

© *M. Romanică*

Exerciţii de învăţare automată

Liviu Ciortuz, Alina Munteanu, Sebastian Ciobanu

Draft, 27 septembrie 2026

Pentru corecţii şi sugestii de îmbunătăţire, vă rugăm să folosiţi adresa
liviu.ciortuz@info.uaic.ro

Motto:

„Tot ce găseşte mâna ta să facă, fă cu toată puterea ta, căci în locuinţa morţilor, în care mergi, nu mai este nici lucrare, nici chibzuială, nici ştiinţă, nici înţelepciune.“

Cartea Eclesiastului 9:10

Cuprins

Mulţumiri 7
Cuvânt înainte 9

0 Fundamente matematice ale învăţării automate **16**
0.1 Probleme rezolvate 23
0.1.1 Evenimente aleatoare şi formula lui Bayes 23
0.1.2 Variabile aleatoare 30
0.1.3 Distribuţii probabiliste uzuale 56
0.1.4 Estimarea parametrilor unor distribuţii probabiliste 93
0.1.5 Elemente de teoria informaţiei 129
0.1.6 Funcţii-nucleu 158
0.1.7 Metode de optimizare în învăţarea automată 170
0.2 Probleme propuse 216
0.2.1 Evenimente aleatoare şi formula lui Bayes 216
0.2.2 Variabile aleatoare 218
0.2.3 Distribuţii probabiliste uzuale 223
0.2.4 Estimarea parametrilor unor distribuţii probabiliste 230
0.2.5 Elemente de teoria informaţiei 241
0.2.6 Funcţii-nucleu 250
0.2.7 Metode de optimizare în învăţarea automată 257

1 Metode de regresie **272**
1.1 Probleme rezolvate 275
1.1.1 Regresia liniară 275
1.1.2 Regresia logistică 304
1.2 Probleme propuse 326
1.2.1 Regresia liniară 326
1.2.2 Regresia logistică 338
1.2.3 Modele liniare generalizate 348

2 Clasificare bayesiană **354**
2.1 Probleme rezolvate 358
2.1.1 Ipoteze de probabilitate maximă a posteriori (MAP) 358
2.1.2 Algoritmii Bayes Naiv şi Bayes Optimal 368
2.1.3 Clasificare bayesiană gaussiană 395
2.2 Probleme propuse 414
2.2.1 Ipoteze de probabilitate maximă a posteriori (MAP) 414
2.2.2 Algoritmii Bayes Naiv şi Bayes Optimal 416
2.2.3 Clasificare bayesiană gaussiană 430

3 Învăţare bazată pe memorare — Algoritmul k-NN **440**
3.1 Probleme rezolvate . . . 443
3.2 Probleme propuse . . . 475

4 Arbori de decizie **486**
4.1 Probleme rezolvate . . . 492
4.1.1 Algoritmul ID3 . . . 492
4.1.2 Algoritmul AdaBoost . . . 541
4.2 Probleme propuse . . . 590
4.2.1 Algoritmul ID3 . . . 590
4.2.2 Algoritmul AdaBoost . . . 612

5 Maşini cu vectori-suport **636**
5.1 Probleme rezolvate . . . 640
5.1.1 SVM cu margine "hard" . . . 640
5.1.2 SVM cu margine "soft" . . . 666
5.1.3 Alte *probleme de optimizare* de tip SVM . . . 707
5.2 Probleme propuse . . . 727
5.2.1 SVM cu margine "hard" . . . 727
5.2.2 SVM cu margine "soft" . . . 735
5.2.3 Alte *probleme de optimizare* de tip SVM . . . 745

6 Reţele neuronale artificiale **756**
6.1 Probleme rezolvate . . . 759
6.1.1 Chestiuni introductive . . . 759
6.1.2 Unităţi neuronale — algoritmi de antrenare . . . 775
6.1.3 Reţele "feed-forward" — algoritmul de retropropagare . . . 797
6.2 Probleme propuse . . . 815
6.2.1 Chestiuni introductive . . . 815
6.2.2 Unităţi neuronale — algoritmi de antrenare . . . 819
6.2.3 Reţele "feed-forward" — algoritmul de retropropagare . . . 828

7 Clusterizare **840**
7.1 Probleme rezolvate . . . 847
7.1.1 Clusterizare ierarhică . . . 847
7.1.2 Algoritmul K-means . . . 861
7.1.3 Algoritmul EM pentru modele de mixturi gaussiene . . . 881
7.2 Probleme propuse . . . 928
7.2.1 Clusterizare ierarhică . . . 928
7.2.2 Algoritmul K-means . . . 937
7.2.3 Algoritmul EM pentru modele de mixturi gaussiene . . . 950

8 Schema algoritmică EM **974**
8.1 Probleme rezolvate . . . 977
8.1.1 Fundamente teoretice . . . 977
8.1.2 Mixturi de distribuţii Bernoulli / categoriale . . . 987
8.1.3 Distribuţii binomiale / multinomiale . . . 1033
8.1.4 Sume de variabile aleatoare . . . 1040
8.1.5 Alte instanţe ale schemei algoritmice EM . . . 1051
8.2 Probleme propuse . . . 1056
8.2.1 Fundamente teoretice . . . 1056

8.2.2 Mixturi de distribuţii Bernoulli / categoriale 1060
8.2.3 Distribuţii binomiale / multinomiale 1071
8.2.4 Alte câteva instanţe ale schemei algoritmice EM 1078

9 Modele Markov ascunse **1086**
9.1 Probleme rezolvate 1087
9.2 Probleme propuse 1110

Mulţumiri

O mare parte dintre exerciţiile şi problemele din această culegere au fost date la examenele sau temele pentru acasă de la cursurile de învăţare automată ţinute în anii 1994-2026 la Carnegie Mellon University (CMU) din Pittsburgh, Statele Unite ale Americii, de către profesorii Tom Mitchell, Andrew Moore, Carlos Guestrin, Geoff Gordon, Eric Xing, William Cohen, Ziv Bar-Joseph, Aarti Singh, Roni Rosenfeld, Alex Smola, Barnabás Póczos, Seyoung Kim, Nina Balcan şi Matt Gormley. Alte exerciţii şi probleme provin de la cursul de învăţare automată al profesorului Andrew Ng de la Universitatea Stanford, precum şi cel al profesorilor Tommi Jaakkola, Regina Barzilai şi Leslie Käelbling de la MIT, S.U.A.

Facem menţiunea că, în general, rezolvările care apar în prezenta culegere au un nivel de detaliere semnificativ mai elaborat decât rezolvările originale, prezentate adeseori doar în mod esenţializat de către profesorii mai sus menţionaţi şi/sau asistenţii dânşilor.

Tuturor acestora le aducem pe această cale cele mai sincere mulţumiri pentru generozitatea cu care au pus la dispoziţie pe internet aceste materiale didactice.

Mai mulţi studenţi de la Facultatea de Informatică a Universităţii „Alexandru Ioan Cuza“ din Iaşi ne-au ajutat de-a lungul ultimilor ani la redactarea textelor sau a imaginilor pentru unele probleme. Îi menţionăm în ordinea alfabetică a numelui de familie: Luisa Apachiţei, Gheorghe Balan, Ciprian Băetu, Călin Bălan, Cristian Budăianu, Giorgiana Caltais, Ionuţ Călin, Petru Cehan, Ivona Chili, Sebastian Ciobanu, Lidia Corciova, Oana Cotoman, Denis Crusos, Corina Dimitriu, Sergiu Dinu, Ahmad Cezar El-Nazli, Oana Florean, Eugen Goriac, Irina Grosu, Mihaela Hudişteanu, Andrei-Constantin Iacob, Sergiu Iacob, Anca Luca, Diana Lucaci, Ştefan Matcovici, Dorin Miron, Alexandru Mitan, Andi Munteanu, Georgiana Ojoc, Oriana Oniciuc, Mihaela Potlog, Andreea Prodan, Mădălina Racoviţă, Elena-Irina Radu, Ciprian Recianu, Cristian Rotaru, Irina Roznovăţ, Alexandra Sbiera, Marius Spânu, Andreea Stanciu, Cristina Şerban, Octavian Tamaş, Călin-Rareş Turliuc şi Liana-Ştefania Ţucăr. (Ne cerem sincer scuze pentru eventualele omisiuni.)

De asemenea, alături de Ştefan Panţiru, Anca Ignat, Mihaela Breabăn şi Adrian Zălinescu, care au ţinut seminarii, mai mulţi studenţi — dintre care îi menţionăm pe Ramona Albert, Luisa Apachiţei, Vlad Aioanei, Mihai Baboi, Ştefan Bălăucă, Diana Carcea, Ştefan Catîru, Nicolae Căpăţînă, Sebastian Ciobanu, Andrei Cioromilă, Cristian Cojocaru, Laura Cornei, Rareş Dima, Corina Dimitriu, Alexandru Dobranici, Alina Duca, Marius Georgică, Mihai Grigoriţă, Sorin Griţco, Mihaela Hudişteanu, Mircea Irimescu, Diana Lucaci, Cecilia Mariciuc Mălina Marin, Ştefan Matcovici, Alexandru Mărtinaş, Mihaela Menghereş, Alexandra Minghel, Ionuţ-Vlad Modoranu, Petru Munteanu, Lucian Neştian, Lucian Nevoe, Axenia Niţă, Alexandru Oloieri, Alexandra Pal, Andreea Pădurariu, Iulian Pichiu, Codrina Prisecaru, Emanuel Puşcaşu, Mihaela Radu, Carol Rameder, Ştefan-Vladimir Sbârcea, Raluca Scorţanu, Cristian Simionescu, Nicolae Şoitu, Ionel Ştefănucă, Silviu Şutea-Drăgan, Cezar Todirişcă, Iulian Vâscu, Silviu Viţel şi Ina Vivdici —, precum şi domnul Liviu Olaru şi conf. dr. Elena Nenciu, au identificat anumite erori în versiuni preliminare ale prezentei culegeri.

Mulţumim domnilor Ştefan Panţiru, Daniel Munteanu şi Dorin Miron pentru instalarea programelor necesare pentru tehnoredactare. Mulţumim redactorilor de carte Eduard Dulman şi Cerasela Cirimpei, precum şi doamnelor Dana Lungu şi Luminiţa Răducanu de la Editura Universităţii „Alexandru Ioan Cuza“ din Iaşi

pentru asistenţa acordată în vederea publicării primei ediţii a acestei cărţi. Mulţumiri speciale se cuvin Mihaelei Romanică din Antwerpen, Belgia, care ne-a pus la dispoziţie desenele care apar în deschiderea fiecărui capitol, precum şi desenul de pe copertă.

Liviu Ciortuz îi mulţumeşte pe această cale profesorului Cristian Preda de la Universitatea din Lille şi INRIA – Nord Europe pentru posibilitatea de a lucra acolo la finalizarea primei ediţii a acestei culegeri, în luna iulie 2015.

Cuvânt înainte — de Liviu Ciortuz

Aşa cum se cuvine, vom preciza aici următoarele chestiuni: de ce anume a fost scrisă această culegere, cum a fost ea alcătuită în timp, care au fost criteriile de selecţie atât la nivel tematic — adică, de ce au fost alese respectivele capitole de învăţare automată şi nu altele — cât şi la nivelul tipului / tipurilor de exerciţii incluse, cum se situează această culegere în raport cu alte culegeri din aria învăţării automate, care este suportul bibliografic,[1] documentele on-line care o însoţesc — site-ul asociat, slide-uri, un index tematic şi un „companion" de tip probleme de implementare —, precum şi ideile pe care le avem acum cu privire la o posibilă extindere ulterioară a culegerii.

Cartea aceasta s-a născut adunând mai întâi exerciţiile care au fost date la examenele şi testele „pe parcurs" din cadrul cursului de *Învăţare automată* de la Facultatea de Informatică a Universităţii „Alexandru Ioan Cuza" din Iaşi. Cursul acesta s-a ţinut începând din anul 2003.[2] Atunci când s-au acumulat un număr semnificativ de exerciţii traduse în limba română, a fost limpede că ar fi de mare ajutor ca rezolvările acestor exerciţii să fie puse la dispoziţia studenţilor din seriile noi, sub forma unei culegeri. Pentru aceasta, am socotit că este benefic să apelez la ajutorul unora dintre cei mai buni studenţi care au absolvit deja cursul. Astfel, mai întâi Elena Bădărău (studentă la master, seria 2008–2010) şi apoi, într-o măsură considerabil sporită, Alina Munteanu (studentă la master, seria 2009–2011, iar după aceea doctorand) au redactat soluţii (în general, semnificativ mai extinse faţă de original), la multe dintre problemele pe care le culesesem anterior.

De atunci şi până acum am cules, am tradus şi chiar am formulat multe alte exerciţii, cu scopul acoperirii unei arii suficient de largi pentru fiecare dintre capitolele predate la curs. Fiecare capitol din culegere a fost împărţit într-o secţiune de probleme în probleme rezolvate şi o secţiune de probleme propuse. În ultimii ani, după ce cele două co-autoare au plecat din facultate,[3] am redactat multe soluţii şi am realizat şlefuirea întregului material. De-a lungul acestor ani, am prezentat la cursurile de învăţare automată următoarele capitole (selecţia şi ordinea lor de succesiune variind uşor de la an la an): fundamente matematice — şi anume, probabilităţi şi statistică, estimarea parametrilor unor distribuţii probabiliste, teoria informaţiei, funcţii-nucleu şi metode de optimizare (toate aceste secţiuni având rol de suport pentru capitolele următoare) —, modele de regresie, clasificare bayesiană, învăţare bazată pe memorare, arbori de decizie, maşini cu vectori-suport,

[1] Este vorba despre cartea sau cărţile care sunt mai potrivite pentru a fi folosite în tandem cu această culegere şi, foarte posibil, pentru cursul aferent.

[2] În perioada 2003–2008 el a fost curs opţional la ciclul de licenţă, anul IV. Apoi, odată cu trecerea la sistemul Bologna, el a devenit curs obligatoriu la ciclul de master. Ulterior, două dintre cele cinci specializări / programe de master de la facultatea noastră l-au pus la dispoziţia studenţilor drept curs obligatoriu, iar celelalte trei l-au păstrat drept curs opţional.

Începând din anul 2015, cursul acesta s-a scindat într-un curs de bază (obligatoriu) în cadrul ciclului de licenţă, la anul III, şi un curs avansat (sub denumirea de *Capitole speciale de Învăţare automată*) la master.

În anii universitari 2018–2019 şi 2019–2020 cursul de bază de *Învăţare automată* a fost ţinut sub formă de curs opţional şi la masterul de *Matematici aplicate* de la Facultatea de Matematică a Universităţii „Alexandru Ioan Cuza" din Iaşi. De asemenea, în semestrul întâi al anului universitar 2012–2013 am ţinut cursul de *Învăţare automată* în regim modular (timp de 4 săptămâni) şi la studenţii de la master, specializarea informatică, de la Facultatea de Matematică-Informatică a Universităţii de Vest din Timişoara.

[3] După absolvirea masterului, Elenă Bădărău a lucrat pentru un timp la compania Amazon din Iaşi. În perioada 2015-2017, dânsa a avut amabilitatea de a ţine seminarii la cursul de *Învăţare automată* de la ciclul de licenţă din facultatea noastră. Alina Munteanu a început doctoratul la Iaşi şi apoi l-a continuat la *Max Delbrück Center for Molecular Medecine* din Berlin, Germania, folosind tehnici de învăţare automată în rezolvarea unor probleme de cercetare în bioinformatică. Dânsa şi-a susţinut teza de doctorat la *Universitatea Humboldt* din Berlin în octombrie 2017.

reţele neuronale artificiale, clusterizare, schema algoritmică EM, modele Markov ascunse, şi, în sfârşit, teoria învăţării computaţionale.

În actuala versiune a culegerii apar exerciţii pentru primele zece capitole dintre cele unsprezece enumerate mai sus. Dintre acestea, capitolul 0 (anumite secţiuni) şi capitolele 2-4 şi 7 corespund acum programei cursului de *Învăţare automată* de la licenţă, iar restul capitolelor corespund programei cursului de *Capitole speciale de Învăţare automată* de la programul de master de la facultatea noastră.[4] Pe lângă adăugarea de noi capitole, ne propunem ca pe viitor să extindem culegerea prin introducerea de noi exerciţii care să acopere „segmente" care nu sunt prezente în capitolele deja elaborate. Ne gândim, de exemplu, la unele exerciţii legate de analiza componentelor principale (engl., *Principal Component Analysis*), clusterizare spectrală (engl., spectral clustering), precum şi analiza prin factori (engl., factor analysis).

Majoritatea exerciţiilor din această culegere sunt destinate a fi rezolvate „cu creionul pe hârtie". Ele solicită în general fie *aplicarea* principalilor algoritmi din capitolele / domeniile de învăţare automată care au fost menţionate mai sus, fie *demonstrarea* unor proprietăţi (teoretice) ale acestor algoritmi sau ale conceptelor utilizate de ei.

Remarcăm modul în care au fost şi sunt în general concepute exerciţiile cu caracter teoretic de la cursurile de *Machine Learning* de la CMU. Şi anume, ele călăuzesc pas cu pas studentul în aşa fel încât să ajungă să demonstreze el însuşi (cu încredere crescândă în forţele proprii) rezultate importante. În acest fel se micşorează sau chiar se elimină efortul necesar pentru memorarea integrală a acestor demonstraţii, care devin astfel mai degrabă obiect de lucru acasă sau în grup decât de prezentare abstractă şi aridă la curs. Pe această cale se realizează o conlucrare foarte bine mediată între profesor (şi / sau asistent) pe de o parte şi student pe de altă parte, pe baza acestor texte de tip problemă / exerciţiu, bine concepute şi elaborate având în minte acest scop didactic.

În alcătuirea acestei culegeri am lăsat în mod intenţionat de o parte problemele cu specific de implementare. Am procedat aşa din două motive. În primul rând, există suficient de multe cărţi de specialitate care se ocupă de astfel de chestiuni. În al doilea rând, am început deja să creăm pentru studenţii noştri un repertoar (engl., *repository*) de seturi de date, implementări şi scripturi pentru cele mai instructive probleme practice date la lucrările pentru acasă (engl., *homeworks*) de la cursurile de *Machine Learning* de la CMU şi eventual de la alte universităţi. Corespunzător, este în curs avansat de elaborare (în limba engleză) o fasciculă care constituie un *Companion practic* pentru prezenta culegere.

Culegerea are un *site* asociat: http://profs.info.uaic.ro/∼ciortuz/ML.ex-book/**. Acolo se găsesc — pe lângă unele *seturi de date* folosite la exerciţii pentru care au fost create grafice ce pun în evidenţă comportamentul unor algoritmi de învăţare automată pe care îi studiem — *slide*-uri pentru cele mai utile exerciţii din culegere. Menţionăm că multe dintre aceste slide-uri au fost folosite nu doar la seminarii ci şi la curs, pentru exemplificarea aplicării algoritmilor predaţi, demonstrarea unor proprietăţi importante ale lor etc. Sperăm că acest set de slide-uri se va mări în timp şi că ele vor fi folosite şi de cadre didactice de la alte facultăţi. Majoritatea slide-urilor au fost redactate *în limba engleză* (spre deosebire de culegere!), pentru ca de ele să beneficieze un număr cât mai mare de studenţi.[5]**

[4] Capitolul de *Teoria învăţării computaţionale* (engl., Computational learning theory) se află acum într-un stadiu incipient de elaborare. Sperăm că vom avea răstimpul necesar pentru finalizarea lui şi atunci acest capitol va fi adăugat la culegere într-o ediţie viitoare.

[5] La fiecare dintre exerciţiile pentru care am făcut deja slide-uri, am pus în antet — mai precis, acolo unde

Ca *suport bibliografic* la *curs* am folosit cu preponderenţă în primii ani de predare cartea *Machine Learning* a profesorului Tom Mitchell de la CMU (ed. McGraw-Hill, 1997). Deşi nu mai este de dată recentă, această carte are un stil uşor accesibil studenţilor noştri. Menţionăm că, dintre cele unsprezece capitole enumerate mai înainte, cartea profesorului Tom Mitchell nu conţine capitolele de fundamente, clusterizare, maşini cu vectori-suport şi modele Markov ascunse (dar conţine alte capitole, precum învăţare bazată pe ranforsare şi algoritmi genetici). Pentru trei dintre cele patru capitole pe care tocmai le-am menţionat,[6] am folosit cartea *Foundations of Statistical Natural Language Processing* (Christopher Manning, Hinrich Schütze, MIT Press, 2002).[7] Pe parcursul acestor ani de predare, am făcut referire din ce în ce mai mult şi la notiţele de curs (engl., *Lecture notes*) ale renumitului profesor Andrew Ng de la Stanford University, în care se folosesc adeseori formalizări matematice mai sofisticate.[8]

La începutul fiecărui capitol din carte am alcătuit un *sumar*, sau o *privire de ansamblu*, un conspect (engl., overview) asupra capitolului respectiv. El poate fi văzut şi ca un *index de probleme*. Ansamblul acestor conspecte constituie un rezumat al cursului / cursurilor de *Învăţare automată* de la facultatea noastră. Pentru aproape fiecare chestiune care se pretează a fi predată la curs, acest *sumar* / *index* indică unul sau mai multe exerciţii care se referă la (sau folosesc, exemplifică) chestiunea respectivă.

Sintetizând, putem spune că această culegere este o încercare de a pune împreună în mod unitar exerciţii cu caracter preponderent conceptual şi teoretic pentru principalele capitole de învăţare automată. Ea foloseşte un aparat matematic care este bazat în special pe noţiunile de matematică din liceu şi de la cursurile de matematică şi de probabilităţi din primul an de facultate.

Remarcăm că, pe de o parte, există cărţi conţinând exerciţii specializate pe [câte] un capitol de învăţare automată, de exemplu *Regression Analysis by Example* (Samprit Chatterjee, Ali S. Hadi, Wiley-Interscience, 2006) şi *Reinforcement Learning: An Introduction* (Richard S. Sutton, Andrew G. Barto, MIT Press, 1998). Însă, pe de altă parte, nu ştim să existe o culegere similară culegerii noastre, având o arie suficient de vastă, cu exerciţii situate pe două paliere, unul exemplificativ, iar celălalt conceptual şi teoretic, în aşa fel încât (foarte important!) să fie adecvate pentru nivelul studenţilor de la facultatea noastră. Trebuie spus că exerciţiile din cărţile consacrate în domeniul învăţării automate, care acoperă un număr mare de capitole — de exemplu *Pattern Classification* (Richard O. Duda, Peter E. Hart, David G. Stork, Wiley-Interscience, 2003), *Pattern Recognition and Machine Learning* (Christopher M. Bishop, Springer, 2006), *The Elements of Statistical Learning. Data Mining, Inference and Prediction* (Trevor Hastie, Robert Tibshirani, Jerome Friedman, Springer, 2009), precum şi *Pattern Recognition* (Sergios Theodoridis, Konstantinos Koutroumbas, Academic Press, 2009) —, folosesc adeseori un aparat matematic destul de complex. Rolul culegerii noastre este de

se indică *subiectul* şi *sursa* respectivului exerciţiu — semnul ■.

[6] Excepţie face capitolul despre maşini cu vectori-suport (SVM). O foarte bună prezentare a acestui domeniu este făcută în cartea *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods* de Nello Cristianini şi John Shawe-Taylor, publicată la editura Cambridge University Press, în anul 2000.

[7] Alegerea acestei cărţi se datorează experienţei pe care am acumulat-o mai înainte în acest domeniu (NLP), asupra căruia învăţarea automată a avut un foarte puternic impact în ultimele două decenii, iar acest fapt este bine reflectat în cartea menţionată.

[8] Pentru studenţii cei mai sârguincioşi, care vor să se iniţieze în cercetarea din domeniul învăţării automate de tip „shallow“, recomandăm acum cartea *Machine Learning — A Probabilistic Perspective* de Kevin P. Murphy, publicată la editura MIT Press în anul 2012.

a fi o „punte“ înspre nivelul cărţilor pe care tocmai le-am menţionat,[9] astfel încât studenţii noştri cei mai buni să poată să acceadă la acel nivel.[10]

Experienţa pe care am acumulat-o în ultimii ani, în care am folosit deja culegerea, dotând studenţii de la cursurile noastre de *Învăţare automată* cu drafturi preliminare ale ei, ne arată că exerciţiile din această carte sunt suficiente nu doar pentru două cursuri de câte un semestru (având săptămânal două ore de predare propriu-zisă şi două ore de seminar), ci şi pentru seminarii suplimentare / avansate, cu un grup de studenţi de "top".

Înainte de a încheia, trebuie precizate câteva *detalii tehnice*:
1. La folosirea ghilimelelor, am utilizat " " pentru expresii în limba engleză, respectiv „ “ pentru expresii româneşti, conform regulilor de ortografie corespunzătoare.
2. Pentru numere zecimale, din pură convenienţă, am folosit în general notaţia de tip anglo-saxon (adică, de exemplu, 1.5 în loc de $1,5$, cum ar fi trebuit), datorită faptului că am tehnoredactat în paralel două versiuni (engleză şi română) ale culegerii, iar formulelele matematice sunt (deocamdată) comune pentru cele două versiuni.
3. Menţionăm că, tot din motive de convenienţă la tehnoredactare, figurile (dar şi tabelele) din carte nu au fost numerotate şi, în consecinţă, fiecare figură a fost plasată în imediata vecinătate a textului care se referă la ea.
4. Precizăm că atunci când, în cadrul unei probleme (fie al enunţului, fie al soluţiei), ne vom referi la un anumit punct al problemei respective, îl vom desemna printr-unul din caracterele italice a, b, c etc, deşi la începutul punctului respectiv apare litera scrisă sub forma uzuală: a, b, c etc. De asemenea, atunci când, într-o anumită problemă, ne vom referi la un punct al unei alte probleme, vom scrie pe scurt, de exemplu, „vedeţi problema 14.b“; se va înţelege că facem trimitere la punctul b al problemei 14.

Atât eu cât şi ceilalţi doi coautori ai acestei culegeri ne dorim ca vastul efort depus pentru alcătuirea ei — în primul rând, de către autorii exerciţiilor originale dar, cu modestia de rigoare, şi de către noi — să permită generaţiilor actuale şi viitoare de studenţi şi doctoranzi să-şi însuşească cunoştinţelele de bază din domeniul învăţării automate, care a ajuns azi la un nivel impresionant de aplicabilitate, atât în industria IT cât şi în cercetarea ştiinţifică.

Liviu Ciortuz
Iaşi, decembrie 2024

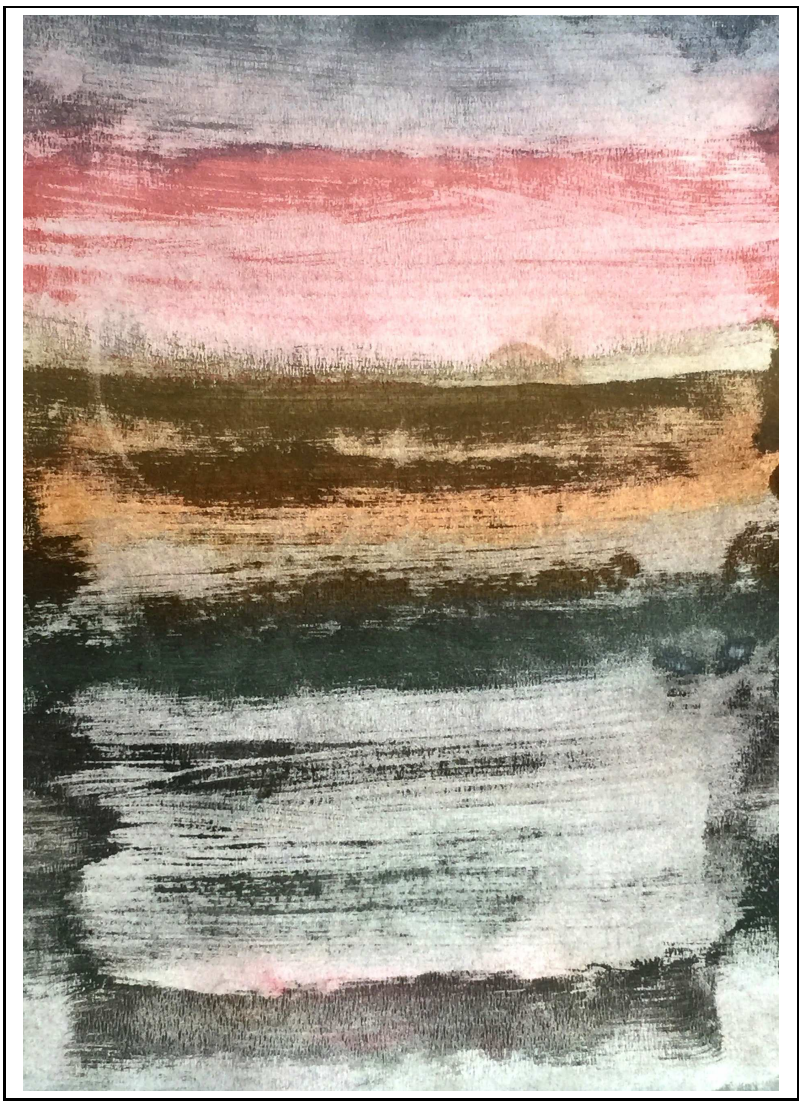
[9] Pentru soluţii la exerciţiile din cărţile enumerate, vedeţi: *Solution Manual to accompany Pattern Classification (2nd ed.) by R. O. Duda, P. E. Hart and D. G. Stork* (David G. Stork, 2001), *Pattern Recognition and Machine Learning Solutions to the Exercises: Web-Edition* (Markus Svens én and Christopher M. Bishop, 2009), *A Solution Manual and Notes for: The Elements of Statistical Learning by Jerome Friedman, Trevor Hastie,and Robert Tibshirani* (John L. Weatherwax, David Epstein, http://waxworksmath.com/Authors/G_M/Hastie/hastie.html, 21 June 2013), *Notes and Solutions for: Pattern Recognition by Sergios Theodoridis and Konstantinos Koutroumbas* (John L. Weatherwax, http://waxworksmath.com/Authors/N_Z/Theodoridis/theodoridis.html, October 17, 2015).

[10] O culegere de exerciţii de tip implementare în domeniul învăţării automate este următoarea: *Introduction to Pattern Recognition: A Matlab Approach* (Sergios Theodoridis, Aggelos Pikrakis, Konstantinos Koutroumbas, Dionisis Cavouras, Academic Press, 2010). Pentru exemplificări pe aceeaşi „linie“, vedeţi şi *Computational Statistics Handbook with Matlab* (Wendy Martinez, Angel Martinez, CRC Press, 2007, 2015).

"Puerile as such an exercise may seem, it sharpens the faculties of observation, and teaches one where to look and what to look for."

Sherlock Holmes in *Study in Scarlet*,

cited by
Wolfgang Karl Härdle and Zdeněk Hlávka in
***Multivariate Statistics — Exercises and Solutions*,**
second edition, Springer, 2015



© *M. Romanică*

0 Fundamente matematice: probabilităţi, variabile aleatoare, estimarea parametrilor unor distribuţii probabiliste, teoria informaţiei, funcţii-nucleu şi metode de optimizare

Sumar

Evenimente aleatoare şi formula lui Bayes

– ***funcţia de probabilitate*** – **câteva proprietăţi care derivă din definiţia ei: ex. 1, ex. 93;**

– **calcularea unor *probabilităţi* elementare: ex. 2.a, ex. 89;**

– **calcularea unor *probabilităţi condiţionate*: ex. 2.b, ex. 3, ex. 91.a;**

– ***regula de înmulţire*: ex. 93.b;**

– ***formula probabilităţii totale*: ex. 90, ex. 93.e;**
formula probabilităţii totale – varianta condiţională: ex. 96.cd;

– ***independenţa* evenimentelor aleatoare: ex. 4, ex. 5, ex. 91.bc;**

– ***independenţa condiţională* a evenimentelor aleatoare – legătura dintre forma „tare“ şi forma „slabă“ a definiţiei pentru această noţiune: ex. 92;**

– ***formula lui Bayes*** – **aplicare: ex. 6, ex. 7, ex. 94, ex. 95;**
formula lui Bayes – varianta condiţională: ex. 96.b;

– **recapitulare: probabilităţi elementare şi probabilităţi condiţionate – câteva proprietăţi (A/F): ex. 8, ex. 96.**

Variabile aleatoare

– **funcţie de *distribuţie cumulativă* (engl., cumulative distribution function, c.d.f.), un exemplu: ex. 33 (vedeţi şi ex. 9 de la capitolul *Învăţare bazată pe memorare*);**

– **proprietatea de *liniaritate* a *mediei*: ex. 9.a;**

– ***varianţa* şi *covarianţa*: proprietăţi de tip *caracterizare*: ex. 9.bc;**

– **covarianţa oricăror două variabile aleatoare independente este 0: ex. 10;**
reciproca acestei afirmaţii nu este adevărată în general: ex. 11.a, ex. 100;
totuşi ea are loc dacă variabilele sunt de tip binar (adică iau doar valorile 0 şi 1): ex. 11.b;

– **o *condiţie suficientă* pentru ca $Var[X + Y] = Var[X] + Var[Y]$: independenţa variabilelor X şi Y: ex. 23.c**

• **pentru *variabile aleatoare discrete*:**

– **calcularea mediilor şi a varianţelor – exemplificare: ex. 12, ex. 97, ex. 99.a;**

– **definirea unei *variabile-indicator* cu ajutorul unui eveniment aleator; calcularea mediei acestei variabile: ex. 98;**

– **regula de *înlănţuire* (pt. var. aleat.) – aplicare: ex. 13;**

– **regula de *înmulţire* (pt. var. aleat.), varianta condiţională – demonstraţie: ex. 14;**

– ***independenţă*: ex. 99.b, ex. 101.a, ex. 16.a;**
***independenţă condiţională*: ex. 15, ex. 101.b, ex. 16.b, ex. 102.b;**
o condiţie suficientă pentru independenţa condiţională: ex. 103;

• **pentru *variabile aleatoare continue*:**

– **dată fiind o funcţie care depinde de un parametru real, să se calculeze valoarea respectivului parametru astfel încât funcţia respectivă să fie o *funcţie de densitate de probabilitate* (engl., probability density function, p.d.f.): ex. 17.a, ex. 18, ex. 104;**

– **dat fiind un p.d.f., să se calculeze o anumită probabilitate: ex. 17.c, ex. 105;**

– **variabile aleatoare discrete vs. variabile aleatoare continue; p.m.f. vs. p.d.f.: ex. 107;**

– **variabile aleatoare discrete şi variabile aleatoare continue; independenţa şi calcul de medii: ex. 108;**

• ***coeficientul de corelaţie* pentru două variabile aleatoare: ex. 19;**

• ***vector de variabile aleatoare*:**
o proprietate: matricea de covarianţă este simetrică şi pozitiv definită: ex. 20;
calculul matricei de covarianţă când asupra vectorului de variabile aleatoare operăm transformări liniare: ex. 109;

• **câteva inegalităţi de bază în teoria probabilităţilor: margini superioare pentru probabilităţi de forma $P(Z \geq t)$ şi $P(Z - E[Z] \geq t)$: ex. 21, ex. 22;**

• **recapitulare: ex. 23, ex. 110.**

Distribuţii probabiliste uzuale

• **distribuţii discrete**

– **distribuţia *Bernoulli*:**
suma de variabile identic distribuite; media sumei: ex. 24;
***mixturi* de distribuţii Bernoulli: ex. 113, ex. 114;**

– **distribuţia *binomială*:**
verificarea condiţiilor de definiţie pentru p.m.f.: ex. 25.a;
calculul mediei şi al varianţei: ex. 25.b, ex. 26.bf;
calcularea unor probabilităţi (simple şi respectiv condiţionale): ex. 111, ex. 26.c;

– **distribuţia *categorială*:**
calcularea unor probabilităţi şi a unor medii: ex. 112;
***mixtură* de distribuţii categoriale: calculul mediei şi al varianţei: ex. 29;**

– **distribuţia *geometrică*:**
calcularea numărului „aşteptat“ / mediu de „observaţii“ necesare pentru ca un anumit eveniment să se producă: ex. 28;

– **distribuţia *Poisson*:**
verificarea condiţiilor de definiţie pentru p.m.f., calculul mediei şi al varianţei: ex. 27;

- **distribuţii continue**

– **distribuţia *continuă uniformă*:**
exemplu de distribuţie continuă uniformă unidimensională; calcularea mediei şi a varianţei: ex. 115;
calculul unei p.d.f. comune, pornind de la două variabile [urmând distribuţii continue uniforme unidimensionale] independente; calculul unei anumite probabilităţi, folosind p.d.f. comună: ex. 30;
exemplu de distribuţie continuă uniformă bidimensională; calcularea unei p.d.f. condiţionale; verificarea independenţei celor două distribuţii marginale; calculul mediilor unor p.d.f.-uri condiţionale: ex. 106, ex. 108;

– **distribuţia *exponenţială*:**
verificarea condiţiilor de definiţie pentru p.d.f.,
calculul mediei şi al varianţei: ex. 31.a;

– **distribuţia *Gamma*:**
verificarea condiţiilor de definiţie pentru p.d.f., calculul mediei şi al varianţei: ex. 31.b;

– **distribuţia *gaussiană unidimensională*:**
verificarea condiţiilor de definiţie pentru p.d.f., calculul mediei şi al varianţei: ex. 32;
„standardizare“ (i.e., reducerea cazului nestandard la cazul standard): ex. 33;

– **distribuţia *gaussiană bidimensională*: exemplificare; calcularea explicită a p.d.f.-ului, dat fiind vectorul de medii şi matricea de covarianţă: ex. 35;**
o *proprietate* pentru distribuţia *gaussiană bidimensională*: distribuţia condiţională a unei componente în raport cu cealaltă componentă este tot de tip gaussian; calculul parametrilor acestei distribuţii condiţionale: ex. 38;

– **distribuţia *gaussiană multidimensională*:**
matrice simetrice şi pozitiv definite[11] o proprietate de tip *factorizare* folosind *vectori proprii*: ex. 36;
densitatea distribuţiei gaussiene multidimensionale este într-adevăr p.d.f.:[12] ex. 37;
calculul mediei şi al matricei de covarianţei: ex. 118;
o *proprietate* importantă, în cazul în care matricea de covarianţă este diagonală: p.d.f.-ul comun este produsul p.d.f.-urilor marginale (care sunt independente): ex. 34;

- **În ce priveşte specificul datelor generate de distribuţii gaussiene multidimensionale:[13]**
◦ în cazul cel mai general (deci când matricea Σ nu este neapărat diagonală), datele generate de acest tip de distribuţie tind să se grupeze în elipse (corpuri elipsoidale) cu *axele de simetrie* [desigur, perpendiculare, dar] în general neparalele cu *axele de coordonate*;
◦ dacă matricea de covarianţă Σ este diagonală, atunci datele generate tind să se grupeze în elipse (dacă se lucrează în $\mathbb{R}^2$) sau, mai general, corpuri elipsoidale având axele de simetrie paralele cu axele sistemului de coordonate;
◦ dacă matricea Σ este de forma $\sigma^2 I$, unde I este matricea identitate, datele generate de respectiva distribuţie tind să se grupeze în sfere;

[11] Aşa sunt matricele de covarianţă ale variabilelor gaussiene multidimensionale.
[12] Adică satisface condiţiile din definiţia noţiunii de p.d.f.
[13] Vedeţi corespondenţa imediată cu alura curbelor de izocontur / nivel determinate de aceste distribuţii.

– ***mixturi*** **de distribuţii** ***gaussiene multidimensionale*****:**
exprimarea vectorului de medii şi a matricei de covarianţă în funcţie de mediile şi matricele de covarianţă ale distribuţiilor componente: ex. 119;

- ***mixturi*** **de distribuţii oarecare:**
calculul mediilor şi al varianţelor (în funcţie de mediile şi matricele de covarianţă ale distribuţiilor componente): ex. 120;
în cazul unei distribuţii multidimensionale de tip mixtură $p(x) = \sum_{k=1}^{K} \pi_k \, p(x|z_k)$, dacă partiţionăm variabila $x \in \mathbb{R}^d$ în două, sub forma $x \overset{not.}{=} (x_1, x_2)$, atunci variabila condiţionată $x_2|x_1$ urmează tot o distribuţie de tip mixtură: ex. 39;
- **distribuţia Bernoulli şi distribuţia normală standard:**
intervale de încredere**,** ***legea numerelor mari*****,** ***teorema limită centrală*****; aplicaţie la calculul erorii reale a unui clasificator: ex. 40;**
- **funcţia generatoare de** ***momente*** **pentru o variabilă aleatoare: ex. 121;**
- **familia de distribuţii exponenţiale: ex. 41 şi ex. 123 (precum şi ex. 42.a şi ex. 43.a de la capitolul** ***Metode de regresie*****);**
- **chestiuni recapitulative (corespondenţa dintre nume de distribuţii şi expresiile unor p.d.f.-uri date): ex. 122.**

Estimarea parametrilor unor distribuţii probabiliste uzuale

– **distribuţia Bernoulli: ex. 43 (+MAP, folosind distr. Beta), ex. 42, ex. 126 (un caz particular), ex. 127.a, ex. 125.bcd (bias-ul şi varianţa estimatorului MLE);**

– **estimarea parametrului de selecţie dintr-o mixtură de [alte două] distribuţii Bernoulli: ex. 45;**

– **distribuţia binomială: ex. 127.bc şi ex. 128 (ultimul, folosind şi metoda gradientului şi metoda lui Newton);**

– **distribuţia categorială: ex. 44, ex. 129 (+MAP, folosind distr. Dirichlet);**

– **distribuţia geometrică: ex. 130 (+MAP, folosind distr. Beta);**

– **distribuţia Poisson: ex. 46 (+MAP, folosind distr. Gamma);**

– **distribuţia uniformă continuă: calcul de probabilităţi şi MLE:**
în $\mathbb{R}$: ex. 47, ex. 131, ex. 132; în $\mathbb{R}^2$: ex. 48, ex. 133;

– **distribuţia gaussiană unidimensională:**
MLE pt. μ, considerând σ^2 cunoscut: ex. 50 (+MAP, folosind distribuţia gaussiană);
MLE pt. σ^2, atunci când nu se impun restricţii asupra lui μ: ex. 51 (+deplasare);
MLE pt. σ^2, atunci când $\mu = 0$: ex. 135 (+nedeplasare);

– **distribuţia gaussiană multidimensională: ex. 53, ex. 136 (+MAP, folosind distr. Gauss-Wishart); un estimator nedeplasat: ex. 138;**

– **distribuţia exponenţială: ex. 49, ex. 134 (+MAP, folosind distr. Gamma);**

– **distribuţia Gamma: ex. 52 şi ex. 137 (ultimul, folosind metoda gradientului şi metoda lui Newton);**

- **existenţa şi unicitatea MLE: ex. 54;**
- **MLE şi parametrizarea alternativă: ex. 139.**

Elemente de teoria informaţiei

– ***definiţii*** **şi** ***proprietăţi imediate*** **pentru entropie, entropie comună, entropie condiţională specifică, entropie condiţională medie, câştig de informaţie: ex. 55;**

exemplificarea **acestor noţiuni (varianta discretă): ex. 56, ex. 57, ex. 140, ex. 141, ex. 142;**

exemple **de calculare a entropiei unor variabile aleatoare continue: distribuţia continuă uniformă (ex. 60.a), distribuţia gaussiană unidimensională (ex. 60.b) şi distribuţia gaussiană multidimensională (ex. 60.c), distribuţia exponenţială (ex. 61.a), distribuţia Gamma (ex. 61.b);**

o ***proprietate*** **a entropiei: nenegativitatea: ex. 55.a, ex. 151.a;**
o margine superioară **pentru valorea entropiei unei variabile aleatoare discrete: ex. 143;**
două ***proprietăţi*** **ale câştigului de informaţie: nenegativitatea (ex. 65.c, ex. 146) şi $IG(X;Y) = 0 \Leftrightarrow$ variabila X este independentă de Y (ex. 65.c); pentru o demonstraţie imediată a implicaţiei directe, vedeţi ex. 141.b;**

– **re-descoperirea definiţiei entropiei, pornind de la un set de** ***proprietăţi dezirabile*****: ex. 62;**

– **o** ***aplicaţie*** **pentru câştigul de informaţie:** ***selecţia de trăsături*****: ex. 58;**

– ***entropia comună*****: exemplu de calculare: ex. 142.d**
forma particulară a relaţiei de „înlănţuire“ în cazul variabilelor aleatoare independente: ex. 59, ex. 144, ex. 151.c;
entropie comună şi condiţionată: formula de „înlănţuire“ condiţională: ex. 145;

– ***cross-entropie*****: definiţie, o proprietate (nenegativitatea) şi un exemplu simplu de calculare a valorii cross-entropiei: ex. 63;**
un exemplu de ***aplicaţie*** **pentru cross-entropie: selecţia modelelor probabiliste: ex. 147;**

– ***inegalitatea lui Gibbs*****: un caz particular; comparaţie între valorile entropiei şi ale cross-entropiei: ex. 64;**

– ***entropia relativă*** **(divergenţa Kullback-Leibler / KL):**
definiţie şi proprietăţi elementare: ex. 65.a;
exprimarea câştigului de informaţie cu ajutorul entropiei relative: ex. 65.b;
o proprietate de tip „relaţie de înlănţuire“: ex. 148;
echivalenţa dintre minimizarea entropiei relative şi maximizarea funcţiei de log-verosimilitate: ex. 149;

– **entropia văzută ca o** ***funcţională*** **în raport cu p; calculul** ***derivatei funcţionale*** **a entropiei în raport cu p: ex. 66;**

– **determinarea distribuţiilor probabiliste unidimensionale care — în anumite condiţii — au entropii maxime, folosind metoda dualităţii Lagrange: ex. 150.**

Funcţii-nucleu

– **definiţie: ex. 67;**

– **aflarea funcţiei de „mapare“ a trăsăturilor care corespunde unei funcţii-nucleu date: ex. 67, ex. 153, ex. 154.a;**
comparaţii asupra numărului de operaţii efectuate la calcularea valorii unor funcţii-nucleu (în spaţiul iniţial vs. spaţiul nou de „trăsături“): ex. 154.b;

calculul distanţelor euclidiene în spaţii de „trăsături“ folosind doar funcţii-nucleu: ex. 71;

- ***teorema lui Mercer* (1909): condiţii necesare şi suficiente pentru ca o funcţie să fie funcţie-nucleu: ex. 68.ab;**
- **rezultate de tip „constructiv“ pentru [obţinerea de noi] funcţii-nucleu: ex. 68.c, 69, ex. 70, ex. 77.b, ex. 156, ex. 157.bcd; contraexemple: ex. 77.ac, ex. 158; „normalizarea“ funcţiilor-nucleu: ex. 155;**

– **o inegalitate [derivată din inegalitatea Cauchy-Buniakovski-Schwarz], care furnizează o margine superioară pentru $K(x, x')$, valoarea absolută a unei funcţii-nucleu oarecare: ex. 159;**

– **un exemplu de funcţie-nucleu care serveşte la a măsura similaritatea dintre două imagini oarecare: ex. 72;**

– **un exemplu de funcţie-nucleu care serveşte la a măsura similaritatea dintre două documente oarecare: ex. 157.a;**

– **exemple de [funcţii de] *mapare a atributelor* care asigură separabilitate liniară [în spaţiul de trăsături] pentru orice set de instanţe de antrenament: ex. 73, ex. 160;**

– **funcţia-nucleu gaussiană / *funcţia cu baza radială* (engl., Radial Basis Function, RBF):**
  - **demonstraţia faptului că RBF este într-adevăr funcţie-nucleu: ex. 74;**
  - **funcţia de „mapare“ corespunzătoare funcţiei-nucleu RBF ia valori într-un spaţiu [de „trăsături“] de dimensiune infinită: ex. 75;**
  - **2 proprietăţi simple ale nucleului RBF: ex. 76, ex. 161;**
  - **RBF menţine monotonia distanţei euclidiene dintre perechi de instanţe:**[14] **ex. 15 de la capitolul *Învăţare bazată pe memorare*;**
  - **orice mulţime de instanţe distincte, având orice etichetare posibilă, este separabilă liniar în spaţiul de „trăsături“ dacă se foloseşte nucleul RBF: ex. 26.a de la capitolul *Maşini cu vectori-suport* (aici parametrul nucleului RBF este ales în mod convenabil);**[15]

– **trecerea de la ne-separabilitate liniară [a unor exemple de antrenament] în *spaţiul de intrare* la separabilitate liniară în *spaţiul de trăsături*: ex. 152, ex. 162;**

– **recapitulare (A/F): ex. 164.**

Metode de optimizare în învăţarea automată

– **(P0) definiţii, caracterizări şi câteva proprietăţi pentru funcţii convexe: ex. 78;**

– **analiza convexităţii unor funcţii de cost folosite în învăţarea automată [profundă]: ex. 166;**

– **inegalitatea lui Jensen: ex. 79;**

[14] Relativ la k-NN vecinătăţi, această proprietate se enunţă astfel: pentru orice x, y şi z din $\mathbb{R}^d$, are loc relaţia echivalenţă

$$\|x - y\| \leq \|x - z\| \Leftrightarrow \|\phi(x) - \phi(y)\| \leq \|\phi(x) - \phi(z)\|,$$

unde ϕ este funcţia de „mapare“ corespunzătoare nucleului RBF.

[15] Problema 5.A de la capitolul *Maşini cu vectori-suport* arată că această proprietate este valabilă pentru orice valoare a parametrului nucleului RBF.

– ***metoda gradientului***, **exemplificare: ex. 80.c, ex. 165, ex. 169, ex.128, ex.137.a; implementare: ex. 168.**
subderivată, ***subdiferenţială*** **şi** ***subgradient*****: ex. 81, ex. 170;**
algoritmul Perceptron[16] poate fi văzut ca o instanţă a algoritmului subgradientului descendent ciclic (varianta stochastică): ex. 171;
metoda lui Newton, **exemplificare: ex. 80.d, ex.128, ex.137.b;**

(P1) condiţii suficiente pentru convergenţa metodei gradientului: ex. 167;
(P2) o proprietate interesantă a metodei lui Newton: în cazul oricărei funcţii de gradul al doilea (de una sau mai multe variabile), aplicarea acestei metode de optimizare implică / necesită execuţia unei singure iteraţii: ex. 172;
(P3) ***reparametrizarea liniară*** **a atributelor nu afectează [rezultatele obţinute cu] metoda lui Newton, însă afectează metoda gradientului: ex. 173;**

– ***metoda dualităţii Lagrange*****:**

(P4) demonstrarea proprietăţii de ***dualitate slabă*****: ex. 82;**
(P5) demonstrarea unei părţi din ***teorema Karush-Kuhn-Tucker*****: ex. 83;**

◦ exemple de aplicare: ex. 84, ex. 85, ex. 86, ex. 143, ex. 174, ex. 175, ex. 176, ex. 177, ex. 178, ex. 179;
◦ un exemplu de problemă de optimizare convexă pentru care condiţiile Karush-Kuhn-Tucker nu sunt satisfăcute: ex. 180;
◦ folosirea metodei multiplicatorilor lui Lagrange la estimarea în sens MLE a parametrilor distribuţiei categoriale: ex. 44;
folosirea metodei multiplicatorilor lui Lagrange pentru determinarea marginii superioare pentru valorile entropiei unei variabile aleatoare discrete: ex. 143;
• aplicarea metodei dualităţii Lagrange la determinarea distribuţiilor probabiliste unidimensionale care — în anumite condiţii — au entropii maxime: ex. 150;

– **două variante a algoritmului Perceptron,[17] pentru care relaţia de actualizare a ponderilor se obţine rezolvând [câte] o problemă de optimizare [convexă] cu restricţii: ex. 87, ex. 181;**

– ***metoda descreşterii pe coordonate***, **în contextul regresiei liniare cu regularizare L_1: ex. 11 de la capitolul** ***Metode de regresie*****;**
metoda subgradientului, **tot în contextul regresiei liniare cu regularizare L_1: ex. 28 de la capitolul** ***Metode de regresie*****;**

– **teorema de reprezentare: ex. 88, ex. 182.**

[16] Vedeţi pr. 16 de la capitolul *Reţele neuronale artificiale*.
[17] Vedeţi problema 16 de la capitolul *Reţele neuronale artificiale*.

0.1 Fundamente — Probleme rezolvate

0.1.1 Evenimente aleatoare şi formula lui Bayes

1. **(Proprietăţi derivate din definiţia funcţiei de probabilitate)**

• ***CMU, 2009 spring, Tom Mitchell, HW2, pr. 1.1***

Fie două evenimente A şi B.

a. Folosind doar proprietăţile din definiţia funcţiei de probabilitate, arătaţi că $P(A \cap \bar{B}) = P(A) - P(A \cap B)$.

b. Demonstraţi *inegalitatea lui Bonferroni*: $P(A \cap B) \geq P(A) + P(B) - 1$.

c. Spunem că evenimentele A şi B sunt *incompatibile* dacă $P(A \cap B) = 0$.

În ipoteza în care $P(A) = 1/3$ şi $P(B) = 5/6$, este posibil ca evenimentele A şi B să fie incompatibile? Justificaţi răspunsul.

Răspuns:

a. $A = A \cap \Omega = A \cap (B \cup \bar{B}) = (A \cap B) \cup (A \cap \bar{B})$.

Cum evenimentele $(A \cap B)$ şi $(A \cap \bar{B})$, văzute ca mulţimi, sunt disjuncte, conform proprietăţii de „aditivitate numărabilă“ din definiţia funcţiei de probabilitate putem scrie $P(A) = P(A \cap B) + P(A \cap \bar{B})$, deci $P(A \cap \bar{B}) = P(A) - P(A \cap B)$.

b. Întrucât $A \cup B = (A \cap \bar{B}) \cup (A \cap B) \cup (\bar{A} \cap B)$, este imediat că $P(A \cup B) = P(A \cap \bar{B}) + P(A \cap B) + P(\bar{A} \cap B)$. Deci,

$$\begin{aligned} P(A \cap B) &= P(A \cup B) - P(A \cap \bar{B}) - P(\bar{A} \cap B) \\ &= [P(A \cup B) - P(A \cap \bar{B})] + [P(A \cup B) - P(\bar{A} \cap B)] - P(A \cup B) \\ &= P(A) + P(B) - P(A \cup B). \end{aligned}$$

Cum $P(A \cup B) \leq 1$ (fiind o probabilitate), putem scrie că: $P(A \cap B) = P(A) + P(B) - P(A \cup B) \geq P(A) + P(B) - 1$, deci $P(A \cap B) \geq P(A) + P(B) - 1$.

***Observaţie*: Din egalitatea $P(A \cap B) = P(A) + P(B) - P(A \cup B)$ dedusă mai sus rezultă**

$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

Această egalitate se numeşte *regula de adunare / însumare a probabilităţilor* (engl., additivity rule).

c. Pentru a studia posibilitatea ca evenimentele A şi B să fie incompatibile vom aplica inegalitatea lui Bonferroni:

$$P(A) + P(B) = \frac{1}{3} + \frac{5}{6} = \frac{7}{6} \Rightarrow P(A \cap B) \geq \frac{1}{6} > 0$$

Deci $P(A \cap B)$ nu poate fi 0 şi, în consecinţă, evenimentele A şi B nu sunt incompatibile.

2. **(Calcularea de probabilităţi elementare şi probabilităţi condiţionate)**

• ***CMU, 2009 spring, Tom Mitchell, HW2, pr. 1.4***

În acest exerciţiu vom arăta că probabilitatea unui eveniment se poate schimba (într-un anumit sens) dacă ştim probabilitatea unui alt eveniment, legat de cel dintâi.

Se aruncă simultan două zaruri. Notăm cu S variabila aleatoare care desemnează suma numerelor rezultate din aruncarea celor două zaruri.

a. Calculaţi $P(S = 11)$.

b. Dacă ştim că S este număr prim, cât devine probabilitatea de mai sus?

Răspuns:

a. Probabilitatea unui eveniment precum cel din enunţ este dată de raportul dintre numărul de cazuri favorabile şi numărul cazurilor posibile.

La aruncarea simultană a două zaruri, fiecare zar poate cădea pe una dintre cele 6 feţe, independent de celălalt zar. Prin urmare, există $6 \cdot 6 = 36$ posibilităţi (cazuri posibile). Pentru a se obţine $S = 11$ cele două zaruri trebuie să aibă fie valorile $(5, 6)$, fie $(6, 5)$, deci există 2 posibilităţi (cazuri favorabile). Prin urmare,

$$P(S = 11) = \frac{2}{36} = \frac{1}{18}$$

b. Probabilitatea căutată este:

$$P(S = 11 \mid S = prim) = \frac{P(S = 11 \cap S = prim)}{P(S = prim)} = \frac{P(S = 11)}{P(S = prim)}$$

Trebuie calculată probabilitatea ca S să fie număr prim. Pentru aceasta este necesar numărul de cazuri favorabile, adică numărul cazurilor pentru care $S \in \{2, 3, 5, 7, 11\}$. Există următoarele 15 posibilităţi:

– $S = 2 : (1, 1)$

– $S = 3 : (1, 2), (2, 1)$

– $S = 5 : (1, 4), (4, 1), (2, 3), (3, 2)$

– $S = 7 : (1, 6), (6, 1), (2, 5), (5, 2), (3, 4), (4, 3)$

– $S = 11 : (5, 6), (6, 5)$

Deci $P(S = prim) = 15/36$. Prin urmare,

$$P(S = 11 \mid S = prim) = \frac{2/36}{15/36} = \frac{2}{15}$$

Întrucât $2/15 > 1/18$, putem spune că probabilitatea evenimentului $S = 11$ a crescut după ce am aflat un fapt suplimentar, şi anume că suma rezultată la aruncarea celor două zaruri este un număr prim ($S = prim$).

Ca o *observaţie* cu caracter general: dacă $A \subseteq B$ (aşa cum a fost cazul în această problemă), atunci rezultă imediat că $P(A) \leq P(A|B)$.

3. **(Spaţiu de eşantionare – exemplificare; calcul de probabilităţi condiţionate)**

• ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW1, pr. 1.4***

O cutie conţine trei carduri. Un card este roşu pe ambele părţi, un alt card este verde pe ambele părţi, iar cel care a rămas este roşu pe o parte şi verde pe partea cealaltă. Selectăm în mod aleatoriu un card din această cutie; presupunem că nu-i vedem decât culoarea de pe faţa superioară. Dacă această faţă este verde, care este probabilitatea ca şi cealaltă faţă să fie verde?

Răspuns:

Pentru a rezolva această problemă este foarte util să stabilim mai întâi care sunt elementele ce compun *spaţiul de eşantionare* (engl., sample space), Ω.[18] Contrar intuiţiei primare, Ω nu va fi constituit din cele trei carduri, ci din feţele lor, fiindcă ceea ce observăm după o extragere este doar o faţă a unui card, nu ambele feţe ale cardului extras.

Din punct de vedere formal, vom folosi următoarea *notaţie* pentru carduri:

$$C_1 = (R1, R2),\ C_2 = (R3, V4),\ C_3 = (V5, V6),$$

unde $R1, R2, R3, V4, V5$ şi $V6$ desemnează cele 6 feţe ale cardurilor. Aşadar, $\Omega = \{R1, R2, R3, V4, V5, V6\}$.

***Observaţie*: Am fi putut nota feţele cardurilor folosind pur şi simplu numerele $1, \ldots, 6$, însă am preferat să însoţim fiecare dintre aceste numere cu litera R sau V care desemnează culoarea feţei respective.**

După ce am făcut această pregătire, probabilitatea cerută în enunţul problemei se calculează simplu, folosind regula clasică: $p = m/n$, unde m este numărul de cazuri favorabile, iar n este numărul de cazuri posibile.

Cazurile posibile sunt $V4, V5, V6$, iar cazurile favorabile sunt $V5$ şi $V6$ deoarece doar pentru ele faţa cealaltă a cardului este verde (este vorba de $V6$ şi respectiv $V5$). Aşadar, probabilitatea cerută este $\frac{2}{3}$.

4. **(Evenimente aleatoare independente)**

• ***CMU, 2009 spring, Tom Mitchell, HW2, pr. 1.2.1***
CMU, 2009 spring, Ziv Bar-Joseph, HW1, pr. 1.1

Două evenimente A şi B sunt independente statistic dacă $P(A \cap B) = P(A) \cdot P(B)$.

a. Arătaţi că dacă A şi B sunt evenimente independente, atunci:

– A şi $\bar{B}$ sunt independente;

– $\bar{A}$ şi $\bar{B}$ sunt independente.

b. Dacă evenimentul A este independent în raport cu el însuşi, ce puteţi spune despre $P(A)$?

Răspuns:

[18] A se vedea noţiunea prezentată la curs.

a. $P(A\cap\bar{B}) = P(A)-P(A\cap B) = P(A)-P(A)\cdot P(B) = P(A)\cdot(1-P(B)) = P(A)\cdot P(\bar{B})$, **deci** A **şi** $\bar{B}$ **sunt independente. (S-a folosit independenţa evenimentelor** A **şi** B**.)**

Pentru independenţa evenimentelor $\bar{A}$ **şi** $\bar{B}$ **se procedează analog:** $P(\bar{A}\cap\bar{B}) = P(\bar{B})-P(A\cap\bar{B}) = P(\bar{B})-P(A)\cdot P(\bar{B}) = P(\bar{B})\cdot(1-P(A)) = P(\bar{A})\cdot P(\bar{B})$**, deci** $\bar{A}$ **şi** $\bar{B}$ **sunt independente. (La cea de-a doua egalitate s-a folosit independenţa evenimentelor** A **şi** $\bar{B}$ **demonstrată mai sus.)**

b. Condiţia de independenţă a evenimentului A **faţă de el însuşi se scrie astfel:**

$$P(A\cap A) = P(A)\cdot P(A) \Rightarrow P(A) = [P(A)]^2 \Rightarrow P(A)[P(A)-1] = 0$$

Ţinând cont şi de restricţia $P(A)\in[0,1]$**, rezultă că** $P(A) = 0$ **sau** $P(A) = 1$**. (Atenţie: nu rezultă neapărat că** $A = \emptyset$ **respectiv** $A = \Omega$**!)**

5. **(Evenimente aleatoare independente; aplicarea proprietăţilor din definiţia funcţiei de probabilitate)**

• ***CMU, 2009 spring, Tom Mitchell, HW2, pr. 1.2.3***

Robert şi Alina dau cu banul alternativ. Cel dintâi dintre ei care va obţine stema (în engleză: head) câştigă jocul. Alina este prima care va da cu banul.

a. Dacă $P(\textit{stemă}) = 1/2$**, care este probabilitatea ca Alina să câştige jocul?**

***Indicaţie* (1): Încercaţi să enumeraţi toate situaţiile în care Alina poate câştiga.**

***Indicaţie* (2): Pentru orice** $a\in[0,1)$**, avem** $\sum_{i=0}^{i=\infty} a^i = 1+a+a^2+\ldots+a^n+\ldots = \lim_{n\to+\infty}\dfrac{1-a^n}{1-a} = \dfrac{1}{1-a}$**.**

b. Dacă $P(\textit{stemă}) = p\in(0,1]$**, care este probabilitatea ca Alina să câştige jocul?**

c. Ţinând cont de expresia obţinută la punctul b**, dacă ar fi ca tu să joci acest joc, cum ai decide să intri în joc: primul ori al doilea (presupunând, bineînţeles, că ai avea posibilitatea să alegi)?**

Răspuns:

a. Alina aruncă moneda în „paşii" impari. Ea câştigă jocul dacă la pasul $2n+1$ **obţine stema *şi* până la pasul respectiv nimeni nu a obţinut stema.**

Notăm cu A **evenimentul ca Alina să câştige jocul şi cu** A_i **evenimentul ca Alina să câştige jocul la a** $i-$**a aruncare a banului. Întrucât evenimentele** A_i**, văzute ca mulţimi, sunt mutual disjuncte (**$A_i\cap A_j = \emptyset$ **pentru orice** $i\neq j$**) şi** $A = A_1\cup A_3\cup\ldots$**, conform proprietăţii de aditivitate numărabilă din definiţia funcţiei de probabilitate rezultă că**

$$P(A) = P(A_1)+P(A_3)+P(A_5)+\ldots$$

Întrucât $p = \dfrac{1}{2}$**, ţinând cont [şi] de faptul că toate aruncările sunt independente, probabilităţile corespunzătoare evenimentelor** A_i **se calculează astfel:**

$$P(A_1) = P(\textit{stemă}) = \frac{1}{2}$$
$$P(A_3) = (1 - P(\textit{stemă})) \cdot (1 - P(\textit{stemă})) \cdot P(\textit{stemă}) = \left(\frac{1}{2}\right)^3$$
$$P(A_5) = (1 - \frac{1}{2}) \cdot (1 - \frac{1}{2}) \cdot (1 - \frac{1}{2}) \cdot (1 - \frac{1}{2}) \cdot P(\textit{stemă}) = \left(\frac{1}{2}\right)^5$$
$$\ldots$$
$$P(A_{2i+1}) = (1 - P(\textit{stemă}))^{2i} \cdot P(\textit{stemă}) = \left(\frac{1}{2}\right)^{2i+1}$$

Prin urmare,

$$P(A) = \sum_{i=0}^{\infty} P(A_{2i+1}) = \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^{2i+1} = \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^{2i} \cdot \frac{1}{2} = \frac{1}{2} \cdot \sum_{i=0}^{\infty} \left(\frac{1}{4}\right)^{i} = \frac{1}{2} \cdot \frac{1}{1 - 1/4}$$
$$= \frac{1}{2} \cdot \frac{4}{3} = \frac{2}{3}.$$

Aşadar, pentru $p = 1/2$ probabilitatea ca Alina să câştige jocul este $2/3$ (deci de două ori mai mare decât probabilitatea ca Robert să câştige jocul), pur şi simplu datorită faptului că ea este prima care dă cu banul. (Avantajul primului jucător!)

b. Dacă $P(\textit{stemă}) = p \in (0, 1]$, se urmează acelaşi raţionament ca mai sus, cu modificarea valorilor $P(A_i)$ astfel:

$$P(A_1) = P(\textit{stemă}) = p$$
$$P(A_3) = (1 - P(\textit{stemă})) \cdot (1 - P(\textit{stemă})) \cdot P(\textit{stemă}) = (1 - p)^2 \cdot p$$
$$P(A_5) = (1 - p) \cdot (1 - p) \cdot (1 - p) \cdot (1 - p) \cdot P(\textit{stemă}) = (1 - p)^4 \cdot p$$
$$\ldots$$
$$P(A_{2i+1}) = (1 - P(\textit{stemă}))^{2i} \cdot P(\textit{stemă}) = (1 - p)^{2i} \cdot p$$

Prin urmare, probabilitatea ca Alina să câştige devine:

$$P(A) = \sum_{i=0}^{\infty} A_{2i+1} = \sum_{i=0}^{\infty} (1-p)^{2i} \cdot p = p \cdot \sum_{i=0}^{\infty} (1-p)^{2i} \overset{p \neq 0}{=} p \cdot \frac{1}{1 - (1-p)^2} = \frac{p}{2p - p^2} = \frac{1}{2 - p}$$

c. Jucătorul care aruncă primul banul câştigă jocul cu o probabilitate de $1/(2-p)$ (calculată mai sus). Cum $p \geq 0$, rezultă că $\frac{1}{2-p} \geq \frac{1}{2}$. Aşadar, în cazul în care există posibilitatea de a alege, este de preferat să arunci primul.

6. (Formula lui Bayes)

- ***CMU, 2001 fall, Andrew Moore, midterm exam, pr. 5***

Se consideră două variabile aleatoare A şi B despre care ştim următoarele informaţii:

a. $P(A \mid B) = 2/3$
b. $P(A \mid B) = 2/3$ **şi** $P(A \mid \bar{B}) = 1/3$
c. $P(A \mid B) = 2/3$, $P(A \mid \bar{B}) = 1/3$ **şi** $P(B) = 1/3$
d. $P(A \mid B) = 2/3$, $P(A \mid \bar{B}) = 1/3$, $P(B) = 1/3$ **şi** $P(A) = 4/9$.

În care din cele patru cazuri informaţiile date sunt suficiente pentru a calcula $P(B \mid A)$? Există vreun caz în care apar informaţii superflue (i.e, informaţii care pot fi deduse din celelalte informaţii furnizate în cazul respectiv)?

Răspuns:

Conform teoremei lui Bayes, vom avea:

$$\begin{aligned} P(B \mid A) &= \frac{P(A \mid B) \cdot P(B)}{P(A)} = \frac{P(A \mid B) \cdot P(B)}{P(A \mid B) \cdot P(B) + P(A \mid \bar{B}) \cdot P(\bar{B})} \\ &= \frac{P(A \mid B) \cdot P(B)}{P(A \mid B) \cdot P(B) + P(A \mid \bar{B}) \cdot (1 - P(B))} \end{aligned}$$

Devine astfel evident că în cazurile c şi d informaţiile deţinute sunt suficiente, pe când în celelalte două cazuri nu. În cazul d, informaţia $P(A) = 4/9$ este superfluă.

7. **(Formula lui Bayes)**

• ○ ***CMU, 2008 spring, T. Mitchell, W. Cohen, midterm, pr. 1.5***

Presupunem că, răspunzând la o întrebare cu răspuns de genul adevărat / fals, un student fie cunoaşte răspunsul, fie ghiceşte răspunsul. Probabilitatea de a cunoaşte răspunsul este p, iar probabilitatea de a ghici răspunsul este $1 - p$.

Presupunem că probabilitatea de a răspunde corect la întrebare este

1 în cazul în care studentul cunoaşte răspunsul

şi 0.5 dacă studentul ghiceşte răspunsul.

Exprimaţi în funcţie de p care este probabilitatea ca studentul examinat să cunoască răspunsul la întrebare, în ipoteza că el a răspuns corect (notaţie: $P(knew \mid correct)$).

Răspuns:

Evenimentele de interes în problema dată sunt:
correct **= studentul a răspuns corect la întrebare,**
knew **= studentul cunoştea răspunsul corect**
şi complementarul acestuia din urmă:
guess $\stackrel{not.}{=} \neg$ ***knew*** **= studentul ghiceşte răspunsul.**

Aplicând formula lui Bayes obţinem:

$$\begin{aligned} P(knew \mid correct) &= \frac{P(correct \mid knew) \cdot P(knew)}{P(correct \mid knew) \cdot P(knew) + P(correct \mid guess) \cdot P(guess)} \\ &= \frac{1 \cdot p}{1 \cdot p + 0.5 \cdot (1 - p)} = \frac{p}{0.5p + 0.5} = \frac{p}{0.5(p+1)} = \frac{2p}{p+1} \end{aligned}$$

8. **(Probabilităţi, chestiuni elementare: Adevărat sau Fals?)**

□ • ○ ***CMU, 2016 fall, N. Balcan, M. Gormley, HW1, pr. 6.1.1-4***

În cele de mai jos vom nota cu Ω spaţiul de eşantionare, iar cu $\bar{A}$ complementul evenimentului A.
Marcaţi cu *adevărat* sau *fals* fiecare dintre afirmaţiile următoare:

a. Pentru orice $A, B \subseteq \Omega$ astfel încât $P(A) > 0$ şi $P(B) > 0$, are loc egalitatea $P(A|B)P(B) = P(B|A)P(A)$.

b. Pentru orice $A, B \subseteq \Omega$ astfel încât $P(B) > 0$, are loc egalitatea $P(A \cup B) = P(A) + P(B) - P(A|B)$.

c. Pentru orice $A, B, C \subseteq \Omega$ astfel încât $P(B \cup C) > 0$, urmează că $\frac{P(A \cup B \cup C)}{P(B \cup C)} \geq P(A|B \cup C)P(B \cup C)$

d. Pentru orice $A, B \subseteq \Omega$ astfel încât $P(A) > 0$ şi $P(\bar{A}) > 0$, urmează că $P(B|A) + P(B|\bar{A}) = 1$.

***Indicaţie*:**
Pentru fiecare afirmaţie adevărată, faceţi demonstraţia proprietăţii respective. Pentru fiecare afirmaţie falsă, daţi fie un contraexemplu fie o justificare riguroasă.

Răspuns:

a. Adevărat. Demonstraţia este imediată.

b. Fals.
Ştim că pentru orice $A, B \subseteq \Omega$, are loc egalitatea $P(A \cup B) = P(A) + P(B) - P(A \cap B)$. Aşadar, egalitatea din enunţ ar fi adevărată dacă pentru orice $A, B \subseteq \Omega$ am avea $P(A \cap B) = P(A|B)$. Însă,

$$P(A \cap B) = P(A|B) \Leftrightarrow P(A \cap B) = \frac{P(A \cap B)}{P(B)} \Leftrightarrow P(B) = 1 \text{ sau } P(A \cap B) = 0.$$

Deci egalitatea nu este adevărată pentru orice A şi B.

c. Adevărat.
Observăm că dacă se notează $D = B \cup C$, atunci inegalitatea din enunţ devine mai uşor de manipulat: $\frac{P(A \cup D)}{P(D)} \geq P(A|D) \cdot P(D)$. Este imediat că ea este echivalentă cu $\frac{P(A \cup D)}{P(D)} \geq P(A \cap D)$, ceea ce implică $P(A \cup D) \geq P(A \cap D) \cdot P(D)$. Ultima inegalitate este adevărată fiindcă pe de o parte $P(A \cup D) \geq P(A \cap D)$ şi pe de altă parte $P(D) \in [0,\ 1]$.

d. Fals.
Ştim că pentru orice $A, B \subseteq \Omega$ cu $P(A) > 0$, are loc egalitatea $P(B|A) + P(\bar{B}|A) = 1$ (o puteţi demonstra imediat). Aşadar, egalitatea din enunţ ar fi adevărată dacă pentru orice $A, B \subseteq \Omega$ astfel încât $P(A) > 0$ şi $P(\bar{A}) > 0$ am avea $P(B|\bar{A}) = P(\bar{B}|A)$.

Vom construi următorul contraexemplu: considerăm că la aruncarea unui zar perfect, evenimentul A este apariţia unei feţe pare, iar evenimentul B este apariţia feţei 1. Urmează că

$$P(B|\bar{A}) = \frac{1}{3} \text{ şi } P(\bar{B}|A) = 1.$$

Deci egalitatea nu este adevărată pentru orice evenimente A (cu $P(A) > 0$ şi $P(\bar{A}) > 0$) şi B.

0.1.2 Variabile aleatoare

9. **(Variabile aleatoare: proprietăţi de bază pentru medii, varianţă, covarianţă)**

Fie variabila aleatoare $X : \Omega \to \mathbb{R}$, cu funcţia de probabilitate P.

Dacă X este variabilă aleatoare *discretă*, atunci prin definiţie $P(x) \overset{not.}{=} P(X = x) \overset{not.}{=} P(\{\omega \mid X(\omega) = x\}) \geq 0$ pentru orice $x \in \mathbb{R}$, şi $\sum_{x_i \in Val(X)} P(x_i) = 1$, unde $Val(X)$ este mulţimea valorilor variabilei aleatoare X.

Dacă X este variabilă aleatoare *continuă*, având funcţia densitate de probabilitate p, atunci prin definiţie $p(X = x) \geq 0$ pentru orice $x \in \mathbb{R}$, şi $\int_{-\infty}^{\infty} p(X = x)dx = 1$ (sau, scris mai simplu: $\int_{-\infty}^{+\infty} p(x)dx = 1$).

a. ■ ***CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, HW1, pr. 1.1***

Dacă X este variabilă aleatoare discretă, *media* sa se defineşte ca fiind numărul real $E[X] = \sum_{x_i \in Val(X)} x_i \cdot P(X = x_i)$. Dacă X este variabilă aleatoare continuă, media sa este $E[X] = \int_{-\infty}^{\infty} x \cdot p(X = x)dx$.

Arătaţi că pentru orice două variabile aleatoare W şi Z de acelaşi tip (adică fie ambele discrete fie ambele continue), având acelaşi domeniu de definiţie (Ω), avem

$$E[W + Z] = E[W] + E[Z].$$

De asemenea, demonstraţi că pentru orice constantă $a \in \mathbb{R}$, are loc egalitatea

$$E[aX] = aE[X]. \tag{1}$$

Notaţi că aX este o variabilă aleatoare definită pe acelaşi domeniu (Ω) ca şi variabila X, cu proprietatea că $(aX)(\omega) \overset{def.}{=} a\,X(\omega)$ pentru orice $\omega \in \Omega$.

***Observaţie*: Cele două egalităţi de mai sus se pot combina sub o formă mai generală: pentru orice variabile aleatoare (fie toate discrete fie toate continue) $X_1, \ldots, X_n$ şi pentru orice constante $a_1, \ldots, a_n \in \mathbb{R}$, cu $n \geq 1$, are loc egalitatea**

$$E[a_1X_1 + \ldots + a_nX_n] = a_1E[X_1] + \ldots + a_nE[X_n]. \tag{2}$$

Această egalitate este cunoscută sub numele de *proprietatea de liniaritate a mediei*.

b. ***CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, HW1, pr. 1.3***

Fie X o variabilă aleatoare. Notăm $\bar{X} = E[X]$. *Varianţa* lui X se defineşte ca fiind $Var(X) = E[(X - \bar{X})^2]$. Arătaţi că:

$$Var(X) = E[X^2] - (E[X])^2.$$

***Observaţie importantă*: Veţi vedea că această proprietate este adeseori folosită (în locul definiţiei varianţei) în diverse demonstraţii care vor urma.**

De asemenea, demonstraţi că pentru orice constantă $a \in \mathbb{R}$, are loc egalitatea

$$Var(aX) = a^2 Var(X). \tag{3}$$

Prin urmare, în cazul varianţei nu avem o proprietate de liniaritate similară cu cea din cazul mediei.

***Indicaţie*: La acest punct nu este necesar să faceţi demonstraţiile separat pentru cele două cazuri, discret şi respectiv continuu.**

c. ***CMU, 2009 spring, Tom Mitchell, HW2, pr. 1.3.1***

***Covarianţa* a două variabile aleatoare X şi Y care au acelaşi domeniu de definiţie (Ω) se defineşte astfel: $Cov(X, Y) = E[(X - E[X])(Y - E[Y])]$, unde $E[X]$ este media lui X. (Este imediat faptul că noţiunea de covarianţă generalizează noţiunea de varianţă.)**

Demonstraţi egalitatea:

$$Cov(X, Y) = E[XY] - E[X] \cdot E[Y]. \tag{4}$$

***Consecinţă imediată* (din relaţiile (1) şi (4)):**

$$Cov(aX, bY) = ab\, Cov(X, Y)\ \forall a, b \in \mathbb{R}. \tag{5}$$

***Observaţii*:**

1. Este imediat faptul că proprietatea (5) generalizează proprietatea (3).

2. Spre deosebire de varianţă, care poate lua doar valori mai mari sau egale cu 0 (ceea ce decurge imediat din definiţia de la punctul b), covarianţa poate lua şi valori negative. Mai mult, la problema 19 se va demonstra că atunci când $Var(X) \neq 0$ şi $Var(Y) \neq 0$, avem următoarele margini (una inferioară şi cealaltă superioară) pentru $Cov(X, Y)$:[19]

$$-\sqrt{Var(X) \cdot Var(Y)} \leq Cov(X, Y) \leq +\sqrt{Var(X) \cdot Var(Y)}. \tag{6}$$

Răspuns:

a. Pentru cazul discret[20] **vom folosi o formă echivalentă a formulei pentru media unei variabile aleatoare şi anume $E[X] = \sum_{\omega \in \Omega} X(\omega) \cdot P(\omega)$. Prin urmare, putem scrie:**

[19] Atunci când $Var(X) = 0$ sau $Var(Y) = 0$, rezultă că variabila X (respectiv Y) este funcţie constantă (fie peste tot domeniul de definiţie, fie cu excepţia unei mulţimi de probabilitate 0), ceea ce implică imediat $Cov(X, Y) = 0$. Aşadar, dubla inegalitate (6) este satisfăcută şi în acest caz.

[20] *Observaţie importantă*: În demonstraţia care urmează — şi, de asemenea, în toate problemele din acest capitol în care se fac / cer demonstraţii pentru variabile aleatoare discrete — se presupune că mulţimile de valori ale acestor variabile sunt finite. Aşadar, tratăm doar cazul discret finit.

$$
\begin{aligned}
E[W+Z] &= \sum_{u \in Val(W+Z)} u \cdot P(W+Z=u) \\
&= \sum_{w \in Val(W), z \in Val(Z)} (w+z) \cdot P(W+Z=w+z) \\
&= \sum_{w \in Val(W)} \sum_{z \in Val(Z)} (w+z) \cdot P(\{\omega \in \Omega \mid (W+Z)(\omega) = w+z\}) \\
&= \sum_{w \in Val(W)} \sum_{z \in Val(Z)} w \cdot P(\{\omega \in \Omega \mid (W+Z)(\omega) = w+z\}) + \\
&\quad \sum_{w \in Val(W)} \sum_{z \in Val(Z)} z \cdot P(\{\omega \in \Omega \mid (W+Z)(\omega) = w+z\}) \\
&= \sum_{w \in Val(W)} w \sum_{z \in Val(Z)} P(\{\omega \in \Omega \mid W(\omega) = w, Z(\omega) = z\}) + \\
&\quad \sum_{z \in Val(Z)} z \sum_{w \in Val(W)} P(\{\omega \in \Omega \mid W(\omega) = w, Z(\omega) = z\}) \\
&= \sum_{w \in Val(W)} wP(\{\omega \in \Omega \mid W(\omega) = w\}) + \sum_{z \in Val(Z)} zP(\{\omega \in \Omega \mid Z(\omega) = z\}) \\
&= E[W] + E[Z].
\end{aligned}
$$

Pentru cazul continuu:

$$
\begin{aligned}
E[W+Z] &= \int_w \int_z (w+z)\, p_{WZ}(w,z) dz dw \\
&= \int_w \int_z w\, p_{WZ}(w,z) dz dw + \int_w \int_z z\, p_{WZ}(w,z) dz dw \\
&= \int_w w \int_z p_{WZ}(w,z) dz dw + \int_z z \int_w p_{WZ}(w,z) dw dz \\
&= \int_w w p_W(w) dw + \int_z z p_Z(z) dz = E[W] + E[Z].
\end{aligned}
$$

În cazul în care variabila aleatoare X este discretă, egalitatea $E[aX] = aE[X]$ se poate demonstra imediat, aplicând definiţia mediei. Vom considera mai întâi subcazul $a \neq 0$:

$$
\begin{aligned}
E[aX] &= \sum_{u \in Val(aX)} u\, P(aX = u) = \sum_{x \in Val(X)} ax\, P(X = x), \quad \textbf{unde } x = \frac{1}{a}u \\
&= a \sum_{x \in Val(X)} x\, P(X = x) = aE[X].
\end{aligned}
$$

Pentru subcazul $a = 0$, egalitatea $E[aX] = aE[X]$ este trivială.

Similar se face demonstraţia egalităţii $E[aX] = aE[X]$ în cazul în care variabila aleatoare X este continuă.

b. Pentru a demonstra această proprietate vom ţine cont de liniaritatea mediei (vedeţi *Observaţia* de la punctul a):

$$
\begin{aligned}
Var(X) &= E[(X - \bar{X})^2] = E[X^2 - 2X\bar{X} + \bar{X}^2] \\
&= E[X^2] - E[2X\bar{X}] + E[\bar{X}^2] = E[X^2] - 2\bar{X}E[X] + \bar{X}^2 \\
&= E[X^2] - 2\bar{X}^2 + \bar{X}^2 = E[X^2] - \bar{X}^2 = E[X^2] - (E[X])^2.
\end{aligned}
$$

Egalitatea $Var\,(aX) = a^2\,Var\,(X)$ rezultă imediat, aplicând definiţia varianţei şi ţinând cont de proprietatea $E[aX] = aE[X]$ pe care am demonstrat-o la punctul a.

$$\begin{aligned} Var\,(aX) &\overset{def.}{=} E[(aX - \underbrace{E[aX]}_{aE[X]})^2] = E[(a(X - E[X]))^2] \\ &= E[a^2(X - E[X])^2] = a^2 E[(X - E[X])^2] \overset{def.}{=} a^2\,Var\,(X). \end{aligned}$$

Alternativ, putem folosi proprietatea pe care am demonstrat-o mai sus:

$$\begin{aligned} Var\,(aX) &= E[(aX)^2] - (E[aX])^2 = E[a^2X^2] - (aE[X])^2 = a^2E[X^2] - a^2(E[X])^2 \\ &= a^2(E[X^2] - (E[X])^2) = a^2\,Var\,(X). \end{aligned}$$

c. Se foloseşte acelaşi gen de raţionament ca la punctul precedent (prima parte):

$$\begin{aligned} Cov\,(X,Y) &= E[(X - E[X])(Y - E[Y])] \\ &= E[XY - XE[Y] - E[X]Y + E[X]E[Y]] \\ &= E[XY] - E[XE[Y]] - E[E[X]Y] + E[E[X]E[Y]] \\ &= E[XY] - E[X]E[Y] - E[X]E[Y] + E[X]E[Y] = E[XY] - E[X]E[Y]. \end{aligned}$$

10. **(Rezultat teoretic: covarianţa oricăror 2 variabile aleatoare independente este 0)**

■ ***CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, HW1, pr. 1.2***

În mod intuitiv, două variabile aleatoare X şi Y sunt *independente* atunci când cunoaşterea valorii uneia dintre ele (de exemplu X) nu furnizează niciun indiciu despre valoarea celeilalte variabile (Y, în acest caz).

Formal, dacă X şi Y sunt variabile aleatoare discrete, independenţa lor revine la egalitatea $P(X = x, Y = y) = P(X = x) \cdot P(Y = y)$ pentru orice $x \in Val(X)$ şi orice $y \in Val(Y)$.

Similar, dacă X şi Y sunt variabile aleatoare continue, atunci $p(X = x, Y = y) = p(X = x) \cdot p(Y = y)$ pentru orice valori x şi y posibile.

Arătaţi că dacă X şi Y sunt variabile aleatoare independente de acelaşi tip (adică fie discret fie continuu), atunci

$$E[XY] = E[X] \cdot E[Y].$$

Echivalent: X, Y independente $\Rightarrow Cov(X, Y) = 0$. (A se vedea problema 9 punctul c.)

***Observaţie*: Reciproca implicaţiei de mai sus nu este în general adevărată. A se vedea problema 11.**

Răspuns:

Pentru cazul în care variabilele aleatoare independente X şi Y sunt discrete, putem scrie:

$$\begin{aligned}
E[XY] &= \sum_{x\in Val(X)}\sum_{y\in Val(Y)} xyP(X=x,Y=y) \\
&\overset{indep.}{=} \sum_{x\in Val(X)}\sum_{y\in Val(Y)} xyP(X=x)\cdot P(Y=y) \\
&= \sum_{x\in Val(X)}\Big(xP(X=x)\sum_{y\in Val(Y)} yP(Y=y)\Big) \\
&= \sum_{x\in Val(X)} xP(X=x)E[Y] = \Big(\sum_{x\in Val(X)} xP(X=x)\Big)E[Y] = E[X]\cdot E[Y]
\end{aligned}$$

Pentru cazul continuu demonstraţia este similară:

$$\begin{aligned}
E[XY] &= \int_x\int_y xy\; p(X=x,Y=y)dydx \\
&\overset{indep.}{=} \int_x\int_y xy\; p(X=x)\cdot p(Y=y)dydx \\
&= \int_x x\; p(X=x)\int_y y\; p(Y=y)dy\; dx \\
&= \int_x x\; p(X=x)E[Y]dx = E[Y]\cdot\int_x x\; p(X=x)\; dx \\
&= E[Y]\cdot E[X] = E[X]\cdot E[Y]
\end{aligned}$$

11. **(Covarianţa nulă nu implică în mod neapărat independenţa variabilelor aleatoare)**

CMU, 2009 spring, Tom Mitchell, HW2, pr. 1.3.2
CMU, 2009 fall, Geoff Gordon, HW1, pr. 3.1

a. Reciproca afirmaţiei din problema 10 nu este în general adevărată, deci $Cov(X,Y)=0 \not\Rightarrow X$ şi Y sunt independente. Arătaţi aceasta folosind ca exemplu variabilele aleatoare ale căror distribuţii sunt date în tabelul alăturat.

X	Y	$P(X,Y)$
0	0	1/3
1	0	0
2	0	1/3
0	1	0
1	1	1/3
2	1	0

b. Totuşi, dacă X şi Y sunt variabile aleatoare binare luând valori în mulţimea $\{0,1\}$, iar $E[XY]=E[X]\cdot E[Y]$, atunci X şi Y sunt independente. Justificaţi.

(Aşadar, două variabile aleatoare binare cu valori în mulţimea $\{0,1\}$ sunt independente atunci şi numai atunci când au covarianţa nulă.)

Răspuns:

a. Din datele furnizate în exemplul a rezultă următoarele probabilităţi:

$$\begin{array}{lll}
P(X=0)=1/3 & P(Y=0)=2/3 & P(XY=0)=2/3 \\
P(X=1)=1/3 & P(Y=1)=1/3 & P(XY=1)=1/3 \\
P(X=2)=1/3 & & P(XY=2)=0.
\end{array}$$

Putem calcula mediile acestor variabile aleatoare folosind formula de definiţie pentru variabile discrete $E[X]=\sum_x x\cdot P(X=x)$:

$$\begin{aligned} E[X] &= 0 \cdot \frac{1}{3} + 1 \cdot \frac{1}{3} + 2 \cdot \frac{1}{3} = 1 \\ E[Y] &= 0 \cdot \frac{2}{3} + 1 \cdot \frac{1}{3} = \frac{1}{3} \\ E[XY] &= 0 \cdot \frac{2}{3} + 1 \cdot \frac{1}{3} + 2 \cdot 0 = \frac{1}{3}. \end{aligned}$$

Conform formulei care a fost demonstrată la problema 10, covarianţa variabilelor X şi Y este: $Cov(X,Y) = E[XY] - E[X] \cdot E[Y] = \frac{1}{3} - 1 \cdot \frac{1}{3} = 0$. Cu toate acestea, variabilele X şi Y nu sunt independente. Într-adevăr, pentru $X = 0$ şi $Y = 0$ se observă că $P(X = 0, Y = 0) = 1/3$ dar $P(X = 0)P(Y = 0) = 1/3 \cdot 2/3 = 2/9 \neq \frac{1}{3}$.

Prin acest contraexemplu am arătat că implicaţia „$Cov(X,Y) = 0 \Rightarrow X$ şi Y sunt independente“ nu este în general adevărată.

b. În continuare vom demonstra că în cazul în care X şi Y iau valori în mulţimea $\{0,1\}$ implicaţia de mai sus este adevărată.

Dacă X şi Y sunt variabile aleatoare binare, atunci:

$$\begin{aligned} E[X] &= 0 \cdot P(X = 0) + 1 \cdot P(X = 1) = P(X = 1) \\ E[Y] &= P(Y = 1) \\ E[XY] &= P(X = 1, Y = 1). \end{aligned}$$

Covarianţa nulă înseamnă — conform problemei 9.c — că $E[XY] = E[X] \cdot E[Y]$, adică:

$$P(X = 1, Y = 1) = P(X = 1)P(Y = 1).$$

Însă, a demonstra independenţa variabilelor aleatoare X şi Y revine la a arăta că $P(X,Y) = P(X)P(Y)$ pentru toate combinaţiile posibile de valori ale variabilelor. Unul din cazuri ($X = 1, Y = 1$) este deja demonstrat, deci mai există încă 3 cazuri. Pentru acestea vom utiliza formulele: $P(A \cap B) = P(A) - P(A \cap \bar{B})$ şi $P(\bar{A}) = 1 - P(A)$.

Cazul $X = 1, Y = 0$:

$$\begin{aligned} P(X = 1, Y = 0) &= P(X = 1) - P(X = 1, Y = 1) \\ &= P(X = 1) - P(X = 1)P(Y = 1) \\ &= P(X = 1)(1 - P(Y = 1)) \\ &= P(X = 1)P(Y = 0). \end{aligned}$$

Cazul $X = 0, Y = 1$ se tratează similar cu cazul anterior:

$$\begin{aligned} P(X = 0, Y = 1) &= P(Y = 1) - P(X = 1, Y = 1) \\ &= P(Y = 1) - P(X = 1)P(Y = 1) \\ &= P(Y = 1)(1 - P(X = 1)) \\ &= P(Y = 1)P(X = 0). \end{aligned}$$

Cazul $X = 0, Y = 0$:

$$\begin{aligned} P(X = 0, Y = 0) &= P(X = 0) - P(X = 0, Y = 1) \\ &= P(X = 0) - P(X = 0)P(Y = 1) \\ &= P(X = 0)(1 - P(Y = 1)) \\ &= P(X = 0)P(Y = 0). \end{aligned}$$

Prin urmare, egalitatea $P(X,Y) = P(X)P(Y)$ este adevărată pentru toate cazurile, deci variabilele X şi Y sunt independente.

12. **(Variabile aleatoare: calcul de medii)**

Fie X o variabilă aleatoare pentru care $E(X) = \mu$ şi $Var(X) = \sigma^2$.

a. ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW1, pr. 2.4***

Cât este $E[X(X-1)]$ în funcţie de μ şi σ?

b. ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, midterm, pr. 1.c***

Fie $c \in R$. Care din următoarele variante sunt adevărate?

A. $E[(X-c)^2] = (\mu - c)^2 + \sigma^2$
B. $E[(X-c)^2] = (\mu - c)^2$
C. $E[(X-c)^2] = (\mu - c)^2 - \sigma^2$
D. $E[(X-c)^2] = (\mu - c)^2 + 2\sigma^2$
E. $E[(X-c)^2] = \mu^2 + c^2 + 2\sigma^2$
F. $E[(X-c)^2] = \mu^2 + c^2 - 2\sigma^2$.

Răspuns:

a. De la problema 9.a ştim că media sumei a două variabile aleatoare este suma mediilor variabilelor respective. De asemenea, este imediat demonstrabil că $E[c \cdot X] = c \cdot E[X]$, unde X, Y sunt variabile aleatoare, iar c este o constantă. Ţinând cont că $Var(X) = E[X^2] - (E[X])^2$ (vedeţi problema 9.b), putem scrie:

$$\begin{aligned} E[X(X-1)] &= E[X^2 - X] = E[X^2] - E[X] \\ &= E[X^2] - (E[X])^2 + (E[X])^2 - E[X] \\ &= Var(X) + (E[X])^2 - E[X] \\ &= \sigma^2 + \mu^2 - \mu \end{aligned}$$

b. Pentru a găsi varianta adevărată vom calcula $E[(X-c)^2]$:

$$\begin{aligned} E[(X-c)^2] &= E[X^2 - 2cX + c^2] = E[X^2] - 2cE[X] + c^2 \\ &= E[X^2] - (E[X])^2 + (E[X])^2 - 2cE[X] + c^2 \\ &= \sigma^2 + \mu^2 - 2c\mu + c^2 \\ &= \sigma^2 + (\mu - c)^2 \end{aligned}$$

Deci varianta A este adevărată. (Toate celelalte variante sunt, în general, false.)

13. **(Variabile aleatoare discrete: distribuţii de probabilitate comune, marginale, condiţionate; regula de înlănţuire)**

CMU, 2002 fall, Andrew Moore, final exam, pr. 4.a

Considerăm un set de date definit cu ajutorul a 3 variabile aleatoare cu valori booleene X, Y şi Z. Care dintre seturile de informaţii de mai jos sunt suficiente pentru a specifica distribuţia comună $P(x,y,z)$?

A.	**B.**	**C.**	**D.**
$P(\neg X \mid Z)$	$P(\neg X \mid \neg Z)$	$P(X \mid Z)$	$P(X \mid Z)$
$P(\neg X \mid \neg Z)$	$P(X \mid \neg Z)$	$P(X \mid \neg Z)$	$P(X \mid \neg Z)$
$P(\neg Y \mid X, Z)$	$P(Y \mid X, Z)$	$P(Y \mid X, Z)$	$P(Y \mid X, Z)$
$P(\neg Y \mid X, \neg Z)$	$P(Y \mid X, \neg Z)$	$P(Y \mid X, \neg Z)$	$P(Y \mid X, \neg Z)$
$P(\neg Y \mid \neg X, Z)$	$P(Y \mid \neg X, Z)$	$P(Y \mid \neg X, Z)$	$P(\neg Y \mid \neg X, \neg Z)$
$P(\neg Y \mid \neg X, \neg Z)$	$P(Y \mid \neg X, \neg Z)$	$P(\neg Y \mid \neg X, \neg Z)$	$P(Y \mid \neg X, \neg Z)$
$P(Z)$	$P(Z)$	$P(\neg Z)$	$P(Z)$

Răspuns:

Pentru a calcula distribuţia comună a mai multor variabile aleatoare se poate aplica regula de înlănţuire.[21] În cazul nostru, putem scrie:

$$P(X, Y, Z) = P(Z) \cdot P(X \mid Z) \cdot P(Y \mid X, Z)$$

Deoarece variabilele aleatoare X, Y şi Z au valori booleene, pentru a specifica distribuţia comună $P(X, Y, Z)$ este nevoie să se calculeze valoarea acesteia în fiecare din cele 8 cazuri posibile:

$P(X, Y, Z)$	$P(X, Y, \neg Z)$	$P(X, \neg Y, Z)$	$P(X, \neg Y, \neg Z)$
$P(\neg X, Y, Z)$	$P(\neg X, Y, \neg Z)$	$P(\neg X, \neg Y, Z)$	$P(\neg X, \neg Y, \neg Z)$

Cunoaştem de asemenea relaţii de calcul de forma:

$$\begin{aligned} P(\neg X) &= 1 - P(X) \\ P(\neg X \mid Y) &= 1 - P(X \mid Y) \end{aligned}$$

Aşadar, pentru a aplica regula de înlănţuire de mai sus este nevoie să cunoaştem

- $P(Z)$ **sau** $P(\neg Z)$**;**
- $P(X \mid Z)$ **sau** $P(\neg X \mid Z)$**;**
- $P(X \mid \neg Z)$ **sau** $P(\neg X \mid \neg Z)$**;**
- $P(Y \mid X, Z)$ **sau** $P(\neg Y \mid X, Z)$**;**
- $P(Y \mid \neg X, Z)$ **sau** $P(\neg Y \mid \neg X, Z)$**;**
- $P(Y \mid X, \neg Z)$ **sau** $P(\neg Y \mid X, \neg Z)$**;**
- $P(Y \mid \neg X, \neg Z)$ **sau** $P(\neg Y \mid \neg X, \neg Z)$**.**

Cu aceste precizări, putem specifica pentru fiecare dintre seturile de informaţii din enunţ dacă sunt suficiente pentru a calcula distribuţia comună $P(X, Y, Z)$.

Cazul A. Da. Se observă că putem calcula $P(X \mid Z)$ şi $P(X \mid \neg Z)$ din primele două probabilităţi din enunţ. De asemenea, utilizând următoarele 4 probabilităţi se pot deduce: $P(Y \mid X, Z)$, $P(Y \mid X, \neg Z)$, $P(Y \mid \neg X, Z)$ şi $P(Y \mid \neg X, \neg Z)$. Iar din $P(Z)$ se obţine $P(\neg Z)$. Prin urmare, există toate informaţiile necesare distribuţiei comune $P(X, Y, Z)$.

Cazul B. Nu, informaţiile din enunţ nu sunt suficiente. Nu putem deduce valoarea pentru $P(X \mid Z)$.

[21] Pentru variabile aleatoare, regula de înlănţuire

$$P(A_1, A_2, \ldots, A_n) = P(A_1) \cdot P(A_2 \mid A_1) \cdot P(A_n \mid A_1, A_2, \ldots, A_{n-1})$$

se demonstrează imediat pornind de la regula de înlănţuire pentru (probabilităţi de) evenimente aleatoare. A se vedea problema 14.

Cazul C. Da, informaţiile din enunţ sunt suficiente. Din $P(X \mid Z)$ se obţine $P(\neg X \mid Z)$, iar din $P(X \mid \neg Z)$ se obţine $P(\neg X \mid \neg Z)$. Din următoarele 4 probabilităti se obţin celelalte 4 necesare pentru $P(Y \mid X, Z)$, şi anume: $P(\neg Y \mid X, Z)$, $P(\neg Y \mid X, \neg Z)$, $P(\neg Y \mid \neg X, Z)$ şi respectiv $P(Y \mid \neg X, \neg Z)$.

Cazul D. Nu, informaţiile din enunţ nu sunt suficiente. Nu putem deduce valoarea pentru $P(Y \mid \neg X, Z)$.

14. **(Variabile aleatoare discrete: regula de înmulţire, varianta condiţională)**

CMU, 2009 fall, Geoff Gordon, HW2, pr. 2.1

Arătaţi că pentru orice valori x, y şi z ale variabilelor aleatoare X, Y şi Z respectiv, avem:

$$P(X = x, Y = y \mid Z = z) = P(X = x \mid Y = y, Z = z) \cdot P(Y = y \mid Z = z).$$

În notaţie simplificată: $P(X, Y \mid Z) = P(X \mid Y, Z) \cdot P(Y \mid Z)$.

***Indicaţie*: Folosiţi regula de înlănţuire pentru evenimente aleatoare.**

Răspuns:

Folosind definiţia probabilităţii condiţionate şi regula de înlănţuire (cu termenii ordonaţi în mod convenabil), egalitatea cerută se obţine astfel:

$$\begin{aligned} P(X, Y \mid Z) &\overset{def.}{=} \frac{P(X, Y, Z)}{P(Z)} \overset{not.}{=} \frac{P(Z \cap Y \cap X)}{P(Z)} \\ &= \frac{P(Z)P(Y \mid Z)P(X \mid Y, Z)}{P(Z)} = P(Y \mid Z) \cdot P(X \mid Y, Z) \end{aligned}$$

***Observaţie*: O altă metodă de rezolvare constă în a aplica pentru fiecare membru al egalităţii din enunţ definiţia probabilităţii condiţionate, după simplificări obţinându-se pentru ambii membri aceeaşi valoare:**

$$\begin{aligned} P(X, Y \mid Z) &= \frac{P(X, Y, Z)}{P(Z)} \\ P(X \mid Y, Z) \cdot P(Y \mid Z) &= \frac{P(X, Y, Z)}{P(Y, Z)} \cdot \frac{P(Y, Z)}{P(Z)} = \frac{P(X, Y, Z)}{P(Z)} \end{aligned}$$

15. **(Variabile aleatoare discrete: independenţă condiţională)**

CMU, 2005 fall, T. Mitchell, A. Moore, midterm, pr. 4

Fie variabilele aleatoare discrete A, B şi C având distribuţia comună conform tabelului de mai jos.

a. Este variabila A independentă condiţional de B în raport cu variabila C?

b. Dacă aţi răspuns afirmativ la întrebarea a, faceţi o schimbare în primele două linii ale tabelului de mai sus pentru a obţine o distribuţie pentru care răspunsul la aceeaşi întrebare să devină negativ. Invers, dacă aţi răspuns negativ la întrebarea a, faceţi o schimbare în primele două linii ale tabelului încât răspunsul să devină afirmativ.

A	B	C	$P(A, B, C)$
0	0	0	1/8
0	0	1	1/8
0	1	0	1/8
0	1	1	1/8
1	0	0	1/8
1	0	1	1/8
1	1	0	1/8
1	1	1	1/8

Răspuns:

a. Faptul că variabila A este independentă condiţional de B în raport cu variabila C se mai notează prin $A \perp B \mid C$ şi poate fi demonstrat prin una din următoarele două relaţii:

$$P(A = a, B = b \mid C = c) = P(A = a \mid C = c) \cdot P(B = b \mid C = c) \text{ sau}$$
$$P(A = a \mid B = b, C = c) = P(A = a \mid C = c), \text{ dacă } P(B = b, C = c) \neq 0.$$

pentru orice $a \in Val(A), b \in Val(B), c \in Val(C)$. Deşi în general se foloseşte prima relaţie, pentru acest exerciţiu este mai uşor să utilizăm cea de-a doua relaţie. Conform tabelului dat în enunţ, rezultă imediat că $P(B = b, C = c) \neq 0$ pentru orice $b, c \in \{0, 1\}$. Vom demonstra că pentru orice $a, b, c \in \{0, 1\}$ este adevărat că $P(A = a \mid B = b, C = c) = P(A = a \mid C = c)$. Cele două probabilităţi condiţionate vor fi calculate folosind datele din tabel:

Cazul $(0, 0, 0)$: $P(A = 0 \mid B = 0, C = 0) = \dfrac{1 \cdot \frac{1}{8}}{2 \cdot \frac{1}{8}} = \dfrac{1}{2}$ **şi** $P(A = 0 \mid C = 0) = \dfrac{2 \cdot \frac{1}{8}}{4 \cdot \frac{1}{8}} = \dfrac{1}{2}$.

Cazul $(0, 0, 1)$: $P(A = 0 \mid B = 0, C = 1) = \dfrac{1 \cdot \frac{1}{8}}{2 \cdot \frac{1}{8}} = \dfrac{1}{2}$ **şi** $P(A = 0 \mid C = 1) = \dfrac{2 \cdot \frac{1}{8}}{4 \cdot \frac{1}{8}} = \dfrac{1}{2}$.

Se observă că pentru toate celelalte cazuri se obţin aceleaşi valori, deci

$$P(A = a \mid B = b, C = c) = P(A = a \mid C = c), \ \forall a, b, c \in \{0, 1\}.$$

Aşadar, variabila A este independentă condiţional de B în raport cu variabila C.

b. Schimbarea trebuie făcută în aşa fel încât să se păstreze relaţia $\sum P(A, B, C) = 1$. O variantă posibilă este:

A	B	C	$P(A, B, C)$
0	0	0	1/4
0	0	1	0
	...		...

Pentru aceste noi valori se observă că: $P(A = 0 \mid B = 0, C = 0) = \dfrac{1 \cdot 1/4}{1 \cdot 1/4 + 1 \cdot 1/8} = \dfrac{2}{3}$ **şi** $P(A = 0 \mid C = 0) = \dfrac{1 \cdot 1/4 + 1 \cdot 1/8}{1 \cdot 1/4 + 3 \cdot 1/8} = \dfrac{3}{5}$. **Este suficient un singur caz în care probabilităţile respective nu sunt egale, prin urmare variabila A nu este independentă condiţional de variabila B în raport cu a treia variabilă, C.**

16. **(Variabile aleatoare discrete: distribuţii comune, distribuţii marginale, distribuţii condiţionale; independenţă, independenţă condiţională)**

• ***CMU, 2016 fall, N. Balcan, M. Gormley, HW2, pr. 1.4***

Fie trei variabile aleatoare X, Y şi Z care iau valori în mulţimea $\{0, 1\}$. În tabelul următor este dată distribuţia probabilistă comună a acestor trei variabile, $P(X, Y, Z)$.

	$Z=0$		$Z=1$	
	$X=0$	$X=1$	$X=0$	$X=1$
$Y=0$	$1/24$	$1/12$	$1/12$	$5/24$
$Y=1$	$1/12$	p	q	$7/24$

a. Considerând că X şi Y sunt independente, găsiţi valorile lui p şi q.

***Indicaţie*: Este util (deşi nu obligatoriu) să calculaţi mai întâi $P(X,Y)$, distribuţia comună a variabilelor X şi Y, completând tabelul alăturat, după care veţi calcula şi distribuţiile (marginale) pentru X şi Y, de preferinţă pe o linie suplimentară şi respectiv pe o coloană suplimentară la acest tabel.**

	$X=0$	$X=1$
$Y=0$		
$Y=1$		

b. Considerând valorile lui p şi q determinate la punctul a, sunt X şi Y independente condiţional în raport cu Z? De ce?

Răspuns:

a. Conform definiţiei, variabilele aleatoare X şi Y sunt independente dacă şi numai dacă

$$P(X=x,Y=y)=P(X=x)\cdot P(Y=y),\forall x\in Val\,(X),\forall y\in Val\,(Y). \tag{7}$$

***Observaţie*: Evident, pentru a determina p şi q, ar fi de dorit să scriem un sistem de două ecuaţii cu aceste două necunoscute. Cele două ecuaţii pot fi obţinute prin particularizarea relaţiei (7) pentru două perechi distincte de valori $(x,y)\in Val(X)\times Val(Y)$. Alternativ, va fi suficient să reţinem doar o astfel de ecuaţie, pentru că în locul celeilalte putem folosi una din proprietăţile din definiţia funcţiei de probabilitate, şi anume**

$$\sum_{x,y,z} P(X=x,Y=y,Z=z)=1. \tag{8}$$

Aceasta revine practic la a însuma elementele din tabelul dat în enunţ pentru distribuţia $P(X,Y,Z)$.[22]

Pentru a calcula probabilităţile marginale $P(X=x,Y=y)$, vom proceda conform definiţiei: $P(X=x,Y=y)=\sum_{z\in Val\,(Z)}P(X=x,Y=y,Z=z)$. Şi anume:

$$\begin{aligned}
P(X=0,Y=0) &= P(X=0,Y=0,Z=0)+P(X=0,Y=0,Z=1)=\frac{1}{24}+\frac{1}{12}=\frac{1}{8}\\
P(X=0,Y=1) &= P(X=0,Y=1,Z=0)+P(X=0,Y=1,Z=1)=\frac{1}{12}+q\\
P(X=1,Y=0) &= P(X=1,Y=0,Z=0)+P(X=1,Y=0,Z=1)=\frac{1}{12}+\frac{5}{24}=\frac{7}{24}\\
P(X=1,Y=1) &= P(X=1,Y=1,Z=0)+P(X=1,Y=1,Z=1)=\frac{7}{24}+p.
\end{aligned}$$

În mod similar,

[22] Remarcăm că, dacă vom urma *Indicaţia* din enunţ, după ce vom scrie probabilităţile marginale $P(X,Y)$, $P(X)$ şi $P(Y)$, va fi uşor să scriem încă alte trei proprietăţi similare cu (8). Deci, o vom putea alege atunci pe cea mai „directă"/simplă dintre ele.

$$P(X=0) = \sum_{y\in Val\,(Y)} P(X=x,Y=y) = \frac{1}{8}+\frac{1}{12}+q = \frac{5}{24}+q$$
$$P(X=1) = \frac{7}{24}+p+\frac{7}{24} = \frac{7}{12}+p$$

şi

$$P(Y=0) = \sum_{x\in Val\,(X)} P(X=x,Y=y) = \frac{1}{8}+\frac{7}{24} = \frac{5}{12}$$
$$P(Y=1) = 1-P(Y=0) = \frac{7}{12}.$$

Punând acum toate aceste rezultate împreună, obţinem:

	$X=0$	$X=1$	
$Y=0$	$1/8$	$7/24$	$5/12$
$Y=1$	$1/12+q$	$7/24+p$	$7/12$
	$5/24+q$	$7/12+p$	

Acum este simplu de văzut că $5/24+q+p+14/24=1 \Leftrightarrow p+q=5/24$**. (Aceasta constituie prima noastră ecuaţie în** p **şi** q**.) În consecinţă, am putea chiar să eliminăm** q **din tabelul de mai sus, scriind** $P(X=0,Y=1)=1/12+q=1/12+5/24-p=7/24-p$ **şi** $P(X=0)=5/24+q=5/24+5/24-p=5/12-p$**.**

Aplicând definiţia independenţei variabilelor X **şi** Y **pentru** $x=0$ **şi** $y=0$**, vom avea:**

$$P(X=0,Y=0)=P(X=0)\cdot P(Y=0) \Leftrightarrow \frac{1}{8}=\left(\frac{5}{12}-p\right)\cdot\frac{5}{12} \Leftrightarrow \frac{5}{12}-p=\frac{3}{2}\cdot\frac{1}{5}$$
$$\Leftrightarrow p=\frac{5}{12}-\frac{3}{10}=\frac{25-18}{60}=\frac{7}{60},$$

de unde rezultă imediat

$$q=\frac{5}{24}-p=\frac{5}{24}-\frac{7}{60}=\frac{25-14}{120}=\frac{11}{120}.$$

De asemenea, vom avea $P(X=0)=5/12-p=5/12-7/60=3/10$ **şi** $P(X=1)=7/10$**.**

În final, va trebui să verificăm şi celelalte trei egalităţi din definiţia independenţei lui X **şi** Y**, pentru că, în general, este posibil ca sistemul de ecuaţii corespunzător relaţiei (7) să fie supra-restricţionat (deci, incompatibil):**[23]

$P(X=0,Y=1)=P(X=0)\cdot P(Y=1)$ **:**

$$\frac{7}{24}-p=\frac{3}{10}\cdot\frac{7}{12} \Leftrightarrow \frac{7}{24}-\frac{7}{60}=\frac{7}{40} \Leftrightarrow 7\cdot\left(\frac{1}{24}-\frac{1}{60}\right)=\frac{7}{40} \Leftrightarrow 7\cdot\frac{5-2}{120}=\frac{7}{40}\ (A)$$

$P(X=1,Y=0)=P(X=1)\cdot P(Y=0)$ **:**

$$\frac{7}{24}=\frac{7}{10}\cdot\frac{5}{12}\ (A)$$

[23] Ar fi suficient chiar să verificăm doar două dintre cele trei egalităţi, fiindcă ultima decurge automat din celelalte, ţinând cont de proprietăţile $\sum_{x,y}P(X=x,Y=y)=1$, $\sum_x P(X=x)=1$ şi $\sum_y P(Y=y)=1$.

$P(X=1,Y=1)=P(X=1)\cdot P(Y=1)$ **:**

$$\frac{7}{24}+p=\frac{7}{10}\cdot\frac{7}{12}\Leftrightarrow\frac{7}{24}+\frac{7}{60}=7^2\cdot\frac{1}{10}\cdot\frac{1}{12}\Leftrightarrow\frac{1}{24}+\frac{1}{60}=\frac{7}{120}\Leftrightarrow\frac{5+2}{120}=\frac{7}{120}\ (A)$$

Aşadar, date fiind cele două valori ale lui p şi q, variabilele X şi Y sunt independente.

b. Conform definiţiei, variabilele aleatoare X şi Y sunt independente condiţional în raport cu variabila Z dacă şi numai dacă

$$P(X=x,Y=y|Z=z)=P(X=x|Z=z)\cdot P(Y=y|Z=z),\quad \forall x\in Val(X),\forall y\in Val(Y),\forall z\in Val(Z). \tag{9}$$

Dacă X şi Y *nu* sunt independente condiţional în raport cu Z, atunci $\exists x\in Val(X),\exists y\in Val(Y),\exists z\in Val(Z)$ astfel încât $P(X=x,Y=y|Z=z)\neq P(X=x|Z=z)\cdot P(Y=y|Z=z)$.

Pentru a putea exprima probabilităţile condiţionate care apar în formulele de mai sus, vom calcula în prealabil distribuţia marginală a lui Z pornind de la tabelul din enunţ:

$$\begin{aligned}P(Z=0) &= \sum_{x\in Val(X)}\sum_{y\in Val(Y)}P(X=x,Y=y,Z=z)\\ &= 5/24+p=5/24+7/60=39/120=13/40\\ P(Z=1) &= 1-P(Z=0)=27/40.\end{aligned}$$

Vom verifica relaţia (9) pentru $x=y=z=0$:

$$\begin{aligned}&P(X=0,Y=0|Z=0)=P(X=0|Z=0)\cdot P(Y=0|Z=0)\\ \Leftrightarrow\ &\frac{P(X=0,Y=0,Z=0)}{P(Z=0)}=\frac{P(X=0,Z=0)}{P(Z=0)}\cdot\frac{P(Y=0,Z=0)}{P(Z=0)}\\ \Leftrightarrow\ &P(X=0,Y=0,Z=0)\cdot P(Z=0)=P(X=0,Z=0)\cdot P(Y=0,Z=0)\\ \Leftrightarrow\ &\frac{1}{24}\cdot\frac{13}{40}=\left(\frac{1}{24}+\frac{1}{12}\right)\cdot\left(\frac{1}{24}+\frac{1}{12}\right)\\ \Leftrightarrow\ &\frac{13}{24\cdot 40}=\frac{1}{8}\cdot\frac{1}{8}\Leftrightarrow\frac{13}{3\cdot 5}=1\ (F)\end{aligned}$$

Prin urmare, variabilele X şi Y nu sunt independente condiţional în raport cu variabila Z.

***Observaţie*: Această problemă pune în evidenţă următoarea *proprietate*, care merită a fi reţinută: independenţa a două variabile aleatoare nu implică (în general) independenţa lor condiţională în raport cu o a treia variabilă aleatoare.**

17. **(Variabile aleatoare continue: funcţia densitate de probabilitate)**

CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW1, pr. 1.5

Fie variabila aleatoare continuă X a cărei funcţie densitate de probabilitate (în limba engleză: "probability density functions"; scris, sub formă abreviată, p.d.f.) este:

$$p(x) = \begin{cases} cx^2 & \textbf{pentru } 1 \leq x \leq 2 \\ 0 & \textbf{în caz contrar}. \end{cases}$$

a. Ţinând cont de proprietăţile funcţiei densitate de probabilitate (a se vedea notiţele de la curs), cât trebuie să fie valoarea constantei c?

b. Desenaţi graficul funcţiei de mai sus.

c. Calculaţi $P(X > 3/2)$.

Răspuns:

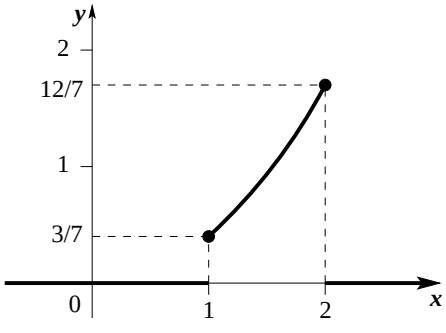
a. Faptul că $p(x)$ este funcţie densitate de probabilitate pentru variabila aleatoare continuă X înseamnă că $p(x) \geq 0, \forall x$ şi că $\int_{-\infty}^{+\infty} p(x)dx = 1$. Pentru a afla constanta c vom calcula valoarea integralei:

$$\begin{aligned} \int_{-\infty}^{+\infty} p(x)dx &= \underbrace{\int_{-\infty}^{1} p(x)dx}_{=0} + \int_1^2 p(x)dx + \underbrace{\int_2^{+\infty} p(x)dx}_{=0} \\ &= \int_1^2 cx^2dx = c \cdot \int_1^2 x^2dx = c \cdot \frac{x^3}{3}\Big|_1^2 = c\left(\frac{2^3}{3} - \frac{1^3}{3}\right) = c \cdot \frac{7}{3} \end{aligned}$$

Prin urmare, $c \cdot \frac{7}{3} = 1 \Rightarrow c = \frac{3}{7}$.

b. Trebuie să reprezentăm grafic funcţia:

$$p(x) = \begin{cases} \frac{3}{7}x^2 & \textbf{pentru } 1 \leq x \leq 2 \\ 0 & \textbf{în caz contrar}. \end{cases}$$



În intervalul $[1, 2]$, această funcţie este un fragment din parabola corespunzătoare funcţiei de gradul al doilea: $\frac{3}{7}x^2$, parabolă care are vârful în punctul $(0, 0)$. Putem calcula $p(1) = 3/7 \approx 0.42$ şi $p(2) = 12/7 \approx 1.71$.

c. Valoarea probabilităţii cerute se poate calcula astfel:

$$\begin{aligned} P(X > 3/2) &\overset{not.}{=} P(\{\omega \mid X(\omega) > \frac{3}{2}\}) \overset{def.}{=} \int_{X(\omega)=x>3/2} p(x)dx = \int_{3/2}^{+\infty} p(x)dx \\ &= \int_{3/2}^{2} p(x)dx + \underbrace{\int_2^{+\infty} p(x)dx}_{=0} = \int_{3/2}^{2} \frac{3}{7}x^2dx = \frac{3}{7} \cdot \int_{3/2}^{2} x^2dx \\ &= \frac{3}{7} \cdot \frac{x^3}{3}\Big|_{3/2}^{2} = \frac{1}{7} \cdot x^3\Big|_{3/2}^{2} = \frac{1}{7} \cdot \left(8 - \frac{27}{8}\right) = \frac{1}{7} \cdot \frac{64 - 27}{8} = \frac{37}{56}. \end{aligned}$$

O altă variantă de rezolvare este bazată pe folosirea *funcţiei cumulative de distribuţie*, care, după cum ştim, se defineşte prin relaţia $F(x) \overset{def.}{=} P(X \leq x)$, pentru orice $x \in \mathbb{R}$:

$$\begin{aligned} P(X > 3/2) &= 1 - P(X \leq 3/2) = 1 - F\left(\frac{3}{2}\right) = 1 - \int_{-\infty}^{3/2} p(x)dx \\ &= 1 - \left(\int_{-\infty}^{1} \underbrace{p(x)}_{0} dx + \int_{1}^{3/2} p(x)dx\right) = 1 - \left(0 + \int_{1}^{3/2} p(x)dx\right) \\ &= 1 - \int_{1}^{3/2} p(x)dx = 1 - \int_{1}^{3/2} \frac{3}{7}x^2 dx = 1 - \frac{1}{7} \cdot x^3\Big|_1^{3/2} \\ &= 1 - \frac{1}{7} \cdot \left(\frac{27}{8} - 1\right) = 1 - \frac{1}{7} \cdot \frac{19}{8} \\ &= 1 - \frac{19}{56} = \frac{37}{56}. \end{aligned}$$

18. **(Variabile aleatoare continue: funcţia densitate de probabilitate)**

CMU, 2008 spring, Eric Xing, HW1, pr. 1.1.b

Fie funcţia

$$p(x) = \begin{cases} cx^{-d} & \textbf{pentru } x > 1 \\ 0 & \textbf{în caz contrar}. \end{cases}$$

Care sunt valorile posibile pentru c şi d în aşa fel ca p să poată reprezenta o funcţie densitate de probabilitate?

Răspuns:

O funcţie p poate reprezenta o funcţie densitate de probabilitate dacă $p(x) \geq 0$ pentru $\forall x$, şi $\int_{-\infty}^{\infty} p(x)dx = 1$. Vom folosi aceste două condiţii pentru a calcula valorile posibile pentru c şi d.

Prima condiţie $p(x) \geq 0$ implică faptul că $c \geq 0$, asupra lui d neimpunând nicio restricţie. Mai mult, ţinând cont de forma lui p şi de cea de-a doua condiţie, vom avea chiar $c > 0$, fiindcă $c = 0$ ar implica $\int_{-\infty}^{\infty} p(x)dx = 0 \neq 1$.

Pentru a aplica cea de-a doua condiţie, calculăm integrala:

$$\int_{-\infty}^{\infty} p(x)dx = \int_{1}^{\infty} cx^{-d}dx = c\int_{1}^{\infty} x^{-d}dx.$$

Vom avea de tratat două cazuri:

$$\textbf{Cazul 1:} \quad d = 1 \quad \Rightarrow c\int_{1}^{\infty} x^{-d}dx = c\int_{1}^{\infty} \frac{1}{x}dx = c \cdot \ln x\Big|_1^{\infty} = \infty$$

$$\textbf{Cazul 2:} \quad d \neq 1 \quad \Rightarrow c\int_{1}^{\infty} x^{-d}dx = c \cdot \frac{x^{-d+1}}{-d+1}\Big|_1^{\infty} = \frac{c}{1-d} \cdot x^{1-d}\Big|_1^{\infty}$$

Subcazul $d > 1$: $x^{1-d}\Big|_1^{\infty} = 0 - 1 = -1$. Deci în acest caz $c\int_1^{\infty} x^{-d}dx = \frac{c}{d-1}$.

Subcazul $d < 1$: $x^{1-d}\Big|_1^{\infty} = +\infty$. Deci în acest caz $c\int_1^{\infty} x^{-d}dx = +\infty$.

Prin urmare, $\int_{-\infty}^{\infty} p(x)dx = 1 \Rightarrow d > 1$ **şi** $\dfrac{c}{d-1} = 1$**.**

În concluzie, p **poate reprezenta o funcţie densitate de probabilitate dacă**

$$c > 0,\ d > 1 \text{ şi } c = d - 1.$$

Referitor la aceste trei restricţii, se observă că ultimele două o implică pe cea dintâi, deci o fac superfluă.

19. (Coeficientul de corelaţie pentru două variabile aleatoare: două proprietăţi)

Liviu Ciortuz, 2019, după
■ · ***Sheldon Ross, A First Course in Probability, 5th ed., Prentice Hall, 1997, pag. 332***

Pentru două variabile aleatoare oarecare X **şi** Y **având** $Var\,(X) \neq 0$ **şi** $Var\,(Y) \neq 0$**,** ***coeficientul de corelaţie*** **se defineşte astfel:**

$$\rho(X,Y) \overset{def.}{=} \frac{Cov(X,Y)}{\sigma_X\,\sigma_Y},$$

unde $\sigma_X \overset{not.}{=} \sqrt{Var(X)}$ **şi** $\sigma_Y \overset{not.}{=} \sqrt{Var(Y)}$ **sunt** ***deviaţiile standard*** **ale celor două variabile aleatoare.**

a. Să se demonstreze că $-1 \leq \rho(X,Y) \leq 1$**.**

Consecinţă**:** $Cov(X,Y) \in [-\sigma_X\,\sigma_Y, +\sigma_X\,\sigma_Y]$**.**

b. Să se arate că dacă $\rho(X,Y) = 1$ **(deci** $Cov(X,Y) = \sigma_X\,\sigma_Y$**), atunci** $Y = aX + b$**, cu** $a = \sigma_Y/\sigma_X > 0$**. Similar, dacă** $\rho(X,Y) = -1$ **(deci** $Cov(X,Y) = -\sigma_X\,\sigma_Y$**), atunci** $Y = aX + b$**, cu** $a = -\sigma_Y/\sigma_X < 0$**.**[24]

Indicaţii**:**

1. La punctul a**, pentru a demonstra inegalitatea** $\rho(X,Y) \geq -1$ **vă sugerăm să dezvoltaţi expresia** $Var\Big(\dfrac{X}{\sigma_X} + \dfrac{Y}{\sigma_Y}\Big)$ **folosind următoarele două proprietăţi, valabile pentru orice variabile aleatoare** X **şi** Y**:** $Var(X+Y) = Var(X) + Var(Y) + 2Cov(X,Y)$**,**[25] **şi** $Cov(aX,bY) = abCov(X,Y)$**, pentru orice** $a, b \in \mathbb{R}$**.**[26] **Apoi veţi proceda similar, dezvoltând expresia** $Var\Big(\dfrac{X}{\sigma_X} - \dfrac{Y}{\sigma_Y}\Big)$**, ca să demonstraţi inegalitatea** $\rho(X,Y) \leq 1$**.**

2. La punctul b**, veţi ţine cont de faptul că pentru o variabilă aleatoare oarecare** X**, avem** $Var(X) = 0$ **dacă şi numai dacă variabila** X **este constantă.**[27]

Răspuns:

[24] Aşadar, coeficientul de corelaţie reprezintă o „măsură“ a gradului de „dependenţă liniară“ dintre X şi Y. LC: Coeficientul a din relaţia $Y = aX + b$ nu poate lua valori în afara intervalului $[-\sigma_Y/\sigma_X, +\sigma_Y/\sigma_X]$ — aşa cum ne-am aştepta dacă facem legătura cu ecuaţia unei drepte oarecare din planul euclidian — din cauza simetriei $Cov(X,Y) = Cov(Y,X)$.

[25] Vedeţi rezolvarea problemei 23.c.

[26] Vedeţi *Consecinţa* de la problema 9.c.

[27] Mai precis, există $c \in \mathbb{R}$ astfel încât $P(X = c) = 1$, unde P este distribuţia de probabilitate considerată la definirea variabilelor din enunţul problemei.

a. Pentru a demonstra inegalitatea $\rho(X,Y) \geq -1$, procedăm conform *Indicaţiei* 1:

$$\begin{aligned} Var\Big(\frac{X}{\sigma_X}+\frac{Y}{\sigma_Y}\Big) &= Var\Big(\frac{X}{\sigma_X}\Big)+Var\Big(\frac{Y}{\sigma_Y}\Big)+2\,Cov\Big(\frac{X}{\sigma_X},\frac{Y}{\sigma_Y}\Big) \\ &= \frac{1}{\sigma_X^2}Var(X)+\frac{1}{\sigma_Y^2}Var(Y)+2\frac{1}{\sigma_X}\frac{1}{\sigma_Y}Cov(X,Y) \\ &= 1+1+2\rho(X,Y)=2[1+\rho(X,Y)]. \end{aligned}$$

Întrucât $Var(X) \geq 0$ pentru orice variabilă aleatoare X, rezultă că $Var\Big(\frac{X}{\sigma_X}+\frac{Y}{\sigma_Y}\Big) \geq 0$, deci $1+\rho(X,Y) \geq 0$, adică $\rho(X,Y) \geq -1$.

În mod similar, putem să arătăm că

$$Var\Big(\frac{X}{\sigma_X}-\frac{Y}{\sigma_Y}\Big)=2[1-\rho(X,Y)] \geq 0,$$

deci $\rho(X,Y) \leq 1$.

b. Dacă $\rho(X,Y)=-1$, atunci din primul calcul de la punctul a va rezulta că $Var\Big(\frac{X}{\sigma_X}+\frac{Y}{\sigma_Y}\Big)=0$. Ştim că aceasta se întâmplă dacă $\frac{X}{\sigma_X}+\frac{Y}{\sigma_Y}$ este o variabilă aleatoare constantă, adică, mai precis, există $a' \in \mathbb{R}$ astfel încât $\frac{X}{\sigma_X}+\frac{Y}{\sigma_Y}=a'$ cu probabilitate 1. Prin urmare, putem scrie $Y=a'\sigma_Y-\frac{\sigma_Y}{\sigma_X}X$. Rezultă că $Y=aX+b$, unde $a=-\frac{\sigma_Y}{\sigma_X}<0$ şi $b=a'\sigma_Y$.

În mod similar, dacă $\rho(X,Y)=1$, atunci din al doilea calcul de la punctul a va rezulta că $Var\Big(\frac{X}{\sigma_X}-\frac{Y}{\sigma_Y}\Big)=0$, deci există $a'' \in \mathbb{R}$ astfel încât $\frac{X}{\sigma_X}-\frac{Y}{\sigma_Y}=a''$ cu probabilitate 1. Aşadar, $Y=-a''\sigma_Y+\frac{\sigma_Y}{\sigma_X}X$. Renotând, obţinem $Y=aX+b$, cu $a=\frac{\sigma_Y}{\sigma_X}>0$ şi $b=-a''\sigma_Y$.

20. **(O proprietate: matricea de covarianţă a oricărui vector de variabile aleatoare este simetrică şi pozitiv semidefinită)**

■ □ ***prelucrare de Liviu Ciortuz, după "The Multivariate Gaussian Distribution", Chuong B. Do, 2008***

Fie variabilele aleatoare $X_1,\ldots,X_n$, cu $X_i:\Omega \to \mathbb{R}$ pentru $i=1,\ldots,n$. *Matricea de covarianţă* a *vectorului de variabile aleatoare* $X=(X_1,\ldots,X_n)$ este o matrice pătratică de dimensiune $n \times n$, ale cărei elemente se definesc astfel: $[Cov(X)]_{ij} \stackrel{def.}{=} Cov(X_i,X_j)$, pentru orice $i,j \in \{1,\ldots,n\}$.

Arătaţi că $\Sigma \stackrel{not.}{=} Cov(X)$ este matrice simetrică şi pozitiv semidefinită, cea de-a doua proprietate însemnând că pentru orice vector $z \in \mathbb{R}^n$ are loc inegalitatea $z^\top \Sigma z \geq 0$. (Vectorii $z \in \mathbb{R}^n$ sunt consideraţi vectori-coloană, iar simbolul $\top$ reprezintă operaţia de transpunere a matricelor.)

Răspuns:

Faptul că matricea Σ este simetrică decurge imediat din definiţia ei: dacă $X = (X_1, \ldots, X_n)$, atunci $[Cov(X)]_{i,j} \stackrel{def.}{=} Cov(X_i, X_j) = E[(X_i - E[X_i])(X_j - E[X_j])] = E[(X_j - E[X_j])(X_i - E[X_i])] = Cov(X_j, X_i) = [Cov(X)]_{j,i}$, pentru orice $i, j \in \{1, \ldots, n\}$.

Apoi, pentru orice vector $z \in \mathbb{R}^n$ de forma $z = (z_1, \ldots, z_n)^\top$, avem:

$$\begin{aligned} z^\top \Sigma z &= \sum_{i=1}^n z_i (\sum_{j=1}^n \Sigma_{ij} z_j) = \sum_{i=1}^n \sum_{j=1}^n (z_i \Sigma_{ij} z_j) = \sum_{i=1}^n \sum_{j=1}^n (z_i \, Cov[X_i, X_j] \, z_j) \\ &= \sum_{i=1}^n \sum_{j=1}^n (z_i \, E[(X_i - E[X_i])(X_j - E[X_j])] \, z_j) \\ &= E\left[\sum_{i=1}^n \sum_{j=1}^n z_i \, (X_i - E[X_i])(X_j - E[X_j]) \, z_j \right] \end{aligned}$$

Ultima dintre egalităţile de mai sus derivă din proprietatea de liniaritate a mediilor (vedeţi ex. 9.a). Mai departe,

$$\begin{aligned} z^\top \Sigma z &= E\Big[\Big(\sum_{i=1}^n z_i \, (X_i - E[X_i]) \Big) \Big(\sum_{j=1}^n (X_j - E[X_j]) \, z_j \Big) \Big] \\ &= E\Big[\Big(\sum_{i=1}^n (X_i - E[X_i]) \, z_i \Big) \Big(\sum_{j=1}^n (X_j - E[X_j]) \, z_j \Big) \Big] \\ &= E\Big[\Big(\sum_{i=1}^n (X_i - E[X_i]) \, z_i \Big)^2 \Big], \end{aligned}$$

ceea ce evident, reprezintă o cantitate nenegativă. Aşadar, $z^\top \Sigma z \geq 0$.

Să mai observăm că ultima expresie obţinută mai sus se scrie mai simplu / compact sub formă vectorială astfel:

$$E[((X - E[X])^\top \cdot z)^2].$$

21. **(Câteva inegalităţi de bază în teoria probabilităţilor: margini superioare pentru probabilităţi de forma $P(Z \geq t)$ şi $P(Z - E[Z] \geq t)$ şi unele consecinţe ale lor)**

□ • · ***Liviu Ciortuz, Andi Munteanu, pornind de la A first course in Probability, Sheldon Ross, 8th edition, 2010, Stanford, Machine Learning course, John Duchi, Supplemental Lecture Notes – Hoeffding's inequality***

a. Inegalitatea lui [Andrey] Markov.[28]
Fie $Z \geq 0$ o variabilă aleatoare cu valori nenegative. Demonstraţi că pentru orice $t > 0$, are loc inegalitatea

[28] Andrey Markov, 1856 - 1922, matematician rus, care l-a avut ca profesor pe Pafnuty Chebyshev. Inegalitatea aceasta a apărut în lucrările lui Chebyshev (şi ale altor matematicieni), însă ea a primit mai târziu numele lui Markov, în onoarea lui. Markov a avut contribuţii în domeniul proceselor stochastice. Cercetările sale au stat la baza creării lanţurilor Markov şi a proceselor Markov. Cf. https://en.wikipedia.org/wiki/Andrey_Markov, accesat la 22 martie 2022.

$$P(Z \geq t) \leq \frac{E[Z]}{t}.$$

b. Inegalitatea lui Chebyshev[29] – consecinţă a inegalităţii Markov.
Fie Z o variabilă aleatoare având atât media ($E[Z]$) cât şi varianţa ($Var[Z]$) finite. Demonstraţi că pentru orice $t > 0$,

$$P(Z - E[Z] \geq t \text{ sau } Z - E[Z] \leq -t) = P(|Z - E[Z]| \geq t) \leq \frac{Var[Z]}{t^2}.$$

c. O consecinţă imediată a inegalităţii Chebyshev.
Fie Z o variabilă aleatoare a cărei medie ($E[Z]$) este finită. Arătaţi că în cazul în care $Var[Z] = 0$, rezultă

$$P(Z = E[Z]) = 1.$$

d. O altă inegalitate de tip Chebyshev — consecinţă a inegalităţii Markov — cunoscută şi sub numele de inegalitatea lui Cantelli.[30]
Fie Z o variabilă aleatoare pentru care media ($E[Z]$) şi varianţa ($Var[Z]$) sunt finite. Demonstraţi că pentru orice $t > 0$, are loc inegalitatea

$$P(Z - E[Z] \geq t) \leq \frac{Var[Z]}{Var[Z] + t^2}.$$

e. Inegalităţile lui Chernoff[31] – consecinţe ale inegalităţii Markov.
Fie Z o variabilă aleatoare a cărei medie ($E[Z]$) este finită. Demonstraţi că pentru orice $t \geq 0$,

$$P(Z \geq E[Z] + t) \leq \min_{\lambda \geq 0}\{E[\exp(\lambda(Z - E[Z]))]\, e^{-\lambda t}\} \overset{not.}{=} \min_{\lambda \geq 0}\{M_{Z-E[Z]}(\lambda)\, e^{-\lambda t}\}$$

şi

$$P(Z \leq E[Z] - t) \leq \min_{\lambda \geq 0}\{E[\exp(\lambda(E[Z] - Z))]\, e^{-\lambda t}\} \overset{not.}{=} \min_{\lambda \geq 0}\{M_{E[Z]-Z}(\lambda)\, e^{-\lambda t}\},$$

unde prin $M(\lambda)$ am desemnat funcţia generatoare de momente (vedeţi pr. 121).

f. Legea numerelor mari, varianta slabă.[32]
Folosiţi inegalitatea Chebyshev pentru a demonstra următoarea proprietate:

[29] Pafnuty Chebyshev, 1821 - 1894, matematician rus; este considerat părintele fondator al matematicii ruseşti. A avut contribuţii în domeniile probabilităţilor, statisticii, mecanicii, precum şi în teoria numerelor. Cf. https://en.wikipedia.org/wiki/Pafnuty_Chebyshev, accesat la 22 martie 2022.

[30] Din inegalitatea lui Chebyshev, deducem că $P(Z - E[Z] \geq t) \leq \frac{Var[Z]}{t^2}$. Vom arăta aici că putem stabili pentru această probabilitate o margine superioară mai bună, şi anume $\frac{Var[Z]}{Var[Z] + t^2}$.

[31] Herman Chernoff, 1923- , matematician american, ai cărui părinţi au fost imigranţi evrei proveniţi din Rusia. Herman Chernoff a predat la University of Illinois Urbana-Champaign, Stanford, MIT şi Harvard University. Cf. https://en.wikipedia.org/wiki/Herman_Chernoff, accesat la 22 martie 2022.

[32] Alături de *teoremele limită centrală*, teoremele cunoscute sub numele de *legi ale numerelor mari* constituie cele mai importante rezultate din teoria probabilităţilor. Legile numerelor mari stabilesc condiţiile în care *media aritmetică* (sau, „empirică“) a unei secvenţe de variabile aleatoare aproximează (într-un anumit sens) *media probabilistă* a variabilelor respective. Teoremele limită centrală studiază condiţiile în care suma unui mare număr de variabile aleatoare urmează o distribuţie aproximativ normală / gaussiană. Cf. *A first course in Probability*, Sheldon Ross, 8th edition, 2010, Prentice Hall, pag. 395.

dacă $Z_1, Z_2, \ldots$ sunt un şir de variabile aleatoare i.i.d., toate având media finită $E[Z_i] = \mu$, atunci urmează că pentru orice $\varepsilon > 0$

$$P\Big(\Big|\frac{Z_1 + \ldots + Z_n}{n} - \mu\Big| \geq \varepsilon\Big) \to 0 \text{ atunci când } n \to \infty.$$

***Sugestie*: În demonstraţie puteţi folosi o *presupoziţie suplimentară*, şi anume că toate variabilele au varianţa finită, $Var[Z_i] \stackrel{not.}{=} \sigma^2$.**

Răspuns:

a. Inegalitatea lui Markov.
Definim variabila-indicator $1_{\{Z \geq t\}}$, care ia valoarea 1 atunci când $Z \geq t$ şi 0 în cazul contrar. Întrucât Z ia doar valori nenegative, are loc inegalitatea $1_{\{Z \geq t\}} \leq \frac{Z}{t}$. Aplicând operatorul E (media) ambilor membrii ai acestei inegalităţi, obţinem $E[1_{\{Z \geq t\}}] \leq E\left[\frac{Z}{t}\right] \Leftrightarrow E[1_{\{Z \geq t\}}] \leq \frac{E[Z]}{t}$, fiindcă t este constant. Din modul în care este definită variabila aleatoare $1_{\{Z \geq t\}}$, rezultă că $E[1_{\{Z \geq t\}}] = P(Z \geq t)$ (vedeţi pr. 98). Înlocuind în ultima inegalitate, obţinem că $P(Z \geq t) \leq \frac{E[Z]}{t}$, ceea ce trebuia demonstrat.

b. Inegalitatea lui Chebyshev.
Variabila aleatoare $(Z - E[Z])^2$ ia doar valori nenegative, deci putem să-i aplicăm inegalitatea lui Markov cu $t = t'^2$:

$$P((Z - E[Z])^2 \geq t'^2) \leq \frac{E[(Z - E[Z])^2]}{t'^2}.$$

Inegalitatea $(Z - E[Z])^2 \leq t'^2$ este satisfăcută dacă şi numai dacă $|Z - E[Z]| \geq t'$. Ajungem deci la următoarea inegalitate:

$$P(|Z - E[Z]| \geq t') \leq \frac{E[(Z - E[Z])^2]}{t'^2} = \frac{Var[Z]}{t'^2}.$$

c. O consecinţă imediată a inegalităţii Chebyshev.
Conform inegalităţii Chebyshev, pentru variabila aleatoare Z are loc inegalitatea $P(|Z - E[Z]| \geq t) \leq \frac{Var[Z]}{t^2}$, pentru orice $t > 0$. Din ipoteză ştim că $Var[Z] = 0$, deci $P(|Z - E[Z]| \geq t) \leq 0$. Ţinând cont de faptul că probabilităţile pot lua valori doar între 0 şi 1, deducem că $P(|Z - E[Z]| \geq t) = 0$, pentru orice $t > 0$.

Întrucât expresia $|Z - E[Z]|$ ia ca valori doar numere reale nenegative, iar evenimentele $|Z - E[Z]| = t$ sunt imposibile pentru orice $t > 0$, rezultă că $P(|Z - E[Z]| = 0) = 1 \Leftrightarrow P(Z = E[Z]) = 1$, adică Z este variabilă aleatoare constantă.

d. O altă inegalitate de tip Chebyshev.
Fie $a > 0$ un termen ce va fi adăugat de o parte şi alta a inegalităţii $Z - E[Z] \geq t$. Folosindu-ne de inegalitatea evidentă $P(X \geq x) \leq P(X^2 \geq x^2)$ pentru orice $x > 0$, obţinem:

$$\begin{aligned} P(Z - E[Z] \geq t) &= P(\underbrace{Z - E[Z] + a}_{not.:X} \geq \underbrace{t + a}_{not.:x}) \leq P((Z - E[Z] + a)^2 \geq (t + a)^2) \\ &\stackrel{Markov}{\leq} \frac{E[(Z - E[Z] + a)^2]}{(t + a)^2}. \end{aligned}$$

În continuare, vom simplifica numărătorul fracţiei:

$$E[(Z-E[Z]+a)^2] = E[(Z-E[Z])^2 + 2a(Z-E[Z]) + a^2] \overset{lin.\ med.}{=}$$
$$= E[(Z-E[Z])^2] + \cancel{2aE[Z]} - \cancel{2aE[Z]} + a^2 = Var[Z] + a^2.$$

Înlocuind în inegalitatea de mai sus, obţinem $P(Z-E[Z]\geq t) \leq \dfrac{Var[Z]+a^2}{(t+a)^2}$**. Această inegalitate este valabilă pentru orice** $a>0$**, deci pentru îmbunătăţirea marginii superioare, vom minimiza membrul drept.**

Fie $f:(0,\infty)\to\mathbb{R}, f(x)=\dfrac{x^2+\sigma^2}{(x+t)^2}$**, unde** $\sigma \overset{not.}{=} \sqrt{Var[Z]}$ **şi** t **sunt constante strict pozitive. Funcţia** f **este derivabilă. Calculând derivata ei de ordinul unu, obţinem:**

$$\begin{aligned} f'(x) &= \frac{(x^2+\sigma^2)'(x+t)^2 - (x^2+\sigma^2)((x+t)^2)'}{(x+t)^4} = \frac{2x(x+t)^2 - (x^2+\sigma^2)2(x+t)}{(x+t)^4} \\ &= \frac{2(xt-\sigma^2)}{(x+t)^3}. \end{aligned}$$

Egalându-l pe $f'(x)$ **cu 0, vom obţine** $\dfrac{2(xt-\sigma^2)}{(x+t)^3} = 0 \Leftrightarrow xt-\sigma^2 = 0 \Leftrightarrow x_0 = \dfrac{\sigma^2}{t}$**. Ştiind că** $x+t>0, \forall x$**, remarcăm faptul că** $f'(x)<0, \forall x<x_0$**, respectiv** $f'(x)>0, \forall x>x_0$**. Deducem astfel că punctul** $x_0=\dfrac{\sigma^2}{t}$ **minimizează funcţia** f**. Revenind la îmbunătăţirea marginii superioare, obţinem:**

$$P(Z-E[Z]\geq t) \leq \min_{a>0}\frac{\sigma^2+a^2}{(t+a)^2} = \frac{\sigma^2+\frac{\sigma^4}{t^2}}{\left(t+\frac{\sigma^2}{t}\right)^2} = \frac{t^2\sigma^2+\sigma^4}{(t^2+\sigma^2)^2} = \frac{\sigma^2(t^2+\sigma^2)}{(t^2+\sigma^2)^2} = \frac{\sigma^2}{\sigma^2+t^2}.$$

e. Inegalităţile lui Chernoff.

***Prima inegalitate*: Ţinând cont de faptul că funcţia exponenţială este strict crescătoare (păstrând deci sensul inegalităţii), inegalitatea** $Z-E[Z]\geq t$ **este echivalentă cu** $\exp(\lambda(Z-E[Z]))\geq e^{\lambda t}$**, pentru orice** $\lambda>0$**, deci** $P(Z\geq E[Z]+t) = P(\exp(\lambda(Z-E[Z]))\geq e^{\lambda t})$**. Conform inegalităţii lui Markov, această nouă probabilitate va fi majorată astfel:**

$$P(\exp(\lambda(Z-E[Z]))\geq e^{\lambda t}) \leq \frac{E[\exp(\lambda(Z-E[Z]))]}{\exp(\lambda t)} \overset{not.}{=} M_{Z-E[Z]}(\lambda)\ e^{-\lambda t}.$$

Această inegalitate fiind valabilă pentru orice $\lambda>0$**, putem îmbunătăţi marginea superioară a probabilităţii prin aplicarea operatorului de minimizare peste parametrul** $\lambda>0$ **(şi chiar pentru** $\lambda\geq 0$**, fiindcă** $M_{Z-E[Z]}(0)\ e^{-0\cdot t}=1$**). Obţinem, astfel:**

$$\begin{aligned} P(Z\geq E[Z]+t) &= P(Z-E[Z]\geq t) = P(\exp(\lambda(Z-E[Z]))\geq e^{\lambda t}) \\ &\leq \min_{\lambda\geq 0}\{M_{Z-E[Z]}(\lambda)\ e^{-\lambda t}\}. \end{aligned}$$

***Cea de-a doua inegalitate*: Inversăm semnul inegalităţii** $Z\leq E[Z]-t$**, ajungând astfel la** $-Z\geq t-E[Z]$**. Ţinând cont de faptul că funcţia exponenţială este strict**

crescătoare, inegalitatea $E[Z]-Z\geq t$ este echivalentă cu $\exp(\lambda(E[Z]-Z))\geq e^{\lambda t}$, pentru orice $\lambda>0$, deci $P(Z\leq E[Z]-t)=P(E[Z]-Z\geq t)=P(\exp(\lambda(E[Z]-Z))\geq e^{\lambda t})$. Conform inegalităţii lui Markov, această nouă probabilitate va fi majorată astfel:

$$P(\exp(\lambda(E[Z]-Z))\geq e^{\lambda t})\leq \frac{E[\exp(\lambda(E[Z]-Z))]}{\exp(\lambda t)} \overset{not.}{=} M_{E[Z]-Z}(\lambda)\, e^{-\lambda t}.$$

Această inegalitate fiind valabilă pentru orice $\lambda>0$, putem îmbunătăţi marginea superioară a probabilităţii prin aplicarea operatorului de minimizare peste parametrul $\lambda>0$ (şi chiar pentru $\lambda\geq 0$, fiindcă $M_{E[Z]-Z}(0)\, e^{-0\cdot t}=1$). Obţinem, astfel:

$$\begin{aligned} P(Z\leq E[Z]-t) &= P(E[Z]-Z\geq t)=P(\exp(\lambda(E[Z]-Z))\geq e^{\lambda t}) \\ &\leq \min_{\lambda\geq 0}\{M_{E[Z]-Z}(\lambda)\, e^{-\lambda t}\}. \end{aligned}$$

f. Legea numerelor mari, varianta slabă.
Vom nota $Z=\frac{Z_1+...+Z_n}{n}$. Ştim din enunţ că variabilele aleatoare Z_i sunt i.i.d. şi au varianţa (σ^2) finită, pentru $i\in\{1,...,n\}$. Ţinând cont de proprietăţile de liniaritate ale mediei $E[aX+bY]=aE[X]+bE[Y]$ şi ale varianţei $Var[aX+bY]\overset{indep.}{=}a^2Var[X]+b^2Var[Y]$, obţinem că:

$$E[Z]=\frac{1}{n}\sum_{i=1}^{n}E[Z_i]=\frac{n\mu}{n}=\mu;\quad Var[Z]=\frac{1}{n^2}\sum_{i=1}^{n}Var[Z_i]=\frac{n\sigma^2}{n^2}=\frac{\sigma^2}{n}.$$

Probabilitatea $P\left(\left|\frac{Z_1+...+Z_n}{n}-\mu\right|\geq\varepsilon\right)$ se poate rescrie ca $P(|Z-E[Z]|\geq\varepsilon)$. Aplicând inegalitatea lui Chebyshev, rezultă:

$$P(|Z-E[Z]|\geq\varepsilon)\leq\frac{Var[Z]}{\varepsilon^2}=\frac{1}{n^2}\cdot n\sigma^2\cdot\frac{1}{\varepsilon^2}=\frac{\sigma^2}{n\varepsilon^2}.$$

Ţinând cont de faptul că $\varepsilon>0$ şi σ nu depind de n, deducem că $\lim_{n\to\infty}\frac{\sigma^2}{n\varepsilon^2}=0$. Întrucât probabilitatea nu poate fi negativă, rezultă prin trecere la limită că $0\leq\lim_{n\to\infty}P(|Z-E[Z]|\geq\varepsilon)\leq 0$. Aşdar, conform teoremei cleştelui obţinem rezultatul

$$\lim_{n\to\infty}P\left(\left|\frac{Z_1+...+Z_n}{n}-\mu\right|\geq\varepsilon\right)\overset{not.}{=}\lim_{n\to\infty}P(|Z-E[Z]|\geq\varepsilon)=0.$$

22. **(Inegalităţile lui Hoeffding)**

□ ***Andi Munteanu, Liviu Ciortuz, pornind de la Stanford, Machine Learning course, John Duchi, Supplemental Lecture Notes – Hoeffding's inequality, University of Massachusetts at Amherst, Statistical ML course, 2010 spring, Justin Domke, Learning theory (Lecture Notes 10)***

a. Demonstraţi următorul rezultat, cunoscut sub numele de *lema lui Hoeffding*:

Dacă Z este o variabilă aleatoare mărginită, cu $Z \in [a', b']$, atunci are loc inegalitatea:

$$M_{Z-E[Z]}(\lambda) \stackrel{def.}{=} E[\exp(\lambda(Z - E[Z]))] \leq \exp\Big(\frac{\lambda^2(b'-a')^2}{8}\Big) \text{ pentru orice } \lambda \in \mathbb{R},$$

unde prin $M(\lambda)$ am desemnat funcţia generatoare de momente (vedeţi pr. 121).

b. Folosind inegalităţile lui Chernoff (vedeţi problema 21.e), precum şi lema lui Hoeffding, demonstraţi *inegalităţile lui Hoeffding*,[33] care sunt probabil inegalităţile cel mai des folosite în *teoria învăţării computaţionale*:
Dacă $Z_1, \ldots, Z_n$ sunt variabile aleatoare independente şi mărginite, cu $Z_i \in [a_i, b_i]$ pentru $i = 1, \ldots, n$ şi notăm $\bar{Z} = \frac{1}{n}\sum_{i=1}^{n} Z_i$, atunci pentru orice $t \geq 0$ au loc inegalităţile următoare:

$$P(\bar{Z} - E[\bar{Z}] \geq t) \leq \exp\left(-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right)$$
$$P(E[\bar{Z}] - \bar{Z} \geq t) \leq \exp\left(-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right),$$

de unde rezultă imediat că

$$P(|\bar{Z} - E[\bar{Z}]| \geq t) \leq 2\exp\left(-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right). \tag{10}$$

c. Arătaţi că dacă se consideră un număr oarecare $\delta \in (0, 1]$ — care poate fi asociat cu un anumit „nivel de încredere“ (engl., confidence) —, atunci pentru orice

$$n \geq \frac{1}{t}\sqrt{\frac{1}{2}\Big(\sum_{i=1}^{n}(b_i - a_i)^2\Big)\ln\frac{2}{\delta}}$$

este satisfăcută inegalitatea $P(|\bar{Z} - E[\bar{Z}]| \geq t) \leq \delta$. În mod alternativ, putem afirma cu un grad de încredere de cel puţin $1 - \delta$ că, atunci când n satisface inegalitatea de mai sus, diferenţa (în modul) dintre $\bar{Z}$ şi $E[\bar{Z}]$ este de cel mult t.

Răspuns:

a. Notăm $Z - E[Z]$ cu $\tilde{Z}$. Rezultă că $E[\tilde{Z}] = E[Z - E[Z]] = 0$. De asemenea, notăm $a = a' - E[Z]$ şi $b = b' - E[Z]$. Evident, $Z \in [a', b'] \Rightarrow E[Z] \in [a', b']$, deci $a \leq 0 \leq b$, iar $\tilde{Z} = Z - E[Z] \in [a, b]$. Pe lângă relaţiile acestea, mai avem şi egalitatea $b - a = b' - a'$, care se poate verifica imediat.

Ştim că $e^{\lambda x}$ este o funcţie convexă pentru orice $\lambda \in \mathbb{R}$. Considerând $x \in [a, b]$, vom rescrie λx sub forma combinaţiei convexe $\frac{b-x}{b-a}\lambda a + \frac{x-a}{b-a}\lambda b$ şi apoi vom aplica inegalitatea lui Jensen:[34]

[33] Wassily Hoeffding (1914 - 1991), statistician, născut în Finlanda (la vremea respectivă Finlanda era ducat în imperiul ţarist). Bunicii lui pe linie paternă au fost de origine daneză. Familia lui a emigrat în 1920 în Danemarca. Wassily Hoeffding a obţinut doctoratul în 1940 la universitatea din Berlin. A emigrat în 1946 în USA şi a activat la universitatea Carolinei de Nord la Chapel Hill. (Cf. https://en.wikipedia.org/wiki/Wassily_Hoeffding, accesat la 22 martie 2022.)

[34] Vedeţi problema 143 de la acest capitol.

$$e^{\lambda x} \leq \frac{b-x}{b-a}e^{\lambda a} + \frac{x-a}{b-a}e^{\lambda b}.$$

Inegalitatea are loc pentru orice $x \in [a,b]$, deci în particular pentru $x = \tilde{Z}$. Aplicând apoi operatorul E (medie) ambelor părţi ale inegalităţii, vom obţine:

$$M_{\tilde{Z}}(\lambda) \overset{def.}{=} E[\exp(\lambda\tilde{Z})] \leq \frac{b-E[\tilde{Z}]}{b-a}e^{\lambda a} + \frac{E[\tilde{Z}]-a}{b-a}e^{\lambda b} \overset{E[\tilde{Z}]=0}{=} \frac{b}{b-a}e^{\lambda a} - \frac{a}{b-a}e^{\lambda b}.$$

Notăm $-\frac{a}{b-a}$ cu θ. Ţinând cont de *Observaţia* de mai sus legată de semnul lui a şi semnul lui b, rezultă că $\theta \geq 0$. Inegalitatea de mai sus poate fi rescrisă astfel:

$$M_{\tilde{Z}}(\lambda) \leq (1-\theta)e^{\lambda a} + \theta e^{\lambda b} = e^{\lambda a}\left(1-\theta+\theta e^{\lambda(b-a)}\right) \overset{def.\ \theta}{=} e^{-\lambda\theta(b-a)}\left(1-\theta+\theta e^{\lambda(b-a)}\right).$$

Notând $u = \lambda(b-a)$, expresia $e^{-\lambda\theta(b-a)}\left(1-\theta+\theta e^{\lambda(b-a)}\right)$ poate fi rescrisă sub forma $e^{-\theta u}(1-\theta+\theta e^u)$. Pentru a studia această expresie într-o manieră convenabilă, vom defini funcţia $f : \mathbb{R} \to \mathbb{R}$, $f(u) = -\theta u + \ln(1-\theta+\theta e^u)$. Verificăm dacă argumentul logaritmului este pozitiv:

$$1-\theta+\theta e^u = \theta\left(\frac{1}{\theta} - 1 + e^u\right) \overset{def.\ \theta}{=} \theta\left(-\frac{b}{a} + e^u\right) \overset{a<0,\ b\geq 0}{>} 0.$$

Studiem derivatele de ordinul unu şi doi ale funcţiei f:

$$\begin{aligned} f'(u) =& -\theta + \frac{\theta e^u}{1-\theta+\theta e^u} \\ f''(u) =& \frac{\theta e^u(1-\theta+\theta e^u) - (\theta e^u)^2}{(1-\theta+\theta e^u)^2} = \frac{\theta e^u}{1-\theta+\theta e^u}\left(1 - \frac{\theta e^u}{1-\theta+\theta e^u}\right) \overset{not.}{=} v(1-v), \\ & \text{unde } v \overset{not.}{=} \frac{\theta e^u}{1-\theta+\theta e^u}. \end{aligned}$$

Observăm faptul că $v - v^2$ este o funcţie de gradul al doilea, concavă, având coeficienţii $c_2 = -1$, $c_1 = 1$ şi $c_0 = 0$. Funcţia îşi atinge maximul în $v_0 = -\frac{c_1}{2c_2} = \frac{1}{2}$. În consecinţă, $f''(u) \leq \frac{1}{2}\cdot\frac{1}{2} = \frac{1}{4}$, pentru orice u.

Aplicând Teorema lui Taylor, rezultă că pentru orice $u \in \mathbb{R}^*$, există un w situat între 0 şi u, astfel încât

$$f(u) = f(0)+(u-0)f'(0)+(u-0)^2\frac{f''(w)}{2!} = (0+\ln 1)+u\left(-\theta+\frac{\theta}{1}\right)+\frac{1}{2}u^2f''(w) = \frac{u^2f''(w)}{2}.$$

Ştiind că $f''(u) \leq \frac{1}{4}$ pentru orice u, rezultă că $f(u) \leq \frac{u^2}{2}\cdot\frac{1}{4} = \frac{u^2}{8}$.

Revenind acum la inegalitatea relativă la $M_{\tilde{Z}}(\lambda)$, putem scrie:

$$\begin{aligned} M_{\tilde{Z}}(\lambda) &\leq e^{-\lambda\theta(b-a)}\left(1-\theta+\theta e^{\lambda(b-a)}\right) = e^{f(u)} \\ &\leq \exp\left(\frac{1}{8}u^2\right) \overset{def.\ u}{=} \exp\left(\frac{1}{8}\lambda^2(b-a)^2\right) = \exp\left(\frac{1}{8}\lambda^2(b'-a')^2\right). \end{aligned}$$

b. *Prima inegalitate*: Aplicăm prima inegalitate a lui Chernoff (varianta fără $\min_{\lambda\geq 0}$) pentru a mărgini superior probabilitatea $P(\bar{Z}-E[\bar{Z}]\geq t)$:

$$P(\bar{Z}-E[\bar{Z}]\geq t)=P\left(\frac{1}{n}\sum_{i=1}^{n}(Z_i-E[Z_i])\geq t\right)=P\left(\sum_{i=1}^{n}(Z_i-E[Z_i])\geq nt\right)$$
$$\overset{Chernoff}{\leq} E\left[\exp\left(\lambda\sum_{i=1}^{n}(Z_i-E[Z_i])\right)\right]e^{-n\lambda t}=E\left[\prod_{i=1}^{n}\exp\left(\lambda(Z_i-E[Z_i])\right)\right]e^{-n\lambda t}$$
$$\overset{pr.\ 121.a}{=} e^{-n\lambda t}\prod_{i=1}^{n}E[\exp(\lambda(Z_i-E[Z_i]))]\overset{not.}{=} e^{-n\lambda t}\prod_{i=1}^{n}M_{Z_i-E[Z_i]}(\lambda).$$

Ne vom folosi în continuare de lema lui Hoeffding, pe care o vom aplica fiecărui factor al produsului şi, prin urmare, vom obţine:

$$\begin{aligned}P(\bar{Z}-E[\bar{Z}]\geq t) &\leq e^{-n\lambda t}\prod_{i=1}^{n}\exp\left(\frac{\lambda^2(b_i-a_i)^2}{8}\right)=\exp\left(-n\lambda t+\sum_{i=1}^{n}\frac{\lambda^2(b_i-a_i)^2}{8}\right)\\ &= \exp\left(-n\lambda t+\frac{\lambda^2}{8}\sum_{i=1}^{n}(b_i-a_i)^2\right).\end{aligned}\tag{11}$$

Inegalitatea aceasta are loc pentru orice $\lambda\geq 0$ (vedeţi pr. 21.e), deci putem să îmbunătăţim marginea superioară prin aplicarea operatorului de minimizare peste parametrul $\lambda\geq 0$. Observăm că argumentul funcţiei $\exp$ este un polinom de gradul al doilea în raport cu parametrul λ, deci de forma $c_2\lambda^2+c_1\lambda+c_0$, având coeficienţii

$$c_2=\frac{1}{8}\sum_{i=1}^{n}(b_i-a_i)^2,\quad c_1=-nt,\quad c_0=0.$$

Coeficientul c_2 este pozitiv, deci funcţia aceasta îşi atinge minimul în $\lambda'=-\dfrac{c_1}{2c_2}=\dfrac{4nt}{\sum_{i=1}^{n}(b_i-a_i)^2}$. Prin urmare, putem deduce că marginea superioară a probabilităţii $P(\bar{Z}-E[\bar{Z}]\geq t)$ este următoarea:

$$\begin{aligned}P(\bar{Z}-E[\bar{Z}]\geq t) &\leq \min_{\lambda\geq 0}\exp\left(-n\lambda t+\frac{\lambda^2}{8}\sum_{i=1}^{n}(b_i-a_i)^2\right)\\ &= \exp\left(-n\lambda' t+\frac{\lambda'^2}{8}\sum_{i=1}^{n}(b_i-a_i)^2\right)=\exp\left(-\frac{4n^2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}+\frac{16n^2t^2}{8\sum_{i=1}^{n}(b_i-a_i)^2}\right)\\ &= \exp\left(-\frac{2n^2t^2}{\sum_{i=1}^{n}(b_i-a_i)^2}\right).\end{aligned}\tag{12}$$

***A doua inegalitate*: Aplicăm inegalitatea doua a lui Chernoff (varianta fără $\min_{\lambda\geq 0}$) pentru a mărgini superior probabilitatea $P(E[\bar{Z}]-\bar{Z}\geq t)$:**

$$P(E[\bar{Z}]-\bar{Z}\geq t)=P\left(\frac{1}{n}\sum_{i=1}^{n}(E[Z_i]-Z_i)\geq t\right)=P\left(\sum_{i=1}^{n}(E[Z_i]-Z_i)\geq nt\right)$$
$$\overset{Chernoff}{\leq} E\left[\exp\left(\lambda\sum_{i=1}^{n}(E[Z_i]-Z_i)\right)\right]e^{-n\lambda t}=E\left[\prod_{i=1}^{n}\exp\left(\lambda(E[Z_i]-Z_i)\right)\right]e^{-n\lambda t}$$
$$\overset{pr.\ 121.a}{=} e^{-n\lambda t}\prod_{i=1}^{n}E[\exp(\lambda(E[Z_i]-Z_i))]\overset{not.}{=} e^{-n\lambda t}\prod_{i=1}^{n}M_{E[Z_i]-Z_i}(\lambda).$$

Vom folosi în din nou de lema lui Hoeffding (ţinem cont de faptul că lema este valabilă pentru orice $\lambda \in \mathbb{R}$ şi de faptul că $M_{E[Z_i]-Z_i}(\lambda)$, cu un λ pozitiv, este echivalent cu $M_{Z_i-E[Z_i]}(\lambda)$, cu un λ negativ), pe care o vom aplica pentru fiecare factor al produsului. Vom obţine:

$$\begin{aligned} P(E[\bar{Z}] - \bar{Z} \geq t) &\leq e^{-n\lambda t} \prod_{i=1}^{n} \exp\left(\frac{\lambda^2 (b_i - a_i)^2}{8}\right) = \exp\left(-n\lambda t + \sum_{i=1}^{n} \frac{\lambda^2 (b_i - a_i)^2}{8}\right) \\ &= \exp\left(-n\lambda t + \frac{\lambda^2}{8} \sum_{i=1}^{n} (b_i - a_i)^2\right). \end{aligned} \tag{13}$$

Ca şi mai sus, inegalitatea aceasta are loc pentru orice $\lambda \geq 0$, deci putem să îmbunătăţim marginea superioară prin aplicarea operatorului de minimizare peste parametrul $\lambda \geq 0$. Observăm că expresiile (13) şi (11) sunt identice, deci au aceeaşi valoare minimă, şi anume (12). Prin urmare,

$$P(E[\bar{Z}] - \bar{Z} \geq t) \leq \min_{\lambda \geq 0} \exp\left(-n\lambda t + \frac{\lambda^2}{8} \sum_{i=1}^{n} (b_i - a_i)^2\right) = \exp\left(-\frac{2n^2 t^2}{\sum_{i=1}^{n} (b_i - a_i)^2}\right).$$

c. Impunând condiţia ca membrul drept al inegalităţii (10) să fie mai mic sau egal cu δ, vom obţine:

$$\begin{gathered} 2\exp\left(-\frac{2n^2 t^2}{\sum_{i=1}^{n} (b_i - a_i)^2}\right) \leq \delta \Leftrightarrow \exp\left(-\frac{2n^2 t^2}{\sum_{i=1}^{n} (b_i - a_i)^2}\right) \leq \frac{\delta}{2} \Leftrightarrow \\ -\frac{2n^2 t^2}{\sum_{i=1}^{n} (b_i - a_i)^2} \leq \ln\frac{\delta}{2} \Leftrightarrow \frac{2n^2 t^2}{\sum_{i=1}^{n} (b_i - a_i)^2} \geq \ln\frac{2}{\delta} \Leftrightarrow \\ n^2 \geq \frac{\sum_{i=1}^{n} (b_i - a_i)^2}{2t^2} \ln\frac{2}{\delta} \Leftrightarrow n \geq \frac{1}{t}\sqrt{\frac{1}{2}\sum_{i=1}^{n} (b_i - a_i)^2 \ln\frac{2}{\delta}}. \end{gathered}$$

23. **(Variabile aleatoare: Adevărat sau Fals?)**

CMU, 2006 fall, E. Xing, T. Mitchell, final exam, pr. 1.b
CMU, 2008 fall, Eric Xing, midterm exam, pr. 1.1, 1.2, 1.3

a. Dacă o variabilă aleatoare continuă X are funcţia densitate de probabilitate p diferită de zero pe tot domeniul de definiţie, atunci probabilitatea ca X să ia o valoare oarecare x (notaţie: $P(X = x)$) este egală cu $p(x)$.

b. $E[X + Y] = E[X] + E[Y]$ pentru orice două variabile aleatoare X şi Y.

c. $Var[X + Y] = Var[X] + Var[Y]$ pentru orice două variabile aleatoare X şi Y.

d. $E[XY] = E[X] \cdot E[Y]$ pentru orice două variabile aleatoare X şi Y.

Răspuns:

a. Fals (în general). Dacă variabila aleatoare continuă X are funcţia densitate de probabilitate p, atunci $P(a \leq X \leq b) = \int_a^b p(x)dx$. Prin urmare, $P(X = x) = 0$ pentru orice $x \in \mathbb{R}$. Enunţul afirmă pe de o parte că $p(x) = P(X = x) = 0$ pentru

un anume $x \in \mathbb{R}$, iar pe de altă parte că p este diferită de zero pe tot domeniul de definiţie, ceea ce este absurd.

b. Adevărat. Demonstraţia este făcută în problema 9 punctul a.

c. Fals (în general). Se poate considera situaţia $Y = -X$, caz în care $Var[X + Y] = 0$, dar $Var[X] + Var[Y] = E[X^2] - (E[X])^2 + E[(-X)^2] - (E[-X])^2 = 2\,Var[X]$. Aşadar, pentru orice variabilă aleatoare X cu $Var[X] \neq 0$, luând $Y = -X$, rezultă că $Var[X + Y] = 0$, iar $Var[X] + Var[Y] \neq 0$. Un exemplu de variabilă aleatoare cu varianţa nenulă este distribuţia gaussiană.

Mai *general*, se poate demonstra că $Var[X + Y] = Var[X] + Var[Y] + 2Cov[X, Y]$ astfel:

$$\begin{aligned} Var[X+Y] &= E[(X+Y)^2] - (E[X+Y])^2 \\ &= E[X^2 + 2XY + Y^2] - (E[X] + E[Y])^2 \\ &= E[X^2] + 2E[XY] + E[Y^2] - (E[X])^2 - 2E[X] \cdot E[Y] - (E[Y])^2 \\ &= (E[X^2] - (E[X])^2) + (E[Y^2] - (E[Y])^2) + (2E[XY] - 2E[X] \cdot E[Y]) \\ &= Var[X] + Var[Y] + 2Cov[X, Y] \end{aligned}$$

Prin urmare, egalitatea din enunţ este adevărată pentru orice două variabile aleatoare X şi Y pentru care $Cov[X, Y] = 0$ (vedeţi problema 9.c), dar este falsă în rest. Egalitatea $Cov[X, Y] = 0$ este adevărată de exemplu atunci când X şi Y sunt variabile independente (vedeţi problema 10).

d. Fals, în general. Afirmaţia din enunţ este echivalentă cu $Cov[X, Y] = 0$. Aşa cum am menţionat la punctul c, ea este adevărată dacă, de exemplu X şi Y sunt variabile aleatoare independente. Dacă, în schimb, vom considera de pildă cazul $Y = X$, cu X variabilă aleatoare binară care ia valoarea 1 cu probabilitatea p şi valoarea 0 cu probabilitatea $1 - p$, se poate deduce imediat că $E[X^2] \neq (E[X])^2$, fiindcă $p(1 - p) \neq 0$ pentru orice $p \in (0, 1)$.

0.1.3 Distribuţii probabiliste uzuale

24. **(Distribuţia Bernoulli; variabile aleatoare identic distribuite: calcul de valori medii folosind proprietatea de liniaritate a mediilor)**

• ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW1, pr. 2.2***

Un iepuraş se joacă „de-a săritelea". Poziţia lui iniţială coincide cu originea axei reale. După aceea, iepuraşul face câte un salt de-a lungul axei, fie la stânga fie la dreapta. Pentru a determina în ce direcţie să sară, iepuraşul dă cu banul. Dacă obţine stema, va sări spre dreapta, iar dacă obţine banul, va sări spre stânga. Probabilitatea de a obţine stema este p. Se presupune că toate salturile iepuraşului au aceeaşi lungime, şi anume 1.

Care va fi poziţia (medie) la care ne aşteptăm să fie iepuraşul după ce face n salturi?

Răspuns:

Fiecare săritură a iepuraşului este modelată de o variabilă aleatoare. Un salt spre dreapta înseamnă o deplasare cu $+1$ pe axă, iar un salt spre stânga -1. Să notăm cu X_i variabila aleatoare corespunzătoare săriturii i. Aceasta este:

$$X_i : \begin{pmatrix} -1 & 1 \\ 1-p & p \end{pmatrix}$$

Media acestei variabile aleatoare este $E[X_i] = -1 \cdot (1-p) + 1 \cdot p = 2p - 1$.

Ţinând cont de proprietatea de liniaritate a mediilor (vedeţi pr. 9.a), poziţia iepuraşului după n salturi este de aşteptat să fie:

$$E[X_1 + X_2 + \ldots + X_n] = E[X_1] + E[X_2] + \ldots + E[X_n] = n(2p-1).$$

De exemplu, pentru $p = 1/2$, se va obţine poziţia 0 pe axa reală (aşa cum este de aşteptat dacă n este număr par), în vreme ce pentru $p = 2/3$ va rezulta poziţia $n/3$ (dacă $n/3 \in \mathbb{N}$), iar pentru $p = 1/3$ poziţia $-n/3$ (similar).

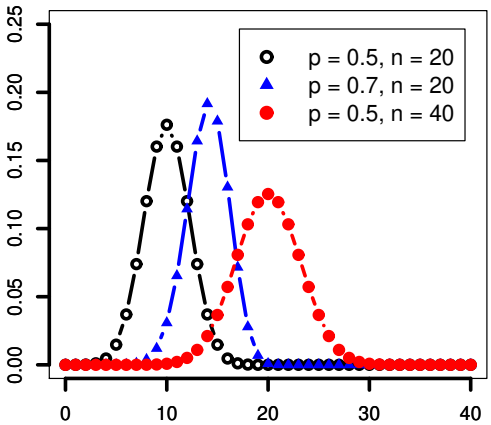
25. (Distribuţia binomială: verificarea condiţiilor de definiţie pentru p.m.f.; calculul mediei şi al varianţei)

■ □ ***Liviu Ciortuz, 2015***

Distribuţia binomială de parametri n şi p are funcţia masă de probabilitate (engl., probability mass function, p.m.f.) definită astfel:

$$b(r; n, p) = C_n^r p^r (1-p)^{n-r} \; \forall r \in \{0, \ldots, n\}.$$

Distribuţia binomială: p.m.f.



Vă reamintim că $b(r; n, p)$ este probabilitatea care corespunde numărului (r) de apariţii ale feţei cu stema (corespondent engl., *head*) obţinute la efectuarea a n aruncări independente ale unei monede, atunci când se presupune că probabilitatea de apariţie a stemei la o aruncare oarecare a acestei monede este p.

a. Verificaţi că funcţia $b(r; n, p)$, aşa cum a fost definită mai sus, reprezintă într-adevăr o funcţie masă de probabilitate. Aceasta revine la a arăta că $b(r; n, p) \geq 0$ pentru orice $p \in [0, 1]$, $n \in \mathbb{N}$ şi $r \in \{0, 1, \ldots, n\}$, iar $\sum_{r=0}^{n} b(r; n, p) = 1$ pentru orice astfel de n şi p, fixaţi.

b. Calculaţi media şi varianţa distribuţiei binomiale.

Răspuns:

a. Evident, $b(r;n,p) \overset{def.}{=} C_n^r p^r (1-p)^{n-r} \geq 0$ **pentru orice** $p \in [0,1]$**,** $n \in \mathbb{N}$ **şi** $r \in \{0,1,\ldots,n\}$**, iar**

$$\begin{aligned}\sum_{r=0}^{n} b(r;n,p) &= (1-p)^n + C_n^1 p(1-p)^{n-1} + \cdots + C_n^{n-1} p^{n-1}(1-p) + p^n \\ &= [p+(1-p)]^n = 1\end{aligned}$$

b. Calculul mediei se poate face pornind de la definiţie:

$$\begin{aligned}&E[b(r;n,p)] \overset{def.}{=} \sum_{r=0}^{n} r \cdot b(r;n,p) = \\ &= 1 \cdot C_n^1 p(1-p)^{n-1} + 2 \cdot C_n^2 p^2 (1-p)^{n-2} + \cdots + (n-1) \cdot C_n^{n-1} p^{n-1}(1-p) + n \cdot p^n \\ &= p\left[C_n^1 (1-p)^{n-1} + 2 \cdot C_n^2 p(1-p)^{n-2} + \cdots + (n-1) \cdot C_n^{n-1} p^{n-2}(1-p) + n \cdot p^{n-1}\right] \\ &= np\left[(1-p)^{n-1} + C_{n-1}^1 p(1-p)^{n-2} + \cdots + C_{n-1}^{n-2} p^{n-2}(1-p) + C_{n-1}^{n-1} p^{n-1}\right] \qquad (14) \\ &= np[p+(1-p)]^{n-1} = np = nE[Bernoulli\,(p)]. \qquad (15)\end{aligned}$$

Pentru egalitatea (14) **am folosit faptul că**

$$\begin{aligned}k\,C_n^k &= k\frac{n!}{k!\,(n-k)!} = \frac{n!}{(k-1)!\,(n-k)!} = \frac{n\,(n-1)!}{(k-1)!\,(n-1-(k-1))!} \\ &= n\,C_{n-1}^{k-1}, \forall k = 1,\ldots,n.\end{aligned}$$

Pentru calculul varianţei, vom folosi formula $Var[X] = E[X^2] - E^2[X]$**, care a fost demonstrată la problema 9.b. Întrucât am calculat deja** $E[b(r;n,p)]$**, rămâne să calculăm** $E[b^2(r;n,p)]$**.**[35] **Notând** $q = 1-p$**, vom avea:**

$$\begin{aligned}E[b^2(r;n,p)] &\overset{def.}{=} \sum_{r=0}^{n} r^2 C_n^r p^r q^{n-r} = \sum_{r=0}^{n} r^2 \frac{n(n-1)\ldots(n-r+1)}{r!} p^r q^{n-r} \\ &= \sum_{r=1}^{n} rn \frac{(n-1)\ldots(n-r+1)}{(r-1)!} p^r q^{n-r} = \sum_{r=1}^{n} rn\,C_{n-1}^{r-1}\,p^r q^{n-r} \\ &= np\sum_{r=1}^{n} r\,C_{n-1}^{r-1}\,p^{r-1} q^{(n-1)-(r-1)}.\end{aligned}$$

Mai departe, notând pentru convenienţă $r-1$ **cu** j**, urmează:**

$$\begin{aligned}E[b^2(r;n,p)] &= np\sum_{j=0}^{n-1}(j+1)\,C_{n-1}^{j}\,p^j q^{(n-1)-j} \\ &= np\Big[\sum_{j=0}^{n-1} j\,C_{n-1}^{j}\,p^j q^{(n-1)-j} + \sum_{j=0}^{n-1} C_{n-1}^{j}\,p^j q^{(n-1)-j}\Big].\end{aligned}$$

Prima sumă din paranteza pătrată de mai sus este chiar $E[b(r;n-1,p)]$**, conform relaţiei** (15)**, iar cea de-a doua sumă este egală cu** 1**, conform unui calcul absolut similar cu cel de la punctul** a**. Prin urmare,**

$$E[b^2(r;n,p)] = np[(n-1)p+1] = n^2p^2 - np^2 + np.$$

[35] Rezolvarea de mai jos urmează îndeaproape linia demonstraţiei găsite pe site-ul www.proofwiki.org/wiki/Variance_of_Binomial_Distribution, accesat la data de 5 octombrie 2015. La rândul său, acest site menţionează ca sursă "Probability: An Introduction" de Geoffrey Grimmett şi Dominic Welsh, Oxford Science Publications, 1986.

Aşadar, $Var[X] = E[b^2(r;n,p)] - (E[b(r;n,p)])^2 = n^2p^2 - np^2 + np - n^2p^2 = np(1-p)$**.**

***Observaţie*: O altă cale de a calcula varianţa distribuţiei binomiale este următoarea:**

– se demonstrează relativ uşor că orice variabilă aleatoare urmând distribuţia binomială $b(r;n,p)$ **poate fi văzută ca o sumă de** n **variabile independente care urmează distribuţia Bernoulli de parametru** p**;**[36]

– se ştie (sau, se poate dovedi imediat) că varianţa distribuţiei Bernoulli de parametru p **este** $p(1-p)$**;**

– ţinând cont de proprietatea $Var[X_1+X_2\ldots+X_n] = Var[X_1]+Var[X_2]\ldots+Var[X_n]$ **atunci când** $X_1, X_2, \ldots, X_n$ **sunt variabile independente, conform demonstraţiei de la problema 23.c, rezultă că** $Var[X] = np(1-p)$**.**

26. **(Distribuţia categorială; distribuţia binomială: calcularea unor probabilităţi, medii, numere „medii“ etc.)**

□ ***UAIC, 2020 fall, Ştefan Bălăucă, student***

Mickey vrea să îşi cumpere un nou zar, dar se teme că şi acesta ar putea fi măsluit. Vânzătorul îi spune că din noul lot, o fracţiune f **dintre zaruri sunt măsluite având distribuţia de probabilitate** $P = [0.5, 0.1, 0.1, 0.1, 0.1, 0.1]$**.**

Pentru a testa dacă zarul pe care l-a ales este sau nu măsluit, Mickey face următorul experiment: aruncă zarul de n **ori şi numără de câte ori cade faţa cu numărul 1. Notează acest număr cu** a**. Pentru a-l ajuta pe Mickey să decidă dacă să cumpere acest zar, ne propunem să răspundem la următoarele întrebări:**

a. Considerăm variabila aleatoare $S =$ **numărul de aruncări la care se obţine faţa / valoarea 1. Care este distribuţia de probabilitate a variabilei** S **dacă zarul este măsluit? Dar dacă zarul este corect?**

b. Care este media variabilei aleatoare S **în fiecare din cele două cazuri?**

c. Care este probabilitatea *a posteriori* ca zarul să fie măsluit, dacă Mickey obţine în exact a **din cele** n **aruncări valoarea 1?**

d. Câte dintre cele n **aruncări trebuie să aibă valoarea 1 pentru ca Mickey să poată afirma că zarul este măsluit, cu probabilitate de cel puţin** p**?**

e. Presupunând că zarul este măsluit, care este numărul *minim* de aruncări (n**) necesare pentru a putea afirma că acesta este măsluit, cu probabilitate de cel puţin** p**? (Veţi da răspunsul în funcţie de** f **şi** p**.)**

f. Presupunând că zarul este măsluit, care este numărul *mediu* de aruncări necesare pentru a putea afirma că acesta este măsluit, cu probabilitate de cel puţin p**? (Veţi da răspunsul în funcţie de** f **şi** p**.)**

g. Aplicaţie numerică pentru punctele e **şi** f**:** $f = 1\%$ **şi** 5%**, iar** $p = 50\%$**,** 95% **şi** 99%**.**

[36] Vedeţi www.proofwiki.org/wiki/Bernoulli_Process_as_Binomial_Distribution, care citează ca sursă "Probability: An Introduction" de Geoffrey Grimmett şi Dominic Welsh, Oxford Science Publications, 1986.

Răspuns:

a. Considerăm variabilele aleatoare X_k = valoarea zarului la aruncarea cu numărul k şi evenimentele aleatoare $Y_k = (X_k$ este $1)$ şi $M =$ zarul este măsluit. Atunci, $P(Y_k|\neg M) = P(X_k = 1|\neg M) = 1/6$ şi $P(Y_k|M) = P(X_k = 1|M) = 1/2$.

Valoarea variabilei aleatoare S este a (adică, $S = a$) dacă şi numai dacă mulţimea $A = \{k|X_k = 1\}$ are cardinalul $|A| = a$. Cum fiecare dintre configuraţiile posibile ale mulţimii A sunt disjuncte (ca evenimente aleatoare), putem scrie

$$P(S = a|\neg M) = \sum_{A,|A|=a} \Big(\prod_{k\in A} P(Y_k|\neg M) \prod_{k\notin A} P(\neg Y_k|\neg M) \Big) = \sum_{A,|A|=a} \Big((1/6)^a (5/6)^{n-a} \Big),$$

deci

$$P(S = a|\neg M) = C_n^a (1/6)^a (5/6)^{n-a}.$$

Similar, putem scrie

$$P(S = a|M) = \sum_{A,|A|=a} \Big(\prod_{k\in A} P(Y_k|M) \prod_{k\notin A} P(\neg Y_k|M) \Big) = \sum_{A,|A|=a} \Big((1/2)^a (1/2)^{n-a} \Big),$$

deci

$$P(S = a|M) = \sum_{A,|A|=a} (1/2)^n = C_n^a (1/2)^n.$$

Putem observa că datorită construcţiei lui S, distribuţiile de probabilitate condiţionale $S = a|\neg M$ şi $S = a|M$ sunt de tip binomial.

b. Folosind formula mediei pentru distribuţiile binomiale,[37] vom avea:

$$E[S|\neg M] = n \cdot P(Y_k|\neg M) = n/6, \text{ iar } E[S|M] = n \cdot P(Y_k|M) = n/2.$$

c. Aplicând formula lui Bayes şi ţinând cont de informaţia din enunţ că probabilitatea ca un zar oarecare să fie măsluit este f, găsim probabilitatea cerută:

$$P(M|S = a) = \frac{P(S = a|M) \cdot P(M)}{P(S = a)} = \frac{P(S = a|M) \cdot f}{P(S = a|M) \cdot f + P(S = a|\neg M) \cdot (1 - f)}.$$

Pe baza relaţiilor deduse la punctul a, putem scrie:

$$\begin{aligned} P(M|S = a) &= \frac{C_n^a (1/2)^n \cdot f}{C_n^a (1/2)^n \cdot f + C_n^a (1/6)^a (5/6)^{n-a} \cdot (1 - f)} \\ &= \frac{f}{f + \dfrac{5^{n-a}}{3^n}(1 - f)} = \frac{3^n f}{3^n f + 5^{n-a}(1 - f)}. \end{aligned}$$

d. Pentru a putea afirma că zarul este măsluit cu probabilitate de cel puţin p trebuie să avem $P(M|S = a) \geq p$. Pe baza relaţiei deduse la punctul c, avem:

$$\begin{aligned} &\frac{3^n f}{3^n f + 5^{n-a}(1 - f)} \geq p \Leftrightarrow 3^n f \geq 3^n fp + 5^{n-a}(1 - f)p \Leftrightarrow 3^n f(1 - p) \geq 5^{n-a}(1 - f)p \\ &\Leftrightarrow 5^a \geq \left(\frac{5}{3}\right)^n \frac{1 - f}{f} \frac{p}{1 - p} \Leftrightarrow a \ln 5 \geq n(\ln 5 - \ln 3) + \ln \frac{1 - f}{f} + \ln \frac{p}{1 - p} \\ &\Leftrightarrow a \geq n \left(1 - \frac{\ln 3}{\ln 5}\right) + \frac{1}{\ln 5} \left(\ln \frac{1 - f}{f} + \ln \frac{p}{1 - p}\right). \end{aligned}$$

[37] Vedeţi ex. 25.b.

Observăm că din punct de vedere *asimptotic*, [adică] atunci când $n \to \infty$, raportul a/n trebuie să fie cel puţin egal cu $1 - \frac{\ln 3}{\ln 5} \approx 0.317$. Din ultima inegalitate de mai sus putem trage concluzia că indiferent de valoarea lui f, alegând n suficient de mare, vom putea efectua teste pentru a verifica dacă zarul este sau nu măsluit, cu orice precizie p.

e. Numărul minim de aruncări se va obţine dacă toate aruncările au valoarea 1 (dacă $a = n$). În acest caz trebuie ca $P(M|S = n) \geq p$. Pe baza relaţiilor de la punctul anterior, putem scrie:

$$3^n f(1-p) \geq 5^{n-n}(1-f)p \Leftrightarrow 3^n \geq \frac{1-f}{f}\frac{p}{1-p} \Leftrightarrow$$
$$n \geq \frac{1}{\ln 3}\left(\ln\frac{1-f}{f} + \ln\frac{p}{1-p}\right).$$

f. La punctul b am arătat că $\mathrm{E}[S|M] = n/2$, deci $a \approx n/2$ în acest caz. Înlocuind în relaţiile de la punctul d, putem scrie:

$$3^n f(1-p) \geq 5^{n-n/2}(1-f)p \Leftrightarrow 3^n f(1-p) \geq 5^{n/2}(1-f)p \Leftrightarrow$$
$$3^n f(1-p) \geq (\sqrt{5})^n(1-f)p \Leftrightarrow \left(\frac{3}{\sqrt{5}}\right)^n \geq \frac{1-f}{f}\frac{p}{1-p} \Leftrightarrow$$
$$n \geq \frac{1}{\ln\frac{3}{\sqrt{5}}}\left(\ln\frac{1-f}{f} + \ln\frac{p}{1-p}\right).$$

g. Începem prin a calcula valorile logaritmilor:

$$f = 5\% \to \ln\frac{1-f}{f} = \ln\frac{95}{5} = \ln 19 \approx 2.94$$
$$f = 1\% \to \ln\frac{1-f}{f} = \ln\frac{99}{1} = \ln 99 \approx 4.60$$
$$p = 50\% \to \ln\frac{p}{1-p} = \ln\frac{50}{50} = \ln 1 = 0$$
$$p = 95\% \to \ln\frac{p}{1-p} = \ln\frac{95}{5} = \ln 19 \approx 2.94$$
$$p = 99\% \to \ln\frac{p}{1-p} = \ln\frac{99}{1} = \ln 99 \approx 4.60.$$

Valorile minime pentru n se calculează folosind formula de la punctul e, iar valorile medii pentru n se calculează folosind formula de la punctul f:

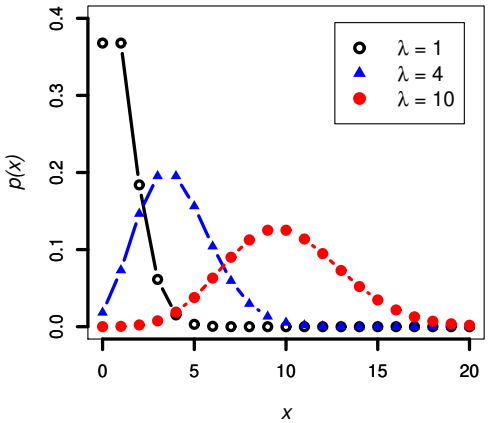
n_{min}	$f = 5\%$	$f = 1\%$
$p = 50\%$	2.68	4.18
$p = 95\%$	5.36	6.86
$p = 99\%$	6.86	8.36

n_{med}	$f = 5\%$	$f = 1\%$
$p = 50\%$	10.02	15.63
$p = 95\%$	20.03	25.65
$p = 99\%$	25.65	31.27

27. **(Distribuţia Poisson: verificarea condiţiilor de definiţie pentru p.m.f.; calculul mediei şi al varianţei)**

Liviu Ciortuz, 2017

Distribuţia Poisson: p.m.f.



Distribuţia *Poisson* este o distribuţie discretă de parametru $\lambda > 0$, a cărei funcţie masă de probabilitate este dată de expresia

$$p(x \mid \lambda) = \frac{1}{e^\lambda} \cdot \frac{\lambda^x}{x!}, \text{ pentru orice } x \in \mathbb{N}.$$

Prin convenţie, se consideră că $0! = 1$. Factorul $\frac{1}{e^\lambda}$, care nu depinde de x, este aşa-numita *constantă de normalizare*.

Demonstraţi mai întâi că funcţia $p(\ )$ este într-adevăr funcţie masă de probabilitate (engl., probability mass function, p.m.f.) şi apoi că media acestei distribuţii este λ, iar varianţa ei este tot λ.

***Sugestie*: Ţineţi cont că $\sum_{x=0}^{\infty} \frac{\lambda^x}{x!} = e^\lambda$ (limită fundamentală).**

Răspuns:

Vom arăta mai întâi că $p(\ )$ este într-adevăr o funcţie masă de probabilitate (p.m.f). Evident, $p(x|\lambda) > 0$ pentru orice $x \in \mathbb{N}$, fiindcă $\lambda > 0$. Apoi,

$$\sum_{x \in \mathbb{N}} \frac{1}{e^\lambda} \frac{\lambda^x}{x!} = \frac{1}{e^\lambda} \sum_{x=0}^{\infty} \frac{\lambda^x}{x!} = \frac{1}{e^\lambda} e^\lambda = 1,$$

ţinând cont de limita fundamentală care apare în *Sugestia* din enunţ.

Pentru calcularea mediei acestei distribuţii, aplicăm definiţia:

$$\sum_{x=0}^{\infty} x \frac{1}{e^\lambda} \frac{\lambda^x}{x!} = \frac{1}{e^\lambda} \sum_{x=0}^{\infty} x \frac{\lambda^x}{x!} = \frac{1}{e^\lambda} \sum_{x=1}^{\infty} \frac{\lambda^x}{(x-1)!} = \frac{\lambda}{e^\lambda} \sum_{x=1}^{\infty} \frac{\lambda^{x-1}}{(x-1)!} = \frac{\lambda}{e^\lambda} \underbrace{\sum_{x=0}^{\infty} \frac{\lambda^x}{x!}}_{e^\lambda} = \frac{\lambda}{e^\lambda} e^\lambda = \lambda.$$

În ce priveşte calculul varianţei pentru distribuţia Poisson, ştim că $Var[X] = E[X^2] - E^2[X]$ (proprietate demonstrată la problema 9.b) pentru orice distribuţie aleatoare X şi, prin urmare, pentru că am calculat deja media acestei distribuţii, trebuie să mai calculăm $\sum_{x=0}^{\infty} x^2 p(x|\lambda)$.

$$\sum_{x=0}^{\infty} x^2 p(x|\lambda) = \sum_{x=0}^{\infty} x^2 \frac{1}{e^\lambda} \frac{\lambda^x}{x!} = \frac{1}{e^\lambda} \sum_{x=1}^{\infty} x^2 \frac{\lambda^x}{x!} = \frac{1}{e^\lambda} \left[\sum_{x=1}^{\infty} [x(x-1) + x] \frac{\lambda^x}{x!} \right]$$

$$= \frac{1}{e^\lambda} \left[\sum_{x=1}^{\infty} x(x-1) \frac{\lambda^x}{x!} + \sum_{x=1}^{\infty} x \frac{\lambda^x}{x!} \right] = \frac{1}{e^\lambda} \left[\sum_{x=2}^{\infty} \frac{\lambda^x}{(x-2)!} + \sum_{x=1}^{\infty} \frac{\lambda^x}{(x-1)!} \right]$$

$$= \frac{1}{e^\lambda}\left[\lambda^2\sum_{x=2}^{\infty}\frac{\lambda^{x-2}}{(x-2)!}+\lambda\sum_{x=1}^{\infty}\frac{\lambda^{x-1}}{(x-1)!}\right] = \frac{1}{e^\lambda}\left[\lambda^2\sum_{x=0}^{\infty}\frac{\lambda^x}{x!}+\lambda\sum_{x=0}^{\infty}\frac{\lambda^x}{x!}\right]$$
$$= \frac{1}{e^\lambda}\left[\lambda^2 e^\lambda+\lambda e^\lambda\right] = \frac{1}{e^\lambda}e^\lambda(\lambda^2+\lambda) = \lambda^2+\lambda.$$

Prin urmare, varianţa distribuţiei Poisson este $\lambda^2+\lambda-\lambda^2=\lambda$.

28. **(Distribuţia geometrică: numărul „aşteptat“ / mediu de „observaţii“ necesare pentru ca un anumit eveniment să se producă)**

□ • ○ ***CMU, 2012 spring, Ziv Bar-Joseph, HW1, pr. 1.4***

În cazul unui zar perfect cu şase feţe, probabilităţile de apariţie pentru fiecare dintre feţele zarului sunt egale.

Mickey se duce la un cazinou şi doreşte ca, folosindu-şi cunoştinţele din domeniul probabilităţilor, să-şi evalueze şansa pe care o are de a obţine la aruncarea unui astfel de zar faţa 6.

Mai precis, Mickey se întreabă care este numărul mediu (sau, numărul „aşteptat“; engl., expected number) de aruncări ale zarului pe care ar trebui să le efectueze până să obţină faţa 6.

Justificaţi răspunsul în detaliu.

Răspuns:

Experimentul lui Mickey poate fi modelat cu ajutorul *distribuţiei geometrice*.[38] **Ştim că distribuţia geometrică poate fi reprezentată de o tabelă de „repartiţie“ de forma**

$$\begin{pmatrix} 1 & 2 & 3 & \dots & n & \dots \\ q & pq & p^2q & \dots & p^{n-1}q & \dots \end{pmatrix},$$

unde $p,q\geq 0$ şi $p+q=1$. Se verifică imediat că $\sum_{i=1}^{\infty}p^{i-1}q=1$ atunci când $p\in[0,1)$. În particular, pentru $p=\frac{5}{6}$ şi $q=\frac{1}{6}$, avem

$$\sum_{i=1}^{\infty}\frac{1}{6}\cdot\left(\frac{5}{6}\right)^{i-1}=1. \tag{16}$$

Vom nota cu E numărul mediu de aruncări ale zarului pe care ar trebui să le efectueze Mickey până să obţină faţa 6.

Conform definiţiei mediei, E se poate exprima astfel:

$$E = 1\cdot\frac{1}{6}+2\cdot\frac{1}{6}\cdot\frac{5}{6}+3\cdot\frac{1}{6}\cdot\left(\frac{5}{6}\right)^2\dots+n\cdot\frac{1}{6}\cdot\left(\frac{5}{6}\right)^{n-1}\dots+\dots.$$

[38] Simplu spus, distribuţia gemetrică poate fi gândită ca modelând următorul experiment aleatoriu: Fie o monedă a cărei probabilitate de apariţie a feţei-stemă este p. Aruncăm moneda o dată sau de mai multe ori, până când apare stema. Notăm numărul de aruncări care au precedat apariţia stemei cu k. Acest număr $k\in\{0,1,\dots\}$ va fi [asociat cu] valoarea unei variabile aleatoare X, despre care spunem că urmează distribuţia geometrică. Evident, $P(X=k)=(1-p)^k p$.

Punând termenul generic $n \cdot \frac{1}{6} \cdot \left(\frac{5}{6}\right)^{n-1}$ **sub forma** $(1+(n-1)) \cdot \frac{1}{6} \cdot \left(\frac{5}{6}\right)^{n-1}$ **şi aplicând apoi distributivitatea înmulţirii faţă de adunare, putem rescrie** E **în mod echivalent astfel:**

$$\begin{aligned} E &= \frac{1}{6} + \frac{5}{6} \cdot \left[\frac{1}{6} + \frac{1}{6} \cdot \frac{5}{6} + \ldots + \frac{1}{6} \cdot \left(\frac{5}{6}\right)^{n-1} + \ldots\right] \\ &\quad + \frac{5}{6} \cdot \left[1 \cdot \frac{1}{6} + 2 \cdot \frac{1}{6} \cdot \frac{5}{6} + \ldots + (n-1) \cdot \frac{1}{6} \cdot \left(\frac{5}{6}\right)^{n-1} + \ldots\right]. \end{aligned}$$

Ţinând cont de relaţia (16)**, egalitatea precedentă se rescrie astfel:**

$$E = \frac{1}{6} + \frac{5}{6} \cdot 1 + \frac{5}{6} \cdot E.$$

Rezultă imediat că $\frac{1}{6}E = \frac{1}{6} + \frac{5}{6}$**, deci** $E = 6$**.**

29. **(O mixtură de distribuţii categoriale: calculul mediei şi al varianţei)**

■ □ • ○ ***CMU, 2010 fall, Aarti Singh, HW1, pr. 2.2.1-2***

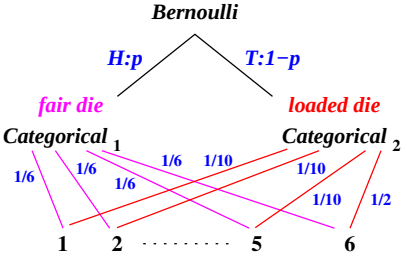
Presupunem că avem două zaruri cu şase feţe. Un zar este perfect — deci vom considera funcţia sa masă de probabilitate (p.m.f.) definită prin $P_1(x) = 1/6$ **pentru** $x = 1, \ldots, 6$ **—, iar celălalt zar este măsluit şi are următoarea funcţie masă de probabilitate:**

$$P_2(x) = \begin{cases} \frac{1}{2} & \text{pentru } x = 6; \\ \frac{1}{10} & \text{pentru } x \in \{1,2,3,4,5\}. \end{cases}$$

Pentru a decide ce zar să aruncăm, vom folosi o monedă; considerăm că probabilitatea să obţinem stema la aruncarea monedei este $p \in (0,1)$**. Dacă obţinem stema (engl., head), vom arunca apoi zarul perfect; în caz contrar vom arunca zarul măsluit.**

Putem reprezenta grafic „mixtura“ formată din cele două distribuţii categoriale ca în figura alăturată.

Notând cu X **variabila aleatoare corespunzătoare acestei mixturi şi cu** P **funcţia ei masă de probabilitate (p.m.f.), putem scrie:**



$P(i) = P(i|fair) \cdot p + P(i|loaded) \cdot (1-p) = P_1(i) \cdot p + P_2(i) \cdot (1-p)$ **pentru** $i = 1, \ldots, 6$.

a. Calculaţi în funcţie de p **media variabilei aleatoare** X**.**

b. Calculaţi în funcţie de p **varianţa variabilei aleatoare** X**.**

Răspuns:

a. Ţinând cont de modul în care a fost definită în enunţ mixtura celor două distribuţii categoriale, putem calcula media variabilei X, care reprezintă această mixtură, în felul următor:

$$\begin{aligned} E[X] &= \sum_{i=1}^{6} i \cdot P(i) = \sum_{i=1}^{6} i \cdot [P_1(i) \cdot p + P_2(i) \cdot (1-p)] \\ &= \Big[\sum_{i=1}^{6} i \cdot P_1(i)\Big]p + \Big[\sum_{i=1}^{6} i \cdot P_2(i)\Big](1-p) = \frac{7}{2} \cdot p + \frac{9}{2} \cdot (1-p) = \frac{9}{2} - p. \end{aligned}$$

Am folosit formula $\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$. La punctul următor vom avea nevoie de o altă formulă: $\sum_{i=1}^{n} i^2 = \frac{n(n+1)(2n+1)}{6}$.

b. Conform formulei $Var(X) = E[X^2] - (E[X])^2$, care a fost demonstrată la problema 9.b, va trebui sa calculăm $E[X^2]$. Procedăm astfel:

$$\begin{aligned} E[X^2] &= \sum_{i=1}^{6} i^2 \cdot P(i) = \sum_{i=1}^{6} i^2 \cdot [P_1(i) \cdot p + P_2(i) \cdot (1-p)] \\ &= \Big[\sum_{i=1}^{6} i^2 \cdot P_1(i)\Big]p + \Big[\sum_{i=1}^{6} i^2 \cdot P_2(i)\Big](1-p) = \frac{91}{6} \cdot p + \Big(\frac{55}{10} + \frac{36}{2}\Big) \cdot (1-p) \\ &= \frac{47}{2} - \frac{25}{3} \cdot p. \end{aligned}$$

Combinând acest rezultat cu cel pe care l-am obţinut la punctul precedent, vom obţine:

$$\begin{aligned} Var(X) &= E[X^2] - (E[X])^2 = \frac{47}{2} - \frac{25}{3}p - \left(\frac{9}{2} - p\right)^2 = \frac{47}{2} - \frac{25}{3}p - \left(\frac{81}{4} - 9p + p^2\right) \\ &= \left(\frac{47}{2} - \frac{81}{4}\right) - \left(\frac{25}{3} - 9\right) \cdot p - p^2 = \frac{13}{4} + \frac{2}{3} \cdot p - p^2. \end{aligned}$$

30. **(Variabile aleatoare uniforme continue, independente; funcţii densitate de probabilitate)**

CMU, 2008 spring, Eric Xing, HW1, pr. 1.5

O persoană pleacă la serviciu între orele 8:00 şi 8:30, iar timpul necesar deplasării este între 40 şi 50 de minute. Considerăm X variabila aleatoare care reprezintă timpul de plecare exprimat în minute scurse după ora 8:00 şi Y variabila aleatoare care reprezintă durata deplasării. Presupunând că aceste două variabile sunt independente şi uniform distribuite, calculaţi:

a. funcţiile densitate de probabilitate $p(x) \overset{not.}{=} P_X(X = x)$, $p(y) \overset{not.}{=} P_Y(Y = y)$ şi $p(x, y) \overset{not.}{=} P_{X,Y}(X = x, Y = y)$.

b. probabilitatea ca acea persoană să ajungă la serviciu înainte de ora 9.

Răspuns:

a. Conform enunţului $Val(X) = [0, 30]$ şi $Val(Y) = [40, 50]$. Pentru a determina funcţia densitate de probabilitate pentru X şi respectiv Y vom impune restricţia ca integrala valorilor pe domeniul de definiţie să fie 1:

$$\int_{-\infty}^{\infty} p(x)dx = 1 \Leftrightarrow \int_{0}^{30} p(x)dx = 1 \overset{p-unif.}{\Leftrightarrow} p(x) = \begin{cases} \frac{1}{30} & \textbf{pentru } 0 \leq x \leq 30 \\ 0 & \textbf{în caz contrar}. \end{cases}$$

$$\int_{-\infty}^{\infty} p(y)dy = 1 \Leftrightarrow \int_{40}^{50} p(y)dy = 1 \overset{p-unif.}{\Leftrightarrow} p(y) = \begin{cases} \frac{1}{10} & \textbf{pentru } 40 \leq y \leq 50 \\ 0 & \textbf{în caz contrar}. \end{cases}$$

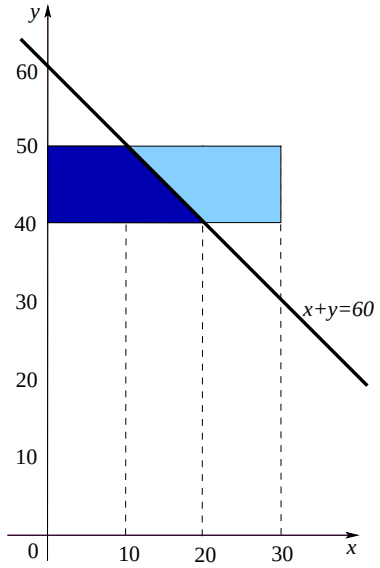
Deoarece variabilele X şi Y sunt independente, funcţia densitate de probabilitate comună este:

$$p(x, y) = p(x) \cdot p(y) = \begin{cases} \frac{1}{300} & \textbf{pentru } 0 \leq x \leq 30 \textbf{ şi } 40 \leq y \leq 50 \\ 0 & \textbf{în caz contrar}. \end{cases}$$

b. Probabilitatea ca persoana respectivă să ajungă la serviciu înainte de ora 9 este $P(X + Y \leq 60)$.

***Grafic*, aşa cum se observă din figura alăturată, această probabilitate este foarte simplu de găsit: $p(x + y \leq 60) = \frac{1}{2}$. Într-adevăr, ştim că aria întregii zone dreptunghiulare pe care $p(x, y) \neq 0$ este**

$$\int_{x=0}^{30} \int_{y=40}^{50} p(x, y)dydx = 1.$$



Probabilitatea urmărită este aria zonei din acest dreptunghi care se găseşte „sub" dreapta de ecuaţie $x + y = 60$. Această zonă corespunde perechilor de valori (x, y) care satisfac condiţia ca persoana sa ajungă la serviciu înainte de ora 9.

***Alternativ*, această arie se poate obţine prin calcul direct:**

$$P(X + Y \leq 60) = \int_{x=0}^{10} \int_{y=40}^{50} \frac{1}{300} dydx + \int_{x=10}^{20} \int_{y=40}^{60-x} \frac{1}{300} dydx$$

$$= \int_{x=0}^{10} \frac{1}{300} \left(\int_{x=40}^{50} dy \right) dx + \int_{x=10}^{20} \frac{1}{300} \left(\int_{y=40}^{60-x} dy \right) dx$$

$$= \int_{x=0}^{10} \frac{1}{300} \cdot y|_{40}^{50} dx + \int_{x=10}^{20} \frac{1}{300} \cdot y|_{40}^{60-x} dx = \int_{x=0}^{10} \frac{1}{300} \cdot 10 dx + \int_{x=10}^{20} \frac{1}{300} \cdot (20 - x)dx$$

$$= \frac{1}{30} \cdot x|_{0}^{10} + \frac{1}{300} \cdot \left(20x - \frac{1}{2}x^2 \right) \Big|_{10}^{20} = \frac{10}{30} + \frac{50}{300} = \frac{1}{2}.$$

31. **(Funcţia Gamma a lui Euler; distribuţia exponenţială şi distribuţia Gamma: verificarea condiţiilor de definiţie pentru p.d.f., calculul mediilor şi al varianţelor)**

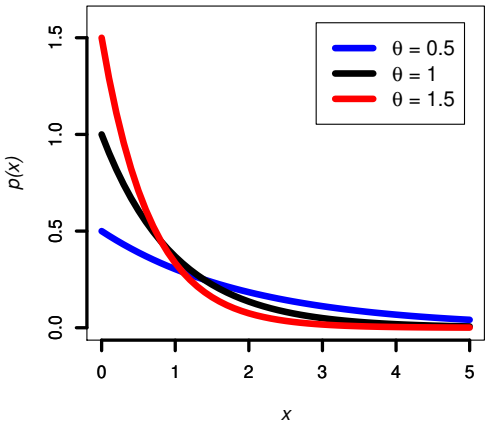
Liviu Ciortuz, 2017

a. Distribuţia *exponenţială* este o distribuţie continuă, care are funcţia densitate de probabilitate

$$p(x \mid \theta) = \begin{cases} \theta e^{-\theta x} & \textbf{pentru } x \geq 0 \\ 0 & \textbf{pentru } x < 0. \end{cases}$$

unde $\theta > 0$ este un parametru real. Arătaţi mai întâi că funcţia $p(\ )$ este într-adevăr funcţie densitate de probabilitate (engl., probability density function, p.d.f.) şi apoi că media şi respectiv varianţa distribuţiei exponenţiale sunt $\frac{1}{\theta}$ şi respectiv $\frac{1}{\theta^2}$.

Distribuţia exponenţială: p.d.f.



b. Funcţia Γ a lui Euler (1707-1783) este o generalizare în $\mathbb{R}^+$ a definiţiei numerelor factoriale din $\mathbb{N}$ (şi anume, $\Gamma(r) = (r-1)!$ pentru orice $r \in \mathbb{N}^*$).

Formula de definiţie a acestei funcţii este următoarea:

$$\Gamma(r) \overset{def.}{=} \int_0^{+\infty} t^{r-1} e^{-t}\, dt \quad \textbf{pentru orice } r > 0.$$

Demonstraţi următoarele proprietăţi ale funcţiei Γ:

i. $\Gamma(r+1) = r\,\Gamma(r)$ **pentru orice** $r > 0$**.**

ii. $\Gamma(1) = 1$ **şi** $\Gamma(1/2) = \sqrt{\pi}$**.**[39]

***Sugestie*: Puteţi folosi următoarea proprietate, care este demonstrată la problema 32 (vedeţi relaţia (21)):**

$$\int_{-\infty}^{\infty} v^2\, e^{-\frac{v^2}{2}}\, dv = 2 \int_0^{\infty} v^2\, e^{-\frac{v^2}{2}}\, dv = \sqrt{2\pi}. \tag{17}$$

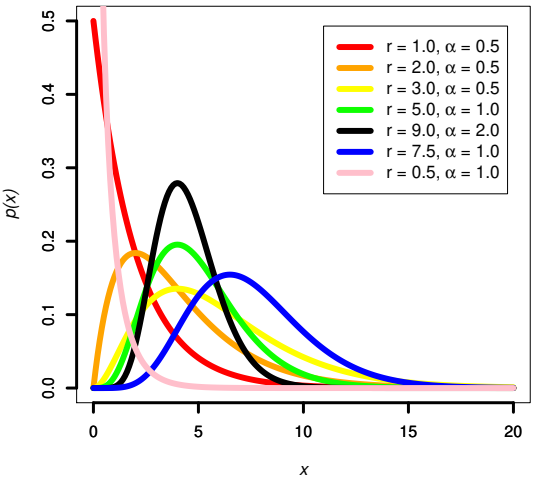
[39] Din relaţia de recurenţă *i.* şi din relaţia $\Gamma(1) = 1$, rezultă $\Gamma(r+1) = r \cdot \Gamma(n) = r \cdot (r-1) \cdot \Gamma(r-1) = \ldots = r \cdot (r-1) \cdot \ldots \cdot 2 \cdot 1 = r!$. Aşadar, într-adevăr funcţia Γ a lui Euler generalizează noţiunea de *produs factorial*.

c. Distribuţia *Gamma* este o distribuţie continuă, de parametri $r > 0$ (care dă *forma* distribuţiei, engl., shape) şi $\alpha > 0$ (numit *rata*, engl., rate), cu funcţia densitate de probabilitate definită pe $\mathbb{R}^+$ prin expresia următoare:

$$p(x) \overset{not.}{=} Gamma(x|r,\alpha) \overset{def.}{=} \frac{\alpha^r}{\Gamma(r)} x^{r-1} e^{-\alpha x} \text{ pentru } x \geq 0,$$

unde *constanta de normalizare* este $\frac{\alpha^r}{\Gamma(r)}$.

Distribuţia Gamma: p.d.f.



Demonstraţi mai întâi că funcţia $p(\ )$ este într-adevăr p.d.f. şi apoi că media distribuţiei Gamma este $\frac{r}{\alpha}$, iar varianţa ei este $\frac{r}{\alpha^2}$.

***Observaţie*: Se poate vedea imediat că $Gamma(x|1,\alpha) = \alpha e^{-\alpha x}$ pentru orice $x \geq 0$, ceea ce corespunde [definiţiei funcţiei de densitate a] distribuţiei exponenţiale (vedeţi punctul a). Aşadar, se poate spune că distribuţia exponenţială este membru al familiei de distribuţii Gamma.**

Răspuns:

a. Verificăm mai întâi că funcţia $p(x|\theta)$ dată în enunţ este într-adevăr o funcţie densitate de probabilitate. Evident, $p(x|\theta) \geq 0$ pentru orice x, fiindcă $\theta > 0$ şi $e^{-\theta x} > 0$ pentru orice x. Apoi,

$$\int_0^\infty \theta e^{-\theta x} dx = -\int_0^\infty (e^{-\theta x})' dx = e^{-\theta x}\Big|_\infty^0 = 1 - 0 = 1.$$

La calculul mediei distribuţiei exponenţiale vom folosi formula de integrare prin părţi:

$$\begin{aligned}\int_0^\infty x\theta e^{-\theta x} dx &= -\int_0^\infty x(e^{-\theta x})' dx = -x(e^{-\theta x})\Big|_0^\infty + \int_0^\infty x' e^{-\theta x} dx \\ &= 0 + \int_0^\infty e^{-\theta x} dx = -\frac{1}{\theta} e^{-\theta x}\Big|_0^\infty = -\frac{1}{\theta}(0-1) = \frac{1}{\theta}.\end{aligned}$$

Facem menţiunea că pentru a deduce $x(e^{-\theta x})\Big|_0^\infty = 0$, am ţinut cont că $\lim_{x\to\infty} \frac{x}{e^{\theta x}} = 0$, conform regulii lui l'Hôpital.

Pentru calculul varianţei distribuţiei exponenţiale, vom apela la formula $Var[X] = E[X^2] - E^2[X]$ (pe care am demonstrat-o la problema 9.b), unde X este o variabilă aleatoare oarecare. Întrucât în cazul nostru cunoaştem deja $E[X]$, calculăm $E[X^2]$:

$$\begin{aligned}&\int_0^\infty x^2 \theta e^{-\theta x} dx = -\int_0^\infty x^2 (e^{-\theta x})' dx = -x^2(e^{-\theta x})\Big|_0^\infty + \int_0^\infty (x^2)' e^{-\theta x} dx \\ &= 0 + 2\int_0^\infty x e^{-\theta x} dx = 2\int_0^\infty x e^{-\theta x} dx = 2\frac{1}{\theta}\int_0^\infty x\theta e^{-\theta x} dx = 2\frac{1}{\theta}E[X] = \frac{2}{\theta^2}.\end{aligned}$$

Ca şi mai înainte, folosind regula lui l'Hôpital, am dedus $x^2(e^{-\theta x})\Big|_0^\infty = 0$. **Aşadar,** $Var[X] = \frac{2}{\theta^2} - \left(\frac{1}{\theta}\right)^2 = \frac{1}{\theta^2}$.

b. Pentru demonstrarea relaţiei *i.*, $\Gamma(r+1) = r\Gamma(r)$, **vom folosi formula de integrare prin părţi:**

$$\Gamma(r+1) = \int_0^\infty t^r e^{-t} dt = \int_0^\infty t^r (-e^{-t})' dt = -e^{-t}t^r\Big|_0^\infty + r\int_0^\infty t^{r-1}e^{-t}dt = r\Gamma(r).$$

Am ţinut cont că $\lim_{t\to\infty} \frac{t^r}{e^t} = 0$ **(pentru** $r \in \mathbb{N}^*$ **se poate folosi teorema lui l'Hôpital, însă pentru** $r \in \mathbb{R}$, $r > 0$ **se foloseşte teorema cleştelui).**

ii. **Valoarea funcţiei** Γ **în** 1 **se calculează astfel:**

$$\Gamma(1) = \int_0^\infty e^{-t}dt = \int_0^\infty (-e^{-t})' dt = -e^{-t}\Big|_0^\infty = 1 - 0 = 1.$$

În sfârşit, calculăm valoarea funcţiei Γ **în** $1/2$**:**

$$\Gamma\left(\frac{1}{2}\right) = \int_0^\infty t^{-\frac{1}{2}}e^{-t}dt = \int_0^\infty \frac{1}{\sqrt{t}}e^{-t}dt = \int_0^\infty (\sqrt{t})' \cdot 2 \cdot e^{-t}dt = 2\underbrace{\sqrt{t}e^{-t}\Big|_0^\infty}_{0} + 2\int_0^\infty \sqrt{t}e^{-t}dt.$$

Făcând schimbarea de variabilă $\sqrt{t} = \frac{v}{\sqrt{2}} \Leftrightarrow t = \frac{v^2}{2}$ **cu** $v \geq 0$, **rezultă** $dt = v\,dv$ **şi**

$$\Gamma\left(\frac{1}{2}\right) = 2\int_0^\infty \sqrt{t}e^{-t}dt = 2\int_0^\infty \frac{v}{\sqrt{2}}\, e^{-\frac{v^2}{2}}\, v\, dv = \sqrt{2}\int_0^\infty v^2\, e^{-\frac{v^2}{2}}\, dv \overset{(17)}{=} \sqrt{2}\cdot\frac{1}{2}\cdot\sqrt{2\pi} = \sqrt{\pi}.$$

c. Folosind notaţia din enunţ, unde $p(\ )$ **desemnează funcţia densitate de probabilitate a distribuţiei Gamma, vom arăta mai întâi că** $p(\ )$ **este într-adevăr funcţie densitate de probabilitate. Evident,** $p(x) > 0$ **pentru orice** $x > 0$, **pentru că** α **şi** r **sunt strict pozitive. Apoi, a verifica condiţia** $\int_0^{+\infty} p(x)dx = 1$ **revine la a arăta că** $\int_0^{+\infty} x^{r-1}e^{-\alpha x}\, dx = \frac{\Gamma(r)}{\alpha^r}$. **Într-adevăr, făcând schimbarea de variabilă** $y = \alpha x$, **care implică** $dy = \alpha\, dx$, **urmează**

$$\int_0^{+\infty} x^{r-1}e^{-\alpha x}\, dx = \int_0^{+\infty} \left(\frac{y}{\alpha}\right)^{r-1} e^{-y}\cdot\frac{1}{\alpha}\, dy = \frac{1}{\alpha^r}\int_0^{+\infty} y^{r-1}e^{-y}\, dy = \frac{\Gamma(r)}{\alpha^r}. \quad (18)$$

Acum arătăm că media distribuţiei *Gamma* **este** $\frac{r}{\alpha}$**:**

$$\int_0^{+\infty} xp(x)\, dx = \frac{\alpha^r}{\Gamma(r)}\int_0^{+\infty} x^r e^{-\alpha x}\, dx \overset{(18)}{=} \frac{\alpha^r}{\Gamma(r)}\cdot\frac{\Gamma(r+1)}{\alpha^{r+1}} \overset{b.i}{=} \frac{\alpha^r}{\Gamma(r)}\cdot\frac{r\,\Gamma(r)}{\alpha^{r+1}} = \frac{r}{\alpha}.$$

În sfârşit, arătăm că varianţa distribuţiei *Gamma* **este** $\frac{r}{\alpha^2}$. **Apelăm încă o dată la formula** $Var[X] = E[X^2] - E^2[X]$. **Pentru că am calculat deja media distribuţiei** *Gamma*, **vom calcula acum** $\int_0^{+\infty} x^2 p(x)\, dx$.

$$\int_0^{+\infty} x^2 p(x)\,dx = \frac{\alpha^r}{\Gamma(r)}\int_0^{+\infty} x^2 x^{r-1}e^{-\alpha x}\,dx = \frac{\alpha^r}{\Gamma(r)}\int_0^{+\infty} x^{r+1}e^{-\alpha x}\,dx$$
$$\overset{(18)}{=} \frac{\alpha^r}{\Gamma(r)}\cdot\frac{\Gamma(r+2)}{\alpha^{r+2}} \overset{b.i}{=} \frac{(r+1)r\,\Gamma(r)}{\Gamma(r)\,\alpha^2} = \frac{r(r+1)}{\alpha^2}.$$

În consecinţă, varianţa distribuţiei *Gamma* este

$$\frac{r(r+1)}{\alpha^2} - \left(\frac{r}{\alpha}\right)^2 = \frac{r}{\alpha^2}.$$

Menţionăm faptul că dând lui r valoarea 1 în formulele pe care tocmai le-am obţinut pentru media şi varianţa distribuţiei Gamma, obţinem media şi respectiv varianţa distribuţiei exponenţiale, ceea ce era de altfel de aşteptat, conform *Observaţiei* din enunţ.

32. **(Distribuţia gaussiană unidimensională: verificarea condiţiilor de definiţie pentru p.d.f., calculul mediei şi al varianţei)**

prelucrare de Liviu Ciortuz, după
■ ***CMU, 2010 spring, T. Mitchel, E. Xing, A. Singh, HW1, pr. 1.3.2***

Considerăm o variabilă aleatoare X care urmează distribuţia normală (gaussiană):[40]

$$\mathcal{N}(X = x) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

unde σ poate fi orice număr real pozitiv, iar μ orice număr real.

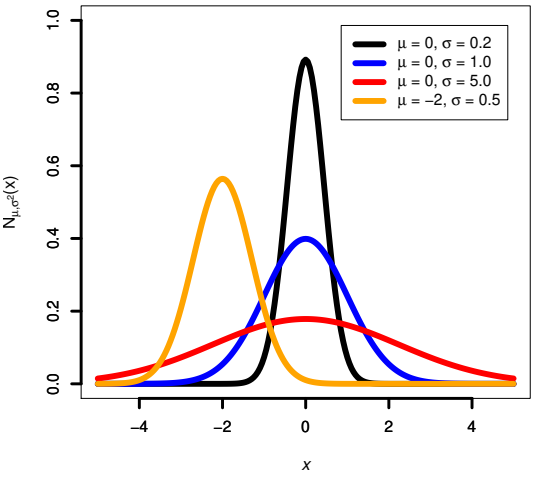
Arătaţi că:

a. $\mathcal{N}$ **este într-adevăr o funcţie de densitate de probabilitate (p.d.f.), adică** $\int_{-\infty}^{+\infty}\mathcal{N}(x)\,dx = 1$.

b. $E[X] = \mu$.

c. $Var[X] = \sigma^2$.

Distribuţia gaussiană: p.d.f.



***Sugestie* (pentru punctul a): Pentru cazul distribuţiei gaussiene *standard* — la care veţi face „reducere", de la cazul *general*, printr-o schimbare liniară de variabilă —, proprietatea de demonstrat devine**

$$\int_{-\infty}^{+\infty}\frac{1}{\sqrt{2\pi}}e^{-\frac{x^2}{2}}\,dx = 1 \text{ sau, echivalent } \int_{-\infty}^{+\infty}e^{-\frac{x^2}{2}}\,dx = \sqrt{2\pi}.$$

[40] După numele lui Karl Friedrich Gauss (1777 – 1855), matematician, fizician şi astronom german.

Pentru demonstrarea ultimei egalităţi vă recomandăm să arătaţi că

$$\left(\int_{-\infty}^{+\infty} e^{-\frac{x^2}{2}}\,dx\right)^2 = 2\pi \text{ sau, echivalent } \left(\int_{-\infty}^{+\infty} e^{-\frac{x^2}{2}}\,dx\right)\cdot\left(\int_{-\infty}^{+\infty} e^{-\frac{y^2}{2}}\,dy\right) = 2\pi,$$

deci

$$\int_{x=-\infty}^{+\infty}\int_{y=-\infty}^{+\infty} e^{-\frac{x^2+y^2}{2}}\,dx\,dy = 2\pi.$$

Ultima egalitate poate fi demonstrată prin trecerea din sistemul de coordonate cartezian în sistemul de coordonate polare. În mod concret, $(x,y)\mapsto(r,\theta)$**, unde** $r\in[0,+\infty)$ **şi** $\theta\in[0,2\pi)$**, cu** $x = r\cos\theta$ **şi** $y = r\sin\theta$**, ceea ce constituie o corespondenţă bijectivă. Vă reamintim că *regula de schimbare de variabilă* pentru cazul vectorial, formulată (aici) pentru cazul probabilist, este următoarea:**[41]

Presupunem că $V = [V_1\ldots V_n]^\top\in\mathbb{R}^n$ **este un vector de variabile aleatoare având funcţia de densitate de probabilitate comună** $f_V:\mathbb{R}^n\to\mathbb{R}$**. Dacă definim un alt vector de variabile aleatoare,** Z**, obţinut prin compunerea** $Z = H(V)$**, unde** $H:\mathbb{R}^n\to\mathbb{R}^n$ **este o funcţie bijectivă şi derivabilă [pe componente] în raport cu fiecare dintre argumentele sale, atunci** Z **va avea funcţia de densitate de probabilitate comună** $f_Z:\mathbb{R}^n\to\mathbb{R}$**, definită prin expresia**

$$f_Z(z) = f_V(v)\cdot\left|\det\left(\begin{bmatrix}\frac{\partial v_1}{\partial z_1} & \cdots & \frac{\partial v_1}{\partial z_n}\\ \vdots & \ddots & \vdots\\ \frac{\partial v_n}{\partial z_1} & \cdots & \frac{\partial v_n}{\partial z_n}\end{bmatrix}\right)\right|. \tag{19}$$

Matricea al cărei determinant este calculat în expresia de mai sus se numeşte matrice *jacobiană*, iar determinantul respectiv se numeşte *determinant jacobian*.[42]

Răspuns:

a. Folosind schimbarea de variabilă *sugerată* în enunţ, integrala dublă

$$\int_{x=-\infty}^{+\infty}\int_{y=-\infty}^{+\infty} e^{-\frac{x^2+y^2}{2}}\,dx\,dy$$

devine

$$\int_{r=0}^{+\infty}\int_{\theta=0}^{2\pi} e^{-\frac{r^2}{2}}\,r\,dr\,d\theta,$$

întrucât avem determinantul jacobian[43]

$$\left|\frac{\partial(x,y)}{\partial(r,\theta)}\right| = \begin{vmatrix}\frac{\partial x}{\partial r} & \frac{\partial x}{\partial\theta}\\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial\theta}\end{vmatrix} = \begin{vmatrix}\frac{\partial r\cos\theta}{\partial r} & \frac{\partial r\cos\theta}{\partial\theta}\\ \frac{\partial r\sin\theta}{\partial r} & \frac{\partial r\sin\theta}{\partial\theta}\end{vmatrix} = \begin{vmatrix}\cos\theta & -r\sin\theta\\ \sin\theta & r\cos\theta\end{vmatrix}$$
$$= r\cos^2\theta + r\sin^2\theta = r\geq 0,$$

[41] Cf. *The Multivariate Gaussian Distribution*, Chuong Do, Stanford University, 2008.
[42] După numele matematicianului german evreu Carl Gustav Jacob Jacobi (1804 – 1851).
[43] În relaţia (19) vom face următoarele înlocuiri: $z_1 = r$, $z_2 = \theta$, $v_1 = x = r\cos\theta$, $v_2 = y = r\sin\theta$.

de unde a rezultat $dx\,dy = r\,dr\,d\theta$**.**

În continuare, vom putea scrie:

$$\int_{r=0}^{\infty}\int_{\theta=0}^{2\pi} e^{-\frac{r^2}{2}}\,(r\,dr\,d\theta) = \int_{r=0}^{\infty} re^{-\frac{r^2}{2}}\left(\int_{\theta=0}^{2\pi} d\theta\right)dr = \int_{r=0}^{\infty} re^{-\frac{r^2}{2}}\,\theta\Big|_0^{2\pi} dr$$

$$= \quad 2\pi\int_{r=0}^{\infty} re^{-\frac{r^2}{2}}\,dr = 2\pi\,(-e^{-\frac{r^2}{2}})\Big|_0^{\infty} = 2\pi\,e^{-\frac{r^2}{2}}\Big|_{\infty}^{0} = 2\pi(1-0) = 2\pi,$$

ceea ce practic finalizează demonstraţia pentru *cazul* distribuţiei gaussiene unidimensionale *standard*.

Pentru *cazul nestandard*, folosim schimbarea de variabilă $v = \dfrac{x-\mu}{\sigma} \Rightarrow x = \sigma v + \mu$ **cu** $dx = \sigma dv$ **şi obţinem:**

$$\int_{-\infty}^{\infty}\mathcal{N}(x)dx \quad = \quad \int_{-\infty}^{\infty}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}\,dx = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty}e^{-\frac{(x-\mu)^2}{2\sigma^2}}\,dx$$

$$= \quad \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty}e^{-\frac{v^2}{2}}\,\sigma dv = \frac{1}{\sqrt{2\pi}\sigma}\sigma\int_{-\infty}^{\infty}e^{-\frac{v^2}{2}}\,dv = \frac{1}{\sqrt{2\pi}}\sqrt{2\pi} = 1,$$

ceea ce era de demonstrat.

b. Vom calcula media variabilei aleatoare X **folosind formula de definiţie:**

$$E[X] \stackrel{def.}{=} \int_{-\infty}^{\infty} xp(x)dx = \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty} x\cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}}\,dx.$$

Facând (din nou) schimbarea de variabilă $v = \dfrac{x-\mu}{\sigma} \Rightarrow x = \sigma v + \mu$ **şi** $dx = \sigma dv$**, vom obţine:**

$$E[X] \quad = \quad \frac{1}{\sqrt{2\pi}\sigma}\int_{-\infty}^{\infty}(\sigma v+\mu)e^{-\frac{v^2}{2}}\,(\sigma dv) = \frac{\sigma}{\sqrt{2\pi}\sigma}\left(\sigma\int_{-\infty}^{\infty} ve^{-\frac{v^2}{2}}\,dv + \mu\int_{-\infty}^{\infty}e^{-\frac{v^2}{2}}\,dv\right)$$

$$= \quad \frac{1}{\sqrt{2\pi}}\left(-\sigma\int_{-\infty}^{\infty}(-v)e^{-\frac{v^2}{2}}\,dv + \mu\int_{-\infty}^{\infty}e^{-\frac{v^2}{2}}\,dv\right)$$

$$= \quad \frac{1}{\sqrt{2\pi}}\left(-\sigma\underbrace{e^{-\frac{v^2}{2}}\Big|_{-\infty}^{\infty}}_{=0} + \mu\int_{-\infty}^{\infty}e^{-\frac{v^2}{2}}\,dv\right) = \frac{\mu}{\sqrt{2\pi}}\underbrace{\int_{-\infty}^{\infty}e^{-\frac{v^2}{2}}\,dv}_{=\sqrt{2\pi}} = \mu.$$

La ultima egalitate am folosit rezultatul obţinut la punctul a **(cazul standard).**

c. Trebuie să arătăm că $Var[X] = \sigma^2$**. Vom calcula varianţa lui** X **utilizând formula** $Var[X] = E[X^2] - (E[X])^2$**. Întrucât am calculat** $E[X]$**, trebuie să mai calculăm valoarea mediei lui** X^2**.**

$$E[X^2] = \int_{-\infty}^{\infty} x^2 p(x) dx = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} x^2 \cdot e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx.$$

Făcând aceeaşi schimbare de variabilă $v = \dfrac{x-\mu}{\sigma} \Rightarrow x = \sigma v + \mu$**, cu** $dx = \sigma dv$**, obţinem:**

$$\begin{aligned} E[X^2] &= \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} (\sigma v + \mu)^2 \, e^{-\frac{v^2}{2}} (\sigma dv) = \frac{\sigma}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} (\sigma^2 v^2 + 2\sigma\mu v + \mu^2) \, e^{-\frac{v^2}{2}} dv \\ &= \frac{1}{\sqrt{2\pi}} \left(\sigma^2 \int_{-\infty}^{\infty} v^2 \, e^{-\frac{v^2}{2}} dv + 2\sigma\mu \int_{-\infty}^{\infty} v \, e^{-\frac{v^2}{2}} dv + \mu^2 \int_{-\infty}^{\infty} e^{-\frac{v^2}{2}} dv \right). \quad (20) \end{aligned}$$

Ştim de la punctul a **că** $\int_{-\infty}^{\infty} e^{-\frac{v^2}{2}} dv = \sqrt{2\pi}$**, iar de la punctul** b **că** $\int_{-\infty}^{\infty} v e^{-\frac{v^2}{2}} dv = 0$**.**

Mai rămâne să calculăm prima integrală din expresia (20)**. Pentru aceasta vom utiliza metoda de integrare prin părţi:**

$$\begin{aligned} \int_{-\infty}^{\infty} v^2 e^{-\frac{v^2}{2}} dv &= \int_{-\infty}^{\infty} (-v) \left(-v e^{-\frac{v^2}{2}} \right) dv = \int_{-\infty}^{\infty} (-v) \left(e^{-\frac{v^2}{2}} \right)' dv \\ &= (-v) \, e^{-\frac{v^2}{2}} \Big|_{-\infty}^{\infty} - \int_{-\infty}^{\infty} (-1) e^{-\frac{v^2}{2}} dv = 0 + \int_{-\infty}^{\infty} e^{-\frac{v^2}{2}} dv = \sqrt{2\pi}. \quad (21) \end{aligned}$$

Pentru a calcula valoarea integralei definite am folosit următoarea proprietate:

$$\lim_{v\to\infty} v \, e^{-\frac{v^2}{2}} = \lim_{v\to\infty} \frac{v}{e^{\frac{v^2}{2}}} \overset{l'Hôpital}{=} \lim_{v\to\infty} \frac{1}{v \, e^{\frac{v^2}{2}}} = 0 = \lim_{v\to-\infty} v \, e^{-\frac{v^2}{2}}.$$

Aşadar,

$$E[X^2] = \frac{1}{\sqrt{2\pi}} \left(\sigma^2 \sqrt{2\pi} + 2\sigma\mu \cdot 0 + \mu^2 \sqrt{2\pi} \right) = \sigma^2 + \mu^2.$$

Deci varianţa variabilei aleatoare $X \sim \mathcal{N}(\mu, \sigma^2)$ **este:**

$$Var[X] = E[X^2] - (E[X])^2 = (\sigma^2 + \mu^2) - \mu^2 = \sigma^2.$$

33. **(Distribuţia gaussiană unidimensională: reducerea cazului nestandard la cazul standard, folosind funcţia cumulativă de distribuţie)**

CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW1, pr. 3

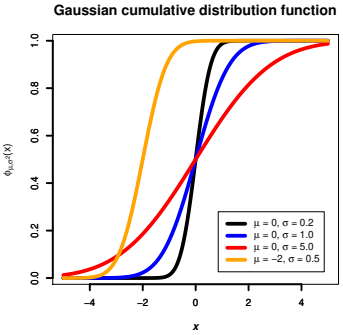
Considerăm variabila aleatoare X **cu distribuţia normală de medie** $\mu = 1$ **şi varianţă** $\sigma^2 = 4$**. Calculaţi următoarele probabilităţi:**

a. $P(X \leq 3)$.

b. $P(|X| \leq 2)$.

***Indicaţie*: Pentru a calcula aceste probabilităţi, soluţia este să transformaţi variabila aleatoare X în distribuţia normală standard Z după formula: $Z = \frac{X-\mu}{\sigma}$. Pentru distribuţia probabilistă normală standard ($\mu = 0$ şi $\sigma = 1$), valorile funcţiei de distribuţie cumulativă (engl., cumulative distribution function), notată Φ şi definită prin relaţia $\Phi(x) \overset{not.}{=} P(Z \leq x)$, se consideră că sunt deja calculate. Puteţi folosi următorul tabel de valori:**

x	$\Phi(x)$	x	$\Phi(x)$	x	$\Phi(x)$	x	$\Phi(x)$
−3.4	0.0003	−1.7	0.0446	0.0	0.5000	1.7	0.9554
−3.3	0.0005	−1.6	0.0548	0.1	0.5398	1.8	0.9641
−3.2	0.0007	−1.5	0.0668	0.2	0.5793	1.9	0.9713
−3.1	0.0010	−1.4	0.0808	0.3	0.6179	2.0	0.9772
−3.0	0.0013	−1.3	0.0968	0.4	0.6554	2.1	0.9821
−2.9	0.0019	−1.2	0.1151	0.5	0.6915	2.2	0.9861
−2.8	0.0026	−1.1	0.1357	0.6	0.7257	2.3	0.9893
−2.7	0.0035	−1.0	0.1587	0.7	0.7580	2.4	0.9918
−2.6	0.0062	−0.9	0.1841	0.8	0.7881	2.5	0.9938
−2.5	0.0062	−0.8	0.2119	0.9	0.8159	2.6	0.9953
−2.4	0.0082	−0.7	0.2420	1.0	0.8413	2.7	0.9965
−2.3	0.0107	−0.6	0.2743	1.1	0.8643	2.8	0.9974
−2.2	0.0139	−0.5	0.3085	1.2	0.8849	2.9	0.9981
−2.1	0.0179	−0.4	0.3446	1.3	0.9032	3.0	0.9987
−2.0	0.0228	−0.3	0.3821	1.4	0.9192	3.1	0.9990
−1.9	0.0287	−0.2	0.4207	1.5	0.9332	3.2	0.9993
−1.8	0.0359	−0.1	0.4602	1.6	0.9452	3.3	0.9995



Notă: Pentru distribuţia normală standard (Φ, în textul problemei), vedeţi curba de culoare albastră.

Răspuns:

a.

$$P(X \leq 3) \overset{not.}{=} P(\{\omega | X(\omega) \leq 3\}) = P\left(\left\{\omega \middle| \frac{X(\omega)-\mu}{\sigma} \leq \frac{3-\mu}{\sigma}\right\}\right)$$
$$\overset{not.}{=} P\left(\frac{X-\mu}{\sigma} \leq \frac{3-\mu}{\sigma}\right) = P\left(Z \leq \frac{3-1}{\sqrt{4}}\right) = P(Z \leq 1) = \Phi(1) = 0.8413.$$

b. Vom proceda analog, descompunând probabilitatea $P(|X| \leq 2) = P(-2 \leq X \leq 2)$ într-o diferenţă de două probabilităţi:

$$\begin{aligned} P(|X| \leq 2) &= P(-2 \leq X \leq 2) = P(X \leq 2) - P(X \leq -2) \\ &= P(X-1 \leq 2-1) - P(X-1 \leq -2-1) \\ &= P\left(\frac{X-1}{\sqrt{4}} \leq \frac{2-1}{\sqrt{4}}\right) - P\left(\frac{X-1}{\sqrt{4}} \leq \frac{-2-1}{\sqrt{4}}\right) \\ &= P\left(Z \leq \frac{2-1}{2}\right) - P\left(Z \leq \frac{-2-1}{2}\right) = \Phi\left(\frac{1}{2}\right) - \Phi\left(-\frac{3}{2}\right) \\ &= 0.6915 - 0.0668 = 0.6247 \end{aligned}$$

34. (Distribuţii gaussiene multidimensionale: o proprietate importantă, în cazul în care matricea de covarianţă este diagonală)

■ □ • ***prelucrare de Liviu Ciortuz, după "The Multivariate Gaussian Distribution", Chuong B. Do, 2008***

Fie o variabilă aleatoare $X : \Omega \to \mathbb{R}^d$. În cele ce urmează elementele din $\mathbb{R}^d$ vor fi considerate vectori-coloană. Vom nota cu $\mathbb{S}^d_+$ spaţiul matricelor simetrice pozitiv definite[44] de dimensiune $d \times d$.

Vom spune că variabila X, reprezentată sub forma $X = [X_1 \dots X_d]^\top$, urmează o distribuţie gaussiană (sau, *normală*) multidimensională, având media $\mu \in \mathbb{R}^d$ şi matricea de covarianţă $\Sigma \in \mathbb{S}^d_+$, dacă funcţia ei de densitate [de probabilitate] are forma analitică următoare:[45]

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\Big(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\Big), \tag{22}$$

unde notaţia $|\Sigma|$ desemnează determinantul matricei Σ, iar $\exp(\ )$ desemnează funcţia exponenţială având baza e.[46] Pe scurt, vom nota această proprietate de definiţie a lui X sub forma $X \sim \mathcal{N}(\mu, \Sigma)$.

***Observaţii*:**
1. Se poate demonstra că orice matrice simetrică şi pozitiv definită Σ este inversabilă (deci $|\Sigma| \neq 0$), iar Σ^{-1} este de asemenea matrice pozitiv definită.[47] Aşadar pentru orice vector nenul z, vom avea $z^\top \Sigma^{-1} z > 0$. Aceasta implică faptul că pentru orice vector $x \neq \mu$, vom avea:

$$(x-\mu)^\top \Sigma^{-1}(x-\mu) > 0, \text{ deci } -\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu) < 0.$$

Similar cazului unidimensional, graficul acestei funcţii exponenţiale va avea forma unui clopot cu vârful având abscisa $x = \mu$ şi cu deschiderea cuadratică[48] îndreptată în jos.

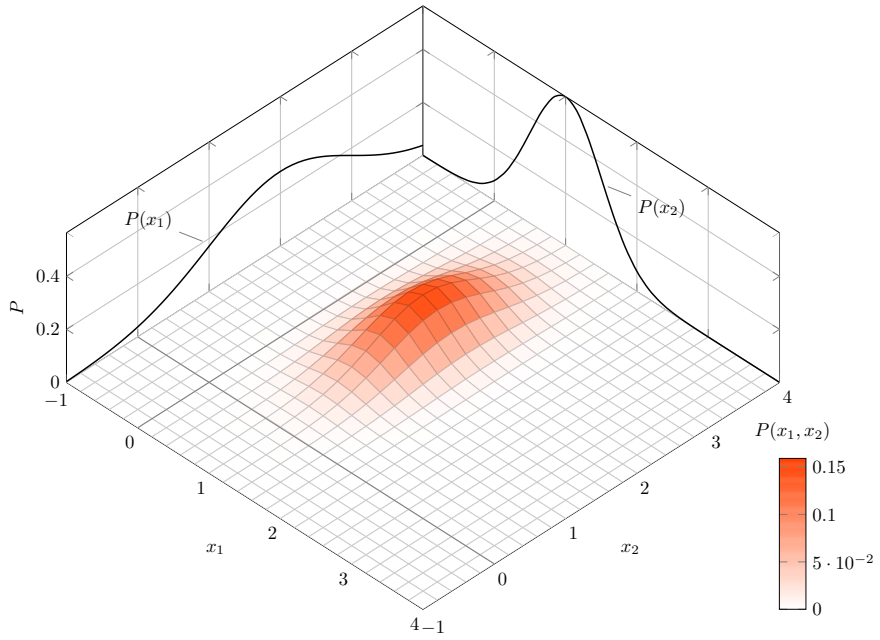
[44] Prin definiţie, o matrice $A \in \mathbb{R}^{d\times d}$ este pozitiv definită dacă $x^\top Ax > 0$ pentru orice $x \neq 0$ din $\mathbb{R}^d$, unde simbolul $\top$ reprezintă operaţia de transpunere a matricelor.

[45] Remarcaţi faptul că expresia din dreapta egalităţii (22) implică un uşor abuz (sau, mai degrabă, o convenţie) de notaţie: într-un astfel de context, o matrice reală de dimensiune 1×1 — în acest caz matricea $(x-\mu)^\top \Sigma^{-1}(x-\mu)$ — este identificată cu un număr real, care este chiar singurul ei element.

[46] În cazul general al unui vector X format din d variabile aleatoare, elementul generic (i, j) al matricei de covarianţă asociate lui X este prin definiţie $Cov(X_i, X_j) = E[(X_i - E[X_i])(X_j - E[X_j])]$ pentru $i, j \in \{1, \dots, d\}$. Pentru demonstraţia faptului că matricea de covarianţă Σ este întotdeauna simetrică şi pozitiv semidefinită vedeţi problema 20.

[47] Vedeţi exerciţiul 36.

[48] Este vorba de o elipsă având axele principale determinate – ca vectori – de vectorii proprii ai matricei de covarianţă Σ. Mărimile axelor pricipale sunt respectiv $\sqrt{\lambda_i}$, unde prin λ_i am notat vectorii proprii ai matricei Σ. (Cf. *Pattern Classification*, Duda, Hart and Stork, Appendix A.5.2, 2001.)



[Credit: http://www.pgfplots.net/tikz/examples/all/]

2. Coeficientul funcţiei exponenţiale, adică $\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}$, nu depinde de x; el este *constanta de normalizare* care ne asigură că

$$\frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\dots\int_{-\infty}^{\infty} \exp\left((x-\mu)^\top \Sigma^{-1}(x-\mu)\right) dx_1\, dx_2\, \dots\, dx_d = 1.$$

Considerăm cazul simplu, în care $d = 2$ şi matricea de covarianţă Σ este diagonală,[49] deci

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \qquad \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \qquad \Sigma = \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2, \end{bmatrix}$$

cu $\sigma_1 > 0$ şi $\sigma_2 > 0$.

Arătaţi că într-un astfel de caz, expresia funcţiei de densitate de probabilitate gaussiană bidimensională este identică cu produsul a două funcţii de densitate de tip gaussian unidimensionale independente, prima funcţie având media μ_1 şi varianţa σ_1^2, iar cea de-a doua având media μ_2 şi varianţa σ_2^2.

Răspuns:

În condiţiile de mai sus, funcţia de densitate de probabilitate gaussiană multidimensională are forma

$$p(x;\mu,\Sigma) = \frac{1}{2\pi \begin{vmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{vmatrix}^{\frac{1}{2}}} \exp\left(-\frac{1}{2}\begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}^\top \begin{bmatrix} \sigma_1^2 & 0 \\ 0 & \sigma_2^2 \end{bmatrix}^{-1} \begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}\right)$$

[49] În aceste condiţii, se poate demostra uşor că elementele de pe diagonala matricei Σ sunt strict pozitive. Într-adevăr, este suficient să se particularizeze vectorul z din formalizarea proprietăţii de *pozitiv-definire* a matricei Σ la valoarea $z = (1, 0)^\top$ şi respectiv $z = (0, 1)^\top$.

Aplicând mai întâi formula pentru determinantul de ordin 2,[50] şi calculând apoi inversa matricei Σ, obţinem:

$$\begin{aligned}
p(x;\mu,\Sigma) &= \frac{1}{2\pi\,\sigma_1\sigma_2}\exp\left(-\frac{1}{2}\begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}^\top \left(\frac{1}{\sigma_1^2\sigma_2^2}\begin{bmatrix} \sigma_2^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix}\right)\begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}\right) \\
&= \frac{1}{2\pi\,\sigma_1\sigma_2}\exp\left(-\frac{1}{2}\begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}^\top \begin{bmatrix} \frac{1}{\sigma_1^2} & 0 \\ 0 & \frac{1}{\sigma_2^2} \end{bmatrix}\begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}\right) \\
&= \frac{1}{2\pi\,\sigma_1\sigma_2}\exp\left(-\frac{1}{2}\begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}^\top \begin{bmatrix} \frac{1}{\sigma_1^2}(x_1-\mu_1) \\ \frac{1}{\sigma_2^2}(x_2-\mu_2) \end{bmatrix}\right) \\
&= \frac{1}{2\pi\,\sigma_1\sigma_2}\exp\left(-\frac{1}{2\sigma_1^2}(x_1-\mu_1)^2-\frac{1}{2\sigma_2^2}(x_2-\mu_2)^2\right) \\
&= \frac{1}{\sqrt{2\pi}\,\sigma_1}\exp\left(-\frac{1}{2\sigma_1^2}(x_1-\mu_1)^2\right)\frac{1}{\sqrt{2\pi}\,\sigma_2}\exp\left(-\frac{1}{2\sigma_2^2}(x_2-\mu_2)^2\right) \\
&= p(x_1;\mu_1,\sigma_1^2)\;p(x_2;\mu_2,\sigma_2^2).
\end{aligned}$$

Generalizând, este imediat că orice variabilă gaussiană d-dimensională de medie $\mu\in\mathbb{R}^d$ şi matrice de covarianţă diagonală $\Sigma \overset{not.}{=} diag(\sigma_1^2,\sigma_2^2,\ldots,\sigma_d^2)$, se comportă ca o colecţie de d variabile aleatoare gaussiene unidimensionale independente, având respectiv mediile μ_i şi varianţele σ_i^2.

***Observaţie*: Ştim de la exerciţiul 10 că dacă X_1 şi X_1 sunt variabile aleatoare independente, atunci $Cov(X_1,X_2)=0$. Deşi reciproca acestei afirmaţii nu este în general valabilă, la exerciţiul 11.b am arătat că ea este adevărată în cazul în care valorile acestor două variabile aleatoare sunt binare. La exerciţiul de faţă am arătat că implicaţia $Cov(X_1,X_2)=0 \Rightarrow X_1 \perp X_2$ are loc şi în cazul în care perechea (X_1,X_2) urmează o distribuţie gaussiană bivariată (sau, mai general, atunci când X_1 şi X_2 sunt componente ale unei distribuţii gaussiene multivariate a cărei matrice de covarianţă este diagonală).**

35. **(Distribuţia gaussiană, cazul bidimensional: explicitarea p.d.f. într-un caz particular)**

prelucrare de Liviu Ciortuz, 2019, după
□ • ○ ***MIT, 2006 fall, Tommi Jaakkola, HW1, pr. A.5.a***

Fie X un vector de variabile aleatoare de tip gaussian, cu

$$E[X]=\begin{pmatrix} 10 \\ 5 \end{pmatrix} \quad \textbf{şi} \quad Cov(X)=\begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix}.$$

a. Verificaţi că matricea $Cov(X)$, care este simetrică, este şi pozitiv definită.[51]

[50] Anume, $\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad-bc$.

[51] Vedeţi problema 20.

b. Pornind de la notaţia matriceală pentru funcţia densitate de probabilitate (p.d.f.) pentru o distribuţie gaussiană multidimensională oarecare,

$$\mathcal{N}(x|\mu,\Sigma)=\frac{1}{(\sqrt{2\pi})^d\sqrt{|\Sigma|}}\exp\left(-\frac{1}{2}(x-\mu)^\top\Sigma^{-1}(x-\mu)\right)$$

şi considerând $x=(x_1,x_2)\in\mathbb{R}^2$, scrieţi expresia funcţiei de densitate de probabilitate comună a variabilei X de mai sus. Veţi nota această funcţie cu $p(x_1,x_2)$.

Răspuns:

a. *Soluţia I-a*: folosind definiţia pentru matrice pozitiv definite.
Matricea $Cov(X)$ este pozitiv definită dacă şi numai dacă pentru orice vector $z\overset{not.}{=}(z_1,z_2)^\top\in\mathbb{R}^2$ care diferă de vectorul $0\in\mathbb{R}^2$ are loc inegalitatea $z^\top Cov(X)\,z>0$.

Fie acum un vector oarecare $z=(z_1,z_2)^\top\in\mathbb{R}^2$ diferit de vectorul 0.

$$(z_1,z_2)\begin{pmatrix}2&1\\1&1\end{pmatrix}\begin{pmatrix}z_1\\z_2\end{pmatrix}=(z_1,z_2)\begin{pmatrix}2z_1+z_2\\z_1+z_2\end{pmatrix}=2z_1^2+2z_1z_2+z_2^2=z_1^2+(z_1+z_2)^2.$$

Evident, $z_1^2+(z_1+z_2)^2\geq 0$ pentru orice z_1 şi $z_2\in\mathbb{R}$. Vom arăta că $z_1^2+(z_1+z_2)^2>0$ pentru orice $(z_1,z_2)^\top\neq 0\in\mathbb{R}^2$. Vom trata două cazuri:

***Cazul 1*: Dacă $z_1\neq 0$, atunci $z_1^2\neq 0$ şi, în consecinţă, $z_1^2+(z_1+z_2)^2>0$, indiferent de valoarea lui $z_2\in\mathbb{R}$.**

***Cazul 2*: Dacă $z_1=0$, atunci din condiţia $(z_1,z_2)\neq 0\in\mathbb{R}^2$ rezultă că $z_2\neq 0$. Prin urmare, $z_1^2+(z_1+z_2)^2=z_2^2>0$.**

***Soluţia a II-a*: Vom folosi următorul rezultat teoretic (de tip *caracterizare*): Fie A o matrice de numere reale, simetrică. Matricea A este pozitiv definită dacă şi numai dacă toate valorile ei proprii (engl., eigenvalues) sunt pozitive.[52]**

Mai întâi vom afla valorile proprii ale matricei $\Sigma\overset{not.}{=}Cov(X)$. Pentru aceasta, vom rezolva ecuaţia $(\Sigma-\lambda I)X=0$, impunând *restricţia* ca soluţia $X\in\mathbb{R}^2$ să fie diferită de vectorul 0. Notând $X=(x_1,x_2)^\top$, ecuaţia $(\Sigma-\lambda I)X=0$ va fi rescrisă astfel:

$$\begin{pmatrix}2-\lambda&1\\1&1-\lambda\end{pmatrix}\cdot\begin{pmatrix}x_1\\x_2\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix}$$

Faptul că soluţia $X\in\mathbb{R}^2$ trebuie să fie diferită de vectorul 0 implică în mod necesar

$$\begin{vmatrix}2-\lambda&1\\1&1-\lambda\end{vmatrix}=0\Leftrightarrow 2-3\lambda+\lambda^2-1=0\Leftrightarrow\lambda^2-3\lambda+1=0\Leftrightarrow\lambda_{1,2}=\frac{3\pm\sqrt{5}}{2}.$$

Se observă imediat că λ_1 şi λ_2 (vectorii proprii ai matricei Σ) sunt pozitivi. Aşadar, ţinând cont de *Indicaţia* din enunţ, rezultă că Σ este matrice pozitiv definită.

b. Avem: $|Cov(X)|=\begin{vmatrix}2&1\\1&1\end{vmatrix}=1$, deci $(Cov(X))^{-1}=\begin{pmatrix}1&-1\\-1&2\end{pmatrix}$.

[52] Cf. http://mathworld.wolfram.com/PositiveDefiniteMatrix.html.

Aşadar, funcţia de densitate de probabilitate comună a variabilei X este următoarea:

$$
\begin{aligned}
p(x_1, x_2) &= \frac{1}{(\sqrt{2\pi})^2} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right) \\
&= \frac{1}{2\pi} \exp\left(-\frac{1}{2}(x_1-10, x_2-5)\begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix}\begin{pmatrix} x_1-10 \\ x_2-5 \end{pmatrix}\right) \\
&= \frac{1}{2\pi} \exp\left(-\frac{1}{2}(x_1-x_2-5, -x_1+2x_2)\begin{pmatrix} x_1-10 \\ x_2-5 \end{pmatrix}\right) \\
&= \frac{1}{2\pi} \exp\left(-\frac{1}{2}(x_1^2 - 10x_1 - x_1x_2 + \cancel{10x_2} - \cancel{5x_1} + 50 - x_1x_2 + \cancel{5x_1} + 2x_2^2 - \cancel{10x_2})\right) \\
&= \frac{1}{2\pi} \exp\left(-\frac{1}{2}(x_1^2 - 10x_1 - 2x_1x_2 + 2x_2^2 + 50)\right).
\end{aligned}
$$

36. **(Matrice simetrice şi pozitiv definite: [o proprietate de tip] factorizare folosind vectori proprii)**

□ ***UAIC Iaşi, 2018 spring, Sebastian Ciobanu, după Appendix A.2, in The Multivariate Gaussian Distribution, by Chuong B. Do, Stanford, 2008***

Fie o variabilă aleatoare $X : \Omega \to \mathbb{R}^d$. În cele ce urmează, elementele din $\mathbb{R}^d$ vor fi considerate vectori-coloană. Vom nota cu $\mathbb{S}_+^d$ mulţimea matricelor simetrice pozitiv definite de dimensiune $d \times d$.[53]

***Definiţie*: Fie $A \in \mathbb{R}^{d\times d}$. Se numeşte *valoare proprie* a matricei A un număr complex $\lambda \in \mathbb{C}$ pentru care există un vector nenul $x \in \mathbb{C}^d$ astfel încât să aibă loc egalitatea $Ax = \lambda x$. Acest vector x se numeşte *vector propriu* asociat valorii proprii λ.**

a. Demonstraţi că pentru orice matrice $\Sigma \in \mathbb{S}_+^d$ există o matrice $B \in \mathbb{R}^{d\times d}$ astfel încât Σ să poată fi factorizată sub forma următoare: $\Sigma = BB^\top$.

***Observaţie*: Factorizarea aceasta nu este unică, adică există mai multe posibilităţi de a scrie matricea Σ sub forma $\Sigma = BB^\top$.**

***Indicaţie*: Vă puteţi folosi de următoarele proprietăţi:**

***i.* Orice matrice $A \in \mathbb{R}^{d\times d}$ care este simetrică poate fi scrisă astfel: $A = U\Lambda U^\top$, unde $U \in \mathbb{R}^{d\times d}$ este o matrice ortonormală conţinând vectorii proprii (pentru care impunem să aibă norma 1) ai lui A drept coloane, iar Λ este matricea diagonală conţinând valorile proprii ale lui A în ordinea corespunzătoare coloanelor (adică, a vectorilor proprii) din matricea U.**[54]

***ii.* Fie $A \in \mathbb{R}^{d\times d}$ o matrice simetrică. A este pozitiv definită dacă şi numai dacă toate valorile sale proprii sunt reale şi pozitive.**[55]

[53] Pentru definiţia noţiunii de matrice pozitiv definită, vedeţi problema 20.

[54] Această proprietate este enunţată în secţiunea 0.8 din documentul *Matrix Identities* de Sam Roweis, 1999. Faptul că U este matrice ortonormală se poate scrie în mod formal astfel: $U^\top U = U U^\top = I$.

[55] https://en.wikipedia.org/wiki/Positive-definite_matrix.

Observaţie: Proprietăţii a.*ii* îi corespund alte trei *proprietăţi similare*: Fie $A \in \mathbb{R}^{d\times d}$ o matrice simetrică. A este pozitiv semidefinită dacă şi numai dacă toate valorile sale proprii sunt reale şi nenegative. A este negativ definită dacă şi numai dacă toate valorile sale proprii sunt reale şi negative. A este negativ semidefinită dacă şi numai dacă toate valorile sale proprii sunt reale şi nepozitive. (Vedeţi https://en.wikipedia.org/wiki/Definiteness_of_a_matrix#Eigenvalues.)

iii. $(AB)^\top = B^\top A^\top$**, pentru orice matrice** $A, B \in \mathbb{R}^{d\times d}$**.**[56]

b. La acest punct vom face o *exemplificare* pentru chestiunile prezentate la punctul a**.**

Considerând matricea

$$\Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix},$$

(deci, $d = 2$**) determinaţi o matrice** $B \in \mathbb{R}^{2\times 2}$ **astfel încât** $\Sigma = BB^\top$**.**

***Indicaţie*: Folosind notaţiile din *Definiţia* dată pentru valorile proprii şi vectorii proprii, vom avea:**

$$Ax = \lambda x \Leftrightarrow (\lambda I_d - A)x = \mathbf{0}, x \neq \mathbf{0} \Leftrightarrow \det(\lambda I_d - A) = 0,$$

unde 0 este vectorul coloană nul d**-dimensional, iar** I_d **este matricea identitate** d**-dimensională.**

c. Demonstraţi că orice matrice B **care satisface proprietatea de la punctul** a **este inversabilă.**

***Indicaţie*: Vă puteţi folosi de următoarele proprietăţi:**

i. **Matricea** $A \in \mathbb{R}^{d\times d}$ **este inversabilă dacă şi numai dacă** $\det(A) \neq 0$**.**

ii. $\det(AB) = \det(A)\det(B)$**, pentru orice matrice** $A, B \in \mathbb{R}^{d\times d}$**.**[57]

iii. $\det(A) = \det(A^\top)$**, unde** $A \in \mathbb{R}^{d\times d}$**.**

iv. **Orice matrice pozitiv definită este inversabilă (iar inversa ei este de asemenea matrice pozitiv definită).**[58]

Răspuns:

a. Ştim, prin ipoteză, că matricea $\Sigma \in \mathbb{S}_+^d$**, deci este simetrică. Conform proprietăţii** $(a.i)$**, putem scrie următoarea factorizare pentru** Σ**:**

$$\Sigma = U\Lambda U^\top,$$

unde U **este matricea ortonormală conţinând vectorii proprii (cu norma** 1**) ai lui** Σ **drept coloane, iar** $\Lambda \in \mathbb{R}^{d\times d}$ **este matricea diagonală conţinând valorile proprii ale lui** Σ**, în ordinea corespunzătoare coloanelor matricei** U**.**

Întrucât $\Sigma \in \mathbb{S}_+^d$ **este matrice pozitiv definită, conform proprietăţii** $(a.ii)$ **rezultă că toate valorile proprii ale lui** Σ **sunt pozitive. Prin urmare, există matricea**

$$\Lambda^{1/2} = \begin{bmatrix} \sqrt{\lambda_1} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sqrt{\lambda_d} \end{bmatrix} \in \mathbb{R}^{d\times d}.$$

***Observaţii*:**
(1) Este imediat faptul că putem factoriza / descompune matricea Λ **în felul următor:**

$$\Lambda^{1/2} \cdot \Lambda^{1/2} = \Lambda. \tag{23}$$

[56] Formula (1c) din acelaşi document (*Matrix Identities*) de Sam Roweis.
[57] Formula (2a) din acelaşi document (*Matrix Identities*) de Sam Roweis.
[58] https://en.wikipedia.org/wiki/Positive-definite_matrix.

(2) $\Lambda^{1/2}$ este matrice diagonală, deci simetrică. Aşadar,

$$(\Lambda^{1/2})^\top = \Lambda^{1/2}. \tag{24}$$

Acum putem relua factorizarea matricei $\Sigma = U\Lambda U^\top$. Elaborând — adică, factorizând matricea Λ — în partea dreaptă a acestei egalităţi, vom putea obţine o nouă factorizare pentru matricea Σ în felul următor:

$$\begin{aligned}
\Sigma &= U\Lambda U^\top \\
&\overset{(23)}{=} U\Lambda^{1/2}\Lambda^{1/2}U^\top \overset{(24)}{=} U\Lambda^{1/2}(\Lambda^{1/2})^\top U^\top \\
&\overset{asoc.}{=} U\Lambda^{1/2}((\Lambda^{1/2})^\top U^\top) \overset{(a.iii)}{=} U\Lambda^{1/2}(U\Lambda^{1/2})^\top \\
&= BB^\top, \text{ unde } B \overset{not.}{=} U\Lambda^{1/2}.
\end{aligned} \tag{25}$$

***Observaţii*:**
(3) O altă modalitate de a factoriza / descompune matricea Σ sub forma $\Sigma = BB^\top$ este dată de *factorizarea Cholesky*: dacă A este o matrice simetrică şi pozitiv definită, atunci există o unică matrice $L \in \mathbb{R}^{d\times d}$, inferior triunghiulară, astfel încât $A = LL^\top$.[59]
(4) Factorizarea (25) este de fapt valabilă şi pentru matrice pozitiv semidefinite. Justificarea se bazează pe proprietatea menţionată la nota de subsol 55. Însă proprietatea de inversabilitate (vedeţi punctul c) *nu* este valabilă decât pentru matrice pozitiv definite.

b. Mai întâi vom calcula valorile proprii ale matricei Σ din enunţ. Fie deci $\lambda \in \mathbb{R}$ şi $x \in \mathbb{R}^d$, $x \neq 0$. Trebuie să rezolvăm ecuaţia $\Sigma x = \lambda x$ şi, conform *Indicaţiei* din enunţ, aceasta revine la a rezolva ecuaţia $\det(\lambda I_d - \Sigma) = 0$.

$$\begin{aligned}
\det(\lambda I_d - \Sigma) = 0 \quad &\Leftrightarrow \quad \det\left(\begin{bmatrix}\lambda - 1 & -0.5\\ -0.5 & \lambda - 1\end{bmatrix}\right) = 0 \Leftrightarrow (\lambda-1)^2 - 0.5^2 = 0 \\
&\Leftrightarrow \quad (\lambda - 1 - 0.5)(\lambda - 1 + 0.5) = 0 \Leftrightarrow (\lambda - 1.5)(\lambda - 0.5) = 0 \\
&\Leftrightarrow \quad \lambda_1 = 0.5,\ \lambda_2 = 1.5.
\end{aligned}$$

Deci valorile proprii ale matricei Σ sunt $\lambda_1 = 0.5$ şi $\lambda_2 = 1.5$.

Se observă că $\lambda_1, \lambda_2 > 0$. Aşadar, conform proprietăţii (a, ii),[60] **matricea Σ dată în enunţ este pozitiv definită. Conform punctului a, ştim acum că există o matrice B astfel încât matricea Σ să poată fi factorizată sub forma $\Sigma = BB^\top$. Pentru a determina matricea B, trebuie sa calculăm vectorii proprii ai matricei Σ. Îi vom obţine rezolvând următorul sistem, care este compatibil nedeterminat (pentru că $\det(\lambda I_d - \Sigma) = 0$):**[61]

$$\begin{cases}(\lambda - 1)v - 0.5u = 0 \Rightarrow v = \dfrac{0.5u}{\lambda - 1} \\ -0.5v + (\lambda - 1)u = 0\end{cases}.$$

Deci, vectorii proprii sunt de forma:

$$x = (v, u)^\top = \left(\frac{0.5}{\lambda - 1}u, u\right)^\top, u \in \mathbb{R}.$$

[59] https://en.wikipedia.org/wiki/Positive-definite_matrix.
[60] Mai exact, ne referim la partea „numai atunci“ a acestei proprietăţi, care reprezintă o caracterizare de tip *echivalenţă* pentru noţiunea de matrice pozitiv definită. (Observaţi că matricea Σ din enunţ este simetrică.)
[61] Vedeţi *Indicaţia* din enunţ.

În mod concret,

$$\lambda_1 = 0.5 \Rightarrow x_1 = (-u, u)^\top, u \in \mathbb{R}$$

şi

$$\lambda_2 = 1.5 \Rightarrow x_2 = (u, u)^\top, u \in \mathbb{R}.$$

Deoarece vrem ca vectorii x_1 şi x_2 să aibă norma 1, îi vom „normaliza". Mai întâi,

$$\frac{x_1}{\|x_1\|} = \frac{1}{\sqrt{2u^2}}(-u, u)^\top = \frac{1}{\sqrt{2}|u|}(-u, u)^\top \tag{26}$$

şi, considerând $u > 0$, deci $|u| = u$, rezultă că

$$\frac{x_1}{\|x_1\|} = \frac{1}{\sqrt{2}u}(-u, u)^\top = \left(-\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)^\top.$$

Apoi, în mod similar obţinem

$$\frac{x_2}{\|x_2\|} = \left(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}\right)^\top.$$

De la punctul a ştim că putem alege $B = U\Lambda^{1/2}$, deci

$$U = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}, \quad \Lambda^{1/2} = \begin{bmatrix} \sqrt{0.5} & 0 \\ 0 & \sqrt{1.5} \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3}}{\sqrt{2}} \end{bmatrix}$$

$$\Rightarrow B = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & \frac{\sqrt{3}}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix}.$$

Verificăm că într-adevăr $\Sigma = BB^\top$:

$$BB^\top = \begin{bmatrix} -\frac{1}{2} & \frac{\sqrt{3}}{2} \\ \frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} \\ \frac{\sqrt{3}}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} = \begin{bmatrix} \frac{1}{4} + \frac{3}{4} & -\frac{1}{4} + \frac{3}{4} \\ -\frac{1}{4} + \frac{3}{4} & \frac{1}{4} + \frac{3}{4} \end{bmatrix} = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} = \Sigma.$$

În ton cu *Observaţia* de la pagina 79, dacă în relaţia (26) vom considera $u < 0$, vom obţine încă o soluţie pentru factorizarea matricei Σ:

$$\frac{x_1'}{\|x_1'\|} = \frac{1}{\sqrt{2}|u|}(-u, u)^\top = \left(\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)^\top = -\frac{x_1}{\|x_1\|}$$

$$\frac{x_2'}{\|x_2'\|} = \frac{1}{\sqrt{2}|u|}(u, u)^\top = \left(-\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}\right)^\top = -\frac{x_2}{\|x_2\|},$$

deci

$$U' = -U, \; B' = U'\Lambda^{1/2} = -U\Lambda^{1/2} = -B$$

şi, în consecinţă,

$$B'(B')^\top = (-B)(-B)^\top = BB^\top = \Sigma.$$

c. De la punctul a rezultă că pentru orice matrice $\Sigma \in \mathbb{S}_+^d$ există o descompunere / factorizare de forma $\Sigma = BB^\top$, deci

$$\det(\Sigma) = \det(BB^\top) \overset{(c.ii)}{=} \det(B)\det(B^\top) \overset{(c.iii)}{=} \det(B)^2. \tag{27}$$

Prin urmare,

$$\Rightarrow \det(B) = \pm\sqrt{\det(\Sigma)}. \tag{28}$$

Pe de altă parte, din faptul că Σ este matrice pozitiv definită, rezultă conform proprietăţii $(c.iv)$ că Σ este matrice inversabilă. Mai departe, conform proprietăţii $(c.i)$, vom avea $\det(\Sigma) \neq 0$. Coroborând aceasta cu relaţia (28), rezultă că $\det(B) \neq 0$ şi în consecinţă (din nou, conform proprietăţii $(c.i)$) că matricea B este inversabilă (ceea ce era de demonstrat).

37. **(Distribuţia gaussiană multidimensională: funcţia ei de densitate de probabilitate satisface într-adevăr proprietăţile de definiţie)**

□ ***UAIC Iaşi, 2018 spring, Sebastian Ciobanu, după Appendix A.2, in The Multivariate Gaussian Distribution, by Chuong B. Do, Stanford, 2008***

***Definiţie*: Notăm cu $\mathbb{S}_+^d$ mulţimea matricelor simetrice pozitiv definite de dimensiune $d \times d$.[62] Spunem că variabila aleatoare vectorială $X : \Omega \to \mathbb{R}^d$, reprezentată sub forma $X = (X_1 \dots X_d)^\top$, urmează o distribuţie gaussiană multidimensională, având media $\mu \in \mathbb{R}^d$ şi matricea de covarianţă $\Sigma \in \mathbb{S}_+^d$, dacă funcţia ei de densitate [de probabilitate] are forma analitică următoare:**

$$p_X(x; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}\det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right), \tag{29}$$

unde notaţia $\det(\Sigma)$ desemnează determinantul matricei Σ, iar $\exp(\ )$ desemnează funcţia exponenţială având baza e. Pe scurt, vom nota această proprietate de definiţie a lui X sub forma $X \sim \mathcal{N}(\mu, \Sigma)$.

***Observaţie* (1): La problema 36 am demonstrat că pentru orice matrice $\Sigma \in \mathbb{R}^d$ simetrică şi pozitiv definită există (însă nu neapărat în mod unic) o matrice $B \in \mathbb{R}^{d\times d}$ cu proprietatea că Σ se poate „factoriza“ sub forma $\Sigma = BB^\top$. Mai mult, am demonstrat că orice matrice B care satisface această proprietate este în mod necesar inversabilă. De asemenea, din relaţia (27) şi din faptul că matricea Σ este pozitiv definită (ceea ce implică $\det(\Sigma) \neq 0$; vedeţi proprietatea $c.iv$ de la problema 36) rezultă că $\det(\Sigma) > 0$, ceea ce justifică faptul că $\det(\Sigma)^{1/2} = \sqrt{\det(\Sigma)}$ din relaţia (29) este bine definit.**

a. Demonstraţi că dacă definim $Z = B^{-1}(X - \mu)$, atunci $Z \sim \mathcal{N}(\mathbf{0}, I_d)$, unde $\mathbf{0}$ este vectorul coloană nul d-dimensional, iar I_d este matricea identitate d-dimensională.

***Observaţie* (2): Proprietatea aceasta este o generalizare a metodei de „standardizare“ pe care am întâlnit-o deja la problema 33, aplicată în cazul distribuţiilor gaussiene *unidimensionale*: $Z = \dfrac{X - \mu}{\sigma}$. În esenţă, acolo ne referream**

[62] Pentru definiţia noţiunii de matrice pozitiv definită, vedeţi problema 20.

la *schimbarea de variabilă* care realizează punerea în corespondenţă a unei distribuţii gaussiene unidimensionale oarecare cu distribuţia gaussiană / normală *standard* (aceasta din urmă având media 0 şi varianţa 1).

***Indicaţie* (1): Vă puteţi folosi de următoarele proprietăţi:**

***i.* Fie $X = (X_1 \dots X_d)^\top \in \mathbb{R}^d$ o variabilă aleatoare de tip vector, cu distribuţia comună dată de funcţia de densitate $p_X : \mathbb{R}^d \to \mathbb{R}$. Dacă $Z = H(X) \in \mathbb{R}^d$, unde H este funcţie bijectivă şi derivabilă [pe componente] în raport cu fiecare dintre argumentele sale, atunci Z are distribuţia comună dată de funcţia de densitate $p_Z : \mathbb{R}^d \to \mathbb{R}$, unde**

$$p_Z(z) = p_X(x) \cdot \left| \det \left(\begin{bmatrix} \frac{\partial x_1}{\partial z_1} & \dots & \frac{\partial x_1}{\partial z_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_d}{\partial z_1} & \dots & \frac{\partial x_d}{\partial z_d} \end{bmatrix} \right) \right| .$$

Matricea $\begin{bmatrix} \frac{\partial x_1}{\partial z_1} & \dots & \frac{\partial x_1}{\partial z_d} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_d}{\partial z_1} & \dots & \frac{\partial x_d}{\partial z_d} \end{bmatrix}$ **se numeşte *matricea jacobiană* a lui x în raport cu z şi se notează, în acest caz, cu $\frac{\partial x}{\partial z}$.**

ii. $\frac{\partial(Ax+b)}{\partial x} = A$**, unde** $x, b \in \mathbb{R}^d$**,** $A \in \mathbb{R}^{d \times d}$**, iar** A **şi** b **nu depind de** x**.**[63]

iii. $(AB)^{-1} = B^{-1}A^{-1}$**, unde** $A, B \in \mathbb{R}^{d \times d}$**.**[64]

iv. $(AB)^\top = B^\top A^\top$**, unde** $A, B \in \mathbb{R}^{d \times d}$**.**[65]

v. $\det(AB) = \det(A)\det(B)$**, unde** $A, B \in \mathbb{R}^{d \times d}$**.**[66]

vi. $\det(A) = \det(A^\top)$**, unde** $A \in \mathbb{R}^{d \times d}$**.**

b. Arătaţi că funcţia $p_X(x; \mu, \Sigma)$ care a fost dată în enunţ este într-adevăr funcţie densitate de probabilitate (p.d.f.).

***Indicaţie* (2): Vă puteţi folosi de următoarele proprietăţi:**

***i.* Pentru cazul $d = 1$, funcţia $p_X(x; \mu, \sigma^2)$ este funcţie densitate de probabilitate. (Demonstraţia a fost făcută la problema 32.a.)**

***ii.* În cazul în care matricea Σ este diagonală,**

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_d^2 \end{bmatrix} \text{, iar } \mu = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_d \end{bmatrix} ,$$

[63] Aceasta este o consecinţă care decurge imediat din formula (6b) din documentul *Matrix Identities* de Sam Roweis, 1999.
[64] Formula (1d) din acelaşi document (*Matrix Identities*) de Sam Roweis.
[65] Formula (1c) din acelaşi document (*Matrix Identities*) de Sam Roweis.
[66] Formula (2a) din acelaşi document (*Matrix Identities*) de Sam Roweis.

expresia funcţiei de densitate gaussiană multidimensională este identică cu produsul a d funcţii de densitate de tip gaussian, unidimensionale şi independente, prima funcţie având media μ_1 şi varianţa σ_1^2, a doua funcţie având media μ_2 şi varianţa σ_2^2, ..., a d-a funcţie având media μ_d şi varianţa σ_d^2. (Demonstraţia acestei proprietăţi a fost făcută la problema 34.)

Răspuns:

a. Sunt imediate următoarele relaţii:

$$z = B^{-1}(x-\mu) \overset{B\cdot}{\Rightarrow} Bz = x - \mu \Rightarrow x = Bz + \mu \tag{30}$$

$$\frac{\partial x}{\partial z} = \frac{\partial (Bz+\mu)}{\partial z} \overset{(a.ii)}{=} B \tag{31}$$

$$(\det(\Sigma))^{1/2} = \sqrt{\det(BB^\top)} \overset{(a.v)}{=} \sqrt{\det(B)\det(B^\top)} \overset{(a.vi)}{=} \sqrt{(\det(B))^2} = |\det(B)|. \tag{32}$$

Vom rescrie acum expresia care constituie argumentul funcţiei $\exp(\,)$ **din definiţia p.d.f.-ului distribuţiei gaussiene multidimensionale (vedeţi relaţia** (29)**), în funcţie de vectorul** z**.**

$$\begin{aligned}
&(x-\mu)^\top \Sigma^{-1}(x-\mu) \\
&\overset{(30)}{=} (Bz+\mu-\mu)^\top (B^\top B)^{-1}(Bz+\mu-\mu) = (Bz)^\top (B^\top B)^{-1}(Bz) \\
&\overset{(a.iii)}{=} (Bz)^\top ((B^\top)^{-1}B^{-1})(Bz) \\
&\overset{(a.iv)}{=} (z^\top B^\top)((B^\top)^{-1}B^{-1})(Bz) \\
&\overset{asoc.}{=} z^\top (B^\top (B^\top)^{-1})(B^{-1}B)z = z^\top I_d\, I_d\, z = z^\top z.
\end{aligned} \tag{33}$$

Aplicând acum proprietatea $(a.i)$**, vom putea calcula p.d.f.-ul distribuţiei gaussiene asociate variabilei** Z**:**

$$\begin{aligned}
p_Z(z) &= p_X(x) \cdot \left|\det\left(\frac{\partial x}{\partial z}\right)\right| \\
&= \frac{1}{(2\pi)^{d/2}\det(\Sigma)^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)\left|\det\left(\frac{\partial x}{\partial z}\right)\right| \\
&\overset{(31)\,(32)\,(33)}{=} \frac{1}{(2\pi)^{d/2}\cancel{|\det(B)|}} \exp\left(-\frac{1}{2}z^\top z\right)\cancel{|\det(B)|} \\
&= \frac{1}{(2\pi)^{d/2}} \exp\left(-\frac{1}{2}z^\top z\right) \overset{z^\top = z^\top I_d}{=} \frac{1}{(2\pi)^{d/2}(\det(I_d))^{1/2}} \exp\left(-\frac{1}{2}z^\top I_d z\right) \\
&= \mathcal{N}(z; \mathbf{0}, I_d).
\end{aligned}$$

b. $p_X(x;\mu,\Sigma)$ **este funcţie densitate de probabilitate (p.d.f.) dacă:**

- $p_X(x;\mu,\Sigma) \geq 0, \forall x \in \mathbb{R}^d$
- $I \overset{not.}{=} \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\ldots\int_{-\infty}^{\infty} p_X(x;\mu,\Sigma)dx_d\ldots dx_2 dx_1 = 1$**.**

Prima condiţie este satisfăcută, pentru că numitorul fracţiei [care este *constanta de normalizare*] din definiţia funcţiei de densitate de probabilitate $p_X(x;\mu,\Sigma)$ **este pozitiv, iar** $\exp(y) > 0$ **pentru orice** $y \in \mathbb{R}$**.**[67]

[67] *Observaţie importantă*: Spre deosebire de problema 32 — unde, la calcularea varianţei pentru distribuţia gaussiană unidimensională, făceam trecerea din sistemul de coordonate carteziene în sistemul de coordonate polare —, aici nu am schimbat sistemul de coordonate, ci practic am translat punctele, conform unei transformări liniare bijective. Am trecut astfel de la un "clopot" cu deschidere de formă eliptică la un "clopot" cu deschidere de formă eliptică circulară, cu centrul în originea sistemului de coordonate.

În continuare vom verifica a doua condiţie.

În integrala I facem substituţia (schimbarea de variabilă): $z = B^{-1}(x-\mu)$, unde $\Sigma = BB^\top, B \in \mathbb{R}^{d\times d}$. Matricea B există şi este inversabilă după cum s-a precizat în enunţ (vedeţi *Observaţia* (1)).

De la punctul a rezultă imediat că:

$$\begin{aligned} I &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\ldots\int_{-\infty}^{\infty} p_Z(z;\mathbf{0},I_d)dz_d\ldots dz_2dz_1 \\ &\overset{(b.ii)}{=} \left(\int_{-\infty}^{\infty} p_{Z_1}(z_1;0,1)dz_1\right)\left(\int_{-\infty}^{\infty} p_{Z_2}(z_2;0,1)dz_2\right)\ldots\left(\int_{-\infty}^{\infty} p_{Z_d}(z_d;0,1)dz_d\right) \\ &\overset{(b.i)}{=} 1\cdot 1\cdot\ldots\cdot 1 = 1. \end{aligned}$$

Deci, şi a doua condiţie este satisfăcută, ceea ce înseamnă că funcţia $p_X(x;\mu,\Sigma)$ este într-adevăr funcţie densitate de probabilitate (p.d.f.).

38. **(Distribuţia gaussiană bidimensională; o proprietate: distribuţia condiţională a unei componente în raport cu cealaltă componentă este tot de tip gaussian; identificarea parametrilor acestei distribuţii condiţionale)**

■ □ ***formulare de Liviu Ciortuz, după "Pattern Classification" (2nd ed.), [Appendix A. Mathematical foundations] R. Duda, P. Hart, D. Stork. John Wiley & Sons Inc., 2001***

Fie X o variabilă aleatoare care urmează o distribuţie gaussiană bidimensională de parametri μ (vectorul de medii) şi Σ (matricea de covarianţă). Aşadar, $\mu = (\mu_1,\mu_2) \in \mathbb{R}^2$, iar $\Sigma \in \mathcal{M}_{2\times 2}(\mathbb{R})$ este matrice simetrică şi pozitiv definită.

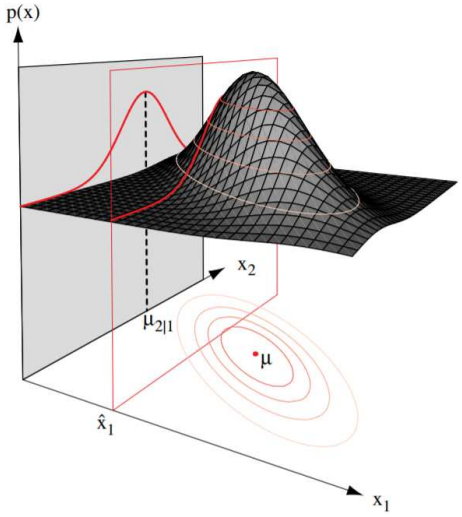
Prin definiţie, $\Sigma = Cov(X,X)$, unde $X \overset{not.}{=} (X_1,X_2)$, aşadar $\Sigma_{ij} = Cov(X_i,X_j)$ pentru $i,j \in \{1,2\}$. De asemenea, $Cov(X_i,X_i) = Var[X_i] \overset{not.}{=} \sigma_i^2 \geq 0$ pentru $i \in \{1,2\}$, în vreme ce pentru $i \neq j$ avem $Cov(X_i,X_j) = Cov(X_j,X_i) \overset{not.}{=} \sigma_{ij}$. În sfârşit, dacă folosim *coeficientul de corelaţie* $\rho \overset{def.}{=} \dfrac{\sigma_{12}}{\sigma_1\sigma_2}$, rezultă că putem scrie matricea de covarianţă Σ sub forma următoare:[68]

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}. \tag{34}$$

[68] Pentru definiţia coeficientului de corelaţie, vedeţi problema 19.

Demonstraţi că ipoteza $X \sim \mathcal{N}(\mu, \Sigma)$ implică faptul că distribuţia condiţională $X_2|X_1$ este de tip gaussian, şi anume $X_2|X_1 = x_1 \sim \mathcal{N}(\mu_{2|1}, \sigma^2_{2|1})$, cu $\mu_{2|1} = \mu_2 + \rho\frac{\sigma_2}{\sigma_1}(x_1 - \mu_1)$ şi $\sigma^2_{2|1} = \sigma^2_2(1-\rho^2)$.

***Observaţie*: Pentru $X_1|X_2$, rezultatul este similar: $X_1|X_2 = x_2 \sim \mathcal{N}(\mu_{1|2}, \sigma^2_{1|2})$, cu $\mu_{1|2} = \mu_1 + \rho\frac{\sigma_1}{\sigma_2}(x_2 - \mu_2)$ şi $\sigma^2_{1|2} = \sigma^2_1(1-\rho^2)$.**



Credit: *Pattern Classification*, 2nd ed., R. Duda, P. Hart and D. Stork, 2001

Răspuns:

Conform definiţiei distribuţiei condiţionale, $X_2|X_1 = x_1$ are funcţia de densitate de probabilitate

$$p(x_2|x_1) = \frac{p_{X_1,X_2}(x_1, x_2)}{p_{X_1}(x_1)}, \tag{35}$$

unde p_{X_1,X_2} este densitatea distribuţiei gaussiene bidimensionale de parametri μ şi Σ, iar p_{X_1} este distribuţia marginală a variabilei X_1. Aşadar,[69]

$$p_{X_1,X_2}(x_1, x_2) = \frac{1}{(\sqrt{2\pi})^2\sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right) \text{ şi} \tag{36}$$

$$p_{X_1}(x_1) = \frac{1}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{1}{2\sigma_1^2}(x_1-\mu_1)^2\right). \tag{37}$$

Se constată imediat din relaţia (34) că *determinantul* matricei Σ este $|\Sigma| = \sigma_1^2\sigma_2^2(1-\rho^2)$. Întrucât din expresia lui p_{X_1,X_2} de mai sus rezultă că Σ trebuie să fie matrice inversabilă, vom avea $\rho \in (-1, 1)$. De asemenea, pentru că se consideră $\sigma_1,\ \sigma_2 > 0$, rezultă că $\sqrt{|\Sigma|} = \sigma_1\sigma_2\sqrt{1-\rho^2}$.

Inversa matricei Σ se calculează astfel:

$$\begin{aligned}\Sigma^{-1} &= \frac{1}{\sigma_1^2\sigma_2^2(1-\rho^2)}\Sigma^* = \frac{1}{\sigma_1^2\sigma_2^2(1-\rho^2)}\begin{bmatrix} \sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix} \\ &= \frac{1}{(1-\rho^2)}\begin{bmatrix} \frac{1}{\sigma_1^2} & -\frac{\rho}{\sigma_1\sigma_2} \\ -\frac{\rho}{\sigma_1\sigma_2} & \frac{1}{\sigma_2^2} \end{bmatrix}\end{aligned}$$

[69] Se poate demonstra că relaţia (37) rezultă din relaţia (36), prin integrare în raport cu x_2. Demonstraţia depăşeşte cadrul de faţă şi este lăsată ca exerciţiu.

Prin urmare,

$$p_{X_1,X_2}(x_1,x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}}\cdot$$
$$\exp\left(-\frac{1}{2(1-\rho)^2}(x_1-\mu_1, x_2-\mu_2)\begin{bmatrix}\frac{1}{\sigma_1^2} & -\frac{\rho}{\sigma_1\sigma_2}\\ -\frac{\rho}{\sigma_1\sigma_2} & \frac{1}{\sigma_2^2}\end{bmatrix}\begin{pmatrix}x_1-\mu_1\\ x_2-\mu_2\end{pmatrix}\right)$$
$$=\frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}}\cdot$$
$$\exp\left(-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 - 2\rho\left(\frac{x_1-\mu_1}{\sigma_1}\right)\left(\frac{x_2-\mu_2}{\sigma_2}\right)+\left(\frac{x_2-\mu_2}{\sigma_2}\right)^2\right]\right) \quad (38)$$

***Observaţie*: Din expresia pe care tocmai am obţinut-o se observă uşor că $p_{X_1,X_2}(x_1,x_2)$ îşi atinge maximul atunci când $(x_1,x_2)=(\mu_1,\mu_2)$. Curbele de ecuaţie $p_{X_1,X_2}(x_1,x_2)=c$, pentru diverse valori ale lui $c\in\mathbb{R}^+$, se numesc *curbe de izocontur*. Ele au formă elipsoidală.**

Înlocuind expresiile (37) şi (38) în definiţia (35), vom obţine:

$$\begin{aligned}p(x_2|x_1) &= \frac{p_{X_1,X_2}(x_1,x_2)}{p_{X_1}(x_1)}\\ &= \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}}\cdot\\ &\quad \exp\left(-\frac{1}{2(1-\rho^2)}\left[\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 - 2\rho\left(\frac{x_1-\mu_1}{\sigma_1}\right)\left(\frac{x_2-\mu_2}{\sigma_2}\right)+\left(\frac{x_2-\mu_2}{\sigma_2}\right)^2\right]\right)\cdot\\ &\quad \left(\sqrt{2\pi}\sigma_1\exp\left(\frac{1}{2}\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2\right)\right)^{-1}\\ &= \frac{1}{\sqrt{2\pi}\sigma_2\sqrt{1-\rho^2}}\exp\left[-\frac{1}{2(1-\rho^2)}\left(\frac{x_2-\mu_2}{\sigma_2}-\rho\frac{x_1-\mu_1}{\sigma_1}\right)^2\right]\\ &= \frac{1}{\sqrt{2\pi}\sigma_2\sqrt{1-\rho^2}}\exp\left[-\frac{1}{2}\left(\frac{x_2-[\mu_2+\rho\frac{\sigma_2}{\sigma_1}(x_1-\mu_1)]}{\sigma_2\sqrt{1-\rho^2}}\right)^2\right].\end{aligned}$$

Din această expresie se observă că $X_2|X_1=x_1$ urmează o distribuţie gaussiană de parametri $\mu_{2|1}\overset{not.}{=}\mu_2+\rho\frac{\sigma_2}{\sigma_1}(x_1-\mu_1)$ şi $\sigma^2_{2|1}\overset{not.}{=}\sigma_2^2(1-\rho^2)$.

***Observaţie*: Rezultatul obţinut la acest exerciţiu se generalizează la [condiţionarea pentru] distribuţii marginale pentru distribuţia gaussiană multidimensională, astfel: dacă $x\sim\mathcal{N}(\mu,\Sigma)$ şi $x=\begin{pmatrix}x_1\\ x_2\end{pmatrix}\in\mathbb{R}^{r+s}$, $x_1\in\mathbb{R}^r$, $x_2\in\mathbb{R}^s$, $\mu=\begin{pmatrix}\mu_1\\ \mu_2\end{pmatrix}$, $\mu_1\in\mathbb{R}^r$, $\mu_2\in\mathbb{R}^s$, $\Sigma=\begin{pmatrix}\Sigma_{11} & \Sigma_{12}\\ \Sigma_{21} & \Sigma_{22}\end{pmatrix}$, $\Sigma_{11}\in\mathbb{R}^{r\times r}$, $\Sigma_{12}=\Sigma_{21}^\top\in\mathbb{R}^{r\times s}$, $\Sigma_{22}\in\mathbb{R}^{s\times s}$, atunci [$x_1\sim\mathcal{N}(\mu_1,\Sigma_{11})$, $x_2\sim\mathcal{N}(\mu_2,\Sigma_{22})$ şi] $x_1|x_2\sim\mathcal{N}(\mu_{1|2},\Sigma_{1|2})$, unde**

$$\mu_{1|2}=\mu_1+\Sigma_{12}\Sigma_{22}^{-1}(x_2-\mu_2),\ \Sigma_{1|2}=\Sigma_{11}-\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}. \quad (39)$$

Similar, $x_{2|1} \sim \mathcal{N}(\mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(x_1 - \mu_1), \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12})$**. Pentru demonstraţie, vedeţi documentul *More on Multivariate Gaussians* de Chuong B. Do (21 November 2008), sect. 3.2 şi 3.3.**

39. **(Mixturi de distribuţii multidimensionale: o proprietate)**

□ • ***CMU, 2015 fall, Z. Bar-Joseph, E. Xing, midterm exam, pr. 7***

Considerăm mixtura de distribuţii

$$p(x) = \sum_{k=1}^{K} \pi_k \, p(x|z_k),$$

unde $x \in \mathbb{R}^d$**, cu** $d \in \mathbb{N}^*$ **şi, ca de obicei,** z_k **sunt valorile unei variabile de tip categorial,** $\pi_k \in [0,1]$ **cu** $\sum_{k=1}^{K} \pi_k = 1$**, iar** $p(x|z_k)$ **sunt distribuţii probabiliste, bineînţeles cu** $k = 1, \ldots, K$**.**

Presupunem că împărţim vectorul x **în două părţi, astfel:** $x = (x_1, x_2)$**.**

Arătaţi că distribuţia condiţionată $p(x_2|x_1)$ **este de asemenea o mixtură de distribuţii,**

$$p(x_2|x_1) = \sum_{k=1}^{K} \varepsilon_k \, p(x_2|x_1, z_k) \tag{40}$$

şi găsiţi expresia *probabilităţii de selecţie* ε_k **în funcţie de** π_k**,** $p(x_1|z_k)$ **şi** $p(x_1)$**.**

Răspuns:

Vom exprima în mod convenabil distribuţia condiţionată $p(x_2|x_1)$ **— care constituie membrul stâng al egalităţii din concluzia problemei — în aşa fel încât să punem în evidenţă relaţia dintre această distribuţie şi distribuţiile condiţionate** $p(x_2|x_1, z_k)$**, care apar în membrul drept al aceleiaşi egalităţi:**

$$\begin{aligned} p(x_2|x_1) &= \sum_{k=1}^{K} p(x_2, z_k|x_1) \overset{form.\ mult.}{=} \sum_{k=1}^{K} p(x_2|z_k, x_1)\, p(z_k|x_1) \\ &\overset{F.\ Bayes}{=} \sum_{k=1}^{K} p(x_2|z_k, x_1)\, p(x_1|z_k)\, \underbrace{p(z_k)}_{\pi_k} / p(x_1) \end{aligned}$$

Prima egalitate este justificată de proprietatea de aditivitate numărabilă din definiţia funcţiei de probabilitate. Pentru obţinerea celei de-a doua egalităţi am folosit regula de înmulţire (în varianta condiţională), iar pentru cea de-a treia am aplicat regula lui Bayes.

Din ultima egalitate rezultă:[70]

$$\varepsilon_k = \pi_k \, p(x_1|z_k)/p(x_1).$$

[70] Se va observa că ε_k nu depinde de x_2, însă depinde de x_1.

40. **(Distribuţia Bernoulli, distribuţia gaussiană standard; intervale de încredere, teorema limită centrală, legea numerelor mari: aplicaţie la calculul erorii reale a unui clasificator)**

■ • ***CMU, 2008 fall, Eric Xing, HW3, pr. 3.3***

Nu demult, Chris s-a decis să folosească un nou clasificator (binar) care filtrează emailurile spam. Ulterior, el a vrut să evalueze cât de bun este acest clasificator. În acest scop, el a testat clasificatorul pe un mic set de date constituit din 100 de emailuri alese în mod aleatoriu dintre toate emailurile sale. Rezultatul pe care l-a obţinut a fost următorul: 83 de emailuri au fost clasificate corect. Aşadar rata erorii produse (sau: eroarea medie produsă) de clasificator pe acest mic set de date este de 17%. Este evident însă că eroarea aceasta este mai mică sau mai mare decât *eroarea reală*, pur şi simplu datorită alegerii aleatoare a celor 100 de emailuri.

Dacă se consideră un nivel de încredere de 95%, în ce interval se va situa eroarea reală (ţinând cont de acest experiment)?

Răspuns:

Vom nota cu X_i variabila aleatoare care reprezintă producerea unui mesaj email ($i = 1, \ldots, n = 100$) şi vom considera că $X_i = 1$ dacă emailul respectiv este clasificat eronat şi 0 în cazul contrar.

Notăm cu $\mu \overset{not.}{=} e_{real}$ media variabilei X_i şi cu σ^2 varianţa variabilei X_i. (Valorile lui μ şi σ nu depind de i).

Conform Teoremei Limită Centrală, eroarea la eşantionare,

$$e_{sample} \overset{not.}{=} \frac{X_1 + \ldots + X_n}{n},$$

văzută ca variabilă aleatoare, este aproximată de $\mathcal{N}(\mu, \sigma^2)$, distribuţia normală de medie μ şi varianţă σ^2. În consecinţă, notând

$$Z_n = \frac{X_1 + \ldots + X_n - n\mu}{\sqrt{n}\,\sigma} = \frac{\frac{X_1 + \ldots + X_n}{n} - \mu}{\frac{\sigma}{\sqrt{n}}}$$

$$\text{fiindcă } Var\left[\frac{X_1 + \ldots + X_n}{n}\right] = \frac{1}{n^2} n\sigma^2 = \frac{\sigma^2}{n},$$

rezultă că $Z_n \sim \mathcal{N}(0, 1)$, deci pentru orice $a \in \mathbb{R}$ vom avea $P(Z_n \leq a) \to \Phi(a)$, unde $\Phi(x) \overset{def.}{=} P(Z \leq x)$ desemnează funcţia de distribuţie cumulativă (engl., cumulative distribution function, c.d.f.) pentru distribuţia normală standard (vedeţi problema 33).

Folosind notaţiile de mai sus, vom scrie următoarele echivalenţe, care au loc pentru orice $a \geq 0$:

$$|\, Z_n \,| \leq a \Leftrightarrow \left|\frac{X_1 + \ldots + X_n - n\mu}{\sqrt{n}\,\sigma}\right| \leq a \Leftrightarrow \left|\frac{X_1 + \ldots + X_n - n\mu}{n\sigma}\right| \leq \frac{a}{\sqrt{n}}$$

$$\Leftrightarrow \left|\frac{X_1 + \ldots + X_n - n\mu}{n}\right| \leq \frac{a\sigma}{\sqrt{n}} \Leftrightarrow \left|\frac{X_1 + \ldots + X_n}{n} - \mu\right| \leq \frac{a\sigma}{\sqrt{n}}$$

$$\Leftrightarrow |e_{sample} - e_{real}| \leq \frac{a\sigma}{\sqrt{n}} \Leftrightarrow |e_{real} - e_{sample}| \leq \frac{a\sigma}{\sqrt{n}}$$

$$\Leftrightarrow \quad -\frac{a\sigma}{\sqrt{n}} \leq e_{real} - e_{sample} \leq \frac{a\sigma}{\sqrt{n}} \Leftrightarrow e_{sample} - \frac{a\sigma}{\sqrt{n}} \leq e_{real} \leq e_{sample} + \frac{a\sigma}{\sqrt{n}}$$
$$\Leftrightarrow \quad e_{real} \in [e_{sample} - \frac{a\sigma}{\sqrt{n}}, e_{sample} + \frac{a\sigma}{\sqrt{n}}] \tag{41}$$

Pentru a determina efectiv invervalul de încredere cerut, trebuie ca mai întâi să mai aflăm a şi σ şi apoi să le înlocuim în relaţia (41).

Constanta a se determină uşor ţinând cont de faptul că în enunţ se precizează *nivelul de încredere* (95%) pentru apartenenţa erorii reale la intervalul specificat. Această restricţie corespunde relaţiei $P(\mid Z_n \mid \leq a) = 0.95$. Pe de altă parte, întrucât $\Phi(-a) + \Phi(a) = 1$, avem:

$$P(\mid Z_n \mid \leq a) = \Phi(a) - \Phi(-a) = 2\Phi(a) - 1,$$

deci

$$P(\mid Z_n \mid \leq a) = 0.95 \Leftrightarrow 2\Phi(a) - 1 = 0.95 \Leftrightarrow \Phi(a) = 0.975 \Leftrightarrow a \cong 1.97$$ **(vedeţi pr. 33).**

Ne-a mai rămas de calculat valoarea lui σ. Vom ţine cont de faptul că

$$\sigma^2 \overset{not.}{=} Var_{real} = e_{real}(1 - e_{real}), \tag{42}$$

ultima egalitate având loc fiindcă toate variabilele X_i, identic distribuite, sunt de tip Bernoulli. În relaţia (42) vom putea aproxima e_{real} cu e_{semple}, conform *legii numerelor mari* (varianta „slabă“ a acestei legi este prezentată la problema 21.f).

În sfârşit, putem scrie:

$$\frac{a\sigma}{\sqrt{n}} = \frac{1.97\sqrt{0.17(1-0.17)}}{\sqrt{100}} \cong 0.07,$$

deci relaţia (41) devine $e_{real} \in [0.10, 0.24]$.

Sumarizând, putem afirma cu încredere de 95% că eroarea reală a clasificatorului folosit de Chris este mai mare sau egală cu 0.1 şi mai mică sau egală cu 0.24.

41. **(Familia de distribuţii exponenţiale: definiţie, exemplificare)**

□ • ○ ***CMU, 2015 fall, Z. Bar-Joseph, E. Xing, HW1, pr. 1.1.a***

Multe dintre distribuţiile probabiliste des folosite în statistică şi învăţarea automată fac parte din *familia exponenţială*. Prin *definiţie*, familia exponenţială este formată din acele distribuţii ale căror funcţii masă de probabilitate (în cazul distribuţiilor discrete), respectiv funcţii densitate de probabilitate (în cazul distribuţiilor continue) pot fi exprimate sub forma următoare:

$$f_X(x) = b(x)\, e^{\eta(\theta)\cdot T(x) - a(\theta)},$$

unde $T(x)$, $b(x)$, $\eta(\theta)$ şi $a(\theta)$ sunt cunoscute.

Demonstraţi că următoarele trei distribuţii

a. distribuţia multinomială, având parametrii $\theta_i > 0$ pentru $i = 1, \ldots, k$, astfel încât $\sum_{i=1}^k \theta_i = 1$ şi fiind definită prin

$$p(x|\theta) = \frac{(\sum_{i=1}^k x_i)!}{\prod_{i=1}^k x_i!} \prod_{i=1}^k \theta_i^{x_i}, \tag{43}$$

unde $x = (x_1, \ldots, x_k)$, cu $x_i \in \mathbb{N}$;

b. distribuţia Dirichlet, [71]având parametrii $\theta_i > 0$ pentru $i = 1, \ldots, k$, astfel încât $\sum_{i=1}^k \theta_i = 1$ şi fiind definită prin

$$p(x|\theta) = \frac{\Gamma(\sum_{i=1}^k \theta_i)}{\prod_{i=1}^k \Gamma(\theta_i)} \prod_{i=1}^k x_i^{\theta_i - 1}; \tag{44}$$

unde Γ este funcţia Gamma a lui Euler,[72] iar $x = (x_1, \ldots, x_k)$, cu $x_i > 0$;

c. distribuţia gaussiană multidimensională, având parametrii $\mu \in \mathbb{R}^d$ şi $\Sigma \in \mathbb{R}^{d \times d}$, matrice simetrică şi pozitiv definită, şi fiind definită prin

$$p(x|\mu, \Sigma) = \frac{1}{(2\pi)^{k/2}\, |\Sigma|^{1/2}} \exp\Big(-\frac{1}{2}(x-\mu)^\top \Sigma (x-\mu)\Big).$$

fac parte din familia exponenţială.

Răspuns:

a. Cazul distribuţiei multinomiale:[73]

$$\begin{aligned} p(x|\theta) &= \frac{(\sum_{i=1}^k x_i)!}{\prod_{i=1}^k x_i!} \prod_{i=1}^k \theta_i^{x_i} = \frac{(\sum_{i=1}^k x_i)!}{\prod_{i=1}^k x_i!} \exp\left(\ln \prod_{i=1}^k \theta_i^{x_i} \right) \\ &= \frac{(\sum_{i=1}^k x_i)!}{\prod_{i=1}^k x_i!} \exp\left(\sum_{i=1}^k \ln \theta_i^{x_i} \right). \end{aligned}$$

Aşadar, luând $b(x) = \dfrac{(\sum_{i=1}^k x_i)!}{\prod_{i=1}^k x_i!}$, $\eta(\theta) = (\ln \theta_1, \ldots, \ln \theta_k)$, $T(x) = x$ şi $a(\theta) = 0$ rezultă că distribuţia multinomială aparţine familiei exponenţiale.

b. Cazul distribuţiei Dirichlet:[74]

$$\begin{aligned} p(x|\theta) &= \frac{\Gamma(\sum_{i=1}^k \theta_i)}{\prod_{i=1}^k \Gamma(\theta_i)} \prod_{i=1}^k x_i^{\theta_i - 1} = \exp\left(\ln \left(\frac{\Gamma(\sum_{i=1}^k \theta_i)}{\prod_{i=1}^k \Gamma(\theta_i)} \prod_{i=1}^k x_i^{\theta_i - 1} \right) \right) \\ &= \exp\left(\ln \frac{\Gamma(\sum_{i=1}^k \theta_i)}{\prod_{i=1}^k \Gamma(\theta_i)} + \sum_{i=1}^k (\theta_i - 1) \ln x_i \right). \end{aligned}$$

[71] Această distribuţie poartă numele lui Johann Peter Gustav Lejeune Dirichlet, matematician prusac (1805-1859). Dirichlet a studiat la universitatea din Paris şi a predat la universităţile din Breslau, Berlin şi Göttinghen. Soţia sa a fost Rebecka Mendelssohn Bartholdy, sora mai mică a compozitorului Felix Mendelssohn Bartholdy.

[72] Vedeţi problema 31.b.

[73] Distribuţia multinomială constituie o generalizare a distribuţiei binomiale (care la rândul ei generalizează distribuţia Bernoulli). Pe de altă parte, distribuţia multinomială poate fi văzută ca o generalizare a distribuţiei categoriale (care este, şi ea, o generalizare a distribuţiei Bernoulli).

[74] Constanta de normalizare din expresia distribuţiei Dirichlet este $1/B(\theta_1, \ldots, \theta_k)$, unde B este funcţia *Beta* multidimensională. Distribuţia Dirichlet — o generalizare a distribuţiei Beta; vedeţi exerciţiile 43, 129 şi 130 — este distribuţie conjugată (a priori) pentru distribuţia categorială şi pentru distribuţia multinomială. (Pentru definiţia noţiunii de *distribuţii conjugate* vedeţi enunţul problemei 43.B.) Ţinând cont de faptul că $\Gamma(x) = (x-1)!$ pentru orice $x \in \mathbb{N}^*$, veţi putea observa că există o mare similaritate între expresiile (43) şi (44).

Considerând $b(x) = 1$, $\eta(\theta) = (\theta_1 - 1, \ldots, \theta_k - 1)$, $T(x) = (\ln x_1, \ldots, \ln x_k)$ **şi** $a(\theta) = -\ln \frac{\Gamma(\sum_{i=1}^k \theta_i)}{\prod_{i=1}^k \Gamma(\theta_i)}$**, rezultă că distribuţia Dirichlet este (şi ea) membră a familiei exponenţiale.**

c. Cazul distribuţiei gaussiene multidimensionale:

$$\begin{aligned} p(x|\mu,\Sigma) &= \frac{1}{(2\pi)^{k/2}\,|\Sigma|^{1/2}} \exp\Big(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\Big) \\ &= \frac{1}{(2\pi)^{k/2}\,|\Sigma|^{1/2}} \exp\Big(-\frac{1}{2}x^\top\Sigma^{-1}x + \mu^\top\Sigma^{-1}x - \frac{1}{2}\mu^\top\Sigma^{-1}\mu\Big) \\ &= \frac{1}{(2\pi)^{k/2}} \exp\Big(-\frac{1}{2}tr(\Sigma^{-1}xx^\top) + \mu^\top\Sigma^{-1}x - \frac{1}{2}\mu^\top\Sigma^{-1}\mu - \frac{1}{2}\ln|\Sigma|\Big). \end{aligned}$$

Şi această distribuţie aparţine familiei exponenţiale, fiindcă putem scrie:

$$\begin{aligned} b(x) &= (2\pi)^{-k/2} \\ \eta(\theta) &= (\Sigma^{-1}\mu; -\frac{1}{2}vec\,(\Sigma^{-1})) \\ T(x) &= (x; vec\,(xx^\top)) \\ a(\theta) &= \frac{1}{2}\mu^\top\Sigma^{-1}\mu + \frac{1}{2}\ln|\Sigma|. \end{aligned}$$

Precizăm că $x^\top\Sigma^{-1}x = tr(x^\top\Sigma^{-1}x)$ **fiindcă** $x^\top\Sigma^{-1}x \in \mathbb{R}$**, apoi** $tr(x^\top\Sigma^{-1}x) = tr(\Sigma^{-1}xx^\top)$ **fiindcă** $tr(ABC) = tr(BCA)$ **pentru orice matrice** A**,** B **şi** C **astfel încât** ABC **este matrice pătratică**[75] **şi, în sfârşit, se poate verifica uşor că** $tr(\Sigma^{-1}xx^\top) = vec\,(\Sigma^{-1}) \cdot vec\,(xx^\top)$ **fiindcă matricea** $xx^\top$ **este simetrică** ($(xx^\top)^\top = xx^\top$)**. Prin** $vec(A)$ **am desemnat vectorul obţinut prin liniarizarea matricei** A**. Similar, am notat cu** $tr(A)$ **suma elementelor de pe diagonala principală a unei matrice pătratice oarecare** A**.**

0.1.4 Estimarea parametrilor unor distribuţii probabiliste

42. **(Distribuţii de tip Bernoulli; calculul verosimilităţii datelor; estimarea parametrilor în sensul MLE)**

CMU, 2005 fall, T. Mitchell, A. Moore, midterm, pr. 1.3

Avem două monede. Probabilitatea de apariţie a stemei este θ în cazul primei monede şi 2θ în cazul celei de-a doua monede.
Presupunem că aruncăm aceste două monede de mai multe ori, în mod independent una de cealaltă, şi obţinem rezultatele din tabelul alăturat.

Moneda	Rezultat
1	stemă
2	ban
2	ban
2	ban
2	stemă

a. Care este log-verosimilitatea acestor date în funcţie de θ?

[75] Vedeţi documentul *Linear algebra review and reference* (29.09.2012) de Zico Kolter de la Stanford Univesity, pag. 9.

b. Cât este estimarea / valoarea de verosimilitate maximă (engl., maximum likelihood, ML) a lui θ?

Răspuns:

a. Dacă notăm pe de o parte cu $stemă_1$ şi ban_1 feţele primei monede şi cu $stemă_2$ şi ban_2 feţele celei de-a doua monede, iar pe de altă parte cu $x_1, x_2, \ldots, x_5$ cele cinci evenimente aleatoare din enunţul problemei, atunci verosimilitatea acestor date în raport cu parametrul θ este:

$$\begin{aligned} L(\theta) &\overset{def.}{=} P(date \mid \theta) = P(x_1 = stemă_1, x_2 = ban_2, x_3 = ban_2, x_4 = ban_2, x_5 = stemă_2 \mid \theta) \\ &\overset{i.i.d.}{=} P(stemă_1 \mid \theta) \cdot [P(ban_2 \mid \theta)]^3 \cdot P(stemă_2 \mid \theta) \\ &= \theta \cdot (1-2\theta)^3 \cdot 2\theta = 2\theta^2 \cdot (1-2\theta)^3. \end{aligned}$$

Notăm log-verosimilitatea acestor date cu $\ell(\theta)$ şi o calculăm astfel:[76]

$$\ell(\theta) \overset{def.}{=} \ln L(\theta) = \ln P(date \mid \theta) = \ln(2\theta^2 \cdot (1-2\theta)^3) = \ln 2 + 2\ln\theta + 3\ln(1-2\theta).$$

b. Estimarea de verosimilitate maximă a lui θ este prin definiţie $\hat{\theta}_{MLE} \overset{not.}{=} \operatorname{argmax}_\theta \ell(\theta)$ — mai exact, ţinând cont de enunţ, $\operatorname{argmax}_{\theta \in (0,1/2)} \ell(\theta)$ — şi se calculează cu ajutorul derivatei de ordinul întâi a funcţiei $\ell(\theta)$:

$$\frac{\partial \ell(\theta)}{\partial \theta} = \frac{\partial}{\partial \theta}(\ln 2 + 2\ln\theta + 3\ln(1-2\theta)) = 0 + \frac{2}{\theta} + \frac{3(-2)}{1-2\theta}$$

$$\frac{\partial \ell(\theta)}{\partial \theta} = 0 \Leftrightarrow \frac{2}{\theta} - \frac{6}{1-2\theta} = 0 \Leftrightarrow 2(1-2\theta) - 6\theta = 0 \Leftrightarrow 2 - 10\theta = 0 \Leftrightarrow \theta_{MLE} = \frac{1}{5} \in (0, \frac{1}{2}).$$

Se poate verifica uşor că $\theta = \frac{1}{5}$ este punct de maxim pentru funcţia ℓ. Într-adevăr, ţinând cont de faptul că din enunţ se poate deduce că $\theta \in (0,\ 1/2)$, rezultă că $\frac{\partial \ell(\theta)}{\partial \theta} \overset{not.}{=} \ell'(\theta) > 0$ pentru $\theta < \frac{1}{5}$ şi $\ell'(\theta) < 0$ pentru $\theta > \frac{1}{5}$, deci funcţia ℓ este strict crescătoare pe intervalul $(0, 1/5]$ şi strict descrescătoare pe intervalul $[1/5, 1/2)$.

43. (Distribuţia Bernoulli: estimarea parametrului în sensul MLE şi, respectiv, MAP)

■ □ ● ○ ***CMU, 2015 spring, T. Mitchell, N. Balcan, HW2, pr. 2***

Presupunem că „observăm" valorile variabilelor aleatoare $X_1, \ldots, X_n$ care sunt distribuite în mod identic şi independent (engl., independent and identically distributed, i.i.d.), conform unei singure distribuţii Bernoulli având parametrul θ. Cu alte cuvinte, pentru fiecare variabilă X_i, ştim că

$$P(X_i = 1) = \theta, \quad \textbf{iar} \quad P(X_i = 0) = 1 - \theta.$$

***Scopul* nostru în cele ce urmează este să estimăm valoarea parametrului θ pornind de la valorile „observate" ale variabilelor $X_1, \ldots, X_n$.**

[76] Am ales ca bază a logaritmului numărul e. De fapt, pentru a asigura consevarea proprietăţilor de monotonie a funcţiei de verosimilitate, ar fi suficient să lucrăm cu o bază oarecare, supraunitară, a. Într-un astfel de caz, la derivare, vom avea în plus (faţă de cazul logaritmului natural, ln) un factor constant, pozitiv.

A. Estimarea în sensul verosimilităţii maxime (engl., Maximum Likelihood Estimation, MLE)

***Introducere*:** Pentru orice valoare $\hat{\theta}$ a parametrului θ, fixată în mod arbitrar, putem să calculăm probabilitatea producerii „observaţiilor" [adică a valorilor variabilelor aleatoare] $X_1, \ldots, X_n$. Această probabilitate a [producerii] datelor observabile este adeseori numită *verosimilitatea datelor*, iar funcţia $L(\hat{\theta})$ care asociază fiecărei valori $\hat{\theta}$ verosimilitatea respectivă a datelor se numeşte *funcţia de verosimilitate*. În mod natural, a „estima" valoarea necunoscută a parametrului θ revine la a identifica acea valoare $\hat{\theta}$ [a lui θ] care maximizează funcţia de verosimilitate. În mod formal, putem scrie acest fapt astfel:

$$\hat{\theta}_{MLE} \stackrel{def.}{=} \operatorname*{argmax}_{\hat{\theta}} L(\hat{\theta}).$$

Această notaţie înseamnă: *i.* $\hat{\theta}_{MLE}$ aparţine domeniului de valori posibile ale parametrului θ, şi *ii.* $L(\hat{\theta}) \leq L(\hat{\theta}_{MLE})$ pentru orice $\hat{\theta}$ care aparţine respectivului domeniu de valori.

a. Scrieţi expresia [pentru calculul valorilor] funcţiei de verosimilitate, $L(\hat{\theta})$. Această expresie ar trebui să depindă de valorile variabilelor aleatoare X_1, $\ldots$, X_n, precum şi de $\hat{\theta}$, valoarea ipotetică a parametrului θ. Depinde oare valoarea acestei expresii de ordinea în care apar variabilele aleatoare?

b. Presupunem că $n = 10$ şi că setul de date conţine şase de 1 şi patru de 0. Scrieţi un mic program de calculator care trasează graficul funcţiei de verosimilitate pentru acest set de date pentru fiecare valoare a lui $\hat{\theta}$ din mulţimea $\{0, 0.01, 0.02, \ldots, 1.0\}$.[77] În acest grafic, axa x-ilor va trebui să-i corespundă lui $\hat{\theta}$, iar axa y-ilor lui $L(\hat{\theta})$. Pe axa y-ilor veţi scala valorile astfel încât să se poată vedea variaţia respectivelor valori. Stabiliţi cât este $\hat{\theta}_{MLE}$, identificând pe axa x-ilor acea valoare a lui $\hat{\theta}$ pentru care se atinge maximul funcţiei de verosimilitate.

c. Găsiţi *expresia analitică* (engl., closed-form formula) pentru $\hat{\theta}_{MLE}$, estimarea de verosimilitate maximă a lui $\hat{\theta}$. Pentru datele de la punctul b, concordă oare rezultatul dat de expresia analitică cu rezultatul obţinut folosind graficul?

d. Generaţi alte trei grafice pentru funcţia de verosimilitate:
unul pentru $n = 5$, setul de date conţinând trei de 1 şi doi de 0;
unul pentru $n = 100$, setul de date conţinând şaizeci de 1 şi patruzeci de 0;
unul pentru $n = 10$, cu cinci de 1 şi cinci de 0.

e. Pentru diferitele seturi de date de mai sus (la punctele b şi d), comparaţi funcţiile de verosimilitate şi estimările în sens MLE.

B. Estimarea în sensul probabilităţii maxime a posteriori (engl., Maximum A posteriori (MAP) Probability Estimation)

***Introducere*:** La estimarea în sens MLE (vedeţi partea A), am tratat valoarea „adevărată" a parametrului θ ca pe un număr fixat (nealeator). Însă alteori — de exemplu, în cazurile în care dispunem de anumite cunoştinţe a priori în legătură cu θ —, este util să-l tratăm pe θ ca fiind o variabilă aleatoare şi să exprimăm aceste cunoştinţe a priori sub forma unei distribuţii de probabilitate a priori peste θ. Să presupunem, de exemplu, că valorile [variabilelor] $X_1, \ldots, X_n$ sunt generate în modul următor:

[77] LC: Alternativ, puteţi folosi cunoştinţele de analiză matematică din liceu pentru a trasa graficul cerut aici (ori pentru a stabili, în principiu, alura lui). Similar pentru punctul d.

– Mai întâi, valoarea lui θ este generată folosind o anumită distribuţie de probabilitate a priori.
– Apoi, valorile [variabilelor] $X_1, \ldots, X_n$ sunt generate în mod independent cu ajutorul unei distribuţii Bernoulli care foloseşte pentru parametrul ei (θ) valoarea generată mai sus.

Întrucât atât θ cât şi $X_1, \ldots, X_n$ sunt văzute ca [fiind] variabile aleatoare, lor li se poate asocia o distribuţie de probabilitate comună (engl., joint probability). În acest context / cadru, o modalitate naturală de a estima valoarea lui θ constă pur şi simplu în a alege valoarea sa cea mai probabilă *ţinând cont* de distribuţia a priori aleasă şi de datele observabile $X_1, \ldots, X_n$:[78]

$$\hat{\theta}_{MAP} \stackrel{def.}{=} \underset{\hat{\theta}}{\operatorname{argmax}}\, P(\theta = \hat{\theta} | X_1, \ldots, X_n).$$

Aceasta se numeşte estimarea de probabilitate maximă a posteriori (engl., maximum a posteriori probability, MAP) a lui θ. Folosind formula lui Bayes, putem rescrie probabilitatea a posteriori a lui θ astfel:

$$P(\theta = \hat{\theta} | X_1, \ldots, X_n) = \frac{P(X_1, \ldots, X_n | \theta = \hat{\theta})\, P(\theta = \hat{\theta})}{P(X_1, \ldots, X_n)}.$$

Întrucât probabilitatea de la numitor nu depinde de $\hat{\theta}$ şi este un număr strict pozitiv, estimarea în sens MAP a lui θ poate fi scrisă astfel:

$$\hat{\theta}_{MAP} = \underset{\hat{\theta}}{\operatorname{argmax}}\, P(X_1, \ldots, X_n | \theta = \hat{\theta})\, P(\theta = \hat{\theta}) = \underset{\hat{\theta}}{\operatorname{argmax}}\, L(\hat{\theta})\, P(\theta = \hat{\theta}).$$

Cu alte cuvinte, estimarea în sens MAP a lui θ este valoarea $\hat{\theta}$ care maximizează produsul dintre funcţia de verosimilitate şi distribuţia a priori a lui θ. În cazul în care această distribuţie a priori este continuă şi funcţia ei de densitate de probabilitate (engl., probability density function, p.d.f.) este p, estimarea în sens MAP a lui θ este dată de formula

$$\hat{\theta}_{MAP} = \underset{\hat{\theta}}{\operatorname{argmax}}\, L(\hat{\theta})\, p(\hat{\theta}).$$

***Definiţie*:** Dacă, la estimarea în sens MAP, distribuţia *a posteriori* a parametrului λ în raport cu datele X, adică $P(\lambda|X)$, face parte din aceeaşi familie ca şi distribuţia *a priori* a parametrului, adică $P(\lambda)$, spunem că *i.* distribuţia a priori $P(\lambda)$ şi distribuţia a posteriori $P(\lambda|X)$ sunt *distribuţii conjugate*, şi *ii.* distribuţia a priori $P(\lambda)$ este o *conjugată* [a priori] pentru funcţia de verosimilitate $P(X|\lambda)$.[79]

În cele ce urmează vom folosi ca distribuţie a priori pentru parametrul θ distribuţia $Beta(3,3)$, care are funcţia densitate de probabilitate următoare:

$$p(\hat{\theta}) = \frac{\hat{\theta}^2 (1-\hat{\theta})^2}{B(3,3)},$$

unde $B(\alpha, \beta)$ desemnează funcţia Beta, iar $B(3,3) = \frac{1}{30}$.[80]

[78] Din punct de vedere matematic, expresia *ţinând cont* se va traduce prin folosirea unei probabilităţi condiţionate.

[79] Pentru un exemplu, vedeţi rezolvarea problemei 46.b.*ii*, unde se arată că distribuţia Gamma este distribuţie conjugată (a priori) pentru distribuţia Poisson.

[80] P.d.f.-ul distribuţiei Beta este definit astfel:

$$f(x; \alpha, \beta) = constant \cdot x^{\alpha-1}(1-x)^{\beta-1} = \frac{1}{B(\alpha,\beta)} \cdot x^{\alpha-1}(1-x)^{\beta-1} = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \cdot x^{\alpha-1}(1-x)^{\beta-1},$$

f. Să presupunem, ca la punctul b, că $n = 10$ şi că „observăm“ şase de 1 şi patru de 0. Scrieţi un mic program care trasează graficul funcţiei $\hat{\theta} \mapsto L(\hat{\theta})\, p(\hat{\theta})$ pentru aceleaşi valori ale lui $\hat{\theta}$ ca la punctul b. Estimaţi $\hat{\theta}_{MAP}$, identificând pe axa x-ilor acea valoare a lui $\hat{\theta}$ pentru care se atinge maximul funcţiei de mai sus.

g. Determinaţi formula analitică pentru $\hat{\theta}_{MAP}$, estimarea în sens MAP a lui $\hat{\theta}$. Pentru datele de la punctul f, concordă oare rezultatul dat de expresia analitică cu rezultatul obţinut folosind graficul?

h. Comparaţi estimările în sens MAP cu cele în sens MLE pentru datele de la punctul b. Explicaţi în mod succint diferenţele semnificative.

i. Comentaţi modul cum evoluează relaţia dintre estimările MAP şi MLE pe măsură ce n (numărul de „observaţii“) tinde la infinit, presupunând că în procesul de trecere la limită raportul $\#\{X_i = 1\}/\#\{X_i = 0\}$ rămâne constant.

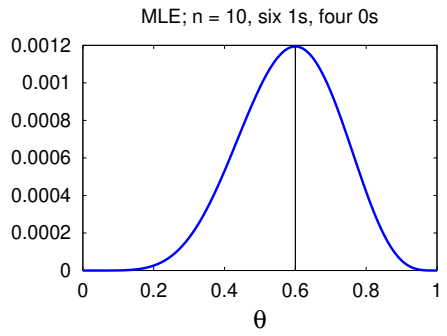
Răspuns:

a. Întrucât variabilele X_i sunt independente, putem scrie:

$$\begin{aligned} L(\hat{\theta}) &\overset{def.}{=} P_{\hat{\theta}}(X_1, \ldots, X_n) \overset{i.i.d.}{=} \prod_{i=1}^{n} P_{\hat{\theta}}(X_i) = \prod_{i=1}^{n} (\hat{\theta}^{X_i} \cdot (1-\hat{\theta})^{1-X_i}) \\ &= \hat{\theta}^{\#\{X_i=1\}} \cdot (1-\hat{\theta})^{\#\{X_i=0\}}, \end{aligned}$$

unde notaţia $\#\{\cdot\}$ desemnează numărul de variabile X_i pentru care este satisfăcută condiţia înscrisă între paranteze. La ultima egalitate am utilizat faptul că $Val(X_i) = \{0, 1\}$.[81] Evident, funcţia de verosimilitate nu depinde de ordinea prezentării datelor.

b. Prezentăm în figura alăturată graficul cerut, pe care l-am obţinut cu ajutorul unui program Matlab relativ simplu [pe care-l puteţi găsi pe site-ul acestei culegeri].



c. Din cauza produselor care apar în funcţia de verosimilitate $L(\theta)$, este mai convenabil să folosim funcţia de log-verosimilitate: $l(\theta) \overset{def.}{=} \ln(L(\theta))$. Întrucât funcţia $\ln$ este crescătoare, acea valoare $\hat{\theta}$ [a argumentului θ] pentru care se atinge maximul funcţiei de log-verosimilitate este identică cu acel $\hat{\theta}$ pentru

unde Γ desemnează funcţia Gamma a lui Euler. Vă reamintim — vedeţi problema 31.b — că $\Gamma(x) = (x-1)!$ pentru orice $x \in \mathbb{N}^*$. Pentru câteva combinaţii de valori ale parametrilor α şi β, graficele p.d.f.-urilor corespunzătoare sunt date la problema 130.

[81] Acesta este aşa-numitul *artificiu al ridicării la putere* (engl., the exponentiation trick). El va fi folosit la unele probleme de la capitolul de *Clasificare Bayesiană*, la secţiunea *Algoritmul EM/GMM* de la capitolul de *Clusterizare*, precum şi la capitolul *Schema algoritmică EM*.

care se atinge maximul funcţiei de verosimilitate. Făcând uz de proprietăţile funcţiei $\ln$**, atunci când** $\hat{\theta} \in (0,\, 1)$ **putem rescrie** $l(\hat{\theta})$ **astfel:**

$$l(\hat{\theta}) = \ln(\hat{\theta}^{n_1} \cdot (1-\hat{\theta})^{n_0}) = n_1 \ln(\hat{\theta}) + n_0 \ln(1-\hat{\theta}),$$

unde $n_1 \overset{not.}{=} \#\{X_i = 1\}$**, iar** $n_0 \overset{not.}{=} \#\{X_i = 0\}$**.**

Derivatele de ordinul întâi şi al doilea ale funcţiei de log-verosimilitate l **pentru** $\hat{\theta} \in (0,\, 1)$ **se scriu astfel:**

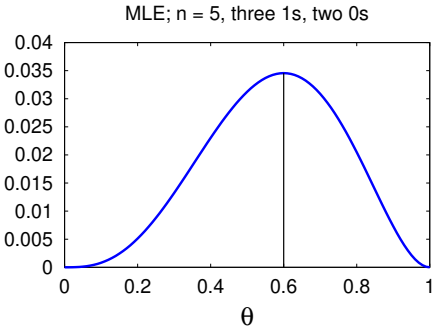
$$l'(\hat{\theta}) = \frac{n_1}{\hat{\theta}} - \frac{n_0}{1-\hat{\theta}} \quad \textbf{şi} \quad l''(\hat{\theta}) = -\Big(\frac{n_1}{\hat{\theta}^2} + \frac{n_0}{(1-\hat{\theta})^2}\Big) \textbf{ pentru } \hat{\theta} \in (0,\, 1).$$

Întrucât derivata de ordin secund a funcţiei l **este negativă pe tot domeniul ei de definiţie (adică, intervalul** $(0,\, 1)$**), rezultă că** l **este funcţie concavă şi vom putea găsi valoarea argumentului pentru care ea îşi atinge maximul rezolvând ecuaţia** $l'(\theta) = 0$**. Soluţia acestei ecuaţii se obţine printr-un calcul algebric simplu şi este**

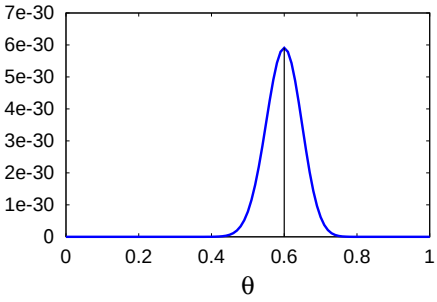
$$\hat{\theta}_{MLE} = \frac{n_1}{n_1 + n_0}.$$

Se constată uşor că această formulă / expresie este în concordanţă cu punctul de maxim din graficul care a fost obţinut la punctul b**:** $6/10 = 0.6$**.**

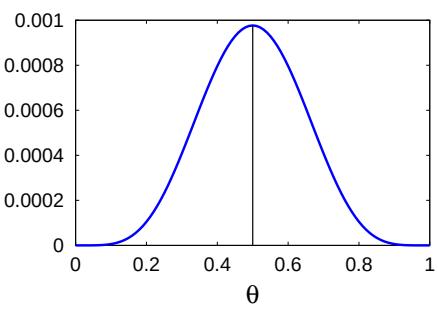
d. Pentru $n = 5$**, cu trei de 1 şi doi de 0, modificând uşor programul scris pentru punctul** b**, se va obţine graficul alăturat.**



În mod similar, pentru $n = 100$**, cu şaizeci de 1 şi patruzeci de 0 obţinem graficul alăturat.**

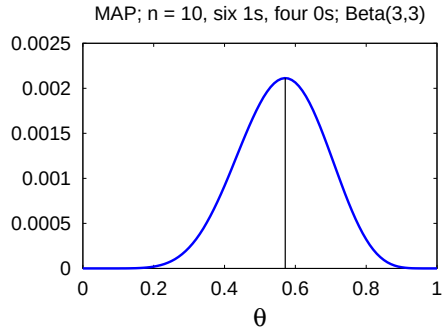


În sfârşit, pentru $n = 10$**, cu cinci de 1 şi cinci de 0 obţinem:**



e. Conform expresiei analitice obţinute la punctul c, estimarea de verosimilitate maximă (MLE) a parametrului θ al distribuţiei Bernoulli este egală cu proporţia „obervaţiilor" 1 în ansamblul datelor. Aşadar, în cazul primelor trei grafice, estimarea în sens MLE a lui θ este 0.6, iar în cazul ultimului grafic este 0.5. Pe măsură ce numărul de instanţe (n) creşte, atunci când proporţia instanţelor 1 se păstrează funcţia de verosimilitate va avea vârful „adunat" din ce în ce mai mult (sub forma unei *fleşe*[82]) în jurul valorii maxime, iar aceste valori maxime vor fi din ce în ce mai mici.

f. Un alt program Matlab relativ simplu — aflat de asemenea pe site-ul acestei culegeri — a produs graficul alăturat. (Este util să-l comparaţi cu graficul de la punctul b.) Abscisa punctului de maxim este $\theta \approx 0.571$.



g. La fel ca în cazul estimării în sensul verosimilităţii maxime (MLE), vom aplica şi aici funcţia $\ln$ înainte de a găsi valoarea lui $\hat{\theta}$ care maximizează funcţia de probabilitate a posteriori. Aşadar, urmărim să maximizăm funcţia

$$l(\hat{\theta}) = \ln(L(\hat{\theta}) \cdot p(\hat{\theta})) = \ln(\hat{\theta}^{n_1+2} \cdot (1-\hat{\theta})^{n_0+2}) - \ln(B(3,3)).$$

Constanta de normalizare pentru distribuţia a priori corespunde aici unei constante adiţionale. Prin urmare, derivatele de ordinul întâi şi al doilea [ale acestei noi funcţii] devin identice cu cele din cazul estimării în sensul MLE, cu singura diferenţă că n_1+2 şi n_0+2 îi vor înlocui pe n_1 şi respectiv n_0. Rezultă că forma analitică (engl., the closed form formula) pentru estimarea în sensul MAP a parametrului θ este următoarea:

$$\hat{\theta}_{MAP} = \frac{n_1+2}{n_1+n_0+4}$$

Se constată uşor că această formulă / expresie este în concordanţă cu punctul de maxim din graficul care a fost obţinut la punctul f: $8/14 \approx 0.571$.

h. Estimarea în sens MAP [a parametrului θ] este identică / egală cu estimarea în sens MLE dacă se mai adaugă patru variabile aleatoare *virtuale*, dintre care două iau valoarea 1, iar două iau valoarea 0. Acestea fac ca valoarea estimatorului MAP să fie împinsă mai aproape de valoarea 0.5 (şi anume, la $8/14 \approx 0.571$); din această cauză $\hat{\theta}_{MAP}$ este mai mic decât $\hat{\theta}_{MLE}$ (care este $6/10 = 0.6$).

i. Este evident că pe măsură ce n creşte, tinzând la infinit, influenţa celor patru variabile aleatoare virtuale dispare, iar cei doi estimatori devin egali:

$$\hat{\theta}_{MAP} = \frac{n_1+2}{\underbrace{n_1+n_0}_{n}+4} = \frac{\frac{n_1}{n}+\frac{2}{n}}{1+\frac{4}{n}} \to \frac{n_1}{n} = const = \hat{\theta}_{MLE}.$$

[82] Cf. DEX, *fleşă* (din fr. flèche) este un acoperiş foarte înalt, sub formă de piramidă sau de con, folosit mai ales în evul mediu la clădirile monumentale ale bisericilor.

44. **(Distribuţia categorială: estimarea parametrilor, în sensul verosimilităţii maxime (MLE))**

prelucrare de Liviu Ciortuz, după
■ □ • ○ ***CMU, 2009 spring, Ziv Bar-Joseph, HW1, pr. 2.3***

În acest exerciţiu veţi calcula estimările în sens MLE pentru parametrii unei distribuţii categoriale. O variabilă aleatoare X care urmează o astfel de distribuţie poate lua k valori (nu doar 2, cum este cazul distribuţiei Bernoulli; vedeţi problema 125), şi anume $a_1, a_2, \ldots, a_k$, probabilitatea de a vedea / „observa“ un eveniment de tip j fiind θ_j, pentru $j = 1, \ldots, k$.

a. Fie D o mulţime formată din n „observaţii“ ale lui X, independente şi identic distribuite, şi anume $\{d_1, \ldots, d_n\}$, fiecare d_i fiind una dintre cele k valori ale lui X. Vom nota cu n_i numărul care desemnează de câte ori variabila X ia / produce valoarea a_i în mulţimea D. Exprimaţi log-verosimilitatea lui D ca o funcţie de $k-1$ parametri (din totalul de k parametri) pentru distribuţia lui X, şi anume $\theta_1, \theta_2, \ldots, \theta_{k-1}$.

***Observaţie*: Calculul log-verosimilităţii în funcţie de $k-1$ parametri (în loc de k) se impune datorită restricţiei $\sum_{i=1}^{k} \theta_i = 1$. Maximizările care vor fi cerute la punctul c trebuie să ţină cont de această restricţie.**

b. Demonstraţi că funcţia de log-verosimilitate $\ln L(\theta)$ este concavă. Pentru aceasta este suficient să arătaţi că matricea hessiană a acestei funcţii este negativ definită.

c. Pentru a determina $\hat{\theta}_j$, estimarea de verosimilitate maximă (MLE) pentru θ_j, un parametru (oarecare) dintre cei $k-1$ parametri de la punctul a, vă cerem ca mai întâi să calculaţi derivata parţială a funcţiei de log-verosimilitate în raport cu θ_j. Apoi, egalând această derivată parţială cu 0, veţi determina [o expresie pentru] $\hat{\theta}_j$, rădăcina acestei ecuaţii.

d. Ca urmare a rezolvării punctelor b şi c, veţi dispune de $k-1$ relaţii de legătură între estimările în sens MLE pentru parametrii $\theta_1, \ldots, \theta_{k-1}$. Arătaţi că aceste relaţii implică faptul că pentru fiecare $j = 1, \ldots, k$, estimatorul în sens MLE pentru parametrul θ_j este egal cu $\frac{n_j}{n}$ (aşa cum era de aşteptat).

***Sugestie*: Pornind de la relaţiile deduse la punctul c pentru $\hat{\theta}_j$ pentru $j = 1, \ldots, k-1$ şi notând $\hat{\theta}_k \overset{not.}{=} 1 - \sum_{i=1}^{k-1} \hat{\theta}_j$, arătaţi că sunt satisfăcute egalităţile $\hat{\theta}_j n_k = n_j \hat{\theta}_k$ pentru $j = 1, \ldots, k-1$, iar aceste egalităţi la rândul lor implică $\hat{\theta}_k = \frac{n_k}{n}$ şi, în consecinţă, $\hat{\theta}_j = \frac{n_j}{n}$ pentru $j = 1, \ldots, k-1$.**

e. ***Comentariu*:** Rezolvarea unei astfel de probleme de estimare a verosimilităţii maxime (MLE) în care intervin şi *restricţii* — în cazul nostru, $\sum_{i=1}^{k} \theta_i = 1$ şi $\theta_i \geq 0$ pentru $i = 1, \ldots, n$ — se poate face (şi de obicei, se face) folosind *metoda multiplicatorilor lui Lagrange*. În cazul nostru, aplicarea acestei metode revine la a rezolva următoarea *problemă de optimizare*:

$$\max_{\theta_1, \ldots, \theta_k} \ln P(d_1, \ldots, d_n | \theta), \text{ a.î } \sum_{i=1}^{k} \theta_i = 1.$$

Observaţi că restricţiile $\theta_i \geq 0$ pentru $i = 1, \ldots, n$ sunt pentru moment ignorate.[83]

[83] Vom arăta ulterior că soluţia corespunzătoare aceastei probleme de optimizare (care este astfel uşor „relaxată“) satisface aceste restricţii de nenegativitate.

Conform rezolvării de la punctul a, aceast*a problemă de optimizare* este echivalentă cu următoarea:

$$\max_{\theta}(n_1 \ln\theta_1 + \ldots + n_k \ln\theta_k) \text{ a.î. } \sum_{i=1}^{k} \theta_i = 1.$$

Lagrangeanul generalizat corespunzător acestei probleme este:[84]

$$\ell(\theta, \lambda) = n_1 \ln\theta_1 + \ldots + n_k \ln\theta_k - \lambda(\theta_1 + \ldots + \theta_k - 1).$$

Aplicaţi *condiţia de staţionaritate / optimalitate*,[85] adică derivaţi $\ell(\theta, \lambda)$ în raport cu θ_j pentru $j \in \{1, \ldots, k\}$ şi egalaţi această derivată parţială cu 0 pentru a obţine expresia lui $\hat{\theta}_j$ în funcţie de [n_j şi] λ. Apoi, înlocuind expresia care tocmai a fost obţinută pentru $\hat{\theta}_j$ în *restricţia* $\sum_{j=1}^{k} \hat{\theta}_j = 1$, aflaţi valoarea implicată pentru λ. În final, veţi înlocui această valoare a lui λ în expresia obţinută anterior pentru $\hat{\theta}_j$.

Răspuns:

a. Verosimilitatea (şi apoi log-verosimilitatea) datelor se calculează astfel:

$$P(d_1, \ldots, d_n|\theta) = \prod_{j=1}^{n} \left(\sum_{i=1}^{k} \theta_i \cdot 1_{\{d_j = a_i\}} \right), \text{ unde } 1_{\{\cdot\}} \text{ este funcţia-indicator}$$

$$\text{şi } \theta \overset{not.}{=} (\theta_1, \ldots, \theta_k)$$

$$\Rightarrow L(\theta) = \prod_{i=1}^{k} \theta_i^{n_i}, \text{ unde } n_i \overset{not.}{=} \sum_{j=1}^{n} 1_{\{d_j = a_i\}}$$

$$\Rightarrow L(\theta) = \Big(1 - \sum_{i=1}^{k-1} \theta_i\Big)^{n_k} \prod_{i=1}^{k-1} \theta_i^{n_i}, \text{ fiindcă } \sum_{i=1}^{k} \theta_i = 1$$

$$\Rightarrow \ln L(\theta) = n_k \ln\Big(1 - \sum_{i=1}^{k-1} \theta_i\Big) + \sum_{i=1}^{k-1} n_i \ln\theta_i. \qquad (45)$$

b. Matricea hessiană a funcţiei de log-verosimilitate $\ln L(\theta)$ se scrie sub forma $-C - diag\left(\frac{n_1}{\theta_1^2}, \ldots, \frac{n_{k-1}}{\theta_{k-1}^2}\right)$, unde C este matricea pătratică de ordin $k-1$ ale cărei elemente sunt (toate) expresia pozitivă $\frac{n_k}{(1 - \sum_{i=1}^{k-1} \theta_i)^2}$, iar notaţia $diag\left(\frac{n_1}{\theta_1^2}, \ldots, \frac{n_{k-1}}{\theta_{k-1}^2}\right)$ desemnează matricea pătratică având pe diagonala principală elementele $\frac{n_1}{\theta_1^2}$, ..., $\frac{n_{k-1}}{\theta_{k-1}^2}$ şi 0 în rest.

Este evident că această matrice hessiană este negativ definită, ceea ce implică faptul că $\ln L(\theta)$ este funcţie [strict] concavă, iar maximul ei se află egalând cu 0 derivatele ei parţiale de ordinul întâi.

c. Vom proceda conform cerinţelor din enunţ, şi anume:

$$\frac{\partial \ln L(\theta)}{\partial \theta_j} = -\frac{n_k}{1 - \sum_{i=1}^{k-1} \theta_i} + \frac{n_j}{\theta_j}$$

[84] Vedeţi problema 82.a.
[85] Vedeţi problema 83.

$$\frac{\partial \ln L(\theta)}{\partial \theta_j} = 0 \Leftrightarrow -\frac{n_k}{1-\hat{\theta}_j - \sum_{i\neq j,k}^{k-1}\theta_i} + \frac{n_j}{\hat{\theta}_j} = 0$$

$$\Leftrightarrow \frac{n_j}{\hat{\theta}_j} = \frac{n_k}{1-\hat{\theta}_j - \sum_{i\neq j,k}^{k-1}\theta_i}$$

$$\Rightarrow n_j\Big(1 - \sum_{i\neq j,k}^{k-1}\theta_i\Big) = (n_k + n_j)\hat{\theta}_j$$

$$\Rightarrow \hat{\theta}_j = \frac{n_j}{n_j+n_k}\Big(1 - \sum_{i\neq j,k}^{k-1}\theta_i\Big) \textbf{ pentru orice } j \in \{1,\ldots,k-1\}. \quad (46)$$

***Observaţie*: Am presupus deja la relaţia** (45) **că** $\theta_i > 0$**, pentru orice** $i = 1,\ldots,k$ **ca să existe** $\ln\theta_i$ **în scrierea funcţiei de log-verosimilitate** $\ln L(\theta)$**. De asemenea, din scrierea lui** $\hat{\theta}_j$ **de pe ultima linie de mai sus, coroborat cu condiţia** $\hat{\theta}_j > 0$**, rezultă că este necesar să presupunem** $n_j > 0$**, pentru orice** $j = 1,\ldots,k$**.**

Ţinând cont de concavitatea funcţiei de log-verosimilitate care a fost demonstrată la punctul b**, în relaţia** (46) **putem înlocui** θ_i **cu** $\hat{\theta}_i$ **(valoarea optimă a parametrului** θ_i**). Aşadar, putem scrie:**

$$\hat{\theta}_j = \frac{n_j}{n_j+n_k}\Big(1 - \sum_{i\neq j,k}^{k-1}\hat{\theta}_i\Big) = \frac{n_j}{n_j+n_k}(\hat{\theta}_j + \hat{\theta}_k), \textbf{ unde } \hat{\theta}_k \overset{not.}{=} 1 - \sum_{i=1}^{k-1}\hat{\theta}_i. \quad (47)$$

***Atenţie*! Nu putem interpreta aici** $\hat{\theta}_k$ **ca fiind valoarea optimă a parametrului** θ_k**, ci pur şi simplu ca [desemnând] valoarea exprsiei** $1 - \sum_{i=1}^{k-1}\hat{\theta}_i$**.**

d. Din primul şi ultimul termen al dublei egalităţi (47) **rezultă:**

$$\hat{\theta}_j\left(1 - \frac{n_j}{n_j+n_k}\right) = \frac{n_j}{n_j+n_k}\hat{\theta}_k$$

$$\Rightarrow \hat{\theta}_j\frac{n_k}{n_j+n_k} = \frac{n_j}{n_j+n_k}\hat{\theta}_k \Rightarrow \hat{\theta}_j n_k = n_j\hat{\theta}_k.$$

Prin urmare, $\hat{\theta}_j = \frac{n_j}{n_k}\hat{\theta}_k$ **pentru orice** $j \in \{1,\ldots,k-1\}$**.**[86] **Mai departe, substituind aceste** $k-1$ **egalităţi în relaţia** $\hat{\theta}_k = 1 - \hat{\theta}_1 - \ldots - \hat{\theta}_{k-1}$**, rezultă**

$$\hat{\theta}_k = 1 - \frac{n_1}{n_k}\hat{\theta}_k - \ldots - \frac{n_{k-1}}{n_k}\hat{\theta}_k$$

$$\Rightarrow n_k\hat{\theta}_k = n_k - (n_1 + \ldots + n_{k-1})\hat{\theta}_k$$

$$\Rightarrow \hat{\theta}_k(\underbrace{n_1 + \ldots + n_{k-1} + n_k}_{n}) = n_k$$

$$\Rightarrow \hat{\theta}_k = \frac{n_k}{n}$$

$$\Rightarrow \hat{\theta}_j = \frac{n_j}{n_k}\cdot\frac{n_k}{n} = \frac{n_j}{n} \in [0,\,1], \textbf{ pentru orice } j \in \{1,\ldots,k-1\}.$$

e. Procedând conform cerinţelor din enunţ, vom avea:

$$\frac{\partial}{\partial\theta_j}\ell(\theta_j,\lambda) = 0 \Rightarrow \frac{n_j}{\theta_j} - \lambda = 0 \Rightarrow \frac{n_j}{\theta_j} = \lambda \Rightarrow \hat{\theta}_j = \frac{n_j}{\lambda}.$$

[86] *Atenţie*! Relaţia $\hat{\theta}_j = \frac{n_j}{n_k}\hat{\theta}_k$ îl conţine în partea dreaptă şi pe $\hat{\theta}_j$.

Întrucât $\sum_{j=1}^{k} \hat{\theta}_j = 1$, rezultă

$$\frac{1}{\lambda}\sum_{j=1}^{k} n_j = \frac{1}{\lambda}\, n = 1.$$

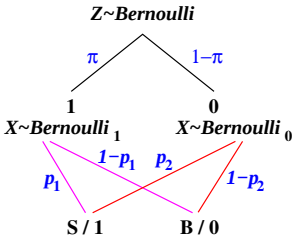
Prin urmare, $\lambda = n$, ceea ce implică $\hat{\theta}_j = \dfrac{n_j}{n} \geq 0$, pentru $j = 1, \ldots, k$. Aceste valori obţinute pentru $\hat{\theta}_j$ reprezintă într-adevăr estimările în sens MLE pentru parametrii $\theta_1, \ldots, \theta_{k-1}$ (şi nu sunt doar *puncte de staţionaritate*) întrucât funcţia $\ln L(\theta)$ este strict concavă.

45. **(O mixtură de distribuţii Bernoulli; estimarea unui parametru prin diferite metode, folosind distribuţia binomială[87])**

formulare de Liviu Ciortuz, pornind de la
■ □ • ***CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, HW1, pr. 2.b***

Presupunem că avem două monede: una este perfectă, având deci probabilitatea de apariţie a feţei cu stema $p_1 = 1/2$, iar cealaltă monedă este măsluită / imperfectă, având probabilitatea de apariţie a stemei $p_2 = 1/3$.

Facem 100 de aruncări după cum urmează. De fiecare dată alegem una dintre cele două monede. Cu o probabilitate necunoscută π, alegem moneda perfectă, iar cu probabilitatea $1 - \pi$ alegem moneda măsluită. Considerăm că am obţinut de 40 de ori stema din totalul celor 100 de aruncări.



a. Calculaţi în manieră directă / analitică estimarea de verosimilitate maximă (MLE) a parametrului π.

b. În locul metodei analitice (de la punctul a), pentru estimarea în sensul verosimilităţii maxime (MLE) a parametrului π se poate folosi medoda gradientului sau metoda lui Newton. Deduceţi regulile de actualizare pentru parametrul π în cazul medodei gradientului şi respectiv în cazul metodei lui Newton. (Vedeţi ex. 80.cd.)

***Observaţie*: Pentru o comparaţie între numărul de iteraţii [şi timpul necesar] pentru a se ajunge la convergenţă pentru metoda gradientului şi metoda lui Newton, vedeţi de exemplu pr. 5 de la capitolul *Schema algoritmică EM*.**

Răspuns:

a. Fie X o variabilă aleatoare având distribuţia probabilistă descrisă de mixtura de distribuţii Bernoulli din enunţ. Vom desemna cu $Z = 1$ (respectiv $Z = 0$) moneda perfectă (respectiv, cea imperfectă), vom nota cu H faţa stemă (engl., head) şi apoi vom considera următoarea probabilitate:

$$\begin{aligned} q &\overset{not.}{=} P(X = H) \overset{F.P.T.}{=} P(X = H, Z = 1) + P(X = H, Z = 0) \\ &= P(X = H | Z = 1) \cdot P(Z = 1) + P(X = H | Z = 0) \cdot P(Z = 0) \\ &= \frac{1}{2} \cdot \pi + \frac{1}{3} \cdot (1 - \pi) = \frac{1}{3} + \frac{\pi}{6}, \end{aligned}$$

[87] Pentru estimarea aceluiaşi parametru folosind algoritmul EM, vedeţi ex. 5 de la capitolul *Schema algoritmică EM*.

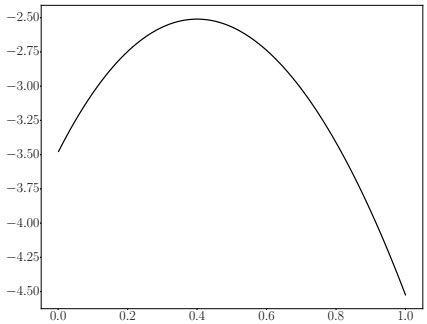
ceea ce implică $1-q=\dfrac{2}{3}-\dfrac{\pi}{6}$**.**

Dacă X' **este o variabilă aleatoare Bernoulli *echivalentă* cu mixtura dată în enunţ, putem scrie** $X' \sim Bernoulli(q)$**. În sfârşit, întrucât nu ni s-a precizat ordinea în care au fost obţinute cele 100 de rezultate / „observaţii“ (de tip stemă sau ban), vom lucra cu o [a treia] variabilă aleatoare,** $X'' \sim binomial(r; n, q)$**, unde** $n=100$**. În cazul nostru, avem** $r=40$**, deci funcţia de verosimilitate a datelor este:**

$$L(\pi)=C_{100}^{40}q^{40}(1-q)^{60}=C_{100}^{40}\Big(\frac{1}{3}+\frac{\pi}{6}\Big)^{40}\cdot\Big(\frac{2}{3}-\frac{\pi}{6}\Big)^{60}.$$

Putem scrie acum funcţia de log-verosimilitate a datelor:

$$\begin{aligned}\ell(\pi) &\overset{def.}{=} \ln(L(\pi))\\ &= \ln C_{100}^{40}+40\ln\Big(\frac{1}{3}+\frac{\pi}{6}\Big)+60\ln\Big(\frac{2}{3}-\frac{\pi}{6}\Big).\end{aligned}$$



Derivata întâi şi derivata a doua pentru funcţia de log-verosimilitate au respectiv următoarele expresii:

$$\begin{aligned}l'(\pi) &= \frac{40}{\frac{1}{3}+\frac{\pi}{6}}\cdot\frac{1}{6}-\frac{60}{\frac{2}{3}-\frac{\pi}{6}}\cdot\frac{1}{6}=\frac{40}{2+\pi}-\frac{60}{4-\pi}=\frac{40-100\pi}{(2+\pi)(4-\pi)}\\ l''(\pi) &= -\frac{40}{(2+\pi)^2}-\frac{60}{(4-\pi)^2}<0,\ \forall\pi\in[0,1],\end{aligned}$$

ceea ce implică faptul că funcţia ℓ **(ca şi** L**) are un singur punct de maxim, iar acesta este obţinut egalând prima derivată cu 0:** $\ell'(\pi)=0\Leftrightarrow\pi=\dfrac{40}{100}=0.4\in[0,1]$**. Aşadar,** $\pi_{MLE}=0.4$**.**

b. În cazul metodei gradientului ascendent, regula de „actualizare“ este următoarea:

$$\pi_{t+1}=\pi_t+\eta\ell'(\pi_t)\overset{a.}{=}\pi_t+\eta\frac{40-100\pi}{(2+\pi)(4-\pi)},$$

unde η **este *rata de învăţare*.**

În cazul metodei lui Newton, regula de „actualizare“ este următoarea:

$$\pi_{t+1}=\pi_t-\frac{\ell'(\pi_t)}{\ell''(\pi_t)}$$

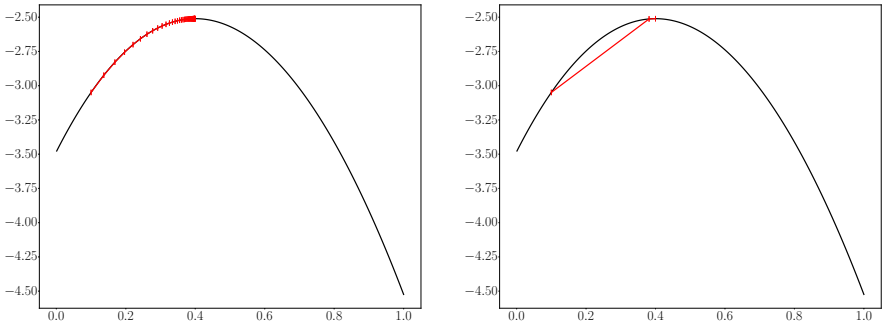
$$\frac{\ell'(\pi_t)}{\ell''(\pi_t)}\overset{a.}{=}-\frac{\cancel{20}(2-5\pi)}{(2+\pi)(4-\pi)}\cdot\frac{(2+\pi)^2(4-\pi)^2}{\cancel{20}[2(4-\pi)^2+3(2+\pi)^2]}=-\frac{(2-5\pi)(2+\pi)(4-\pi)}{5\pi^2-4\pi+44}.$$

Aşadar,

$$\pi_{t+1}=\pi_t+\frac{(2-5\pi_t)(2+\pi_t)(4-\pi_t)}{5\pi_t^2-4\pi_t+44}.$$

Graficele următoare (primul fiind pentru metoda gradientului ascendent, al doilea pentru metoda lui Newton), ca şi graficul precedent, au fost făcute de

către Andi Munteanu. Ele servesc pentru a compara evoluţiile celor două metode. Pe axa orizontală a fost reprezentată valoarea probabilităţii π, iar pe axa verticală funcţia de verosimilitate. În ambele cazuri, punctul de plecare a fost $\pi^{(0)} = 0.1$. Numărul de iteraţii necesare pentru a ajunge la convergenţă (pentru $\epsilon = 10^{-4}$): pentru metoda lui Newton a fost 3, iar în cazul metodei gradientului ascendent a fost 49 dacă rata de învăţare a fost setată la valoarea $\eta = 0.01$. Timpul necesar pentru executarea acestor iteraţii a fost de 0.00041 secunde în cazul metodei lui Newton şi de 0.00054 secunde în cazul metodei gradientului.



46. **(Distribuţia Poisson: estimarea parametrului în sens MLE şi în sens MAP, folosind distribuţia Gamma)**

prelucrare de Liviu Ciortuz, după

• ***CMU, 2005 fall, T. Mitchel, A. Moore, HW1, pr. 5***

La o fabrică de ciocolată, o persoană responsabilă cu controlul calităţii trebuie să estimeze numărul de fragmente de insecte care se găsesc în batoanele de ciocolată. Persoana respectivă procedează astfel:

– Mai întâi, sunt alese şi analizate în mod independent 15 batoane de ciocolată. Să presupunem că numărul de fragmente de insecte găsite în fiecare din aceste batoane este înregistrat în manieră vectorială astfel: $(x_1, \ldots, x_{15}) \overset{not.}{=} (2, 1, 0, 1, 0, 0, 1, 0, 2, 0, 1, 1, 0, 2, 1)$**.**

– Apoi, pentru a estima numărul de fragmente de insecte din (toate) batoanele de ciocolată, este folosită variabila aleatoare X care urmează distribuţia Poisson de parametru real $\lambda > 0$.

***Notă* (1):** Distribuţia *Poisson* de parametru $\lambda > 0$ are funcţia masă de probabilitate

$$p(x \mid \lambda) = \frac{1}{e^{\lambda}} \cdot \frac{\lambda^x}{x!}, \text{ pentru orice } x \in \mathbb{N}.$$

Prin convenţie, se consideră că $0! = 1$.

Factorul $\dfrac{1}{e^{\lambda}}$, care nu depinde de x, este aşa-numita *constantă de normalizare*.[88] Se poate demonstra că media acestei distribuţii este λ, iar varianţa tot λ.[89]

[88] Semnificaţie: $\sum_{x \in \mathbb{N}} \dfrac{\lambda^x}{x!} = e^{\lambda}$.

[89] A se vedea ex. 27.

a. Calculaţi log-verosimilitatea datelor în funcţie de λ şi arătaţi că estimarea de verosimilitate maximă a lui λ este:

$$\lambda_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

unde n este numărul de instanţe considerate. Calculaţi apoi valoarea lui λ_{MLE} corespunzătoare datelor furnizate mai sus.

b. Presupunem că valorile parametrului λ urmează în realitate distribuţia Gamma de parametri $r = 2$ şi $\alpha = 8$.

***Notă* (2):** Vă reamintim că distribuţia *Gamma* de parametri $r > 0$ (care dă *forma* distribuţiei, engl., shape) şi $\alpha > 0$ (numit *rata*, engl., rate) are funcţia densitate de probabilitate definită pe $\mathbb{R}^+$ astfel:

$$p(x) \overset{not.}{=} Gamma(x|r,\alpha) = \frac{\alpha^r}{\Gamma(r)} x^{r-1} e^{-\alpha x}, \tag{48}$$

unde Γ desemnează funcţia [Gamma a] lui Euler, care are proprietatea $\Gamma(r) = (r-1)!$ pentru orice $r \in \mathbb{N}^*$.[90] Aici *constanta de normalizare* este $\frac{\alpha^r}{\Gamma(r)}$.[91] Se poate demonstra că media distribuţiei Gamma este $\frac{r}{\alpha}$,[92] iar dacă $r > 1$ atunci *modul* ei (engl., mode) este $\frac{r-1}{\alpha}$. (*Modul* sau *valoarea dominantă* este, prin definiţie, valoarea cu frecvenţa cea mai mare de apariţie.)

În continuare veţi răspunde la următoarele întrebări:

***i*. Câte fragmente de insecte estimăm că vom găsi, în medie, într-un baton de ciocolată înainte de a face colectarea datelor?**

***Indicaţie*: Veţi lucra cu distribuţia de probabilitate comună $p(x,\lambda)$, unde $x \in \mathbb{N}$ (discret), iar $\lambda \in (0,+\infty)$. Cele două argumente ale funcţiei p nu sunt independente, fiindcă x depinde de λ prin intermediul legii de distribuţie Poisson:**

$$p(x,\lambda) = p(x|\lambda) \cdot p(\lambda).$$

Media cerută în enunţ se calculează însumând toate produsele de forma $x \cdot p(X = x)$, unde prin X am notat variabila aleatoare care modelează numărul de fragmente de insecte găsite în batoanele de ciocolată. Cum $p(x)$ este distribuţie marginală în raport cu distribuţia comună $p(x,\lambda)$, ea se va calcula astfel:

$$p(x) = \int_0^\infty p(x,\lambda)d\lambda = \int_0^\infty p(x|\lambda)p(\lambda)d\lambda.$$

Prin urmare, va trebui să calculaţi (în funcţie de datele din enunţ) valoarea expresiei:

$$E[X] \overset{def.}{=} \sum_x x \cdot p(X = x) = \sum_x x \left(\int_0^\infty p(x|\lambda)p(\lambda)d\lambda \right).$$

[90] Vedeţi problema 31.b.

[91] Semnificaţie: $\int_{x=-\infty}^{+\infty} x^{r-1}e^{-\alpha x}\,dx = \frac{\Gamma(r)}{\alpha^r}$.

[92] A se vedea ex. 31.c.

ii. **Folosind formula lui Bayes, calculaţi expresia funcţiei de densitate de probabilitate a posteriori a lui λ, notată cu $p(\lambda \mid x_1, \ldots, x_n)$.**

***Indicaţie*: Va fi util să observaţi că expresia $\lambda^{\sum x_i + r - 1} e^{-\lambda(n+\alpha)}$ seamănă foarte mult cu funcţia densitate de probabilitate a distribuţiei Gamma [de mai sus] dacă se renunţă la constanta de normalizare şi se înlocuieşte argumentul x cu λ.**

iii. **Calculaţi estimarea de probabilitate maximă a posteriori (MAP) a lui λ în raport cu datele.**

c. Dacă presupunem că valorile lui λ urmează în realitate distribuţia Gamma de parametri $r = 4$ şi $\alpha = 16$, care este estimarea de probabilitate maximă a posteriori a lui λ?

Răspuns:

a. Vom nota log-verosimilitatea datelor cu $\ell(\lambda)$ şi o vom calcula astfel:

$$\begin{aligned}
\ell(\lambda) &\overset{def.}{=} \ln p(x_1, \ldots, x_n \mid \lambda) \overset{i.i.d.}{=} \ln \prod_{i=1}^{n} p(x_i \mid \lambda) = \sum_{i=1}^{n} \ln(p(x_i \mid \lambda)) = \sum_{i=1}^{n} \ln\left(e^{-\lambda} \cdot \frac{\lambda^{x_i}}{x_i!}\right) \\
&= \sum_{i=1}^{n} \left(\ln(e^{-\lambda}) + \ln(\lambda^{x_i}) - \ln(x_i!)\right) = \sum_{i=1}^{n} \left(-\lambda + x_i \ln\lambda - \ln(x_i!)\right) \\
&= -n\lambda + \left(\sum_{i=1}^{n} x_i\right) \cdot \ln\lambda - \sum_{i=1}^{n} \ln(x_i!).
\end{aligned}$$

În cazul datelor din enunţ, avem $\sum_{i=1}^{15} x_i = 12$ şi

$$\sum_{i=1}^{15} \ln(x_i!) = 3 \cdot \ln(2!) + 6 \cdot \ln(1!) + 6 \cdot \ln(0!) = 3 \cdot \ln 2 + 6 \cdot \underbrace{\ln 1}_{=0} + 6 \cdot \underbrace{\ln 1}_{=0} = 3 \cdot \ln 2.$$

Prin urmare, log-verosimilitatea datelor este:

$$\ell(\lambda) = -15\lambda + 12 \ln\lambda - 3 \cdot \ln 2.$$

Această funcţie este concavă (fapt care se poate verifica imediat cu ajutorul derivatei de ordinul al doilea), deci λ_{MLE} este rădăcina derivatei ei de ordinul întâi. În cazul general,

$$\frac{\partial l}{\partial \lambda}\left(-n\lambda + \left(\sum_i x_i\right)\ln\lambda - \sum_{i=1}^{n}(\ln x_i!)\right) = 0 \Leftrightarrow -n + \frac{\sum_i x_i}{\lambda} = 0 \Leftrightarrow \lambda = \frac{\sum_i x_i}{n},$$

cu condiţia ca $\sum_i x_i > 0$, întrucât λ, parametrul distribuţiei Poisson trebuie să fie strict pozitiv. Aşadar, estimarea verosimilităţii maximă a lui λ corespunzătoare datelor din enunţ este:

$$\lambda_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{12}{15} = \frac{4}{5} = 0.8$$

b.*i*. Conform *Indicaţiei* din enunţ,

$$E[X] = \sum_x x\, p(x) = \sum_x x \int_\lambda p(x, \lambda) d\lambda = \sum_x x \left(\int_\lambda p(x \mid \lambda)\, p(\lambda)\, d\lambda\right)$$

$$= \int_\lambda p(\lambda) \cdot \left(\sum_x x\, p(x \mid \lambda) \right) d\lambda = E_\lambda[E_X[X \mid \lambda]],$$

unde $E_X[X \mid \lambda]$ este media variabilei X considerând parametrul λ cunoscut / fixat, iar $E_\lambda[\,\cdot\,]$ reprezintă media obţinută atunci când parametrul λ este lăsat să varieze.

Ştim că pentru orice variabilă X care urmează distribuţia Poisson de parametru λ avem $E[X] = Var(X) = \lambda$ (vedeţi *Nota* (1) din enunţ sau problema 27). Aşadar, $E_X[X \mid \lambda] = \lambda$. Apoi, conform distribuţiei Gamma, $E_\lambda[\lambda] = \frac{r}{\alpha}$ (vedeţi *Nota* (2)). Conform enunţului, $\lambda \sim Gamma(r = 2, \alpha = 8)$. Prin urmare,

$$E[X] = E_\lambda[E_X[\underbrace{X \mid \lambda}_{\sim Poisson(\lambda)}]] = E_\lambda[\underbrace{\lambda}_{\sim Gamma(r=2,\alpha=8)}] = \frac{r}{\alpha} = \frac{2}{8} = 0.25$$

b.*ii*. Folosind formula lui Bayes, vom scrie expresia funcţiei de probabilitate a posteriori astfel:

$$p(\lambda \mid x_1, \ldots, x_n) = \frac{p(x_1, \ldots, x_n \mid \lambda) \cdot p(\lambda)}{p(x_1, \ldots, x_n)} \tag{49}$$

Vom calcula pe rând expresiile care intervin în membrul drept al acestei relaţii:

$$p(x_1, \ldots, x_n \mid \lambda) \overset{i.i.d.}{=} \prod_{i=1}^n p(x_i \mid \lambda) = \prod_{i=1}^n \left(\frac{e^{-\lambda} \lambda^{x_i}}{x_i!} \right) = \frac{e^{-n\lambda} \lambda^{\sum_i x_i}}{\prod_i x_i!}$$

Aşadar,

$$\begin{aligned} p(x_1, \ldots, x_n \mid \lambda) \cdot p(\lambda) &= \frac{e^{-n\lambda} \lambda^{\sum_i x_i}}{\prod_i x_i!} \cdot \frac{\alpha^r}{\Gamma(r)} \lambda^{r-1} e^{-\alpha\lambda} \\ &= \frac{\alpha^r}{\Gamma(r) \cdot \prod_i x_i!} \cdot \lambda^{r-1+\sum_i x_i} \cdot e^{-\lambda(\alpha+n)} \end{aligned} \tag{50}$$

Totodată,

$$\begin{aligned} p(x_1, \ldots, x_n) &= \int_0^\infty p(x_1, \ldots, x_n \mid \lambda) p(\lambda) d\lambda \\ &= \int_0^\infty \frac{\alpha^r}{\Gamma(r) \cdot \prod_i x_i!} \cdot \lambda^{r-1+\sum_i x_i} \cdot e^{-\lambda(n+\alpha)} d\lambda \\ &= \frac{\alpha^r}{\Gamma(r) \cdot \prod_i x_i!} \int_0^\infty \lambda^{r-1+\sum_i x_i} \cdot e^{-\lambda(n+\alpha)} d\lambda \end{aligned}$$

Aşa cum se precizează şi în *Indicaţia* din enunţ, se observă că expresia de sub integrală arată asemănător cu funcţia densitate de probabilitate $Gamma(\lambda | r + \sum_i x_i, n + \alpha)$, însă *constanta de normalizare* lipseşte.[93] Prin urmare, vom continua calculele astfel:

$$\begin{aligned} &p(x_1, \ldots, x_n) = \\ &= \frac{\alpha^r}{\Gamma(r) \cdot \prod_i x_i!} \left(\frac{(n+\alpha)^{r+\sum_i x_i}}{\Gamma(r + \sum_i x_i)} \right)^{-1} \underbrace{\int_0^\infty \frac{(n+\alpha)^{r+\sum_i x_i}}{\Gamma(r + \sum_i x_i)} \lambda^{r-1+\sum_i x_i} \cdot e^{-\lambda(n+\alpha)} d\lambda}_{=1 \text{ (p.d.f.)}} \end{aligned}$$

[93] Altfel spus, în expresia funcţiei de densitate a distribuţiei *Gamma* din enunţ (48) facem substituţiile $x \to \lambda$, $\alpha \to n + \alpha$ şi $r \to r + \sum_i x_i$.

$$= \frac{\alpha^r}{\Gamma(r)\cdot\prod_i x_i!}\cdot\frac{\Gamma(r+\sum_i x_i)}{(n+\alpha)^{r+\sum_i x_i}} \tag{51}$$

Înlocuind cantităţile (50) şi (51) în expresia funcţiei de probabilitate a posteriori (49), vom obţine:

$$\begin{aligned} p(\lambda \mid x_1,\ldots,x_n) &= \\ &= \frac{\alpha^r}{\Gamma(r)\cdot\prod_i x_i!}\lambda^{r-1+\sum_i x_i}\cdot e^{-\lambda(\alpha+n)}\left(\frac{\alpha^r}{\Gamma(r)\cdot\prod_i x_i!}\cdot\frac{\Gamma(r+\sum_i x_i)}{(n+\alpha)^{r+\sum_i x_i}}\right)^{-1} \\ &= \frac{(n+\alpha)^{r+\sum_i x_i}}{\Gamma(r+\sum_i x_i)}\lambda^{r-1+\sum_i x_i}\cdot e^{-\lambda(\alpha+n)} \end{aligned} \tag{52}$$

Aşadar, funcţia de probabilitate a posteriori a parametrului λ are distribuţia $Gamma(\lambda|r+\sum_i x_i, n+\alpha)$.

***Observaţie*: Acest fapt înseamnă că distribuţia Poisson are ca *distribuţie a priori conjugată* distribuţia Gamma.**[94]

iii. **Estimarea de probabilitate maximă a posteriori a lui λ în raport cu datele se calculează după formula:**

$$\lambda_{MAP} = \underset{\lambda}{\operatorname{argmax}}\, p(\lambda \mid x_1,\ldots,x_n)$$

De la punctul *ii* ştim că parametrul λ este distribuit a posteriori conform distribuţiei $Gamma(\lambda|r+\sum_i x_i, n+\alpha)$, deci valoarea căutată este *modul* acestei distribuţii (vedeţi *Nota* (2) din enunţ):

$$\lambda_{MAP} = \frac{(r+\sum_i x_i)-1}{n+\alpha} = \frac{2+12-1}{15+8} = \frac{13}{23} \approx 0.56$$

Prin urmare, în condiţiile de la punctul b se observă că estimarea de probabilitate maximă a posteriori pentru parametrul λ este $\lambda_{MAP} = 0.56$, care este semnificativ mai mică decât $\lambda_{MLE} = 0.8$, estimarea de verosimilitate maximă a lui λ (vedeţi punctul a).

c. Dacă $\lambda \sim Gamma(r=4, \alpha=16)$, atunci estimarea de probabilitate maximă a posteriori a lui λ este:

$$\lambda_{MAP} = \frac{(r+\sum_i x_i)-1}{n+\alpha} = \frac{4+12-1}{15+16} = \frac{15}{31} \approx 0.48$$

Aşadar, în noile condiţii se obţine o valoare şi mai mică pentru λ_{MAP}.

47. **(O distribuţie uniformă (continuă) definită pe $\mathbb{R}$: estimarea parametrului, în sensul verosimilităţii maxime (MLE))**

prelucrare de Liviu Ciortuz, după

□ • ○ ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, midterm, pr. 5***

Presupunem că instanţele $x_1,\ldots,x_n$ au fost generate în mod independent de către o distribuţie uniformă continuă $U(-w,w)$, având parametrul $w>0$. Funcţia densitate de probabilitate (p.d.f.) a acestei distribuţii este:

[94] Pentru definiţia noţiunii de *distribuţii conjugate* vedeţi enunţul problemei 43.B.

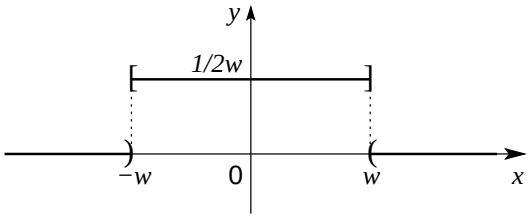
$$p(x) = \begin{cases} 0, & \text{dacă } x < -w; \\ \dfrac{1}{2w}, & \text{dacă } -w \le x \le w; \\ 0, & \text{dacă } x > w. \end{cases}$$

a. Desenaţi graficul funcţiei p şi apoi demonstraţi că într-adevăr p reprezintă o funcţie de densitate de probabilitate.

b. Găsiţi formula de calcul pentru estimarea de verosimilitate maximă (MLE) a lui w.

Răspuns:

a. Graficul funcţiei p este prezentat în figura alăturată.



Pentru ca funcţia p să reprezinte o p.d.f., ea trebuie să satisfacă două cerinţe: *i.* $p(x) \ge 0$ **pentru orice x din domeniul de definiţie şi** *ii.* $\int_{-\infty}^{\infty} p(x)\,dx = 1$**.**

Prima dintre aceste două cerinţe este imediat satisfăcută, fiindcă $w > 0$. Pentru a verifica îndeplinirea celei de-a doua cerinţe, vom calcula integrala definită:

$$\begin{aligned} \int_{-\infty}^{\infty} p(x)\,dx &= \int_{-\infty}^{-w} \underbrace{p(x)}_{0}\,dx + \int_{-w}^{w} p(x)\,dx + \int_{w}^{\infty} \underbrace{p(x)}_{0}\,dx \\ &= \int_{-w}^{w} p(x)\,dx = \int_{-w}^{w} \frac{1}{2w}\,dx = \frac{1}{2w}\int_{-w}^{w} dx \\ &= \frac{1}{2w}\cdot\ x|_{-w}^{w} = \frac{1}{2w}\cdot(w-(-w)) = \frac{1}{2w}\cdot 2w = 1. \end{aligned}$$

b. Vom nota cu $\hat{w}$ estimarea de verosimilitate maximă a lui w. Conform definiţiei, $\hat{w} = \arg\max_{w>0} p(x_1,\ldots,x_n|w)$**. Întrucât instanţele $x_1,\ldots,x_n$ au fost generate de către p în mod independent unele de altele, rezultă că** $\hat{w} = \arg\max_{w>0}\prod_{i=1}^{n} p(x_i|w)$**.**

În cele ce urmează vom nota cu x_M maximul dintre $|x_1|,|x_2|,\ldots,|x_n|$, deci

$$x_M = \max\{|x_1|,|x_2|,\ldots,|x_n|\}.$$

În cazul în care $w < x_M$, conform definiţiei funcţiei p rezultă că $\prod_{i=1}^{n} p(x_i|w) = 0$. În caz contrar, adică atunci când $w \ge x_M$, conform aceleiaşi definiţii a lui p vom avea $\prod_{i=1}^{n} p(x_i|w) = \dfrac{1}{(2w)^n}$, care este o cantitate pozitivă. Prin urmare,

$$\begin{aligned} \hat{w} &= \arg\max_{w\ge x_M} \frac{1}{(2w)^n} = \arg\max_{w\ge x_M} \ln\frac{1}{(2w)^n} = \arg\max_{w\ge x_M} -n\ln(2w) \\ &= \arg\min_{w\ge x_M} n\ln(2w) = \arg\min_{w\ge x_M} \ln w = \arg\min_{w\ge x_M} w = x_M. \end{aligned}$$

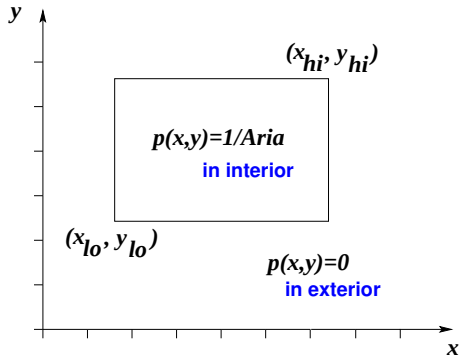
***Observaţie*: La cea de-a doua egalitate din şirul egalităţilor de mai sus, am utilizat faptul că funcţia** $\ln$ **este strict crescătoare; deci, aplicând-o unei expresii oarecare, ea conservă monotonia.**

Aşadar, am obţinut rezultatul $\hat{w} = x_M = \max\{|x_1|, |x_2|, \ldots, |x_n|\}$.

48. (Variabile aleatoare uniforme continue definite pe $\mathbb{R}^2$: funcţia densitate de probabilitate; distribuţii comune, marginale, condiţionale; formula lui Bayes)

CMU, 2002 fall, Andrew Moore, midterm, pr. 2

Figura alăturată ilustrează o clasă simplă de funcţii densitate de probabilitate (p.d.f.) definite peste perechi de variabile continue reale x, y.
Numim această clasă de funcţii *Rectangle-PDF*. O funcţie din această clasă este desemnată în mod generic prin $Rectangle(x_{lo}, y_{lo}, x_{hi}, y_{hi})$.
Aşadar, *parametrii* clasei *Rectangle-PDF* sunt cele 4 coordonate: x_{lo}, y_{lo}, x_{hi} **şi** y_{hi}.



Definiţia funcţiei densitate de probabilitate $Rectangle(x_{lo}, y_{lo}, x_{hi}, y_{hi})$ **este:**

$$p(x,y) = \begin{cases} \dfrac{1}{(x_{hi} - x_{lo})(y_{hi} - y_{lo})} & \text{dacă } x_{lo} \le x \le x_{hi} \text{ şi } y_{lo} \le y \le y_{hi} \\ 0 & \text{în caz contrar.} \end{cases}$$

***Observaţie*: Se poate arăta imediat că** $\int_x \int_y p(x,y)dx\,dy = 1$.

a. Pentru $Rectangle(0, 0, 1/2, 2)$**, calculaţi:** $p(x = 1/4, y = 1/4)$, $p(y = 1/4)$, $p(x = 1/4)$, $p(x = 1/4 \mid y = 1/4)$.

b. Fie D **o mulţime de** R **puncte,** $(x_1, y_1), (x_2, y_2), \ldots, (x_R, y_R)$**, selectate din** $Rectangle(x_{lo}, y_{lo}, x_{hi}, y_{hi})$ **în mod independent unele de altele.**[95] **Pentru simplitate, vom nota** $\theta = (x_{lo}, y_{lo}, x_{hi}, y_{hi})$**. Prin definiţie, verosimilitatea datelor** D **este** $P(D \mid \theta)$.

Dacă vrem să găsim $\theta^{MLE} \overset{not.}{=} (x_{lo}^{MLE}, y_{lo}^{MLE}, x_{hi}^{MLE}, y_{hi}^{MLE})$**, valorile parametrilor care maximizează *verosimilitatea* datelor** D**, atunci este *evident* că vom lua**[96]

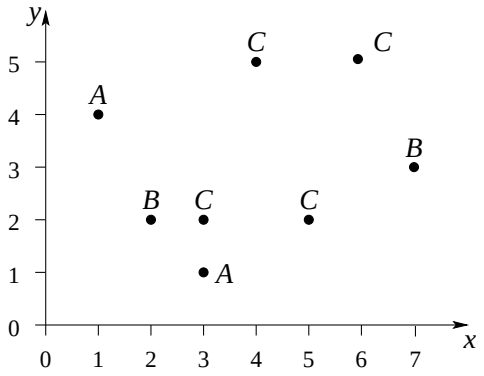
$$x_{lo}^{MLE} = \min_k\{x_k\},\ y_{lo}^{MLE} = \min_k\{y_k\},\ x_{hi}^{MLE} = \max_k\{x_k\},\ y_{hi}^{MLE} = \max_k\{y_k\}.$$

În continuare, ca date concrete, vom folosi punctele $(x_i, y_i), i = 1, ..., 6$**, clasificate în trei „concepte“** A, B, C **care fac parte din clasa *Rectangle-PDF*:**

[95] În mod riguros, în acest context conceptul $Rectangle(x_{lo}, y_{lo}, x_{hi}, y_{hi})$ desemnează mulţimea punctelor din plan pentru care funcţia omonimă $Rectangle(x_{lo}, y_{lo}, x_{hi}, y_{hi})$ care a fost definită mai sus ia valori nenule. Această mulţime este exact dreptunghiul reprezentat în imaginea de mai sus şi punctele din interiorul lui. Altfel spus, cu suprapunere de notaţie, funcţia $Rectangle(x_{lo}, y_{lo}, x_{hi}, y_{hi})$ desemnează conceptul geometric $Rectangle(x_{lo}, y_{lo}, x_{hi}, y_{hi})$.

[96] Pentru justificare riguroasă, vedeţi cazul (mai simplu) al problemei 47.

x	y	*Concept*
1	4	A
3	1	A
2	2	B
7	3	B
3	2	C
4	5	C
5	2	C
6	5	C



Vom avea, de *exemplu*, $p(2.5, 2.5 \mid A) \overset{not.}{=} p_{X,Y}(2.5, 2.5 \mid A) = 1/6$**, fiindcă** A **este estimat în sensul verosimilităţii maxime (MLE) ca fiind** $Rectangle(1, 1, 3, 4)$**, deci cele două laturi ale dreptunghiului** A **sunt de mărime 2 şi respectiv 3.**

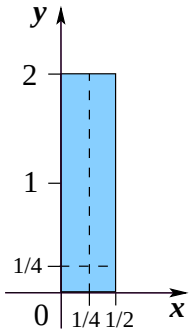
Folosind formula lui Bayes, calculaţi:

– $P(Concept = A \mid x = 1.5, y = 3)$

– $P(Concept = A \mid x = 2.5, y = 2.5)$

– $P(Concept = A \mid y = 5)$**.**

Răspuns:



a. În figura alăturată avem reprezentarea grafică a funcţiei $Rectangle(0, 0, 1/2, 2)$**. Cum punctul** $(1/4, 1/4)$ **se găseşte în interiorul dreptunghiului (**$0 < \frac{1}{4} < \frac{1}{2}$ **şi** $0 < \frac{1}{4} < 2$**), rezultă că valoarea p.d.f. comune cerute este:**

$$p\left(x = \frac{1}{4}, y = \frac{1}{4}\right) = \frac{1}{\left(\frac{1}{2} - 0\right)(2 - 0)} = \frac{1}{\frac{1}{2} \cdot 2} = 1$$

Valorile p.d.f. marginale cerute sunt:

$$\begin{aligned}
p\left(y = \frac{1}{4}\right) &= \int_{-\infty}^{\infty} p(x, y = \frac{1}{4})\, dx \\
&= \int_{-\infty}^{0} p(x, y = \frac{1}{4})\, dx + \int_{0}^{1/2} p(x, y = \frac{1}{4})\, dx + \int_{1/2}^{\infty} p(x, y = \frac{1}{4})\, dx \\
&= \int_{-\infty}^{0} 0\, dx + \int_{0}^{1/2} \frac{1}{\left(\frac{1}{2} - 0\right)(2 - 0)}\, dx + \int_{1/2}^{\infty} 0\, dx \\
&= 0 + \int_{0}^{1/2} 1\, dx + 0 = x|_{0}^{1/2} = \frac{1}{2} - 0 = \frac{1}{2} \\
p\left(x = \frac{1}{4}\right) &= \int_{-\infty}^{\infty} p(x = \frac{1}{4}, y)\, dy \\
&= \int_{-\infty}^{0} p(x = \frac{1}{4}, y)\, dy + \int_{0}^{2} p(x = \frac{1}{4}, y)\, dy + \int_{2}^{\infty} p(x = \frac{1}{4}, y)\, dy
\end{aligned}$$

$$= \int_{-\infty}^{0} 0\,dy + \int_{0}^{2} \frac{1}{\left(\frac{1}{2}-0\right)(2-0)}\,dy + \int_{2}^{\infty} 0\,dy = 0 + \int_{0}^{2} 1\,dy + 0 = y|_0^{1/2}$$

$$= 2 - 0 = 2$$

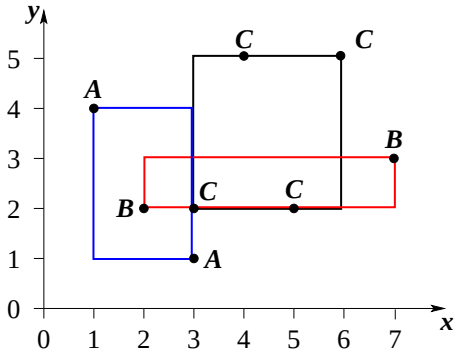
În sfârşit, valoarea p.d.f. condiţionale cerute este:

$$p\left(x = \frac{1}{4} \mid y = \frac{1}{4}\right) = \frac{p(x = \frac{1}{4}, y = \frac{1}{4})}{p(y = \frac{1}{4})} = \frac{\frac{1}{1}}{2} = 2$$

***Observaţie*: Aceeaşi valoare pentru $p(x = \frac{1}{4} \mid y = \frac{1}{4})$ se obţine şi dacă observăm că variabilele x şi y sunt independente, deci $p(x = \frac{1}{4} \mid y = \frac{1}{4}) = p(x = \frac{1}{4}) = 2$.**

b. Ţinând cont de date (vedeţi tabelul din enunţ), cele trei concepte A, B, C vor fi estimate în sensul verosimilităţii maxime ca:

$$Rectangle_A(1, 1, 3, 4)$$
$$Rectangle_B(2, 2, 7, 3)$$
$$Rectangle_C(3, 2, 6, 5)$$



Aşadar, vom avea:

$$p(x, y \mid A) = \begin{cases} \frac{1}{(3-1)(4-1)} = \frac{1}{6}, & \textbf{dacă } 1 \le x \le 3 \textbf{ şi } 1 \le y \le 4 \\ 0, & \textbf{altfel;} \end{cases}$$

$$p(x, y \mid B) = \begin{cases} \frac{1}{(7-2)(3-2)} = \frac{1}{5}, & \textbf{dacă } 2 \le x \le 7 \textbf{ şi } 2 \le y \le 3 \\ 0, & \textbf{altfel;} \end{cases}$$

$$p(x, y \mid C) = \begin{cases} \frac{1}{(6-3)(5-2)} = \frac{1}{9}, & \textbf{dacă } 3 \le x \le 6 \textbf{ şi } 2 \le y \le 5 \\ 0, & \textbf{altfel.} \end{cases}$$

Acum putem calcula cele trei probabilităţi condiţionate cerute, folosind formula lui Bayes. Probabilităţile a priori $P(A) = \frac{2}{8}$, $P(B) = \frac{2}{8}$ şi $P(C) = \frac{4}{8}$ care intervin în aplicarea acestei formule au fost estimate din setul de date D din enunţ, în sensul verosimilităţii maxime. (A se vedea *Observaţia* de mai jos.)

$$P(Concept = A \mid x = 1.5, y = 3)$$
$$= \frac{p(x = 1.5, y = 3 \mid A) \cdot P(A)}{p(1.5, 3 \mid A) \cdot P(A) + p(1.5, 3 \mid B) \cdot P(B) + p(1.5, 3 \mid C) \cdot P(C)}$$
$$= \frac{\frac{1}{6} \cdot \frac{2}{8}}{\frac{1}{6} \cdot \frac{2}{8} + 0 \cdot \frac{2}{8} + 0 \cdot \frac{4}{8}} = 1$$

$$P(Concept = A \mid x = 2.5, y = 2.5)$$
$$= \frac{p(x = 2.5, y = 2.5 \mid A) \cdot P(A)}{p(2.5, 2.5 \mid A) \cdot P(A) + p(2.5, 2.5 \mid B) \cdot P(B) + p(2.5, 2.5 \mid C) \cdot P(C)}$$
$$= \frac{\frac{1}{6} \cdot \frac{2}{8}}{\frac{1}{6} \cdot \frac{2}{8} + \frac{1}{5} \cdot \frac{2}{8} + 0 \cdot \frac{4}{8}} = \frac{\frac{1}{6}}{\frac{1}{6} + \frac{1}{5}} = \frac{5}{5+6} = \frac{5}{11}$$

$$P(Concept = A \mid y = 5) = \frac{p(y = 5 \mid A) \cdot P(A)}{p(y = 5)}$$
$$= \frac{[\int_{-\infty}^{\infty} p(x = t, y = 5 \mid A)dt] \cdot P(A)}{p(y = 5)} = \frac{\int_{-\infty}^{\infty} 0dt \cdot P(A)}{p(y = 5)} = \frac{0 \cdot P(A)}{p(y = 5)} = 0$$

***Observaţie*: Aşa cum am precizat deja, valorile probabilităţilor a priori $P(A) = \frac{2}{8}$, $P(B) = \frac{2}{8}$ şi $P(C) = \frac{4}{8}$ au fost estimate (în sensul verosimilităţii maxime) din tabelul de date din enunţ. Pentru a înţelege mai bine de ce s-a procedat aşa, ar fi fost mai explicit dacă făceam condiţionarea (şi) în funcţie de D, setul de date din enunţ:**

$$P(Concept = A \mid x = 1.5, y = 3, D) =$$
$$\frac{p(x = 1.5, y = 3 \mid A, D) \cdot P(A \mid D)}{p(1.5, 3 \mid A, D) \cdot P(A \mid D) + p(1.5, 3 \mid B, D) \cdot P(B \mid D) + p(1.5, 3 \mid C, D) \cdot P(C \mid D)}$$

Totuşi, din motive legate de simplitate, am optat pentru notaţia folosită mai sus.

49. **(Distribuţia exponenţială: estimarea parametrului în sensul verosimilităţii maxime (MLE))**

□ • ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, final exam, pr. 1.a***

Presupunem că valorile reale $x_1, x_2, \ldots, x_n$ au fost obţinute prin eşantionare stochastică folosind o distribuţie $p(x)$ de forma

$$p(x \mid \lambda) = \begin{cases} \lambda e^{-\lambda x} & \textbf{pentru } x \geq 0 \\ 0 & \textbf{pentru } x < 0. \end{cases}$$

unde $\lambda > 0$ este un parametru necunoscut. Aceasta este *distribuţia exponenţială*.[97]

Care dintre următoarele expresii reprezintă valoarea estimată pentru λ, obţinută prin aplicarea metodei verosimilităţii maxime pe aceste date? (Se va presupune că în mulţimea dată toate elementele x_i sunt mai mari decât 1.)

i. $\frac{\sum_{i=1}^{n} \ln(x_i)}{n}$ *ii.* $\frac{\max_{i=1}^{n} \ln(x_i)}{n}$ *iii.* $\frac{n}{\sum_{i=1}^{n} \ln(x_i)}$ *iv.* $\frac{n}{\max_{i=1}^{n} \ln(x_i)}$

v. $\frac{\sum_{i=1}^{n} x_i}{n}$ *vi.* $\frac{\max_{i=1}^{n} x_i}{n}$ *vii.* $\frac{n}{\sum_{i=1}^{n} x_i}$ *viii.* $\frac{n}{\max_{i=1}^{n} x_i}$

[97] Se poate demonstra că media şi varianţa distribuţiei exponenţiale sunt λ^{-1} şi respectiv λ^{-2}. A se vedea ex. 31.a.

$$ix.\ \frac{\sum_{i=1}^n x_i^2}{n} \quad x.\ \frac{\max_{i=1}^n x_i^2}{n} \quad xi.\ \frac{n}{\sum_{i=1}^n x_i^2} \quad xii.\ \frac{n}{\max_{i=1}^n x_i^2}$$
$$xiii.\ \frac{\sum_{i=1}^n e^{x_i}}{n} \quad xiv.\ \frac{\max_{i=1}^n e^{x_i}}{n} \quad xv.\ \frac{n}{\sum_{i=1}^n e^{x_i}} \quad xvi.\ \frac{n}{\max_{i=1}^n e^{x_i}}$$

Răspuns:

Verosimilitatea datelor $x_1, \ldots, x_n$ în raport cu parametrul λ se calculează astfel:

$$P(x_1, \ldots, x_n \mid \lambda) \overset{i.i.d.}{=} \prod_{i=1}^n p(x_i \mid \lambda) = \prod_{i=1}^n \lambda e^{-\lambda x_i} = \lambda^n \cdot \prod_{i=1}^n e^{-\lambda x_i} = \lambda^n \cdot e^{-\lambda \sum_{i=1}^n x_i}$$

Aşadar, funcţia de log-verosimilitate se va scrie astfel:

$$l(\lambda) \overset{def.}{=} \ln P(x_1, \ldots, x_n \mid \lambda) = \ln \left(\lambda^n \cdot e^{-\lambda \sum_{i=1}^n x_i} \right) = n \ln \lambda - \lambda \sum_{i=1}^n x_i.$$

Derivata acestei funcţii este

$$l'(\lambda) = \frac{n}{\lambda} - \sum_{i=1}^n x_i, \quad \textbf{pentru orice } \lambda > 0.$$

Rădăcina lui l' se obţine uşor:

$$\frac{n}{\lambda} - \sum_{i=1}^n x_i = 0 \Leftrightarrow \lambda = \frac{n}{\sum_{i=1}^n x_i},$$

iar această valoare este strict pozitivă fiindcă $x_i > 1$ pentru $i = 1, \ldots, n$ (vedeţi enunţul). Este imediat că $l'(\lambda) > 0$ pentru $\lambda < \frac{n}{\sum_{i=1}^n x_i}$ şi $l'(\lambda) < 0$ pentru $\lambda > \frac{n}{\sum_{i=1}^n x_i}$. Prin urmare, funcţia l are un punct de maxim. Abscisa acestui punct de maxim este rădăcina primei derivate. Aşadar, valoarea de verosimilitate maximă a lui λ este

$$\lambda_{MLE} = \frac{n}{\sum_{i=1}^n x_i}$$

În concluzie, răspunsul corect este *vii*.

50. (Distribuţia gaussiană unidimensională: estimarea mediei în sensul MLE şi respectiv MAP, atunci când varianţa este cunoscută)

■ • ***CMU, 2011 fall, T. Mitchell, A. Singh, HW2, pr. 1***

În această problemă ne propunem să calculăm *estimatorul de verosimilitate maximă* (engl., maximum likelihood estimator, MLE) şi *estimatorul de probabilitate maximă a posteriori* (engl., maximum a posteriori probability (MAP) estimator) pentru media unei distribuţii gaussiene unidimensionale. Concret, presupunem că avem n instanţe, $x_1, \ldots, x_n$ generate în mod independent de către o distribuţie normală cu varianţa *cunoscută*, σ^2, şi media *necunoscută*, μ.

a. Calculaţi estimatorul MLE pentru media μ. Elaboraţi calculele în mod detaliat.

b. Arătaţi că $E[\mu_{MLE}] = \mu$.

***Observaţie*: Această egalitate înseamnă că estimatorul μ_{MLE} este *fără deplasare* sau *nedeplasat* (engl., unbiased).**

c. Calculaţi $Var[\mu_{MLE}]$. Ce tendinţă au valorile acestei varianţe atunci când numărul de instanţe (n) tinde la infinit?

d. Presupunem că media urmează la rândul ei o distribuţie normală de medie ν şi varianţă β^2. Calculaţi estimatorul MAP pentru media μ. Elaboraţi calculele în mod detaliat.

e. Ce se întâmplă cu estimatorii MLE and MAP atunci când numărul de instanţe (n) tinde la infinit?

Răspuns:

a. Funcţia de verosimilate a datelor se exprimă astfel:

$$L(\mu) \stackrel{def.}{=} p(x_1, \ldots, x_n|\mu) = \prod_{i=1}^{n} p(x_i|\mu) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

Pentru a doua egalitate am ţinut cont de proprietatea de independenţă a datelor, iar ulterior am folosit definiţia funcţiei de densitate a distribuţiei gaussiene de medie μ şi varianţă σ^2. Funcţia de log-verosimilitate este:

$$l(\mu) \stackrel{def.}{=} \ln p(x_1, \ldots, x_n|\mu) = \sum_{i=1}^{n} \left(\ln \frac{1}{\sqrt{2\pi}\sigma} - \frac{(x_i-\mu)^2}{2\sigma^2} \right)$$

În scrierea expresiei funcţiei de log-verosimilitate am aplicat proprietăţile logaritmului. Maximul funcţiei de (log-)verosimilitate se află cu ajutorul derivatei:[98]

$$\frac{\partial}{\partial \mu} l(\mu) = \sum_{i=1}^{n} \frac{x_i - \mu}{\sigma^2}$$

Rădăcina derivatei de ordinul întâi a funcţiei de (log-)verosimilitate se calculează uşor:

$$\begin{aligned} \frac{\partial}{\partial \mu} l(\mu) = 0 \quad &\Leftrightarrow \quad \sum_{i=1}^{n} \frac{x_i - \mu}{\sigma^2} = 0 \Leftrightarrow \sum_{i=1}^{n} (x_i - \mu) = 0 \Leftrightarrow \sum_{i=1}^{n} x_i = n\mu \\ &\Leftrightarrow \quad \mu = \frac{\sum_{i=1}^{n} x_i}{n} \end{aligned}$$

Prin urmare, estimarea de verosimilitate maximă a parametrului μ (media) pentru distribuţia gaussiană este $\mu_{MLE} = \frac{\sum_{i=1}^{n} x_i}{n}$.

b. Considerând instanţele $x_1, \ldots, x_n$ ca fiind generate respectiv de variabilele aleatoare $X_1, \ldots, X_n$, toate având aceeaşi distribuţie gaussiană (cu media μ şi

[98] Se va observa imediat că derivata de ordinul al doilea este strict negativă, deci funcţia de (log-)verosimilitate este strict concavă.

varianţa σ^2), rezultă $\mu_{MLE} = \frac{\sum_{i=1}^n X_i}{n}$. Este natural să considerăm μ_{MLE} ca fiind o variabilă aleatoare. Aplicând proprietatea de liniaritate a mediilor variabilelor aleatoare, vom obţine:

$$E[\mu_{MLE}] = E\left[\frac{X_1 + \ldots + X_n}{n}\right] = \frac{E[X_1] + \ldots + E[X_n]}{n} = \frac{n\mu}{n} = \mu$$

c. Ţinând cont de faptul că $Var[aX] = a^2\, Var[X]$ pentru orice variabilă aleatoare X şi orice constantă $a \in \mathbb{R}$, precum şi de faptul că varianţa unei sume de variabilele independente independente este suma varianţelor lor (vedeţi pr. 23.c), rezultă:

$$Var[\mu_{MLE}] \quad = \quad Var\left[\frac{1}{n}\sum_{i=1}^n X_i\right] = \frac{1}{n^2}\sum_{i=1}^n Var[X_i] = \frac{1}{n^2} n\sigma^2 = \frac{\sigma^2}{n}$$

În consecinţă, $Var[\mu_{MLE}] \to 0$ atunci când $n \to \infty$.

d. Funcţia care ne interesează de data aceasta este $p(\mu|x_1, \ldots, x_n)$, probabilitatea (de fapt, p.d.f.) a posteriori a parametrului μ, date fiind instanţele $x_1, \ldots, x_n$ generate cu ajutorul unei distribuţii gaussiene de medie μ şi varianţă σ^2.[99] Ţinând cont mai întâi de teorema lui Bayes, iar apoi de expresia funcţiei de verosimilitate $L(\mu)$ obţinută la punctul a precum şi de informaţia din enunţ conform căreia parametrul μ urmează o distribuţie gaussiană de medie ν şi varianţă β^2, vom putea scrie:

$$p(\mu|x_1, \ldots, x_n) \overset{T.\ Bayes}{=} \frac{p(x_1, \ldots, x_n|\mu)\, p(\mu)}{p(x_1, \ldots, x_n)} \tag{53}$$

$$= \quad \frac{\left(\prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right)\right) \cdot \frac{1}{\sqrt{2\pi}\beta} \exp\left(-\frac{(\mu-\nu)^2}{2\beta^2}\right)}{C} \tag{54}$$

unde probabilitatea $p(x_1, \ldots, x_n)$ a fost înlocuită cu simbolul C (o constantă), fiindcă această probabilitate nu depinde de parametrul μ.

Ca şi la punctul precedent, pentru studiul maximului acestei funcţii este convenabil ca în prealabil să o logaritmăm:

$$\ln p(\mu|x_1, \ldots, x_n) = -\sum_{i=1}^n \left(\ln\sqrt{2\pi}\sigma + \frac{(x_i-\mu)^2}{2\sigma^2}\right) - \ln\sqrt{2\pi}\beta - \frac{(\mu-\nu)^2}{2\beta^2} - \ln C$$

Maximul acestei funcţii se obţine tot cu ajutorul derivatei:[100]

$$\frac{\partial}{\partial\mu} \ln p(\mu|x_1, \ldots, x_n) = \sum_{i=1}^n \frac{x_i - \mu}{\sigma^2} - \frac{\mu - \nu}{\beta^2}$$

$$\frac{\partial}{\partial\mu} \ln p(\mu|x_1, \ldots, x_n) = 0 \Leftrightarrow \sum_{i=1}^n \frac{x_i - \mu}{\sigma^2} = \frac{\mu - \nu}{\beta^2} \Leftrightarrow \mu\left(\frac{1}{\beta^2} + \frac{n}{\sigma^2}\right) = \frac{\sum_{i=1}^n x_i}{\sigma^2} + \frac{\nu}{\beta^2}$$

[99] De fapt, mai corect ar fi ca şi aici — şi peste tot la acest punct — în loc de $p(\mu|x_1, \ldots, x_n)$ să scriem $p(\mu|x_1, \ldots, x_n, \nu, \beta)$. Similar, în loc de $p(\mu)$ ar trebui să scriem $p(\mu|\nu, \beta)$. Pentru a simplifica redactarea calculelor, vom presupune că se subînţelege de fiecare dată „condiţionarea“ în raport cu parametrii ν şi β, acolo unde este necesară.

[100] Veţi observa că derivata de ordinul al doilea este negativă, ceea ce convine.

Aşadar, estimarea de probabilitate maximă a posteriori a parametrului μ (media) pentru distribuţia gaussiană este $\mu_{MAP} = \dfrac{\sigma^2\nu + \beta^2\sum_{i=1}^n x_i}{\sigma^2 + n\beta^2}$.

O altă soluţie:
În loc să studiem probabilitatea a posteriori $p(\mu|x_1,\ldots,x_n)$ cu ajutorul derivatelor, vom arăta mai întâi că partea dreaptă a expresiei (54) este ea însăşi o gaussiană, iar apoi vom ţine cont de faptul că valoarea maximă a unei gaussiene este atinsă exact în dreptul mediei.[101]

$$
\begin{aligned}
&p(\mu|x_1,\ldots,x_n) = \\
&= \frac{1}{C}\left(\prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}\right)\cdot\frac{1}{\sqrt{2\pi}\beta}e^{-\frac{(\mu-\nu)^2}{2\beta^2}} \\
&= const\cdot e^{-\frac{\sum_{i=1}^n(x_i-\mu)^2}{2\sigma^2}-\frac{(\mu-\nu)^2}{2\beta^2}} \\
&= const\cdot e^{-\frac{\beta^2\sum_{i=1}^n(x_i-\mu)^2+\sigma^2(\mu-\nu)^2}{2\sigma^2\beta^2}} \\
&= const\cdot \exp\left(-\frac{n\beta^2+\sigma^2}{2\sigma^2\beta^2}\mu^2 + \frac{\beta^2\sum_{i=1}^n x_i+\nu\sigma^2}{\sigma^2\beta^2}\mu - \frac{\beta^2\sum_{i=1}^n x_i^2+\nu^2\sigma^2}{2\sigma^2\beta^2}\right) \\
&= const\cdot \exp\left(-\frac{\mu^2 - 2\mu\frac{\beta^2\sum_{i=1}^n x_i+\nu\sigma^2}{n\beta^2+\sigma^2} + \frac{\beta^2\sum_{i=1}^n x_i^2+\nu^2\sigma^2}{n\beta^2+\sigma^2}}{\frac{2\sigma^2\beta^2}{n\beta^2+\sigma^2}}\right) \\
&= const\cdot \exp\left(-\frac{\left(\mu-\frac{\beta^2\sum_{i=1}^n x_i+\nu\sigma^2}{n\beta^2+\sigma^2}\right)^2 - \left(\frac{\beta^2\sum_{i=1}^n x_i+\nu\sigma^2}{n\beta^2+\sigma^2}\right)^2 + \frac{\beta^2\sum_{i=1}^n x_i^2+\nu^2\sigma^2}{n\beta^2+\sigma^2}}{2\frac{\sigma^2\beta^2}{n\beta^2+\sigma^2}}\right) \\
&= const\cdot \exp\left(-\frac{\left(\mu-\frac{\beta^2\sum_{i=1}^n x_i+\nu\sigma^2}{n\beta^2+\sigma^2}\right)^2}{2\frac{\sigma^2\beta^2}{n\beta^2+\sigma^2}}\right) \\
&\qquad\cdot \exp\left(\frac{\left(\frac{\beta^2\sum_{i=1}^n x_i+\nu\sigma^2}{n\beta^2+\sigma^2}\right)^2 - \frac{\beta^2\sum_{i=1}^n x_i^2+\nu^2\sigma^2}{n\beta^2+\sigma^2}}{2\frac{\sigma^2\beta^2}{n\beta^2+\sigma^2}}\right) \\
&= const'\cdot \exp\left(-\frac{\left(\mu-\frac{\beta^2\sum_{i=1}^n x_i+\nu\sigma^2}{n\beta^2+\sigma^2}\right)^2}{2\frac{\sigma^2\beta^2}{n\beta^2+\sigma^2}}\right)
\end{aligned}
$$

Se observă că factorul neconstant din expresia de mai sus reprezintă chiar o

[101] Pentru a asigura o bună lizibilitate, în calculul care urmează am înlocuit — la un moment dat — expresiile de tipul e^x cu $\exp(x)$.

distribuţie gaussiană de medie $\frac{\beta^2 \sum_{i=1}^n x_i + \nu\sigma^2}{n\beta^2 + \sigma^2}$ şi varianţă $\frac{\sigma^2\beta^2}{n\beta^2 + \sigma^2}$. Aşadar, valoarea ei maximă, ca şi cea pentru expresia (54), se realizează pentru $\mu = \frac{\beta^2 \sum_{i=1}^n x_i + \nu\sigma^2}{n\beta^2 + \sigma^2} = \mu_{MAP}$.

e. Startul la acest punct îl constituie rezultatele obţinute mai sus: $\mu_{MLE} = \frac{\sum_{i=1}^n x_i}{n}$ şi $\mu_{MAP} = \frac{\sigma^2\nu + \beta^2 \sum_{i=1}^n x_i}{\sigma^2 + n\beta^2}$. Punând sub o formă convenabilă ultima dintre cele două expresii, vom arăta că μ_{MAP} se poate scrie în funcţie de μ_{MLE}:

$$\begin{aligned}\mu_{MAP} &= \frac{\sigma^2\nu + \beta^2 \sum_{i=1}^n x_i}{\sigma^2 + n\beta^2} = \frac{\sigma^2\nu}{\sigma^2 + n\beta^2} + \frac{\beta^2 \sum_{i=1}^n x_i}{\sigma^2 + n\beta^2} \\ &= \frac{\sigma^2\nu}{\sigma^2 + n\beta^2} + \frac{\frac{1}{n}\sum_{i=1}^n x_i}{1 + \frac{\sigma^2}{n\beta^2}} = \frac{\sigma^2\nu}{\sigma^2 + n\beta^2} + \frac{\mu_{MLE}}{1 + \frac{\sigma^2}{n\beta^2}}\end{aligned}$$

Evident, pentru $n \to \infty$ va rezulta că $\frac{\sigma^2\nu}{\sigma^2 + n\beta^2} \to 0$ şi $\frac{\sigma^2}{n\beta^2} \to 0$, deci $\mu_{MAP} \to \mu_{MLE}$. Aşadar, pentru valori suficient de mari ale lui n, cele două estimări ale lui μ vor avea valori foarte apropiate.

51. **(Distribuţia gaussiană unidimensională: estimarea varianţei în sensul MLE [în cazul când nu se impun restricţii asupra mediei μ])**

□ • ○ ***prelucrare de Liviu Ciortuz, după***
■ ***CMU, 2010 fall, Ziv Bar-Joseph, HW1, pr. 2.1.1-2***
CMU, 2007 fall, Carlos Guestrin, HW1, pr. 3.2.1

Fie $x_1, \ldots, x_n \in \mathbb{R}$ instanţe generate în mod independent şi identic distribuite conform gaussienei $N(x|\mu, \sigma^2)$.

***Comentarii*:**
1. Vă reamintim că funcţia densitate de probabilitate (p.d.f.) a distribuţiei gaussiene unidimensionale este definită de expresia:

$$p(x) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

2. La problema 50.a am calculat estimatorul MLE al mediei μ pentru această distribuţie. (Am notat acest estimator cu μ_{MLE}.)

a. Calculaţi estimatorul MLE al varianţei σ^2 (notat cu σ^2_{MLE}), presupunând că nu se impune nicio restricţie asupra lui μ.[102]

b. Arătaţi că $E[\sigma^2_{MLE}] = \frac{n-1}{n}\sigma^2$.

c. Rezultatul de la punctul precedent ne arată că σ^2_{MLE} este un estimator *cu deplasare* (sau, *deplasat*; engl., biased) în raport cu σ^2.[103] **Totuşi, acelaşi**

[102] Este util de ştiut de asemenea că la problema 135 se cere ca (tot pentru distribuţia gaussiană unidimensională) să se estimeze varianţa σ^2 în sens MLE, presupunând însă că media μ este 0.

[103] Vă readucem aminte că estimatorul σ_{MLE} ar fi fost fără deplasare (sau, nedeplasat; engl., unbiased) dacă ar fi fost adevărată egalitatea $E[\sigma^2_{MLE}] = \sigma^2$.

rezultat ne permite să găsim uşor un estimator *nedeplasat* (engl., unbiased) pentru parametrul σ^2. Care este acest estimator?

Răspuns:

a. Scriem mai întâi funcţia de log-verosimilitate a datelor $x \stackrel{not.}{=} (x_1, \ldots, x_n)$:

$$\begin{aligned}\ell(\mu, \sigma^2) &\stackrel{def.}{=} \ln P(x \mid \mu, \sigma^2) \stackrel{i.i.d.}{=} \ln \prod_{i=1}^{n} p(x_i) = \sum_{i=1}^{n} \left(-\frac{1}{2} \ln(2\pi\sigma^2) - \frac{(x_i - \mu)^2}{2\sigma^2} \right) \\ &= -\frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (x_i - \mu)^2.\end{aligned}$$

Derivata parţială a funcţiei ℓ în raport cu σ^2 este:[104]

$$\frac{\partial \ell(\mu, \sigma^2)}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \sum_{i=1}^{n} (x_i - \mu)^2.$$

Ţinând cont de semnele acestei derivate parţiale şi egalând-o cu 0, vom obţine:

$$\sigma^2_{MLE} = \frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_{MLE})^2.$$

***Observaţie*: Am ţinut cont de faptul că maximul funcţiei $\ell(\mu, \sigma^2)$ se atinge pentru $\mu = \mu_{MLE}$ şi $\sigma^2 = \sigma^2_{MLE}$.**[105] **De aceea, la rezolvarea ecuaţiei de mai sus ($\frac{\partial \ell(\mu, \sigma^2)}{\partial \sigma^2} = 0$), am înlocuit μ cu μ_{MLE}.**

b. Vom folosi acum rezultatul de la punctul precedent pentru a calcula media estimatorului σ^2_{MLE}. Într-adevăr, are sens să vorbim despre media lui σ^2_{MLE} atunci când acesta este văzut ca o variabilă aleatoare (ceea ce este o consecinţă a faptului că instanţele x_i sunt generate în mod aleator). Făcând uz de proprietatea de liniaritate a mediei (a cărei folosire este desemnată mai jos prin simbolul $(*)$),[106] **vom putea scrie:**

$$\begin{aligned}E[\sigma^2_{MLE}] &= E\left[\frac{1}{n} \sum_{i=1}^{n} (x_i - \mu_{MLE})^2 \right] \stackrel{(*)}{=} E[(x_1 - \mu_{MLE})^2] \\ &= E\left[(x_1 - \frac{1}{n} \sum_{i=1}^{n} x_i)^2 \right] = E\left[x_1^2 - \frac{2}{n} x_1 \sum_{i=1}^{n} x_i + \frac{1}{n^2} (\sum_{i=1}^{n} x_i)^2 \right] \\ &= E\left[x_1^2 - \frac{2}{n} x_1 \sum_{i=1}^{n} x_i + \frac{1}{n^2} \sum_{i=1}^{n} x_i^2 + \frac{2}{n^2} \sum_{i<j} x_i x_j \right]\end{aligned}$$

[104] Puteţi constata singuri că este mult mai convenabil să facem derivarea în raport cu σ^2, decât în raport cu σ.

[105] Într-adevăr, la problema 50.a s-a demonstrat de fapt că

$$l(\mu, \sigma^2) \leq l(\mu_{MLE}, \sigma^2) \text{ pentru orice } \sigma > 0 \text{ fixat,}$$

iar la exerciţiul de faţă demonstraţia de mai sus poate fi rescrisă / adaptată imediat pentru a demonstra inegalitatea

$$l(\mu_{MLE}, \sigma^2) \leq l(\mu_{MLE}, \sigma^2_{MLE}).$$

[106] Vedeţi problema 9.a.

$$\begin{aligned} &\overset{(*)}{=} E[x_1^2] + \frac{1}{n^2}\sum_{i=1}^{n} E[x_i^2] - \frac{2}{n}\sum_{i=1}^{n} E[x_1 x_i] + \frac{2}{n^2}\sum_{i<j} E[x_i x_j] \\ &= E[x_1^2] + \frac{1}{n^2} n E[x_1^2] - \frac{2}{n} E[x_1^2] - \frac{2}{n}(n-1)E[x_1 x_2] + \frac{2}{n^2}\frac{n(n-1)}{2} E[x_1 x_2] \\ &= \frac{n-1}{n} E[x_1^2] - \frac{n-1}{n} E[x_1 x_2]. \end{aligned} \tag{55}$$

Ca să finalizăm calculul lui $E[\sigma^2_{MLE}]$, va trebui să calculăm separat mediile $E[x_1^2]$ şi $E[x_1 x_2]$. Pentru prima dintre ele, vom folosi o cunoscută proprietate a varianţei, pe care am demonstrat-o la problema 9.b:

$$\sigma^2 = Var(x_1) = E[x_1^2] - (E[x_1])^2 = E[x_1^2] - \mu^2 \;\Rightarrow\; E[x_1^2] = \sigma^2 + \mu^2.$$

Pentru calculul lui $E[x_1 x_2]$, ţinând cont de faptul că x_1 şi x_2 sunt independente, rezultă $Cov(x_1, x_2) = 0$, conform problemei 10. Deci vom putea scrie:

$$\begin{aligned} 0 &= Cov(x_1, x_2) = E[(x_1 - E[x_1])(x_2 - E[x_2])] = E[(x_1 - \mu)(x_2 - \mu)] \\ &\overset{(*)}{=} E[x_1 x_2] - \mu E[x_1 + x_2] + \mu^2 \overset{(*)}{=} E[x_1 x_2] - \mu(E[x_1] + E[x_2]) + \mu^2 \\ &= E[x_1 x_2] - \mu(2\mu) + \mu^2 = E[x_1 x_2] - \mu^2. \end{aligned}$$

Prin urmare, $E[x_1 x_2] = \mu^2$.

Substituind cele două rezultate intermediare — şi anume, $E[x_1^2] = \sigma^2 + \mu^2$ şi $E[x_1 x_2] = \mu^2$ — în relaţia (55), vom obţine:

$$E[\sigma^2_{MLE}] = \frac{n-1}{n}(\sigma^2 + \mu^2) - \frac{n-1}{n}\mu^2 = \frac{n-1}{n}\sigma^2.$$

c. Din relaţia obţinută la punctul b, ţinând din nou cont de liniaritatea mediei, rezultă că

$$E\left[\frac{n}{n-1}\sigma^2_{MLE}\right] = \sigma^2.$$

Prin urmare, $\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_{MLE})^2$ este un estimator nedeplasat pentru parametrul σ^2.

52. **(Distribuţia Gamma: estimarea parametrilor în sens MLE)**

■ □ ● ○ ***Liviu Ciortuz, 2017***

Fie distribuţia Gamma de parametri $r > 0$ şi $\beta > 0$, a cărei funcţie [de] densitate de probabilitate (p.d.f.) este dată de formula următoare:[107]

$$Gamma\left(x|r, \frac{1}{\beta}\right) = \frac{1}{\beta^r\,\Gamma(r)} x^{r-1} e^{-\frac{x}{\beta}}, \text{ pentru orice } x > 0.$$

În această formulă, am notat cu Γ funcţia lui Euler, care constituie o generalizare pentru [funcţia care defineşte] factorialul unui număr natural ($\Gamma(r) = (r-1)!$ pentru orice $r \in \mathbb{N}^\star$).[108]

[107] Atenţie! Spre deosebire de problemele 31.c şi 46, unde am lucrat cu parametrii r şi α, aici lucrăm cu r şi $\beta = 1/\alpha$.

[108] Vedeţi problema 31.b.

Presupunând că instanţele $x_1, \ldots, x_n \in \mathbb{R}^+$ au fost generate de către o astfel de distribuţie Gamma, cu parametrii r şi β fixaţi, calculaţi estimările de verosimilitate maximă (MLE) pentru cei doi parametri, r şi β.

***Sugestie*: Scrieţi mai întâi $\ell(r, \beta)$, funcţia de log-verosimiliate a datelor x_1, x_2, ..., x_n. Apoi, calculaţi $\hat{\beta}$, rădăcina derivatei parţiale a lui ℓ în raport cu parametrul β. În expresia pe care aţi obţinut-o anterior pentru funcţia de log-verosimilitate ℓ, înlocuiţi β cu expresia lui $\hat{\beta}$. Calculaţi derivata parţială a lui $\ell(r, \hat{\beta})$ în raport cu parametrul r. Egalând cu 0 expresia acestei derivate parţiale veţi obţine o ecuaţie de forma**

$$\ln r - \frac{\Gamma'(r)}{\Gamma(r)} = \ldots \ . \tag{56}$$

Raportul $\dfrac{\Gamma'(r)}{\Gamma(r)}$, notat în general cu $\psi(r)$, se numeşte funcţia digamma.

***Observaţie importantă*: Nu se cunosc soluţii analitice pentru rezolvarea ecuaţiilor de tipul (56). Însă găsirea maximului funcţiei $\ell(r, \hat{\beta})$ în raport cu parametrul r se poate face folosind metode de optimizare precum metoda gradientului ascendent sau metoda lui Newton. (Vedeţi problema 137.)**

Răspuns:

Vom scrie mai întâi expresia funcţiei de verosimilitate:

$$\begin{aligned} L(r,\beta) &\overset{def.}{=} P(x_1,\ldots,x_n|r,\beta) \overset{i.i.d.}{=} \prod_{i=1}^{n} P(x_i|r,\beta) \\ &= \beta^{-rn}(\Gamma(r))^{-n}\left(\prod_{i=1}^{n} x_i\right)^{r-1} e^{-\frac{1}{\beta}\sum_{i=1}^{n} x_i} \end{aligned}$$

Logaritmând, vom obţine funcţia de log-verosimilitate:

$$\ell(r,\beta) \overset{def.}{=} \ln L(r,\beta) = -rn\ln\beta - n\ln\Gamma(r) + (r-1)\sum_{i=1}^{n}\ln x_i - \frac{1}{\beta}\sum_{i=1}^{n} x_i.$$

Vom calcula acum derivata parţială a funcţiei de log-verosimilitate $\ell(r, \beta)$ în raport cu β, iar apoi vom egala această derivată cu 0 pentru a obţine rădăcina ei:

$$\frac{\partial}{\partial\beta}\ell(r,\beta) = -\frac{rn}{\beta} + \frac{1}{\beta^2}\sum_{i=1}^{n} x_i = \frac{1}{\beta^2}\left[\sum_{i=1}^{n} x_i - rn\beta\right]$$

$$\frac{\partial}{\partial\beta}\ell(r,\beta) = 0 \Leftrightarrow \hat{\beta} = \frac{1}{rn}\sum_{i=1}^{n} x_i > 0.$$

Dacă în expresia derivatei parţiale $\dfrac{\partial}{\partial\beta}\ell(r, \beta)$ fixăm valoarea lui r, din analiza semnelor derivatei se constată imediat că $\hat{\beta}$ este punct de maxim şi, mai mult, acest punct de maxim este unic (repetăm, pentru fiecare valoare a lui r, fixată).

Pentru a calcula acum maximul funcţiei de log-verosimilitate $\ell(r, \beta)$ în raport cu r, mai întâi vom înlocui β cu $\hat{\beta}$ în expresia lui $\ell(r, \beta)$ şi vom obţine:

$$\begin{aligned}\ell(r,\hat\beta) &= -rn\ln\hat\beta - n\ln\Gamma(r) + (r-1)\sum_{i=1}^n \ln x_i - \frac{1}{\hat\beta}\sum_{i=1}^n x_i \\ &= rn\ln(rn) - rn\ln\sum_{i=1}^n x_i - n\ln\Gamma(r) + (r-1)\sum_{i=1}^n \ln x_i - \frac{rn}{\sum_{i=1}^n x_i}\cdot\sum_{i=1}^n x_i \\ &= rn\ln(rn) - rn\Big(\ln\sum_{i=1}^n x_i + 1\Big) - n\ln\Gamma(r) + (r-1)\sum_{i=1}^n \ln x_i.\end{aligned}$$

Apoi, încercând să calculăm rădăcina derivatei parţiale a funcţiei $\ell(r,\hat\beta)$ (pe care tocmai am calculat-o mai sus) în raport cu r, scriem următoarele echivalenţe:

$$\begin{gathered}\frac{\partial}{\partial r}\ell(r,\hat\beta) = 0 \Leftrightarrow n\ln(nr) + n - n\Big(\ln\sum_{i=1}^n x_i + 1\Big) - n\cdot\frac{\Gamma'(r)}{\Gamma(r)} + \sum_{i=1}^n \ln x_i = 0 \Leftrightarrow \\ n(\ln r - \psi(r)) = -n\ln n - \sum_{i=1}^n \ln x_i + n\ln\sum_{i=1}^n x_i \Leftrightarrow \\ \ln r - \psi(r) = -\ln n - \frac{1}{n}\sum_{i=1}^n \ln x_i + \ln\sum_{i=1}^n x_i.\end{gathered}$$

Soluţia ultimei ecuaţii este $\hat r$, estimarea de verosimilitate maximă a parametrului r al distribuţiei $Gamma\left(x|r,\frac{1}{\beta}\right)$ şi, conform precizării din enunţ, ea poate fi obţinută nu în mod direct (ca în cazul altor distribuţii probabiliste), ci prin anumite metode / programe de analiză numerică. (Vedeţi problema 137.)

53. (Distribuţia gaussiană multidimensională: estimarea în sens MLE[109] a vectorului de medii μ şi a matricei de precizie $\Lambda = \Sigma^{-1}$)

prelucrare de Liviu Ciortuz, după
■ □ • ○ ***CMU, 2010 fall, Aarti Singh, HW1, pr. 3.2.1***

Funcţia de densitate de probabilitate (p.d.f.) pentru o distribuţie gaussiană d-dimensională este definită prin expresia următoare:[110]

$$\mathcal{N}(x \mid \mu, \Lambda^{-1}) \overset{def.}{=} \frac{\exp\left(-\frac{1}{2}(x-\mu)^\top\Lambda(x-\mu)\right)}{(2\pi)^{d/2}\sqrt{|\Lambda^{-1}|}}, \tag{57}$$

unde $\mu \in \mathbb{R}^d$ este media distribuţiei, iar $\Lambda \in \mathbb{R}^{d\times d}$ este aşa-numita *matrice de precizie* (inversa matricei de covarianţă, pe care am notat-o cu Σ).[111]

[109] Pentru estimarea în sens MAP a parametrilor acestei distribuţii, vedeţi problema 136.

[110] Comparativ cu expresia (29) de la problema 37, aici s-a notat Σ^{-1} cu Λ, iar $\frac{1}{|\Sigma|^{1/2}}$ a devenit $\frac{1}{\sqrt{|\Lambda^{-1}|}}$, care ulterior va fi folosit sub forma $\sqrt{|\Lambda|}$, fiindcă $|\Lambda^{-1}| = |\Lambda|^{-1}$ (vedeţi regula (2b) de mai jos).

Justificarea pentru această schimbare de notaţie este că, din perspectiva calculelor ulterioare (găsirea optimului funcţiei de log-verosimilitate) este convenabil ca în p.d.f.-ul funcţiei $\mathcal{N}$ să nu avem atât Σ cât şi Σ^{-1}. Un pic surprinzător, calculele vor arăta (în mod indirect) că este mai convenabil ca din cele două să păstrăm $\Sigma^{-1} \overset{not.}{=} \Lambda$. Acesta este un „artificiu“ standard în astfel de contexte. (Vedeţi de exemplu problema 24 de la capitolul *Clusterizare*.)

[111] Operatorul $\top$ desemnează transpunerea vectorilor / matricelor. Considerăm că vectorii x şi μ din $\mathbb{R}^d$ sunt vectori-coloană.

Fie $\{x_1, x_2, \ldots, x_n\} \subseteq \mathbb{R}^d$ un eşantion, adică un set de instanţe generate în mod independent şi distribuite identic, conform unei gaussiene d-dimensionale.

a. Presupunem că $n \gg d$ (adică n este mult mai mare decât d).

Calculaţi estimările în sensul verosimilităţii maxime (MLE) pentru parametrii μ şi Λ. (Veţi nota aceste estimări cu $\hat{\mu}$ şi respectiv $\hat{\Lambda}$.)

b. Particularizaţi rezultatele pentru cazul $d = 2$.

***Sugestie*: Întrucât se lucrează în $\mathbb{R}^d$, vă recomandăm să folosiţi derivate [parţiale] vectoriale. Pentru acest tip de derivate, variabila în raport cu care se derivează este din $\mathbb{R}^d$, nu din $\mathbb{R}$ cum suntem obişnuiţi. Unele din următoarele formule (preluate din documentul *Matrix Identities*, de Sam Roweis, 1999) vă pot fi de folos:**

(2b) $|A^{-1}| = \dfrac{1}{|A|}$

(2e) $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$;[112]
mai general, $\mathrm{Tr}(ABC\ldots) = \mathrm{Tr}(BC\ldots A) = \mathrm{Tr}(C\ldots AB) = \ldots$

(3b) $\dfrac{\partial}{\partial X}\mathrm{Tr}(XA) = \dfrac{\partial}{\partial X}\mathrm{Tr}(AX) = A^\top$

(4b) $\dfrac{\partial}{\partial X}\ln|X| = (X^{-1})^\top = (X^\top)^{-1}$

(5c) $\dfrac{\partial}{\partial X}a^\top X b = ab^\top$

(5g) $\dfrac{\partial}{\partial X}(Xa+b)^\top C(Xa+b) = (C+C^\top)(Xa+b)a^\top$

În formulele (2e) şi (3b), notaţia $\mathrm{Tr}(A)$ desemnează *urma* (engl., trace) unei matrice pătratice A, de dimensiune $n \times n$; ea se defineşte ca fiind suma elementelor de pe diagonala principală a matricei,[113] adică $a_{11} + \ldots + a_{nn}$.

Răspuns:

a. Vom scrie mai întâi log-verosimilitatea datelor $x_1, \ldots, x_n$ ca funcţie de cei doi parametri, μ şi Λ:

$$\begin{aligned} l(\mu,\Lambda) &\overset{i.i.d.}{=} \ln\prod_{i=1}^n \mathcal{N}(x_i|\mu,\Lambda^{-1}) = \sum_{i=1}^n \ln\mathcal{N}(x_i|\mu,\Lambda^{-1}) \\ &= -\frac{nd}{2}\ln(2\pi) - \frac{n}{2}\ln|\Lambda^{-1}| - \frac{1}{2}\sum_{i=1}^n (x_i-\mu)^\top\Lambda(x_i-\mu) \\ &\overset{(2b)}{=} -\frac{nd}{2}\ln(2\pi) + \frac{n}{2}\ln|\Lambda| - \frac{1}{2}\sum_{i=1}^n (x_i-\mu)^\top\Lambda(x_i-\mu). \end{aligned} \tag{58}$$

Fixând matricea de precizie Λ la o valoare oarecare (matrice pozitiv definită), se poate demonstra relativ uşor (vedeţi mai jos) că matricea hessiană a acestei funcţii de log-verosimilitate în raport cu variabila μ este negativ definită, deci

[112] Vedeţi teorema 1.3.d din *Matrix Analysis for Statistics*, 2017, James R. Schott.

[113] Diagonala principală a unei matrice este diagonala care porneşte din colţul stânga-sus al matricei şi merge până în colţul din dreapta-jos.

avem de-a face cu o funcţie strict concavă în raport cu μ. Pentru a identifica maximul acestei funcţii, vom urmări să rezolvăm ecuaţia

$$\nabla_\mu l(\mu,\Lambda)=0 \overset{(5g)}{\iff} -\frac{1}{2}(\Lambda+\Lambda^\top)\sum_{i=1}^n(x_i-\mu)(-1)=0 \iff$$
$$\Lambda\sum_{i=1}^n(x_i-\mu)=0 \iff n\Lambda\mu=\Lambda\sum_{i=1}^n x_i, \tag{59}$$

unde notaţia $\nabla_\mu l(\mu,\Lambda)$ desemnează vectorul gradient (adică vectorul de derivate parţiale) al funcţiei $l(\mu,\Lambda)$ în raport cu variabila vectorială μ.

Menţionăm faptul că atunci când, la elaborarea şirului de echivalenţe (59), am aplicat formula (5g), am luat $C=\Lambda$, $X=\mu$, $a=-1$ şi $b=x_i$. În plus, am folosit faptul că Λ este matrice simetrică (adică $\Lambda=\Lambda^\top$), întrucât Λ este inversa matricei Σ, care este matricea de covarianţă a distribuţiei gaussiene multidimensionale considerate şi, conform problemei 20, Σ este matrice simetrică.

Ţinând cont că Λ, ca matrice pozitiv definită, este inversabilă, din relaţia (59) vom obţine următorea valoare (de fapt, exact estimarea în sens MLE) pentru μ:

$$\hat{\mu}=\frac{\sum_{i=1}^n x_i}{n}, \tag{60}$$

iar această expresie coincide cu *media* [aritmetică a] *eşantionului* considerat (engl., sample mean), medie notată îndeobşte cu $\bar{x}$, şi care este constantă în raport cu matricea de precizie Λ.

Ceea ce am arătat până acum este

$$l(\mu,\Lambda)\le l(\hat{\mu},\Lambda) \text{ pentru orice } \mu\in\mathbb{R}^d,$$

unde Λ este o matrice simetrică şi pozitiv definită arbitrară (dar fixată). Observaţi că în inegalitatea aceasta, ambii membri (atât cel stâng cât şi cel drept) au ca al doilea argument aceeaşi matrice Λ.

Urmărind să ne ocupăm acum de Λ — mai precis, să demonstrăm că are loc încă o inegalitate, $l(\hat{\mu},\Lambda)\le l(\hat{\mu},\hat{\Lambda})$, pentru orice matrice pozitiv definită Λ —, vom înlocui mai întâi μ cu $\hat{\mu}$ (adică, cu $\bar{x}$) în expresia pe care am obţinut-o mai sus pentru funcţia de log-verosimilitate, (58):

$$l(\hat{\mu},\Lambda) = -\frac{nd}{2}\ln(2\pi)+\frac{n}{2}\ln|\Lambda|-\frac{1}{2}\sum_{i=1}^n(x_i-\bar{x})^\top\Lambda(x_i-\bar{x}) \tag{61}$$
$$\overset{(2e)}{=} -\frac{nd}{2}\ln(2\pi)+\frac{n}{2}(\ln|\Lambda|-\mathrm{Tr}(S\Lambda)), \tag{62}$$

unde S este *matricea de covarianţă la eşantionare* (engl., sample covariance matrix): $S\overset{not.}{=}\frac{1}{n}\sum_{i=1}^n(x_i-\bar{x})(x_i-\bar{x})^\top$. Oferim mai jos câteva detalii de caclul pentru a arăta modul în care am procedat pentru a obţine expresia lui $l(\hat{\mu},\Lambda)$.

***Explicaţie*: $(x_i-\bar{x})^\top\Lambda(x_i-\bar{x})$ este o matrice de dimensiune 1×1, deci $(x_i-\bar{x})^\top\Lambda(x_i-\bar{x})=\mathrm{Tr}((x_i-\bar{x})^\top\Lambda(x_i-\bar{x}))$. Folosind regula (2e), această expresie poate fi scrisă mai departe ca $\mathrm{Tr}((x_i-\bar{x})(x_i-\bar{x})^\top\Lambda)$.**

Ţinând cont de o altă regulă simplă, $\mathrm{Tr}(A+B)=\mathrm{Tr}(A)+\mathrm{Tr}(B)$ **(care poate fi demonstrată uşor), rezultă că**

$$\begin{aligned}\sum_{i=1}^{n}(x_i-\bar{x})^\top\Lambda(x_i-\bar{x}) &= \sum_{i=1}^{n}\mathrm{Tr}((x_i-\bar{x})^\top\Lambda(x_i-\bar{x})) = \sum_{i=1}^{n}\mathrm{Tr}((x_i-\bar{x})(x_i-\bar{x})^\top\Lambda)\\ &= \mathrm{Tr}\Big(\sum_{i=1}^{n}((x_i-\bar{x})(x_i-\bar{x})^\top\Lambda)\Big) = \mathrm{Tr}\Big(\Big(\sum_{i=1}^{n}(x_i-\bar{x})(x_i-\bar{x})^\top\Big)\Lambda\Big) = \mathrm{Tr}((nS)\Lambda)\\ &= n\mathrm{Tr}(S\Lambda).\end{aligned}$$

Conform relaţiei (3b), avem $\dfrac{\partial}{\partial\Lambda}\mathrm{Tr}(S\Lambda)=S^\top$**. Folosind în continuare relaţia (4b), rezultă că** $\nabla_\Lambda\ell(\hat{\mu},\Lambda)=\dfrac{n}{2}(\Lambda^\top)^{-1}-\dfrac{n}{2}S^\top$**. Prin urmare,**[114] **matricea hessiană a funcţiei** $\ell(\hat{\mu},\Lambda)$ **este** $-\dfrac{n}{2}(\Lambda^{-1})^2=-\dfrac{n}{2}\Sigma^2$**, despre care se poate arăta imediat că este negativ definită.**

Aşadar, vom putea să căutăm maximul expresiei (62) **rezolvând ecuaţia**

$$\nabla_\Lambda l(\hat{\mu},\Lambda)=0,$$

despre care se poate arăta[115] **că este echivalentă cu**

$$\Lambda^{-1}-S=0. \qquad (\textbf{Deci, } \Sigma=\Lambda^{-1}=S.)$$

În general, pentru $n\gg d$ **matricea** S **este inversabilă, aşa că obţinem următoarea estimare în sens MLE pentru** Λ**:**

$$\hat{\Lambda}=S^{-1}.$$

***Observaţii*:**

1. În calculele de mai sus, am urmărit să ne asigurăm că estimările $\hat{\mu}$ **şi** $\hat{\Lambda}$ **satisfac dubla inegalitate**

$$l(\mu,\Lambda)\leq l(\hat{\mu},\Lambda)\leq l(\hat{\mu},\hat{\Lambda})$$

pentru orice $\mu\in\mathbb{R}^d$ **şi orice** Λ **matrice simetrică şi pozitiv definită. Mai mult, se poate demonstra uşor că** S **(deci şi inversa sa) este matrice simetrică şi pozitiv definită. Aşadar, rezultă că** $\hat{\mu}$ **şi** $\hat{\Lambda}$ **sunt estimări în sensul MLE.**

2. În loc să fi folosit relaţia (62)**, adică să lucrăm cu operatorul** Tr**, am fi putut să calculăm derivata parţială a funcţiei** $l(\hat{\mu},\Lambda)$ **în raport cu matricea** Λ **direct din relaţia** (61)**:**

$$\begin{aligned}\nabla_\Lambda l(\hat{\mu},\Lambda) &\overset{(4b),(5c)}{=} \frac{n}{2}(\Lambda^\top)^{-1}-\frac{1}{2}\sum_{i=1}^{n}(x_i-\hat{\mu})(x_i-\hat{\mu})^\top\\ &= \frac{n}{2}\Lambda^{-1}-\frac{1}{2}\sum_{i=1}^{n}(x_i-\hat{\mu})(x_i-\hat{\mu})^\top=\frac{n}{2}\Lambda^{-1}-\frac{n}{2}S. \qquad (63)\end{aligned}$$

[114] Vedeţi relaţia (61) din documentul *The Matrix Cookbook*, K.B. Petersen, M.S. Pedersen, 2012: $\dfrac{\partial}{\partial X}a^\top X^{-1}b=-(X^{-1})^\top ab^\top(X^{-1})^\top$.

[115] $\dfrac{n}{2}(\Lambda^\top)^{-1}-\dfrac{n}{2}S^\top=0\Leftrightarrow(\Lambda^\top)^{-1}-S^\top=0\Leftrightarrow(\Lambda^{-1})^\top=S^\top\Leftrightarrow\Lambda^{-1}=S.$

Aşadar,

$$\nabla_\Lambda l(\hat{\mu}, \Lambda) = 0 \Leftrightarrow \hat{\Lambda}^{-1} \overset{not.}{=} \hat{\Sigma} = S \overset{not.}{=} \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})(x_i - \hat{\mu})^\top. \tag{64}$$

***Observaţie importantă*:**[116]
În enunţ a fost inclusă condiţia $n \gg d$. În cazul în care n (numărul de instanţe) nu este semnificativ mai mare decât d (numărul de dimensiuni / atribute), estimările în sens MLE pentru parametrii μ şi Σ pot fi nesatisfăcătoare (engl., poor). Dacă $d \gg n$ şi încercăm să estimăm media şi matricea de covarianţă folosind relaţiile (60) şi respectiv (64), se poate să obţinem $|\Sigma| = 0$,[117] întrucât cele n instanţe generează (engl., *span*)[118] doar un subspaţiu de dimensiune redusă al lui $\mathbb{R}^d$. În acest caz, Σ^{-1} nu există, deci nu vom putea scrie funcţia de densitate a distribuţiei gaussiene multidimensionale (57). Totuşi, în unele cazuri particulare este posibil ca matricea Σ să fie nesingulară chiar pentru $n \geq 2$, de exemplu atunci când Σ este matrice diagonală sau, şi mai restrictiv, când $\Sigma = \sigma^2 I$. Dezavantajul în cazul în care se lucrează cu Σ matrice diagonală este că se presupune în mod implicit că atributele sunt independente (cf. ex. 34).[119]

b. Pentru cazul distribuţiei gaussiene bidimensionale (adică, pentru $d = 2$), pur şi simplu ca să ne re-familiarizăm cu matricea de covarianţă $\Sigma \overset{not.}{=} \Lambda^{-1}$, mai întâi ne aducem aminte că aceasta poate fi scrisă astfel:[120]

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix},$$

cu $\sigma_1 > 0$, $\sigma_2 > 0$ şi $\rho \in (-1, 1)$.

Urmează apoi că matricea de precizie Λ are forma următoare:

$$\Lambda \overset{not.}{=} \Sigma^{-1} = \frac{1}{\sigma_1^2\sigma_2^2(1-\rho^2)} \begin{pmatrix} \sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2 \end{pmatrix} = \frac{1}{(1-\rho^2)} \begin{pmatrix} \frac{1}{\sigma_1^2} & -\frac{\rho}{\sigma_1\sigma_2} \\ -\frac{\rho}{\sigma_1\sigma_2} & \frac{1}{\sigma_2^2} \end{pmatrix}.$$

Rezultatele pe care le-am obţinut la punctul a se particularizează în felul următor pentru cazul distribuţiei gaussiene bidimensionale ($d = 2$):

$$\hat{\mu} \quad = \quad \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{1}{n}\begin{pmatrix} \sum_{i=1}^{n} x_{i,1} \\ \sum_{i=1}^{n} x_{i,2} \end{pmatrix} = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \end{pmatrix}, \text{ unde}$$

[116] Cf. Andrew Ng, *Factor Analysis* (ML Lecture Notes, Part X), pag. 1-3.
[117] Într-un astfel de caz spunem că Σ este *matrice singulară*.
[118] Cf. documentului *Linear Algebra Review and Reference* de Zico Kolter (şi Chuong Do), 2012, pag. 12, *spaţiul liniar generat* de vectorii $\{x_1, x_2, \ldots, x_n\}$ se defineşte ca fiind mulţimea formată din toţi acei vectori care pot fi scrişi ca o combinaţie liniară de $x_1, x_2, \ldots, x_n$:

$$span(x_1, x_2, \ldots, x_n) = \Big\{v | v = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_n x_n, \text{ cu } \alpha_i \in \mathbb{R}\Big\}.$$

[119] Din această cauză, pentru a putea identifica eventualele dependenţe dintre atribute, atunci când $d \gg n$ se recomandă să se aplice metoda *analizei prin factori* (engl., Factor Analysis), care face estimarea parametrilor folosind *algoritmul EM*. Vedeţi documentul *Factor Analysis* de Andrew Ng (ML Lecture Notes, Part X), pag. 4-5.
[120] Vedeţi problema 38.

$$x_i \overset{not.}{=} \begin{pmatrix} x_{i,1} \\ \\ x_{i,2} \end{pmatrix} \text{ pentru } i = 1, \ldots, n,$$

$$\bar{x}_1 \overset{not.}{=} \frac{1}{n}\sum_{i=1}^{n} x_{i,1} \text{ şi } \bar{x}_2 \overset{not.}{=} \frac{1}{n}\sum_{i=1}^{n} x_{i,2};$$

$$\hat{\Sigma} \overset{not.}{=} \hat{\Lambda}^{-1} = \frac{1}{n}\sum_{i=1}^{n}\left(\begin{pmatrix} x_{i,1}-\bar{x}_1 \\ x_{i,2}-\bar{x}_2 \end{pmatrix}(x_{i,1}-\bar{x}_1, x_{i,2}-\bar{x}_2)\right)$$

$$= \frac{1}{n}\sum_{i=1}^{n}\begin{pmatrix} (x_{i,1}-\bar{x}_1)^2 & (x_{i,1}-\bar{x}_1)(x_{i,2}-\bar{x}_2) \\ (x_{i,1}-\bar{x}_1)(x_{i,2}-\bar{x}_2) & (x_{i,2}-\bar{x}_2)^2 \end{pmatrix}$$

$$= \frac{1}{n}\begin{pmatrix} \sum_{i=1}^{n}(x_{i,1}-\bar{x}_1)^2 & \sum_{i=1}^{n}(x_{i,1}-\bar{x}_1)(x_{i,2}-\bar{x}_2) \\ \sum_{i=1}^{n}(x_{i,1}-\bar{x}_1)(x_{i,2}-\bar{x}_2) & \sum_{i=1}^{n}(x_{i,2}-\bar{x}_2)^2 \end{pmatrix}.$$

Aceasta este exact *matricea de covarianţă la eşantionare*,

$$\begin{pmatrix} Var(X_1) & Cov(X_1, X_2) \\ Cov(X_1, X_2) & Var(X_2) \end{pmatrix},$$

unde am presupus că variabilele X_1 şi X_2 reprezintă cele două componente pentru X, variabila gaussiană bidimensională care a generat instanţele $x_1, \ldots, x_n$.

54. **(Estimatori MLE: existenţa şi unicitatea)**

□ • ○ ***CMU, 2010 fall, Aarti Singh, HW1, pr. 3.3***

În această problemă vom arăta că estimările de verosimilitate maximă (MLE) nu există în mod neapărat. Şi, chiar atunci când există, se poate ca ele să nu fie unice.

a. Puneţi în evidenţă un caz în care nu există estimarea / estimările de verosimilitate maximă (MLE). Vă cerem să specificaţi familia de distribuţii pe care aţi ales-o, precum şi tipul de eşantioane (engl., samples) pentru care MLE nu este [bine] definită.

b. Daţi un exemplu de caz în care MLE există, însă nu în mod unic. Specificaţi familia de distribuţii pe care aţi ales-o, precum şi tipul de eşantioane pentru care pot exista estimări multiple de verosimilitate maximă.

c. Identificând cele două exemple descrise mai sus, sperăm că v-aţi format o anumită intuiţie despre proprietăţile funcţiei de [log-]verosimilitate care sunt cruciale pentru existenţa şi unicitatea MLE. Formulaţi aceste proprietăţi.

Răspuns:

a. La rezolvarea problemei 46.a am demonstrat pentru distribuţia Poisson de parametru $\lambda > 0$ că estimarea lui în sensul MLE este $\lambda_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$, pentru orice set de instanţe $x_1, \ldots, x_n \geq 0$. Evident, această estimare MLE există doar dacă $\sum_{i=1}^{n} x_i > 0$. Atunci când $\sum_{i=1}^{n} x_i = 0$, funcţia de log-verosimilitate

$$l(\lambda) \overset{def.}{=} \ln P(x_1, \ldots, x_n|\lambda) = -n\lambda + \left(\sum_{i=1}^{n} x_i\right) \cdot \ln\lambda - \sum_{i=1}^{n} \ln(x_i!)$$

nu-şi atinge maximul în interiorul domeniului de existenţă pentru λ, şi anume $(0, +\infty)$.

b. Considerăm familia mixturilor de două distribuţii gaussiene unidimensionale:

$$f(x|\theta, \mu_1, \sigma_1, \mu_2, \sigma_2) = \theta \mathcal{N}(x|\mu_1, \sigma_1) + (1-\theta)\mathcal{N}(x|\mu_2, \sigma_2),$$

unde $0 < \theta < 1,\ 0 < \sigma_1,\ 0 < \sigma_2$, iar μ_1, μ_2 sunt numere reale oarecare, distincte. Este imediat că pentru orice estimări bine-definite $\hat{\theta}, \hat{\mu}_1, \hat{\mu}_2, \hat{\sigma}_1$ şi $\hat{\sigma}_2$ obţinute pe un set oarecare de instanţe $x_1, \ldots, x_n$, următorul set de estimări

$$\hat{\theta}' = 1 - \hat{\theta},\ \hat{\mu}'_1 = \hat{\mu}_2,\ \hat{\mu}'_2 = \hat{\mu}_1,\ \hat{\sigma}'_1 = \hat{\sigma}_2,\ \hat{\sigma}'_2 = \hat{\sigma}_1$$

vor produce întotdeauna aceeaşi verosimilitate ca şi estimările iniţiale. Şi totuşi $(\hat{\theta}', \hat{\mu}'_1, \hat{\mu}'_2, \hat{\sigma}'_1, \hat{\sigma}'_2) \neq (\hat{\theta}, \hat{\mu}_1, \hat{\mu}_2, \hat{\sigma}_1, \hat{\sigma}_2)$. Aşadar, estimările în sensul MLE nu sunt în mod necesar unice.

c. În exemplul de la punctul a, domeniul de existenţă pentru parametrul distribuţiei pe care am ales-o este, ca şi în multe alte cazuri, un interval deschis. Însă valoarea acestui parametru pentru care se atinge maximul funcţiei de log-verosimilitate este situată la „marginea" [inferioară a] intervalului, nu în interiorul intervalului respectiv. Această problemă apare adeseori atunci când mărimea eşantionului (adică numărul instanţelor x_i) este mică în comparaţie cu numărul parametrilor care trebuie estimaţi.

În exemplul de la punctul b, funcţia de log-verosimilitate nu este concavă — pentru exemplificare, vedeţi problema 17 de la capitolul de *Clusterizare*, în special graficele care reprezintă curbele de izocontur ale funcţiei de log-verosimilitate ale mixturilor de gaussiene, la pag. 894 şi pag. 895 —, ceea ce implică faptul că pot exista mai multe puncte de maxim local.

0.1.5 Elemente de teoria informaţiei[121]

55. **(Entropie, entropie comună, entropie condiţională, câştig de informaţie: definiţii şi proprietăţi imediate)**

■ • ***Liviu Ciortuz, pornind de la CMU, 2005 fall, T. Mitchell, A. Moore, HW1, pr. 2***

Fie X şi Y variabile aleatoare discrete. Dăm pe scurt următoarele *definiţii*:

- **Entropia variabilei X:**
  $H(X) \stackrel{def.}{=} -\sum_i P(X = x_i) \log P(X = x_i) \stackrel{not.}{=} E_X[-\log P(X)]$.
  Prin *convenţie*, dacă $p(x) = 0$ atunci vom considera $p(x) \log p(x) = 0$.

[121] Observaţie importantă: În toate problemele care urmează, referitor la entropie / teoria informaţiei se va considera în mod implicit că notaţia 'log' desemnează logaritmul în baza 2. De asemenea, prin convenţie, se va considera $p \cdot \log p = 0$ pentru $p = 0$.

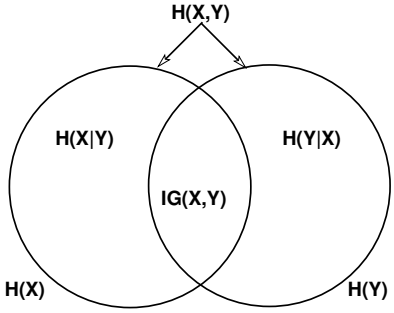
- **Entropia condiţională specifică a variabilei Y în raport cu valoarea x_k a variabilei X:**
  $H(Y \mid X = x_k) \stackrel{def.}{=} -\sum_j P(Y = y_j \mid X = x_k) \log P(Y = y_j \mid X = x_k)$
  $\stackrel{not.}{=} E_{Y|X=x_k}[-\log P(Y \mid X = x_k)]$**.**
- **Entropia condiţională medie a variabilei Y în raport cu variabila X:**
  $H(Y \mid X) \stackrel{def.}{=} \sum_k P(X = x_k) H(Y \mid X = x_k) \stackrel{not.}{=} E_X[H(Y \mid X)]$**.**
- **Entropia comună a variabilelor X şi Y:**
  $H(X,Y) \stackrel{def.}{=} -\sum_i \sum_j P(X = x_i, Y = y_j) \log P(X = x_i, Y = y_j)$
  $\stackrel{not.}{=} E_{X,Y}[-\log P(X,Y)]$**.**
- ***Câştigul de informaţie* al variabilei X în raport cu variabila Y (sau invers), numit de asemenea *informaţia mutuală* (engl., mutual information) a variabilelor X şi Y:**
  $IG(X,Y) \stackrel{not.}{=} MI(X,Y) \stackrel{def.}{=} H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$**.**
  (Observaţie: ultima egalitate de mai sus are loc datorită rezultatului de la punctul c de mai jos.)

Arătaţi că:

a. $H(X) \geq 0$**. În particular,** $H(X) = 0$ **dacă şi numai dacă variabila X este constantă.**

b. $H(Y \mid X) = -\sum_i \sum_j P(X = x_i, Y = y_j) \log P(Y = y_j \mid X = x_i)$**.**

c. $H(X,Y) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y)$**.**
Mai general: $H(X_1, \ldots, X_n) = H(X_1) + H(X_2 \mid X_1) + \ldots + H(X_n \mid X_1, \ldots, X_{n-1})$ **(regula de înlănţuire a entropiilor).**
***Observaţie*: Relaţia precedentă, precum şi relaţia de definiţie pentru câştigul de informaţie sunt ilustrate în figura alăturată.**



Răspuns:

a. Este uşor să arătăm că $H(X) = -\sum_i P(X = x_i) \log P(X = x_i) \geq 0$**.**

Ştim că $\log x \leq 0$ **pentru** $\forall x \leq 1$ **şi** $\log x \geq 0$ **pentru** $\forall x \geq 1$**. De asemenea ştim că** $P(X = x_i) \in [0,1]$ **(fiind o probabilitate). Aşadar,**

$$H(X) = -\sum_i P(X = x_i) \log P(X = x_i) = \sum_i \underbrace{P(X = x_i)}_{\geq 0} \underbrace{\log \frac{1}{P(X = x_i)}}_{\geq 0} \geq 0.$$

Pentru a arăta că $H(X) = 0$ **dacă şi numai dacă X este constantă vom demonstra că ambele implicaţii au loc:**

„⇒“ Presupunem că $H(X) = 0$**, adică** $\sum_i P(X = x_i) \log \frac{1}{P(X = x_i)} = 0$**. Datorită faptului că fiecare termen din această sumă este mai mare sau egal cu** 0**,**

rezultă că $H(X) = 0$ doar dacă pentru $\forall i$, $P(X = x_i) = 0$ sau $\log \dfrac{1}{P(X = x_i)} = 0$, adică dacă pentru $\forall i$, $P(X = x_i) = 0$ sau $P(X = x_i) = 1$. Cum însă $\sum_i P(X = x_i) = 1$ rezultă că există o singură valoare x_1 pentru X astfel încât $P(X = x_1) = 1$, iar $P(X = x) = 0$ pentru orice $x \neq x_1$. Altfel spus, variabila aleatoare discretă X este constantă.[122]

„⇐“ Presupunem că variabila X este constantă, ceea ce înseamnă că X ia o singură valoare x_1, cu probabilitatea $P(X = x_1) = 1$. Prin urmare, $H(X) = -1 \cdot \log 1 = 0$.

b. Pentru a demonstra egalitatea cerută vom porni de la definiţia lui $H(Y \mid X)$ şi apoi vom efectua câteva transformări elementare:

$$\begin{aligned}
H(Y \mid X) &= \sum_i P(X = x_i) H(Y \mid X = x_i) \\
&= \sum_i P(X = x_i) \Big[- \sum_j P(Y = y_j \mid X = x_i) \log P(Y = y_j \mid X = x_i) \Big] \\
&= - \sum_i \sum_j \underbrace{P(X = x_i) P(Y = y_j \mid X = x_i)}_{= P(X = x_i, Y = y_j)} \log P(Y = y_j \mid X = x_i) \\
&= - \sum_i \sum_j P(X = x_i, Y = y_j) \log P(Y = y_j \mid X = x_i).
\end{aligned}$$

c. În primul rând, trebuie să demonstrăm că

$$H(X, Y) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y).$$

Din definiţia entropiei comune ştim că $H(X, Y) = -\sum_i \sum_j P(X = x_i, Y = y_j) \log P(X = x_i, Y = y_j)$. Vom aplica mai întâi regula de înmulţire, $P(X, Y) = P(X) \cdot P(Y \mid X)$, după care vom transforma logaritmul produsului în sumă de logaritmi. Pentru claritatea demonstraţiei vom nota prescurtat $p(x_i) = P(X = x_i)$, $p(x_i, y_j) = P(X = x_i, Y = y_j)$ etc.

$$\begin{aligned}
H(X, Y) &= - \sum_i \sum_j p(x_i, y_j) \log p(x_i, y_j) \\
&= - \sum_i \sum_j p(x_i) \cdot p(y_j \mid x_i) \log [p(x_i) \cdot p(y_j \mid x_i)] \\
&= - \sum_i \sum_j p(x_i) \cdot p(y_j \mid x_i) [\log p(x_i) + \log p(y_j \mid x_i)] \\
&= - \sum_i \sum_j p(x_i) \cdot p(y_j \mid x_i) \log p(x_i) - \sum_i \sum_j p(x_i) \cdot p(y_j \mid x_i) \log p(y_j \mid x_i) \\
&= - \sum_i p(x_i) \log p(x_i) \cdot \underbrace{\sum_j p(y_j \mid x_i)}_{=1} - \sum_i p(x_i) \sum_j p(y_j \mid x_i) \log p(y_j \mid x_i) \\
&= H(X) + \sum_i p(x_i) H(Y \mid X = x_i) = H(X) + H(Y \mid X).
\end{aligned}$$

[122] Mai corect spus, X este constantă pe tot domeniul de definiţie, eventual cu excepţia unei mulţimi de probabilitate 0.

Egalitatea $\sum_j p(y_j \mid x_i) = 1$ se justifică uşor ţinând cont de *proprietatea de aditivitate numărabilă* din definiţia funcţiei / distribuţiei de probabilitate.

Pentru a demonstra egalitatea $H(X,Y) = H(Y) + H(X \mid Y)$, se procedează analog, înlocuind $p(x_i, y_j)$ nu cu $p(x_i) \cdot p(y_j \mid x_i)$, ci cu $p(y_i) \cdot p(x_j \mid y_i)$.

Pentru cazul general

$$H(X_1, \ldots, X_n) = H(X_1) + H(X_2 \mid X_1) + \ldots + H(X_n \mid X_1, \ldots, X_{n-1}),$$

vom folosi regula de înlănţuire de la variabile aleatoare

$$P(X_1, \ldots, X_n) = P(X_1) \cdot P(X_2 \mid X_1) \cdot P(X_3 \mid X_1, X_2) \cdot \ldots \cdot P(X_n \mid X_1, \ldots, X_{n-1}),$$

precum şi scrierea entropiei sub formă de medie, $H(X) = E\left[\log \frac{1}{P(X)}\right]$:

$$\begin{aligned}
H(X_1, \ldots, X_n) &= E\left[\log \frac{1}{p(x_1, \ldots, x_n)}\right] \\
&= -E_{p(x_1,\ldots,x_n)}[\log \underbrace{p(x_1, \ldots, x_n)}_{p(x_1)\cdot p(x_2|x_1)\cdot \ldots \cdot p(x_n|x_1,\ldots,x_{n-1})}] \\
&= -E_{p(x_1,\ldots,x_n)}[\log p(x_1) + \log p(x_2 \mid x_1) + \ldots + \log p(x_n \mid x_1, \ldots, x_{n-1})] \\
&= -E_{p(x_1)}[\log p(x_1)] - E_{p(x_1,x_2)}[\log p(x_2 \mid x_1)] - \ldots \\
&\quad - E_{p(x_1,\ldots,x_n)}[\log p(x_n \mid x_1, \ldots, x_{n-1})] \\
&\overset{(b)}{=} H(X_1) + H(X_2 \mid X_1) + \ldots + H(X_n \mid X_1, \ldots, X_{n-1}).
\end{aligned}$$

La penultima egalitate am ţinut cont de definiţia distribuţiei marginale pornind de la distribuţia comună, iar la ultima egalitate am folosit rezultatul de la punctul b.

56. **(Entropie, entropie condiţională specifică, câştig de informaţie: exemplificare)**

■ □ • ○ ***CMU, 2012 spring, Roni Rosenfeld, HW2, pr. 2***

Problema aceasta se referă la aruncarea a două zaruri perfecte, cu 6 feţe.

a. Calculează distribuţia probabilistă a sumei numerelor de pe cele două feţe care au fost obţinute / „observate“ în urma aruncării zarurilor.

În continuare, suma aceasta va fi asimilată cu o variabilă aleatoare, notată cu S.

b. Cantitatea de *informaţie* obţinută (sau: *surpriza* pe care o resimţim) la „observarea“ producerii valorii x a unei variabile aleatoare X oarecare este prin *definiţie*

$$\mathit{Information}(P(X = x)) = \mathit{Surprise}(P(X = x)) = \log_2 \frac{1}{P(X = x)} = -\log_2 P(X = x).$$

Această cantitate este exprimată (numeric) în *biţi de informaţie*.

Cât de surprins vei fi atunci când vei „observa“ $S = 2$, respectiv $S = 11$, $S = 5$ şi $S = 7$? (Vei exprima de fiecare dată rezultatul în biţi. Puteţi folosi $\log_2 3 = 1.584962501$.)

c. Calculează entropia variabilei S.

d. Să presupunem acum că vei arunca aceste două zaruri pe rând, iar la aruncarea primului zar se obţine numărul 4. Cât este entropia lui S în urma acestei „observaţii"? S-a pierdut, ori s-a câştigat informaţie în acest proces? Calculează cât de multă informaţie (exprimată în biţi) s-a pierdut ori s-a câştigat.

Răspuns:

a. Redăm distribuţia lui S (uşor de calculat) în următorul tabel:

S	2	3	4	5	6	7	8	9	10	11	12
$P(S)$	1/36	2/36	3/36	4/36	5/36	6/36	5/36	4/36	3/36	2/36	1/36

b. Conform definiţiei date, vom avea:

$$\begin{aligned}
\mathit{Information}(S=2) &= -\log_2(1/36) = \log_2 36 = 2\log_2 6 = 2(1+\log_2 3)\\
&= 5.169925001 \text{ biţi}\\
\mathit{Information}(S=11) &= -\log_2 2/36 = \log_2 18 = 1+2\log_2 3 = 4.169925001 \text{ biţi}\\
\mathit{Information}(S=5) &= -\log_2 4/36 = \log_2 9 = 2\log_2 3 = 3.169925001 \text{ biţi}\\
\mathit{Information}(S=7) &= -\log_2 6/36 = \log_2 6 = 1+\log_2 3 = 2.584962501 \text{ biţi}
\end{aligned}$$

c. Conform definiţiei pentru entropie (vedeţi problema 55), $H(S)$ este media ponderată (cu ajutorul probabilităţilor) a „surprizelor" / cantităţilor de informaţie produse la „observarea" tuturor valorilor variabilei S. Făcând calculele, vom obţine:

$$\begin{aligned}
H(S) &= -\sum_{i=1}^{n} p_i \log_2 p_i\\
&= -\left(2\cdot\frac{1}{36}\log_2\frac{1}{36} + 2\cdot\frac{2}{36}\log_2\frac{2}{36} + 2\cdot\frac{3}{36}\log_2\frac{3}{36} + 2\cdot\frac{4}{36}\log_2\frac{4}{36} + \right.\\
&\quad \left. 2\cdot\frac{5}{36}\log_2\frac{5}{36} + \frac{6}{36}\log_2\frac{6}{36}\right)\\
&= \frac{1}{36}\left(2\log_2 36 + 4\log_2 18 + 6\log_2 12 + 8\log_2 9 + 10\log_2\frac{36}{5} + 6\log_2 6\right)\\
&= \frac{1}{36}\left(2\log_2 6^2 + 4\log_2 6\cdot 3 + 6\log_2 6\cdot 2 + 8\log_2 3^2 + 10\log_2\frac{6^2}{5} + 6\log_2 6\right)\\
&= \frac{1}{36}(40\log_2 6 + 20\log_2 3 + 6 - 10\log_2 5)\\
&= \frac{1}{36}(60\log_2 3 + 46 - 10\log_2 5) = 3.274401919 \text{ biţi.}
\end{aligned}$$

d. Distribuţia variabilei S condiţionată de observarea feţei 4 la prima aruncare este:

S	2	3	4	5	6	7	8	9	10	11	12
$P(S\|...)$	0	0	0	1/6	1/6	1/6	1/6	1/6	1/6	0	0

În consecinţă, folosind definiţia entropiei condiţionale specifice (vedeţi de asemenea problema 55), vom avea:

$$H(S|\textit{First-die-shows-4}) = -6\cdot\frac{1}{6}\log_2\frac{1}{6} = \log_2 6 = 2.58 \text{ biţi,}$$

ceea ce înseamnă că se obţine următorul *câştig* de informaţie:

$$IG(S; \textit{First-die-shows-4}) = H(S) - H(S|\textit{First-die-shows-4}) = 3.27 - 2.58 = 0.69 \text{ biţi}.$$

Altfel spus, atunci când ni se comunică faptul că la aruncarea celor două zaruri primul dintre ele produce faţa 4, această informaţie va reduce ulterior entropia variabilei S (sau, am putea spune, „surpriza“ medie provocată de valorile ei) cu 0.69 biţi.

57. **(Probabilităţi marginale, entropii, entropii condiţionale medii)**

■ □ • ***CMU, 2012 spring, Roni Rosenfeld, HW2, pr. 3***

Un doctor trebuie să pună un diagnostic unui pacient care are simptome de răceală (C, de la engl. cold). Factorul principal pe care doctorul îl ia în considerare pentru a elabora diagnosticul este timpul, adică starea vremii de afară (T). Variabila aleatoare C ia două valori, *yes* şi *no*, iar variabila aleatoare T ia 3 valori: *sunny* (însorit), *rainy* (ploios) şi *snowy* (foarte rece, să zicem). Distribuţia comună a celor două variabile este dată în tabelul următor:

	$T = sunny$	$T = rainy$	$T = snowy$
$C = no$	0.30	0.20	0.10
$C = yes$	0.05	0.15	0.20

a. Calculaţi probabilităţile marginale $P(C)$ şi $P(T)$.

***Sugestie*: Folosiţi formula $P(X = x) = \sum_y P(X = x; Y = y)$. De exemplu,**

$$P(C = no) = P(C = no, T = sunny) + P(C = no, T = rainy) + P(C = no, T = snowy).$$

b. Calculaţi entropiile $H(C)$ şi $H(T)$.

c. Calculaţi entropiile condiţionale medii $H(C|T)$ şi $H(T|C)$.

Răspuns:

a. Folosind formula dată, vom obţine: $P_C = (0.6, 0.4)$ şi $P_T = (0.35, 0.35, 0.30)$.

b. Aplicând definiţia pentru entropie (vedeţi problema 55), rezultă:

$$\begin{aligned} H(C) &= 0.6 \log_2 \frac{5}{3} + 0.4 \log_2 \frac{5}{2} = \log_2 5 - 0.6 \log_2 3 - 0.4 = 0.971 \text{ biţi} \\ H(T) &= 2 \cdot 0.35 \log_2 \frac{20}{7} + 0.3 \log_2 \frac{10}{3} \\ &= 0.7(2 + \log_2 5 - \log_2 7) + 0.3(1 + \log_2 5 - \log_2 3) \\ &= 1.7 + \log_2 5 - 0.7 \log_2 7 - 0.3 \log_2 3 = 1.581 \text{ biţi}. \end{aligned}$$

c. Aplicând definiţia pentru entropie condiţională medie (vedeţi de asemenea problema 55), vom avea:

$$\begin{aligned} H(C|T) &\stackrel{def.}{=} \sum_{t \in Val(T)} P(T = t) \cdot H(C|T = t) \\ &= P(T = sunny) \cdot H(C|T = sunny) + P(T = rainy) \cdot H(C|T = rainy) + \\ &\quad P(T = snowy) \cdot H(C|T = snowy) \end{aligned}$$

$$
\begin{aligned}
&= 0.35 \cdot H\left(\frac{0.30}{0.30+0.05}, \frac{0.05}{0.30+0.05}\right) + 0.35 \cdot H\left(\frac{0.20}{0.20+0.15}, \frac{0.15}{0.20+0.15}\right) + \\
&\quad 0.30 \cdot H\left(\frac{0.10}{0.10+0.20}, \frac{0.20}{0.20+0.10}\right) \\
&= \frac{7}{20} \cdot H\left(\frac{6}{7}, \frac{1}{7}\right) + \frac{7}{20} \cdot H\left(\frac{4}{7}, \frac{3}{7}\right) + \frac{3}{10} \cdot H\left(\frac{1}{3}, \frac{2}{3}\right) \\
&= \frac{7}{20} \cdot \left(\frac{6}{7}\log_2\frac{7}{6} + \frac{1}{7}\log_2 7\right) + \frac{7}{20} \cdot \left(\frac{4}{7}\log_2\frac{7}{4} + \frac{3}{7}\log_2\frac{7}{3}\right) + \frac{3}{10} \cdot \left(\frac{1}{3}\log_2 3 + \frac{2}{3}\log_2\frac{3}{2}\right) \\
&= \frac{7}{20} \cdot \left(\log_2 7 - \frac{6}{7} - \frac{6}{7}\log_2 3\right) + \frac{7}{20} \cdot \left(\log_2 7 - \frac{8}{7} - \frac{3}{7}\log_2 3\right) + \frac{3}{10} \cdot \left(\log_2 3 - \frac{2}{3}\right) \\
&= \frac{7}{10}\log_2 7 - \left(\frac{3}{10} + \frac{4}{10} + \frac{2}{10}\right) - \left(\frac{6}{20} + \frac{3}{20} - \frac{3}{10}\right) \cdot \log_2 3 \\
&= \frac{7}{10}\log_2 7 - \frac{3}{20}\log_2 3 - \frac{9}{10} = 0.82715 \text{ biţi.}
\end{aligned}
$$

Similar,

$$
\begin{aligned}
H(T|C) &\stackrel{def.}{=} \sum_{c \in Val(C)} P(C=c) \cdot H(T|C=c) \\
&= P(C=no) \cdot H(T|C=no) + P(C=yes) \cdot H(T|C=yes) \\
&= 0.60 \cdot H\left(\frac{0.30}{0.30+0.20+0.10}, \frac{0.20}{0.30+0.20+0.10}, \frac{0.10}{0.30+0.20+0.10}\right) \\
&\quad +0.40 \cdot H\left(\frac{0.05}{0.05+0.15+0.20}, \frac{0.15}{0.05+0.15+0.20}, \frac{0.20}{0.05+0.15+0.20}\right) \\
&= \frac{3}{5} \cdot H\left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right) + \frac{2}{5} \cdot H\left(\frac{1}{8}, \frac{3}{8}, \frac{1}{2}\right) \\
&= \frac{3}{5}\left(\frac{1}{2} + \frac{1}{3}\log_2 3 + \frac{1}{6}(1+\log_2 3)\right) + \frac{2}{5}\left(\frac{1}{8}\cdot 3 + \frac{3}{8}(3-\log_2 3) + \frac{1}{2}\right) \\
&= \frac{3}{5}\left(\frac{2}{3} + \frac{1}{2}\log_2 3\right) + \frac{2}{5}\left(2 - \frac{3}{8}\log_2 3\right) = \frac{6}{5} + \frac{3}{20}\log_2 3 = 1.43774 \text{ biţi.}
\end{aligned}
$$

58. **(Câştigul de informaţie / informaţia mutuală, o aplicaţie: selecţia de trăsături)**

■ □ • ○ ***CMU, 2009 spring, Ziv Bar-Joseph, HW5, pr. 6***

În tabelul următor se dă un set de opt observaţii / instanţe, reprezentate ca tupluri de valori ale variabilelor aleatoare binare de „intrare“ X_1, X_2, X_3, X_4, X_5 şi ale variabilei aleatoare binare de „ieşire“ Y.

Am dori să reducem spaţiul de trăsături $\{X_1, X_2, X_3, X_4, X_5\}$ folosind o metodă de selecţie de tip *filtru*.

a. Calculaţi câştigul de informaţie / informaţia mutuală $MI(X_i, Y)$ pentru fiecare i.

b. Ţinând cont de rezultatul de la punctul precedent, alegeţi cel mai mic subset de trăsături în aşa fel încât cel mai bun clasificator antrenat pe acest spaţiu (redus) de trăsături să fie cel puţin la fel de bun ca şi cel mai bun clasificator antrenat pe întreg spaţiul de trăsături. Justificaţi alegerea pe care aţi făcut-o.

X_1	X_2	X_3	X_4	X_5	Y
0	1	1	0	1	0
1	0	0	0	1	0
0	1	0	1	0	1
1	1	1	1	0	1
0	1	1	0	0	1
0	0	0	1	1	1
1	0	0	1	0	1
1	1	1	0	1	1

Răspuns:

a. Pentru calculul informaţiei mutuale putem folosi formula din problema 141 sau problema 65.b:

$$MI(X,Y) = \sum_x \sum_y p_{X,Y}(x,y) \log \left(\frac{p_{X,Y}(x,y)}{p_X(x) p_Y(y)} \right)$$

Probabilităţile marginale, estimate în sensul verosimilităţii maxime (MLE), sunt:

	P_{X_1}	P_{X_2}	P_{X_3}	P_{X_4}	P_{X_5}	P_Y
0	1/2	3/8	1/2	1/2	1/2	1/4
1	1/2	5/8	1/2	1/2	1/2	3/4

iar probabilităţile comune sunt:

X_i	Y	$P_{X_1,Y}$	$P_{X_2,Y}$	$P_{X_3,Y}$	$P_{X_4,Y}$	$P_{X_5,Y}$
0	0	1/8	1/8	1/8	1/4	0
0	1	3/8	1/4	3/8	1/4	1/2
1	0	1/8	1/8	1/8	0	1/4
1	1	3/8	1/2	3/8	1/2	1/4

Se poate observa că X_1 şi Y sunt independente, deci $MI(X_1, Y) = 0$, conform proprietăţii care este demonstrată la problema 141 (sau la problemele 65.c şi 146.b). Similar, $MI(X_3, Y) = 0$. În rest, efectuând calculele obţinem $MI(X_2, Y) = 0.01571$, $MI(X_4, Y) = 0.3113$ şi $MI(X_5, Y) = 0.3113$.

b. La selecţia de trăsături vom alege acele trăsături X_i care au informaţie mutuală nenulă în raport cu Y. Acestea sunt X_2, X_4 şi X_5. Celelalte două trăsături, X_1 şi X_3 sunt independente în raport cu Y.

Totuşi, inspectând datele, observăm că dacă vom selecta doar trăsăturile X_2, X_4 şi X_5 vom avea două instanţe (vedeţi prima şi ultima linie din tabel) care au aceleaşi trăsături ($X_2 = 1, X_4 = 0, X_5 = 1$) dar au etichete / ieşiri diferite: $Y = 0$, respectiv $Y = 1$. Aşadar, vom adăuga la setul de trăsături selectate anterior şi variabila X_1, care va permite dezambiguizarea în cazul acestor două instanţe, menţinând astfel „consistenţa“ setului de date.

***Observaţie*: Deşi $MI(X_1, Y) = 0$ — sau, echivalent spus, X_1 este independent de Y —, nu rezultă că variabila X_1 combinată cu una sau mai multe variabile X_j, cu $j \in \{2, 4, 5\}$, şi formând astfel o nouă variabilă aleatoare, rămâne independentă de Y. Noua variabilă poate avea câştig de informaţie nenul (în unele cazuri chiar maxim!) în raport cu Y.**

59. **(Entropia comună: forma particulară a relaţiei de „înlănţuire“ în cazul variabilelor aleatoare independente)**

□ ***prelucrare de Liviu Ciortuz, după CMU, 2012 spring, Roni Rosenfeld, HW2, pr. 7.b***

Conform problemei 55.c, *formula de înlănţuire* a entropiilor pentru cazul general (adică, indiferent dacă X şi Y sunt sau nu independente) este:

$$H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y). \tag{65}$$

Demonstraţi că dacă X şi Y sunt variabile aleatoare discrete independente, atunci $H(X,Y) = H(X) + H(Y)$, şi reciproc: atunci când are loc egalitatea $H(X,Y) = H(X) + H(Y)$ rezultă că variabilele X şi Y sunt independente.

Răspuns:

Conform definiţiei câştigului de informaţie (vedeţi problema 55),

$$IG(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{66}$$

De asemenea, conform problemei 146.b (sau, conform problemei 65.c),

$$IG(X,Y) = 0 \Leftrightarrow X \text{ şi } Y \text{ sunt independente.} \tag{67}$$

Din relaţiile (66) şi (67) rezultă că

$$H(Y) = H(Y|X) \Leftrightarrow X \text{ şi } Y \text{ sunt independente.} \tag{68}$$

Aşadar, dacă X şi Y sunt independente, coroborând relaţiile (68) şi (65) vom avea $H(X,Y) = H(Y) + H(X)$.

Invers, dacă $H(X,Y) = H(X)+H(Y)$, din relaţia (65) rezultă că $H(Y) = H(Y|X)$, ceea ce implică faptul că X şi Y sunt independente, conform relaţiei (68).

***Observaţie*: Din egalităţile (65) şi (66), rezultă**

$$H(X,Y) = H(X) + H(Y) - IG(X,Y).$$

Conform proprietăţii de nenegativitate a câştigului de informaţie ($IG(X,Y) \geq 0$; vedeţi problema 146.a sau problema 65.c), rezultă

$$H(X,Y) \leq H(X) + H(Y).$$

Această ultimă relaţie este ilustrată în figura de la finalul enunţului problemei 55.

60. **(Entropia distribuţiei continue uniforme şi a distribuţiilor gaussiene uni- şi multidimensionale)**

□ ***Liviu Ciortuz, 2019***

Pentru o variabilă aleatoare X care urmează o distribuţie continuă având funcţia densitate de probabilitate (p.d.f.) p, entropia se defineşte astfel:

$$H(X) = \int_{-\infty}^{+\infty} p(x) \log_2 \frac{1}{p(x)} dx \overset{p(x)\neq 0}{=} - \int_{-\infty}^{+\infty} p(x) \log_2 p(x) dx.$$

***Indicaţie*: Dacă $p(x) = 0$, veţi presupune că $-p(x)\log_2 p(x) = 0$.**

a. Calculaţi entropia distribuţiei uniforme continue definite pe intervalul $[a, b]$.

b. Calculaţi entropia distribuţiei gaussiene unidimensionale de parametri $\mu \in \mathbb{R}$ şi $\sigma^2 \in \mathbb{R}_+$, pentru care funcţia de densitate de probabilitate (p.d.f.) este:

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad \textbf{pentru } x \in \mathbb{R}.$$

c. Calculaţi entropia distribuţiei gaussiene multidimensionale de parametri $\mu \in \mathbb{R}^d$ şi $\Sigma \in \mathbb{R}^{d\times d}$ (matrice simetrică şi pozitiv definită), pentru care funcţia de densitate de probabilitate (p.d.f.) este:

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\Big(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\Big) \quad \textbf{pentru } x \in \mathbb{R}^d.$$

***Sugestie*: Următoarea formulă, preluată din documentul *Matrix Identities*, de Sam Roweis, 1999, vă poate fi de folos:**

$$\textbf{(2e)} \quad \mathrm{Tr}(AB) = \mathrm{Tr}(BA).$$

Răspuns:

a. Dacă f este funcţia densitate de probabilitate (p.d.f.) a distribuţiei uniforme continue pe intervalul $[a, b]$, atunci prin definiţie există o constantă $c \in \mathbb{R}_+$ cu proprietatea că $f(x) = c$ pentru orice $x \in [a, b]$. Din condiţia [care ţine tot de definiţie] $\int_a^b f(x)\,dx = 1$ rezultă imediat că valoarea constantei c este $\frac{1}{b-a}$. Aşadar,

$$f(x) = \begin{cases} \frac{1}{b-a} & \textbf{pentru } \forall x \in [a, b]; \\ 0 & \textbf{altfel}. \end{cases}$$

Entropia unei variabile aleatoare X care urmează această distribuţie uniformă continuă se calculează astfel:

$$\begin{aligned} H(X) &= -\int_a^b f(x)\log_2 f(x)\,dx = \int_a^b \frac{1}{b-a}\log_2(b-a)\,dx = \frac{1}{b-a}\log_2(b-a)\cdot x\Big|_a^b \\ &= \frac{1}{\cancel{b-a}}\cdot\log_2(b-a)\cdot\cancel{(b-a)} = \log_2(b-a). \end{aligned}$$

b.[123] **Pentru a calcula entropia distribuţiei gaussiene unidimensionale, ne vom folosi de următoarele rezultate:**

$$\int_{-\infty}^{\infty} \mathcal{N}(x)dx = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx = 1 \quad \textbf{(deoarece } \mathcal{N} \textbf{ este p.d.f.)} \tag{69}$$

$$\int_{-\infty}^{\infty} v^2 e^{-\frac{v^2}{2}}\,dv = \sqrt{2\pi} \quad \textbf{(din rezolvarea ex. 32.c)} \tag{70}$$

[123] Soluţie redactată iniţial de Andi Munteanu (std., FII, 2019f).

Fie $X \sim \mathcal{N}(x|\mu, \sigma^2)$. Entropia lui X se calculează astfel:

$$H(X) \overset{def.}{=} \int_{-\infty}^{\infty} \mathcal{N}(x) \log_2 \frac{1}{\mathcal{N}(x)} dx = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \log_2 \left(\sqrt{2\pi}\sigma \cdot e^{+\frac{(x-\mu)^2}{2\sigma^2}} \right) dx$$

$$= \log_2(\sqrt{2\pi}\sigma) \underbrace{\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx}_{1} + \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \log_2 e^{+\frac{(x-\mu)^2}{2\sigma^2}} dx$$

$$\overset{(69)}{=} \frac{1}{\ln 2} \left(\ln \sqrt{2\pi}\sigma + \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} \frac{(x-\mu)^2}{2\sigma^2} e^{-\frac{(x-\mu)^2}{2\sigma^2}} dx \right).$$

Pentru a calcula integrala, vom face schimbarea de variabilă $v = \frac{x-\mu}{\sigma}$, din care rezultă $x = \sigma v + \mu$ şi $dx = \sigma dv$. Mai departe, reluând calculul entropiei $H(X)$, vom avea:

$$H(X) = \frac{1}{\ln 2} \left(\ln \sqrt{2\pi}\sigma + \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{\infty} \frac{v^2}{2} e^{-\frac{v^2}{2}} \sigma dv \right) \overset{(70)}{=} \frac{1}{\ln 2} \left(\frac{1}{2} \ln(2\pi\sigma^2) + \frac{1}{2\cancel{\sqrt{2\pi}}} \cancel{\sqrt{2\pi}} \right)$$

$$= \frac{1}{2} \frac{\ln(2\pi\sigma^2 e)}{\ln 2} = \frac{1}{2} \log_2(2\pi\sigma^2 e) = \log_2 \sqrt{2\pi\sigma^2 e}.$$

c.[124] **Fie $X \sim \mathcal{N}(x \,|\, \mu, \Sigma)$ şi $x \in \mathbb{R}^d$. Vom folosi următoarele proprietăţi:**

$$E[\ln p(x)] \overset{def.}{=} \int_{-\infty}^{+\infty} p(x) \ln p(x)\, dx \tag{71}$$

$$(x-\mu)^\top \in \mathbb{R}^{1\times d},\ \Sigma^{-1} \in \mathbb{R}^{d\times d},\ (x-\mu) \in \mathbb{R}^{d\times 1} \Rightarrow (x-\mu)^\top \Sigma^{-1} (x-\mu) \in \mathbb{R}$$

$$\Rightarrow (x-\mu)^\top \Sigma^{-1} (x-\mu) = \mathrm{Tr}((x-\mu)^\top \Sigma^{-1} (x-\mu)) \tag{72}$$

$$\Sigma\mu = E[X] \tag{73}$$

$$\Sigma = Var(X) \overset{def.}{=} Cov(X,\, X) \overset{def.}{=} E[(X - E[X])(X - E[X])^\top] \overset{(73)}{=} E[(X-\mu)(X-\mu)^\top]. \tag{74}$$

***Notă*: Pentru demonstrarea egalităţilor $\mu = E[X]$ şi $\Sigma = Var(X)$, vedeţi** www.statlect.com/probability-distributions/multivariate-normal-distribution**.**

Entropia lui X se calculează astfel:

$$H(X) \overset{def.}{=} -\int_{-\infty}^{+\infty} p(x) \log_2 p(x)\, dx = -\frac{1}{\ln 2} \int_{-\infty}^{+\infty} p(x) \ln p(x)\, dx$$

$$\overset{(71)}{=} -\frac{1}{\ln 2} E[\ln p(x)]\, dx = -\frac{1}{\ln 2} E[\ln (\frac{1}{(\sqrt{2\pi})^d \sqrt{|\Sigma|}} e^{-\frac{1}{2}(x-\mu)^\top \Sigma^{-1} (x-\mu)})]\, dx$$

$$= -\frac{1}{\ln 2} E[-\frac{d}{2} \ln(2\pi) - \frac{1}{2} \ln(|\Sigma|) - \frac{1}{2}(x-\mu)^\top \Sigma^{-1} (x-\mu)]$$

$$= \frac{1}{\ln 2} \left(\frac{d}{2} \ln(2\pi) + \frac{1}{2} \ln(|\Sigma|) + \frac{1}{2} E[(x-\mu)^\top \Sigma^{-1} (x-\mu)] \right)$$

$$\overset{(72)}{=} \frac{1}{\ln 2} \left(\frac{d}{2} \ln(2\pi) + \frac{1}{2} \ln(|\Sigma|) + \frac{1}{2} E[\mathrm{Tr}((x-\mu)^\top \Sigma^{-1} (x-\mu))] \right)$$

[124] Soluţie redactată iniţial de Georgiana Ojoc (std., FII, 2020f).

$$\begin{aligned}
&\overset{(2e)}{=} \quad \frac{1}{\ln 2}\Big(\frac{d}{2}\ln(2\pi)+\frac{1}{2}\ln(|\Sigma|)+\frac{1}{2}E[\mathrm{Tr}(\Sigma^{-1}(x-\mu)(x-\mu)^\top)]\Big)\\
&\overset{lin.\ med.}{=} \quad \frac{1}{\ln 2}\Big(\frac{d}{2}\ln(2\pi)+\frac{1}{2}\ln(|\Sigma|)+\frac{1}{2}\mathrm{Tr}(\Sigma^{-1}E[(x-\mu)(x-\mu)^\top)]\Big)\\
&\overset{(74)}{=} \quad \frac{1}{\ln 2}\Big(\frac{d}{2}\ln(2\pi)+\frac{1}{2}\ln(|\Sigma|)+\frac{1}{2}\mathrm{Tr}(\Sigma^{-1}\Sigma)\Big)\\
&= \quad \frac{1}{\ln 2}\Big(\frac{d}{2}\ln(2\pi)+\frac{1}{2}\ln(|\Sigma|)+\frac{1}{2}\mathrm{Tr}(I_d)\Big)\\
&= \quad \frac{1}{\ln 2}\Big(\frac{d}{2}\ln(2\pi)+\frac{1}{2}\ln(|\Sigma|)+\frac{d}{2}\Big).
\end{aligned}$$

61. **(Calcularea entropiei unor variabile aleatoare continue: cazul distribuţiei exponenţiale şi al distribuţiei Gamma)**

a. ■ ***CMU, 2011 spring, Roni Rosenfeld, HW2, pr. 2.c***

Calculaţi entropia *distribuţiei* continue *exponenţiale* de parametru $\lambda > 0$. Vă reamintim că definiţia p.d.f.-ului acestei distribuţii este următoarea:[125]

$$p(x) = \begin{cases} \lambda e^{-\lambda x}, & \textbf{dacă } x \geq 0; \\ 0, & \textbf{dacă } x < 0. \end{cases}$$

b. ***UAIC, 2020, Georgiana Ojoc, Liviu Ciortuz***

Calculaţi entropia distribuţiei Gamma, pentru care funcţia de densitate de probabilitate este:[126]

$$p(x) \overset{not.}{=} Gamma(x \,|\, r, \alpha) \overset{def.}{=} \frac{\alpha^r}{\Gamma(r)} x^{r-1} e^{-\alpha x} \textbf{ pentru } x \geq 0,$$

cu $r > 0$ (forma), $\alpha > 0$ (rata) şi $\Gamma(r) \overset{def.}{=} \int_0^{+\infty} x^{r-1} e^{-x}\, dx$ pentru orice $r > 0$ (funcţia lui Euler). (Vedeţi ex. 31.b.)

***Indicaţie*:**

Este posibil să fie nevoie să folosiţi o proprietate specială, numită *regula de derivare sub semnul de integrală* (engl., differentiation under the integral sign), care constituie obiectul pentru următoarea

***Teoremă*: Dacă $f(r,t)$ este o funcţie cu valori reale, continuă şi derivabilă în raport cu r pe intervalul (a,b), iar derivata parţială $\frac{\partial}{\partial r} f(r,t)$ este de asemenea continuă pe intervalul (a,b), atunci**[127]

$$\int_a^b \frac{\partial}{\partial r} f(r,t)\, dt = \frac{\partial}{\partial r} \int_a^b f(r,t)\, dt.$$

Menţionăm că regula aceasta, datorată lui Gottfried Leibnitz (1646-1716), se generalizează — sub o formă semnificativ mai elaborată — la cazul când limitele de integrare a şi b depind de r, adică sunt de forma $a(r)$ şi respectiv $b(r)$, iar aceste două funcţii sunt continue şi au derivatele continue.

[125] La ex. 31.a puteţi vedea graficul acestei funcţii de densitate pentru câteva valori ale parametrului λ.

[126] La ex. 31.c puteţi vedea graficul acestei funcţii de densitate pentru câteva valori ale parametrillor r şi α.

[127] Pentru demonstraţie, vedeţi „Proof of basic form“ pe site-ul https://en.wikipedia.org/wiki/Leibniz_integral_rule#Alternative_Proof_of_General_Form_with_Variable_Limits,_using_the_Chain_Rule (accesat la data de 15.12.2020).

Răspuns:

a. Dat fiind faptul că funcţia p se anulează pe intervalul $(-\infty, 0)$, este natural ca mai întâi să „rupem“ intervalul de integrare pentru $\int_{-\infty}^{\infty} p(x) \log_2 \frac{1}{p(x)}\, dx$ în două: $(-\infty, 0)$ şi $[0, \infty)$. Aşadar,

$$\begin{aligned} H(X) &= \int_{-\infty}^{0} p(x) \log_2 \frac{1}{p(x)}\, dx + \int_{0}^{\infty} p(x) \log_2 \frac{1}{p(x)}\, dx \\ &\overset{\text{def. } p}{=} -\int_{-\infty}^{0} 0 \log_2 0\, dx + \int_{0}^{\infty} \lambda e^{-\lambda x} \log_2 \frac{1}{\lambda e^{-\lambda x}}\, dx. \end{aligned}$$

Prima dintre aceste două ultime integrale este 0, conform *indicaţiei* din enunţ. Pentru a putea calcula mai uşor cea de-a doua integrală (în expresia căreia apare numărul e), vom schimba baza logaritmului, şi anume vom trece din baza 2 în baza e (baza logaritmului natural, $\ln$).[128]

Prin urmare,

$$\begin{aligned} H(X) &= \frac{1}{\ln 2} \int_{0}^{\infty} \lambda e^{-\lambda x} \ln \frac{1}{\lambda e^{-\lambda x}}\, dx \\ &= \frac{1}{\ln 2} \int_{0}^{\infty} \lambda e^{-\lambda x} \left(\ln \frac{1}{\lambda} + \ln \frac{1}{e^{-\lambda x}} \right) dx \\ &= \frac{1}{\ln 2} \int_{0}^{\infty} \lambda e^{-\lambda x} \left(-\ln \lambda + \ln e^{\lambda x} \right) dx \\ &= \frac{1}{\ln 2} \int_{0}^{\infty} \lambda e^{-\lambda x} \left(-\ln \lambda + \lambda x \right) dx \\ &= \frac{1}{\ln 2} \int_{0}^{\infty} \lambda e^{-\lambda x} (-\ln \lambda)\, dx + \frac{1}{\ln 2} \int_{0}^{\infty} \lambda e^{-\lambda x} \lambda x\, dx \\ &= \frac{-\ln \lambda}{\ln 2} \int_{0}^{\infty} \lambda e^{-\lambda x}\, dx + \frac{\lambda}{\ln 2} \int_{0}^{\infty} \lambda e^{-\lambda x} x\, dx. \end{aligned}$$

Prima integrală are valoarea 1, întrucât $p(\,)$ este p.d.f.-ul distribuţiei exponenţiale (vedeţi ex. 31.a). Cea de-a doua integrală are valoarea $1/\lambda$, întrucât reprezintă media distribuţiei exponenţiale (vedeţi enunţul aceluiaşi ex. 31.a). Prin urmare,

$$H(X) = -\frac{\ln \lambda}{\ln 2} + \frac{\lambda}{\ln 2} \cdot \frac{1}{\lambda} = -\frac{\ln \lambda}{\ln 2} + \frac{1}{\ln 2} = \frac{1 - \ln \lambda}{\ln 2}.$$

b. Fie $X \sim Gamma(x \mid r, \alpha)$. Entropia lui X se calculează astfel:

$$\begin{aligned} H(X) &\overset{def.}{=} -\int_{0}^{+\infty} p(x) \log_2 p(x)\, dx = -\frac{1}{\ln 2} \int_{0}^{+\infty} p(x) \ln p(x)\, dx \\ &= -\frac{1}{\ln 2} \int_{0}^{+\infty} p(x) \ln \Big(\frac{\alpha^r}{\Gamma(r)} x^{r-1} e^{-\alpha x} \Big)\, dx \\ &= -\frac{1}{\ln 2} \int_{0}^{+\infty} p(x) \Big(\ln \frac{\alpha^r}{\Gamma(r)} + \ln x^{r-1} + \ln e^{-\alpha x} \Big)\, dx \end{aligned}$$

[128] Pentru aceasta, vom folosi formula $\log_a b = \frac{\log_c b}{\log_c a}$, valabilă pentru orice $a > 0$, $b > 0$ şi $c > 0$, cu $a \neq 1$ şi $c \neq 1$. În calculele care urmează vom folosi şi alte formule specifice logaritmilor.

$$= -\frac{1}{\ln 2}\Big(\ln\frac{\alpha^r}{\Gamma(r)}\underbrace{\int_0^{+\infty} p(x)\,dx}_{I_1} + (r-1)\underbrace{\int_0^{+\infty} p(x)\ln x\,dx}_{I_2} - \alpha\underbrace{\int_0^{+\infty} x\,p(x)\,dx}_{I_3}\Big).$$

Integrala I_1 are valoarea 1, deoarece $p(\,)$ este p.d.f. Integrala I_3, care reprezintă media distribuţiei Gamma, are valoarea $\frac{r}{\alpha}$ (vedeţi exerciţiul 31.c). Pentru a calcula integrala I_2, vom folosi următoarele egalităţi:

$$\frac{\partial}{\partial r}(e^{-\alpha x}x^{r-1}) = e^{-\alpha x}x^{r-1}\ln x \tag{75}$$

$$\frac{\partial}{\partial r}\ln\frac{\Gamma(r)}{\alpha^r} = \frac{\alpha^r}{\Gamma(r)}\cdot\frac{\partial}{\partial r}\Big(\frac{\Gamma(r)}{\alpha^r}\Big) \tag{76}$$

$$\psi(x) \overset{def.}{=} \frac{\Gamma'(r)}{\Gamma(r)} = \frac{\partial}{\partial r}\ln\Gamma(r) \text{ pentru orice } r>0 \text{ (funcţia digamma)} \tag{77}$$

$$\frac{\partial}{\partial r}\Gamma(r) = \frac{\partial}{\partial r}\int_0^{+\infty} t^{r-1}e^{-t}\,dt = \int_0^{+\infty}\frac{\partial}{\partial r}(t^{r-1}e^{-t})\,dt = \int_0^{+\infty} t^{r-1}e^{-t}\ln t\,dt. \tag{78}$$

Interschimbarea simbolilor de derivare şi respectiv de integrare de pe ultima linie este posibilă întrucât sunt îndeplinite condiţiile de aplicare pentru regula de „derivare sub semnul de integrală“ a lui Leibnitz.

Acum,

$$\begin{aligned}
I_2 &= \int_0^{+\infty}\frac{\alpha^r}{\Gamma(r)}x^{r-1}e^{-\alpha x}\ln x\,dx = \frac{\alpha^r}{\Gamma(r)}\int_0^{+\infty} x^{r-1}e^{-\alpha x}\ln x\,dx\\
&\overset{(75)}{=} \frac{\alpha^r}{\Gamma(r)}\int_0^{+\infty}\frac{\partial}{\partial r}(x^{r-1}e^{-\alpha x})\,dx\\
&= \frac{\alpha^r}{\Gamma(r)}\int_0^{+\infty}\frac{\partial}{\partial r}\Big(\frac{1}{\alpha^{r-1}}\cdot(\alpha x)^{r-1}e^{-\alpha x}\Big)\,dx = \frac{\alpha^r}{\Gamma(r)}\frac{\partial}{\partial r}\int_0^{+\infty}\frac{1}{\alpha^{r-1}}\cdot(\alpha x)^{r-1}e^{-\alpha x}\,dx\\
&= \frac{\alpha^r}{\Gamma(r)}\cdot\frac{\partial}{\partial r}\Big(\frac{1}{\alpha^{r-1}}\int_0^{+\infty}(\alpha x)^{r-1}e^{-\alpha x}\,dx\Big).
\end{aligned}$$

Egalitatea de pe penultima linie are loc întrucât putem aplica tot regula de „derivare sub semnul de integrală“ a lui Leibnitz (aşa cum s-a procedat şi la calcularea lui $\Gamma'(r) \overset{not.}{=} \frac{\partial}{\partial r}\Gamma(r)$, la egalitatea (78)).

Folosind schimbarea de variabilă $v=\alpha x$ care implică $dx=\frac{dv}{\alpha}$, obţinem:

$$\begin{aligned}
I_2 &= \frac{\alpha^r}{\Gamma(r)}\cdot\frac{\partial}{\partial r}\Big(\frac{1}{\alpha^r}\int_0^{+\infty} v^{r-1}e^{-v}\,dv\Big) = \frac{\alpha^r}{\Gamma(r)}\cdot\frac{\partial}{\partial r}\Big(\frac{\Gamma(r)}{\alpha^r}\Big) \overset{(76)}{=} \frac{\partial}{\partial r}\ln\frac{\Gamma(r)}{\alpha^r}\\
&= \frac{\partial}{\partial r}(\ln\Gamma(r) - \ln\alpha^r) = \frac{\partial}{\partial r}\ln\Gamma(r) - \frac{\partial}{\partial r}(r\,\ln\alpha) \overset{(77)}{=} \psi(r) - \ln\alpha.
\end{aligned}$$

Folosind toate aceste rezultate intermediare, în final vom obţine:

$$\begin{aligned}
H(X) &= -\frac{1}{\ln 2}\Big(\ln\frac{\alpha^r}{\Gamma(r)} + (r-1)\,(\psi(r)-\ln\alpha) - \cancel{\alpha}\frac{r}{\cancel{\alpha}}\Big)\\
&= -\frac{1}{\ln 2}\,(\cancel{r\ln\alpha} - \ln\Gamma(r) + (r-1)\,\psi(r) - \cancel{r\ln\alpha} + \ln\alpha - r)\\
&= \frac{1}{\ln 2}\,(r - \ln\alpha + \ln\Gamma(r) + (1-r)\,\psi(r)).
\end{aligned}$$

62. **(Redescoperirea definiţiei entropiei pornind de la un set de *proprietăţi dezirabile* ale ei)**

■ □ • ***CMU, 2005 fall, T. Mitchell, A. Moore, HW1, pr. 2.2***

Prin definiţie, *entropia* (în sens Shannon) a unei variabile aleatoare discrete X ale cărei valori sunt luate cu probabilităţile $p_1, p_2, \ldots, p_n$ este $H(X) = -\sum_i p_i \log p_i$. Însă legătura dintre această definiţie formală şi obiectivul avut în vedere — şi anume, acela de a exprima gradul de *incertitudine* cu care se produc valorile unei astfel de variabile aleatoare — nu este foarte intuitivă.

Scopul acestui exerciţiu este de a arăta că orice funcţie $\psi_n(p_1, \ldots, p_n)$ care satisface trei proprietăţi dezirabile (sau, axiome) pentru entropie este în mod necesar de forma $-K \sum_i p_i \log p_i$ unde K este o constantă reală pozitivă. Iată care sunt aceste *proprietăţi*:[129]

A1. Funcţia $\psi_n(p_1, \ldots, p_n)$ este continuă în fiecare din argumentele ei şi *simetrică*.
Din punct de vedere formal, în acest caz simetria se traduce prin egalitatea $\psi_n(p_1, \ldots, p_i, \ldots, p_j, \ldots, p_n) = \psi_n(p_1, \ldots, p_j, \ldots, p_i, \ldots, p_n)$ pentru orice $i \neq j$. Informal spus, dacă două dintre valorile care sunt luate de variabila aleatoare X (şi anume x_i şi x_j) îşi schimbă între ele probabilităţile (p_i şi respectiv p_j), valoarea entropiei lui X nu se schimbă.

A2. Funcţia $\psi_n(1/n, \ldots, 1/n)$ este strict *crescătoare* în raport cu n.
Altfel spus, dacă toate evenimentele sunt echiprobabile, atunci entropia creşte odată cu numărul de evenimente posibile.

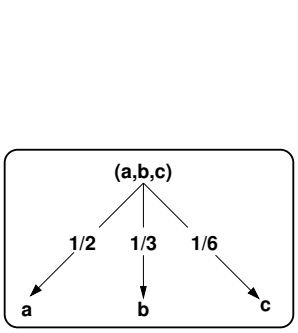
A3. Dacă faptul de a alege între mai multe evenimente posibile poate fi realizat prin mai multe alegeri succesive, atunci $\psi_n(p_1, \ldots, p_n)$ trebuie să se poată scrie ca o sumă ponderată a entropiilor calculate la fiecare stadiu / alegere.[130]
De *exemplu*, dacă evenimentele (a, b, c) se produc respectiv cu probabilităţile $(1/2, 1/3, 1/6)$, atunci acest fapt poate fi echivalat cu

- **a alege mai întâi cu probabilitate de $1/2$ între a şi (b, c),**
- **urmat de a alege între b şi c cu probabilităţile $2/3$ şi $1/3$ respectiv.**
  (A se vedea imaginile de mai jos, Encoding 1 şi Encoding 2.)

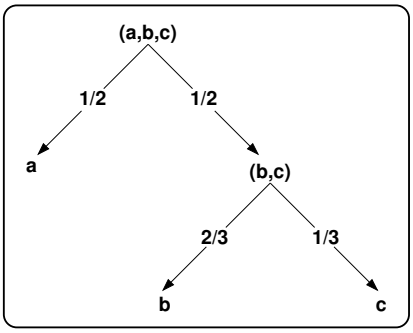
Din punct de vedere formal, proprietatea A3 impune ca, pe acest exemplu, $\psi_3(1/2, 1/3, 1/6)$ să fie egal cu $\psi_2(1/2, 1/2) + 1/2 \cdot \psi_2(2/3, 1/3)$.

[129] LC: Deşi nu se specifică în enunţul original al problemei, este necesar / natural să considerăm şi proprietatea următoare: [A0.] $\psi_1(1) = 0$ pentru că ψ_n este văzută că măsură a *dezordinii* / *incertitudinii*, iar în cazul particular $n = 1$ nu există niciun fel de dezordine / incertitudine.

[130] Această proprietate poate fi folosită pentru a elabora o [variantă de] rezolvare pentru problema 50 de la capitolul *Arbori de decizie*.



Encoding 1



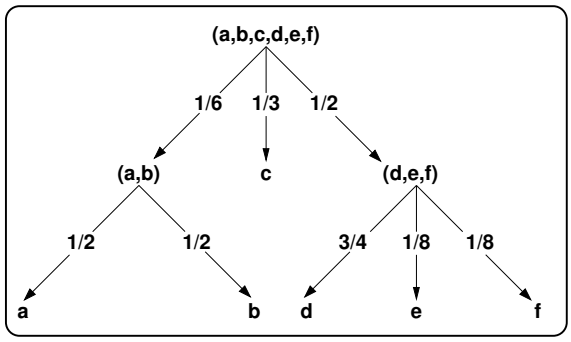
Encoding 2

Aşadar, în acest exerciţiu vi se cere să arătaţi că dacă o funcţie de n variabile $\psi_n(p_1, \ldots, p_n)$ satisface proprietăţile A1, A2 şi A3 de mai sus, atunci există $K \in \mathbb{R}^+$ astfel încât $\psi_n(p_1, \ldots, p_n) = -K \sum_i p_i \log p_i$ unde $K \in \mathbb{R}_+$ este o constantă.[131]

***Indicaţie*:**
Veţi face rezolvarea acestei probleme în mod gradual, parcurgând următoarele puncte (dintre care primele două puncte au rolul de a vă acomoda cu noţiunile din enunţ):

a. Arătaţi că $H(1/2, 1/3, 1/6) = H(1/2, 1/2) + \frac{1}{2} H(2/3, 1/3)$. Altfel spus, verificaţi faptul că definiţia clasică a entropiei, $H(X) = \sum_i p_i \log 1/p_i$, satisface proprietatea A3 pe *exemplul* care a fost dat mai sus.

b. Calculaţi entropia în cazul distribuţiei / „codificării" din figura alăturată, folosind din nou proprietatea A3.



Următoarele întrebări tratează cazul particular $A(n) \stackrel{not.}{=} \psi_n(1/n, 1/n, \ldots, 1/n)$.

c. Arătaţi că

$$A(s^m) = m\, A(s) \text{ pentru orice } s, m \in \mathbb{N}^*. \tag{79}$$

La punctele $d - g$ de mai jos, pentru orice număr $t \in \mathbb{N}^*$ (fixat), vom considera — pe lângă $n \in \mathbb{N}^*$, care a fost de fapt introdus atunci când am spus că aici ne ocupăm de $A(n)$ — numerele $s, m \in \mathbb{N}$, cu $s > 1$ astfel încât[132]

$$s^m \leq t^n \leq s^{m+1}. \tag{80}$$

***Precizare*: Odată ce am fixat t, s şi n, numărul m se alege astfel încât să aibă loc dubla inegalitate (80).**

[131] În demonstraţie, vom vedea, va rezulta $K = \frac{1}{\log s} \psi_s \left(\frac{1}{s}, \ldots, \frac{1}{s} \right)$ pentru un număr oarecare $s \in \mathbb{N}^* \backslash \{1\}$, fixat.

[132] LC: În idee, am putea să fixăm $s = 2$ şi apoi să alegem $m \in \mathbb{N}$ (în funcţie de t şi n) astfel încât relaţia (80) să fie satisfăcută. Însă raţionamentul următor nu depinde (în esenţă) de valoarea aleasă (şi fixată) pentru s.

d. Verificaţi că, prin logaritmarea[133] dublei inegalităţi (80) **şi apoi prin rearanjare, obţinem** $\frac{m}{n} \leq \frac{\log t}{\log s} \leq \frac{m}{n} + \frac{1}{n}$ **pentru** $s \neq 1$**, şi deci**

$$\left| \frac{m}{n} - \frac{\log t}{\log s} \right| \leq \frac{1}{n}. \tag{81}$$

e. Explicaţi de ce $A(s^m) \leq A(t^n) \leq A(s^{m+1})$**.**

f. Combinând ultima inegalitate de mai sus cu egalitatea (79)**, avem** $A(s^m) \leq A(t^n) \leq A(s^{m+1}) \Rightarrow m\,A(s) \leq n\,A(t) \leq (m+1)\,A(s)$**. Verificaţi că**

$$\left| \frac{m}{n} - \frac{A(t)}{A(s)} \right| \leq \frac{1}{n} \text{ pentru } s \neq 1. \tag{82}$$

g. Combinând inegalităţile (81) **şi** (82)**, arătaţi că**

$$\left| \frac{A(t)}{A(s)} - \frac{\log t}{\log s} \right| \leq \frac{2}{n} \text{ pentru } s \neq 1 \tag{83}$$

şi, în consecinţă

$$A(t) = K \log t \text{ cu } K > 0 \text{ (din cauza proprietăţii A2)}. \tag{84}$$

h. Arătaţi că rezultatul de mai sus ($A(t) = K \log t \Leftrightarrow \psi_t(1/t, \ldots, 1/t) = Kt\frac{1}{t}\log t$) **se generalizează uşor la cazul** $\psi_k(p_1, \ldots, p_k)$ **cu** $p_i \in \mathbb{Q}^+$ **pentru** $i = 1, \ldots, k$ **şi** $\sum_{i=1}^{k} p_i = 1$**:**

$$\psi_k(p_1, \ldots, p_k) = -K \sum_i p_i \log p_i.$$

i. În sfârşit, trataţi şi cazul $p_i \in \mathbb{R}^+,\ i = 1, \ldots, k$**, cu** $\sum_i p_i = 1$**.**

Răspuns:

a. Facem calculele:

$$H\left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right) = \frac{1}{2}\log 2 + \frac{1}{3}\log 3 + \frac{1}{6}\log 6 = \left(\frac{1}{2} + \frac{1}{6}\right)\log 2 + \left(\frac{1}{3} + \frac{1}{6}\right)\log 3 = \frac{2}{3} + \frac{1}{2}\log 3$$

$$H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{1}{2}H\left(\frac{2}{3}, \frac{1}{3}\right) = 1 + \frac{1}{2}\left(\frac{2}{3}\log\frac{3}{2} + \frac{1}{3}\log 3\right) = 1 + \frac{1}{2}\left(\log 3 - \frac{2}{3}\right) = \frac{2}{3} + \frac{1}{2}\log 3$$

şi rezultă că $H\left(\frac{1}{2}, \frac{1}{3}, \frac{1}{6}\right) = H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{1}{2}H\left(\frac{2}{3}, \frac{1}{3}\right)$**.**

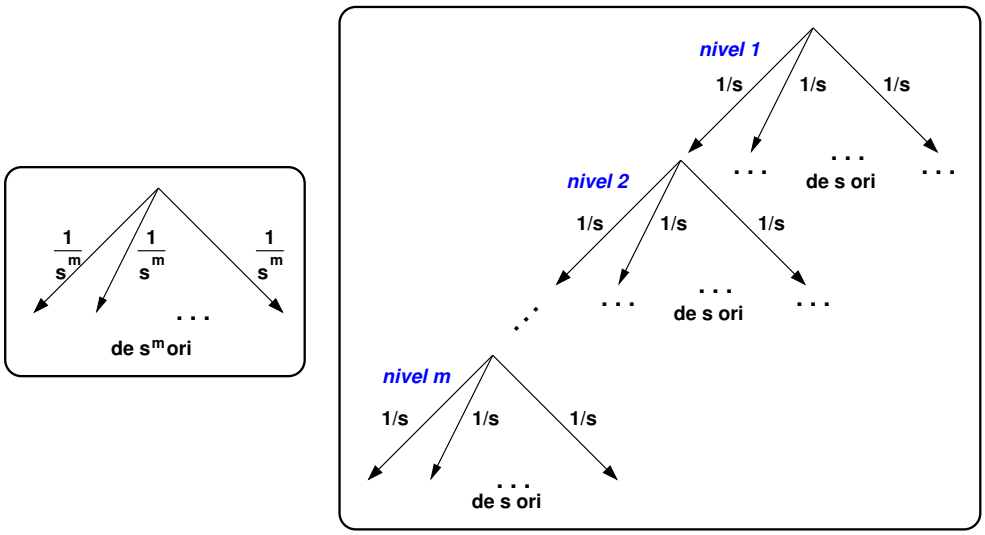
b. Folosind proprietatea A3, entropia „codificării“ din figura dată în enunţ este:

$$\begin{aligned} & H\left(\frac{1}{6}, \frac{1}{3}, \frac{1}{2}\right) + \frac{1}{6}H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{1}{2}H\left(\frac{3}{4}, \frac{1}{8}, \frac{1}{8}\right) \\ = \;& \frac{1}{6}\log 6 + \frac{1}{3}\log 3 + \frac{1}{2}\log 2 + \frac{1}{6} + \frac{1}{2}\left(\frac{3}{4}\log\frac{4}{3} + \frac{2}{8}\log 8\right) \\ = \;& \frac{1}{6}\log 2 + \frac{1}{6}\log 3 + \frac{1}{3}\log 3 + \frac{1}{2} + \frac{1}{6} + \frac{3}{8}(2 - \log 3) + \frac{3}{8} \end{aligned}$$

[133] Puteţi folosi ca bază a logaritmului orice număr supra-unitar, arbitrar ales, dar fixat.

$$= \frac{1}{6}+\frac{1}{2}+\frac{1}{6}+\frac{3}{4}+\frac{3}{8}+\left(\frac{1}{6}+\frac{1}{3}-\frac{3}{8}\right)\log 3$$
$$= \frac{47}{24}+\frac{1}{8}\log 3 = 1.958+0.125\log 3 = 2.156$$

c. Pentru calculul lui $A(s^m)$ se poate folosi atât o „codificare“ imediată cât şi una (des)compusă, ca în figura următoare:



Aplicând proprietatea A3 pe „codificarea“ din figura de mai sus, partea dreaptă, avem:

$$A(s^m) = A(s)+s\cdot\frac{1}{s}A(s)+s^2\cdot\frac{1}{s^2}A(s)+\ldots+s^{m-1}\cdot\frac{1}{s^{m-1}}A(s)$$
$$= \underbrace{A(s)+A(s)+A(s)+\ldots+A(s)}_{\text{de } m \text{ ori}} = mA(s)$$

d. Aplicând funcţia $\log$ **fiecărui termen al inegalităţii** $s^m \le t^n \le s^{m+1}$ **obţinem** $m\log s \le n\log t \le (m+1)\log s$**. Apoi, pentru** $s \ne 1$**, împărţind prin** $n\log s$**, rezultă:**

$$\frac{m}{n} \le \frac{\log t}{\log s} \le \frac{m}{n}+\frac{1}{n} \Rightarrow 0 \le \frac{\log t}{\log s}-\frac{m}{n} \le \frac{1}{n} \Rightarrow \left|\frac{\log t}{\log s}-\frac{m}{n}\right| \le \frac{1}{n}$$

e. Datorită proprietăţii A2 din enunţ, inegalitatea $s^m \le t^n \le s^{m+1}$ **implică**

$$\psi_{s^m}\left(\frac{1}{s^m},\ldots,\frac{1}{s^m}\right) \le \psi_{t^n}\left(\frac{1}{t^n},\ldots,\frac{1}{t^n}\right) \le \psi_{s^{m+1}}\left(\frac{1}{s^{m+1}},\ldots,\frac{1}{s^{m+1}}\right)$$

ceea ce reprezintă exact $A(s^m) \le A(t^n) \le A(s^{m+1})$**.**

f. Datorită proprietăţii (79), dubla inegalitate $A(s^m) \le A(t^n) \le A(s^{m+1})$ **devine** $m\,A(s) \le n\,A(t) \le (m+1)\,A(s)$**. Împărţind această inegalitate prin** $nA(s)$**, despre care se poate spune că este nenul pentru orice** $s \ne 1$**,**[134] **rezultă:**

$$\frac{m}{n} \le \frac{A(t)}{A(s)} \le \frac{m}{n}+\frac{1}{n} \Rightarrow 0 \le \frac{A(t)}{A(s)}-\frac{m}{n} \le \frac{1}{n} \Rightarrow \left|\frac{A(t)}{A(s)}-\frac{m}{n}\right| \le \frac{1}{n}$$

[134] Putem considera în mod natural $A(1) = 0$ (vedeţi proprietatea A0). Conform proprietăţii A2, urmează că $A(s) > A(1) = 0$ pentru orice $s > 1$.

g. Inegalităţile duble de mai jos rescriu convenabil proprietăţile (81) **şi** (82)**:**

$$-\frac{1}{n} \leq \frac{m}{n} - \frac{\log t}{\log s} \leq \frac{1}{n} \quad \text{şi} \quad -\frac{1}{n} \leq \frac{A(t)}{A(s)} - \frac{m}{n} \leq \frac{1}{n}$$

Însumându-le membru cu membru, rezultă

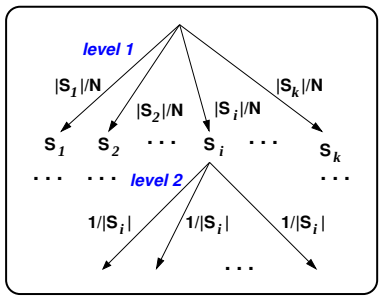
$$-\frac{2}{n} \leq \frac{A(t)}{A(s)} - \frac{\log t}{\log s} \leq \frac{2}{n} \Rightarrow \left|\frac{A(t)}{A(s)} - \frac{\log t}{\log s}\right| \leq \frac{2}{n}$$

Dacă trecem la limită inegalitatea $\left|\ \frac{A(t)}{A(s)} - \frac{\log t}{\log s}\ \right| \leq \frac{2}{n}$ **pentru** $n \to \infty$**, rezultă** $\left|\ \frac{A(t)}{A(s)} - \frac{\log t}{\log s}\ \right| \to 0$**, de unde avem** $\left|\ \frac{A(t)}{A(s)} - \frac{\log t}{\log s}\ \right| = 0$ **şi deci** $\frac{A(t)}{A(s)} = \frac{\log t}{\log s}$**. Rezultă că** $A(t) = \frac{A(s)}{\log s} \log t = K \log t$**. Evident, constanta** $K = \frac{A(s)}{\log s} = \frac{1}{\log s}\psi_s\left(\frac{1}{s}, \ldots, \frac{1}{s}\right)$ **nu depinde de** t**. Variind valorile lui** t**, rezultă că** $A(t) = K \log t$ **pentru orice** $t \in \mathbb{N}^*$ **astfel încât relaţia** (80) **are loc.**

***Observaţie*: Pentru** $t \in \mathbb{N}^*$ **şi** $p_1 = \ldots = p_t = \frac{1}{t}$**, este imediat că** $-\sum_i p_i \log p_i = \log t$**. Aşadar, rezultatul** $A(t) = K \log t$ **pe care tocmai l-am obţinut mai sus implică faptul că egalitatea** $\psi_t(p_1, \ldots, p_t) = -K \sum_i p_i \log p_i$ **are loc pentru cazul** $p_1 = \ldots = p_t = \frac{1}{t}$**.**

h. Considerăm o mulţime de N **evenimente echiprobabile. Fie** $\mathcal{P} = (S_1, S_2, \ldots, S_k)$ **o partiţionare a acestei mulţimi de evenimente. Notăm** $p_i = |\ S_i\ |/N$**.**

Propunem *codificarea* din figura alăturată. Vom alege mai întâi S_i**, una dintre submulţimile din partiţia** $\mathcal{P}$**, în funcţie de probabilităţile** $p_1, \ldots, p_k$**. Extragem apoi unul dintre elementele mulţimii** S_i**, cu probabilitate uniformă.**



Conform egalităţii (84)**, avem** $A(N) = K\ logN$**. Folosind proprietatea A3 şi *codificarea* în doi paşi propusă mai sus, rezultă că**

$$A(N) = \psi_k(p_1, \ldots, p_k) + \textstyle\sum_i p_i A(|\ S_i\ |).$$

Aşadar,

$$K \log N = \psi_k(p_1, \ldots, p_k) + K \textstyle\sum_i p_i \log |\ S_i\ |.$$

Prin urmare,

$$\begin{aligned} \psi_k(p_1, \ldots, p_k) &= K[\ \log N - \sum_i p_i \log |\ S_i\ |\ ] = K[\ (\log N) \sum_i p_i - \sum_i p_i \log |\ S_i\ |\ ] \\ &= -K \sum_i p_i \log \frac{|\ S_i\ |}{N} = -K \sum_i p_i \log p_i. \end{aligned}$$

i.[135] **Considerăm pentru fiecare** $p_i,\ i = 1, \ldots, k-1$ **un şir de aproximări succesive** $\{q_i^{(n)}\}_{n \geq 1} \subset \mathbb{Q}^+$ **astfel încât** $\lim_{n \to \infty} q_i^{(n)} = p_i$**. Definim de asemenea şirul**

[135] Rezolvare redactată de Ştefan Bălăucă, student, FII, 2020.

$\{q_k^{(n)}\}_{n\geq 1} \subset \mathbb{Q}$ astfel: $q_k^{(n)} = 1 - \sum_{i=1}^{k-1} q_i^{(n)}$. Se pot alege aproximările $\{q_i^{(n)}\}$ astfel încât cel puţin de la un loc încolo $q_k^{(n)} \in \mathbb{Q}^+$.[136]

Trecând la limită în relaţia de definiţie $q_k^{(n)} = 1 - \sum_{i=1}^{k-1} q_i^{(n)}$, obţinem:

$$\lim q_k^{(n)} = 1 - \sum_{i=1}^{k-1} \lim q_i^{(n)} = 1 - \sum_{i=1}^{k-1} p_i = p_k.$$

Aşadar, avem $\lim_{n\to\infty} q_i^{(n)} = p_i,\ \forall i = 1, \ldots, k$. Deoarece $q_i^{(n)} \in \mathbb{Q}^+$, pentru $i = 1, \ldots, k$, cu $k \geq 1$, este valabilă relaţia

$$\psi_k(q_1^{(n)}, \ldots, q_k^{(n)}) = -K \sum_i q_i^{(n)} \log q_i^{(n)},\ \forall n \geq 1.$$

Trecând la limită în această relaţie şi ţinând cont de continuitatea funcţiei log**, vom avea:**

$$\begin{aligned} \lim_{n\to\infty} \psi_k(q_1^{(n)}, \ldots, q_k^{(n)}) &= \lim_{n\to\infty} \Big(-K \sum_i q_i^{(n)} \log q_i^{(n)} \Big) \\ &= -K \sum_i \lim_{n\to\infty} q_i^{(n)} \log q_i^{(n)} = -K \sum_i p_i \log p_i. \end{aligned} \tag{85}$$

Datorită proprietăţii A1, funcţia ψ_k este continuă în fiecare dintre argumentele ei, astfel că

$$\lim_{n\to\infty} \psi_k(q_1^{(n)}, \ldots, q_k^{(n)}) = \psi_k(\lim_{n\to\infty} q_1^{(n)}, \ldots, \lim_{n\to\infty} q_k^{(n)}) = \psi_k(p_1, \ldots, p_k). \tag{86}$$

Din relaţiile (85) **şi** (86) **putem concluziona că**

$$\psi_k(p_1, ..., p_k) = -K \sum_i p_i \log p_i, \text{ pentru } \forall p_i \in [0, 1], \text{ a.î. } \sum_i p_i = 1.$$

Astfel se atinge obiectivul propus, acela de a arăta că $\psi_k(p_1, \ldots, p_k)$ este de forma $-K \sum_i p_i \log p_i$, pentru situaţiile în care probabilităţile p_i sunt numere reale.

O *observaţie* finală: Egalitatea $A(t) = K \log t$ implică $A(1) = K \log 1 = 0$, ceea ce este în concordanţă cu proprietatea A0. În baza proprietăţii A2 va rezulta $\psi_n(p_1, \ldots, p_n) \geq 0$ pentru orice $n \in \mathbb{N}^*$ şi orice $p_1, \ldots, p_n \in [0, 1]$ astfel încât $\sum_i p_i = 1$.

63. **(Cross-entropie: definiţie, o proprietate (nenegativitatea) şi un exemplu simplu de calculare a valorii cross-entropiei)**

■ □ • ○ ***CMU, 2011 spring, Roni Rosenfeld, HW2, pr. 3.c***

Cross-entropia **(sau entropia încrucişată) a două distribuţii p şi q, desemnată prin *CH(p,q)*, reprezintă numărul mediu de biţi necesari pentru a codifica un eveniment dintr-o mulţime oarecare de posibilităţi, atunci când schema de cod[ific]are folosită se bazează pe o distribuţie de probabilitate dată q, în loc**

[136] De exemplu, putem defini $q_i^{(n)}$ = numărul format luând în considerare doar primele n zecimale ale lui p_i.

să se bazeze pe distribuţia „adevărată“ p. În cazul în care distribuţiile p şi q sunt discrete, această noţiune se defineşte formal astfel:[137]

$$CH(p,q) = -\sum_x p(x)\ \log q(x).$$

În cazul distribuţiilor continue, definiţia se obţine / construieşte prin analogie:

$$CH(p,q) = -\int_X p(x)\ \log q(x)\, dx.$$

***Observaţie*: În formulele de mai sus, se consideră implicit baza 2 pentru funcţia logaritm.**

Cross-entropia — ca şi entropia relativă; vedeţi problema 65 —, spre deosebire de entropia comună, nu este simetrică în raport cu cele două distribuţii / argumente: în general, $CH(p,q) \neq CH(q,p)$.

a. Poate oare cross-entropia să ia valori negative? Faceţi o demonstraţie sau daţi un contraexemplu.

b. În multe experimente, pentru a stabili calitatea diferitelor ipoteze / modele, se procedează la evaluarea / compararea lor pe un set de date. Să presupunem că, urmărind să faci predicţia *funcţiei de probabilitate* asociate unei anumite *variabile aleatoare* care are 7 valori posibile, ai obţinut (printr-un procedeu oarecare) două *modele* diferite, iar *distribuţiile de probabilitate* prezise de către aceste două modele sunt respectiv:

$$q_1 = \left(\frac{1}{10}, \frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{1}{5}, \frac{1}{20}, \frac{1}{20}\right) \text{ şi } q_2 = \left(\frac{1}{20}, \frac{1}{10}, \frac{3}{20}, \frac{7}{20}, \frac{1}{5}, \frac{1}{10}, \frac{1}{20}\right).$$

Să zicem că pentru evaluare foloseşti un set de date caracterizat de următoarea distribuţie *empirică*:

$$p_{empiric} = \left(\frac{1}{20}, \frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{1}{5}, \frac{1}{10}, \frac{1}{20}\right).$$

Calculează cross-entropiile $CH(p_{empiric}, q_1)$ şi $CH(p_{empiric}, q_2)$.
Care dintre aceste două modele va conduce la o cross-entropie mai mică? Putem oare garanta că acest model este într-adevăr [cel] mai bun? Explică / justifică răspunsul [pe care l-ai] dat.

Răspuns:

a. Nu, cross-entropia nu poate lua valori negative. Iată cum demonstrăm:
Ştim că pentru orice funcţii de probabilitate p şi q şi pentru orice x (care aparţine domeniului de valori al unei variabile aleatoare care are o astfel de distribuţie de probabilitate), valorile $p(x)$ şi $q(x)$ satisfac inegalităţile $0 \leq p(x) \leq 1$ and $0 \leq q(x) \leq 1$. Inegalitatea $q(x) \leq 1$ implică faptul că $\log q(x) \leq 0$. Din $0 \leq p(x)$ şi $-\log q(x) \geq 0$, rezultă că $0 \leq -p(x) \log q(x)$. În consecinţă, suma tuturor acestor termeni va fi de asemenea mai mare sau egală cu 0, deci cross-entropia nu poate fi niciodată negativă.

[137] Pentru exemple de folosire a cross-entropiei în învăţarea automată, vedeţi problema 15 de la capitolul *Reţele neuronale artificale*, precum şi problema 2 de la capitolul *Schema algoritmică EM*.

***Observaţie importantă*: Spre deosebire de entropie (vedeţi problema 143), cross-entropia nu este mărginită superior. Ea poate creşte la infinit; vedeţi cazul când pentru o anumită valoare x sunt adevărate simultan relaţiile $p(x) \neq 0$ şi $q(x) = 0$.**[138]

b. Facem calculele, folosind formula cross-entropiei:

$$\begin{aligned}
&CH(p_{empiric}, q_1) = \\
&\quad -\left(\frac{1}{20}\log_2\frac{1}{10} + \frac{1}{10}\log_2\frac{1}{10} + \frac{1}{5}\log_2\frac{1}{5} + \frac{3}{10}\log_2\frac{3}{10} + \frac{1}{5}\log_2\frac{1}{5} + \frac{1}{10}\log_2\frac{1}{20}\right. \\
&\quad \left. + \frac{1}{20}\log_2\frac{1}{20}\right) = \frac{3}{20}\log_2 10 + \frac{2}{5}\log_2 5 + \frac{3}{10}\log_2\frac{10}{3} + \frac{3}{20}\log_2 20 = \\
&= \frac{3}{20}\log_2 2\cdot 5 + \frac{2}{5}\log_2 5 + \frac{3}{10}\log_2\frac{2\cdot 5}{3} + \frac{3}{20}\log_2 2^2\cdot 5 \\
&= \left(\frac{3}{20} + \frac{3}{10} + 2\cdot\frac{3}{20}\right) + \left(\frac{3}{20} + \frac{2}{5} + \frac{3}{10} + \frac{3}{20}\right)\log_2 5 - \frac{3}{10}\log_2 3 \\
&= \frac{3}{4} + \log_2 5 - \frac{3}{10}\log_2 3 = 2.596439345 \text{ biţi}
\end{aligned}$$

$$\begin{aligned}
&CH(p_{empiric}, q_2) = \\
&\quad -\left(\frac{1}{20}\log_2\frac{1}{20} + \frac{1}{10}\log_2\frac{1}{10} + \frac{1}{5}\log_2\frac{3}{20} + \frac{3}{10}\log_2\frac{7}{20} + \frac{1}{5}\log_2\frac{1}{5} + \frac{1}{10}\log_2\frac{1}{10}\right. \\
&\quad \left. + \frac{1}{20}\log_2\frac{1}{20}\right) = \\
&= \frac{1}{10}\log_2 20 + \frac{1}{5}\log_2 10 + \frac{1}{5}\log_2\frac{20}{3} + \frac{3}{10}\log_2\frac{20}{7} + \frac{1}{5}\log_2 5 \\
&= \frac{1}{10}\log_2 2^2\cdot 5 + \frac{1}{5}\log_2 2\cdot 5 + \frac{1}{5}\log_2\frac{2^2\cdot 5}{3} + \frac{3}{10}\log_2\frac{2^2\cdot 5}{7} + \frac{1}{5}\log_2 5 \\
&= \left(2\cdot\frac{1}{10} + \frac{1}{5} + 2\cdot\frac{1}{5} + 2\cdot\frac{3}{10}\right) + \left(\frac{1}{10} + 3\cdot\frac{1}{5} + \frac{3}{10}\right)\log_2 5 - \frac{1}{5}\log_2 3 - \frac{3}{10}\log_2 7 \\
&= \frac{7}{5} + \log_2 5 - \frac{1}{5}\log_2 3 - \frac{3}{10}\log_2 7 = 2.562729118 \text{ biţi.}
\end{aligned}$$

Se observă că distribuţia $p_{empiric}$ are o cross-entropie mai mică în [raport cu] modelul q_2. Este deci rezonabil să afirmăm că alegerea modelului q_2 este mai bună.

Totuşi, nu putem garanta că acest model este întotdeauna cel mai bun, fiindcă aici lucrăm cu o distribuţie „empirică", iar distribuţia „adevărată" nu neapărat se reflectă în mod complet / perfect în această distribuţie empirică.

De obicei, *bias-ul de eşantionare* (engl., sampling bias), precum şi *insuficienţa datelor de antrenament* vor contribui la lărgirea „spaţiului" care diferenţiază distribuţia adevărată de distribuţia empirică. Prin urmare, în practică, atunci când concepem un [astfel de] experiment de evaluare a mai multor distribuţii probabiliste, trebuie să avem permanent în minte faptul acesta şi, dacă este posibil, să folosim tehnici care reduc / minimizează aceste riscuri.

[138] Mai precis, $\lim_{q(x)\to+0}(-p(x)\cdot\log_2 q(x)) = -p(x)(-\infty) = +\infty$ dacă $p(x) \neq 0$.

64. **(Inegalitatea lui Gibbs:[139] un caz particular; comparaţie între valorile entropiei şi ale cross-entropiei)**

■ □ ***Liviu Ciortuz, 2012, după www.en.wikipedia.org şi CMU, T. Mitchell, A. Moore, 2003 fall, HW1, pr. 1.2***

Fie $P = \{p_1, \dots p_n\}$ o distribuţie de probabilitate discretă.

a. Arătaţi că pentru orice distribuţie de probabilitate $Q = \{q_1, \dots q_n\}$ are loc inegalitatea:

$$-\sum_{i=1}^{n} p_i \log_2 p_i \leq -\sum_{i=1}^{n} p_i \log_2 q_i$$

Altfel spus, $H(P) \leq CH(P,Q)$, unde $H(P)$ este entropia distribuţiei P, iar $CH(P,Q)$ este *cross-entropia* lui P în raport cu Q.

***Observaţie*: Pentru un exemplu de utilizare a acestei inegalităţi, cunoscută sub numele de *inegalitatea lui Gibbs*, în învăţarea automată, puteţi vedea problema 2 de la capitolul *Schema algoritmică EM*.**

b. Arătaţi că în formula de mai sus egalitatea are loc dacă şi numai dacă $p_i = q_i$ pentru $i = 1, \dots, n$.

***Observaţie*: În formula din enunţ, în locul bazei 2 pentru logaritm poate fi folosită orice bază supraunitară.**

***Indicaţie*: Puteţi folosi inegalitatea lui Jensen (vedeţi problema 79).**

Răspuns:

a. Vom folosi inegalitatea lui Jensen în versiune probabilistă (cf. problemei 79.c): dacă φ este o funcţie convexă, iar X este variabilă aleatoare de distribuţie p, atunci $E[\varphi(X)] \geq \varphi(E[X])$. Particularizând această inegalitate pentru $\varphi = -\log_a$, cu $a > 1$, şi înlocuind X cu $\frac{q(x)}{p(x)}$, vom avea:

$$E\left[-\log_a \frac{q(x)}{p(x)}\right] \geq -\log_a E\left[\frac{q(x)}{p(x)}\right] \Leftrightarrow -\sum_i p(x_i) \log_a \frac{q(x_i)}{p(x_i)} \geq -\log_a \Big(\underbrace{\sum_i \cancel{p(x_i)} \cdot \frac{q(x_i)}{\cancel{p(x_i)}}}_{1} \Big)$$

$$\Leftrightarrow -\sum_i p(x_i) \log_a \frac{q(x_i)}{p(x_i)} \geq 0 \Leftrightarrow -\sum_i p(x_i) \log_a q(x_i) \geq -\sum_i p(x_i) \log_a p(x_i)$$

$$\overset{not.}{\Leftrightarrow} -\sum_i p_i \log_a q_i \geq -\sum_i p_i \log_a p_i.$$

b. Vom folosi *Observaţia* 1 de la problema 79, care afirmă următoarele: în cazul unei funcţii *strict* convexe inegalitatea lui Jensen se realizează egalitatea dacă şi numai dacă $x_1 = \dots = x_n$. În cazul nostru, rezultă că în inegalitatea lui Gibbs se realizează egalitatea dacă şi numai dacă $\frac{p_i}{q_i} = \alpha$ (constant), pentru orice i. Cum $\sum_i p_i = \sum_i q_i = 1$, rezultă imediat că $\alpha = 1$, deci distribuţiile probabiliste P şi Q sunt identice.

[139] Josiah Willard Gibbs (1839-1903), om de ştiinţă american, a avut contribuţii teoretice majore în fizică, chimie şi matematică. A fost unul dintre părinţii fondatori ai domeniului mecanicii statistice. A inventat teoria modernă a calculului vectorial. A predat la universitatea Yale în perioada 1871-1903.

65. **(Entropia relativă: definiţie şi proprietăţi elementare; exprimarea câştigului de informaţie cu ajutorul entropiei relative)**

■ □ • ***prelucrare de Liviu Ciortuz, după CMU, 2007 fall, Carlos Guestrin, HW1, pr. 1.2***

Entropia relativă **sau** ***divergenţa Kullback-Leibler*** **(KL) a unei distribuţii p în raport cu o altă distribuţie q — ambele distribuţii fiind discrete — se defineşte astfel:**[140]

$$KL(p||q) \stackrel{def.}{=} -\sum_{x \in X} p(x) \log \frac{q(x)}{p(x)}$$

Observaţie**: În formula de mai sus, se consideră implicit baza 2 pentru funcţia logaritm.**

Din perspectiva teoriei informaţiei, divergenţa KL specifică numărul de *biţi adiţionali* care sunt necesari în medie pentru a transmite valorile variabilei X atunci când presupunem că aceste valori sunt distribuite conform distribuţiei („model“) q, dar în realitate ele urmează o altă distribuţie, p.[141]

Observaţie**: Ţinând cont de definiţia cross-entropiei (vedeţi pr. 63), putem scrie:**

$$KL\,(p||q) = CH(p, q) - H(p).$$

a. Demonstraţi inegalitatea $KL(p||q) \geq 0$ şi apoi arătaţi că egalitatea are loc dacă şi numai dacă $p = q$.[142]

Indicaţie**:**
Pentru a demonstra aceste două proprietăţi puteţi folosi fie *inegalitatea lui Gibbs* (vedeţi problema 64) fie *inegalitatea lui Jensen*:[143]
Dacă $f : \mathbb{R} \to \mathbb{R}$ este o *funcţie convexă*, atunci pentru orice $a_i \geq 0,\ i = 1, \ldots, n$ cu $\sum_i a_i = 1$ şi orice $x_i \in \mathbb{R},\ i = 1, \ldots, n$, avem $f(\sum_i a_i x_i) \leq \sum_i a_i f(x_i)$. Dacă f este strict convexă, atunci egalitatea are loc doar dacă $x_1 = \ldots = x_n$. Pentru funcţii concave, semnul inegalităţii este $\geq$.

b. Câştigul de informaţie poate fi (re)definit ca fiind entropia relativă dintre distribuţia comună observată a lui X şi Y pe de o parte, şi produsul distribuţiilor marginale p_X şi p_Y de cealaltă parte:

---

[140] În învăţarea automată, divergenţa KL este folosită de exemplu la fundamentarea schemei algoritmice EM. Vedeţi problemele 1.b şi 2 de la capitolul *Schema algoritmică EM*. Mai general, maximizarea verosimilităţii datelor — în vederea estimării parametrilor distribuţiilor probabiliste — poate fi văzută ca fiind echivalentă cu minimizarea divergenţei KL (vedeţi problema 149).

[141] *Atenţie*: Divergenţa KL nu este o măsură de *distanţă* între două distribuţii probabiliste, fiindcă în general ea nu este simetrică ($KL(p||q) \neq KL(q||p)$) şi nici nu satisface inegalitatea triunghiului. Pentru „simetrizare“, se consideră $M(p, q) = \frac{1}{2}(p + q)$, apoi se defineşte funcţia $JSD(p||q) = \frac{1}{2} KL(p||M) + \frac{1}{2} KL(q||M)$, care se numeşte *divergenţa Jensen-Shannon*. În sfârşit, se poate arăta că $\sqrt{JSD(p||q)}$ defineşte o măsură de distanţă (metrică), adică este nenegativă, simetrică, implică identitatea indiscernabililor şi satisface inegalitatea triunghiului; ea este numită *distanţa Jensen-Shannon*.

Variaţia informaţiei, definită prin

$$VI(X, Y) \stackrel{def.}{=} H(X, Y) - IG(X, Y) = H(X) + H(Y) - 2IG(X, Y) = H(X \mid Y) + H(Y \mid X),$$

este de asemenea o măsură de distanţă.

[142] Mai general, $KL(p||q)$ este cu atât mai mică cu cât „asemănarea“ dintre distribuţiile p şi q este mai mare.

[143] Vedeţi problema 79.

$$IG(X,Y) \stackrel{def.}{=} KL(p_{X,Y} \,||\, (p_X\, p_Y)) = -\sum_x \sum_y p_{X,Y}(x,y) \log \left(\frac{p_X(x) p_Y(y)}{p_{X,Y}(x,y)} \right)$$
$$\stackrel{not.}{=} -\sum_x \sum_y p(x,y) \log \left(\frac{p(x)p(y)}{p(x,y)} \right)$$

Arătaţi că această nouă definiţie a câştigului de informaţie este echivalentă cu definiţia dată anterior (vedeţi problema 55). Cu alte cuvinte, demonstraţi egalitatea

$$KL(p_{X,Y} \,||\, (p_X\, p_Y)) = H[X] - H[X \mid Y].$$

***Observaţie*: Din noua definiţie introdusă mai sus pentru câştigul de informaţie, rezultă imediat că**

$$\begin{aligned} IG(X,Y) &= \sum_y p(y) \sum_x p(x \mid y) \log \frac{p(x \mid y)}{p(x)} = \sum_y p(y) KL(p_{X|Y} \,||\, p_X) \\ &= E_Y[KL(p_{X|Y} \,||\, p_X)] \end{aligned}$$

ceea ce înseamnă că $IG(X,Y)$ poate fi văzută ca o medie (în raport cu distribuţia lui Y) a divergenţei KL dintre distribuţia condiţională a lui X în raport cu Y pe de o parte, şi distribuţia lui X pe de altă parte.

c. O consecinţă imediată a punctelor a şi b este faptul că $IG(X,Y) \geq 0$ (deci $H(X) \geq H(X|Y)$ şi $H(Y) \geq H(Y|X)$) pentru orice variabile aleatoare discrete X şi Y. Folosind din nou rezultatele de la punctele a şi b, arătaţi că $IG(X,Y) = 0$ dacă şi numai dacă X şi Y sunt independente.[144]

Răspuns:

a. Dacă în inegalitatea lui Gibbs (vedeţi problema 64.a)

$$-\sum_{i=1}^{n} p_i \log_2 p_i \leq -\sum_{i=1}^{n} p_i \log_2 q_i$$

se trece termenul din partea stângă în partea dreaptă, obţinem

$$0 \leq \sum_{i=1}^{n} p_i \log_2 p_i - \sum_{i=1}^{n} p_i \log_2 q_i \Leftrightarrow 0 \leq -\sum_{i=1}^{n} p_i \log_2 \frac{q_i}{p_i}.$$

Expresia din partea dreaptă a ultimei inegalităţi de mai sus este chiar $KL(p||q)$. Aşadar, am obţinut $KL(p||q) \geq 0$. De asemenea, conform problemei 64.b, rezultă că în această expresie egalitatea se realizează dacă şi numai dacă distribuţiile probabiliste p şi q sunt identice.

Alternativ, vom dovedi inegalitatea $KL(p||q) \geq 0$ în manieră directă folosind inegalitatea lui Jensen, în expresia căreia vom înlocui f cu funcţia convexă $-\log_2$, pe a_i cu $p(x_i)$ şi pe x_i cu $\frac{q(x_i)}{p(x_i)}$. (Pentru convenienţă, în cele ce urmează vom renunţa la indicele variabilei x.) Vom avea:

$$KL\,(p \,||\, q) \stackrel{def.}{=} -\sum_x p(x) \log \frac{q(x)}{p(x)}$$

[144] Demonstrarea implicaţiei $X \perp Y \Rightarrow IG(X;Y) = 0$ în manieră directă (adică, fără a apela la divergenţa KL) face obiectul problemei 146.

$$\overset{Jensen}{\geq} \quad -\log\Big(\sum_x p(x)\frac{q(x)}{p(x)}\Big) = -\log\big(\underbrace{\sum_x q(x)}_{1}\big) = -\log 1 = 0.$$

Aşadar, $KL(p \,||\, q) \geq 0$, oricare ar fi distribuţiile (discrete) p şi q.

Vom demonstra acum, tot în manieră directă, echivalenţa $KL(p||q) = 0 \Leftrightarrow p = q$. Egalitatea $p(x) = q(x)$ implică $\frac{q(x)}{p(x)} = 1$, deci $\log\frac{q(x)}{p(x)} = 0$ pentru orice x, de unde rezultă imediat $KL(p||q) = 0$.
Pentru a demonstra implicaţia inversă, se ţine cont că în inegalitatea lui Jensen, în cazul funcţiilor strict convexe (cum este $-\log_2$) are loc egalitatea doar în cazul în care $x_i = x_j$ pentru orice i şi j. În cazul de faţă, această condiţie se traduce prin faptul că raportul $\frac{q(x)}{p(x)}$ este acelaşi (α) pentru orice valoare a lui x. Ţinând cont că $\sum_x p(x) = 1$ şi $\sum_x q(x) = \sum_x p(x)\frac{q(x)}{p(x)} = 1$, rezultă că $\alpha = \frac{q(x)}{p(x)} = 1$, deci $p(x) = q(x)$ pentru orice x, ceea ce înseamnă că distribuţiile p şi q sunt identice.

b. Vom folosi regula de înmulţire, conform căreia $p(x, y) = p(x \mid y)p(y)$:

$$\begin{aligned}
KL\left(p_{X,Y} \,||\, (p_X\, p_Y)\right) &\overset{def.\ KL}{=} -\sum_x\sum_y p(x,y)\log\left(\frac{p(x)p(y)}{p(x,y)}\right)\\
&= -\sum_x\sum_y p(x,y)\log\left(\frac{p(x)}{p(x\mid y)}\right) = -\sum_x\sum_y p(x,y)[\log p(x) - \log p(x\mid y)]\\
&= -\sum_x\sum_y p(x,y)\log p(x) - \Big(-\sum_x\sum_y p(x,y)\log p(x\mid y)\Big)\\
&\overset{pr.\ 55.b}{=} -\sum_x \log p(x)\underbrace{\sum_y p(x,y)}_{=p(x)} - H[X\mid Y] = \sum_x p(x)\log p(x) - H[X\mid Y]\\
&= H[X] - H[X\mid Y] = IG(X,Y)
\end{aligned}$$

c. Conform punctului b, egalitatea $IG(X, Y) = 0$ este echivalentă cu egalitatea $KL\left(p_{X,Y} \,||\, p_X\, p_Y\right) = 0$. Conform punctului a, această a doua relaţie este adevărată dacă şi numai dacă distribuţiile $p_{X,Y}$ şi $p_X\, p_Y$ sunt identice, or aceasta este exact definiţia independenţei variabilelor X şi Y.

66. **(Entropia văzută ca o *funcţională* în raport cu p (p.d.f. sau p.m.f); calculul *derivatei funcţionale* a entropiei în raport cu p)**

□ ***Liviu Ciortuz, 2021, pornind de la CMU, 2013 spring, A. Smola, B. Poczos, HW2, pr. 1***
şi https://en.wikipedia.org/wiki/Functional_derivative

***Comentariu*:**

Ştim că entropia unei variabile aleatoare X continue se defineşte după cum urmează: dacă funcţia de densitate de probabilitate (p.d.f.) a lui X este p,

atunci $H(X) = \int p(x)\, \log_2 p(x)$.[145] Abstractizând, noţiunea de entropie H poate fi interpretată [şi] ca o funcţie care ia ca argument funcţia p. În matematică, astfel de „funcţii de funcţii“ — acestea din urmă îndeplinind anumite proprietăţi, lăsate nespecificate aici — se numesc *funcţionale*.

Ca şi pentru funcţiile de variabilă reală, şi în cazul funcţionalelor se poate defini o noţiune de tip derivată, numită *derivată funcţională*.

Fie F o funcţională definită peste mulţimea de funcţii M. Considerăm f şi h din M, iar $t \in \mathbb{R}$. Noţiunea de *derivată funcţională* se defineşte pornind de la calculul următoarei expresii, care reprezintă valoarea unei *derivate numerice* — adică, definită în sensul clasic — în punctul $t = 0$:

$$\Big[\frac{d}{dt}F(f+th)\Big]_{t=0} \stackrel{def.}{=} \Big[\lim_{t\to 0}\frac{F(f+th)-F(f)}{t}\Big]_{t=0} \tag{87}$$

Dacă această expresie se poate calcula ca integrala unui produs de două funcţii, dintre care una este h, adică

$$\int \Big[\frac{\delta F(f)}{\delta f}(x)\Big] h(x)\, dx, \tag{88}$$

atunci funcţia $\frac{\delta F(f)}{\delta f}(x)$ este considerată *derivata funcţională* a lui F în raport cu funcţia f.

Derivata funcţională $\frac{\delta F(f)}{\delta f}$ poate fi interpretată ca fiind gradientul lui F în „punctul“ f. Integrala $\int \frac{\delta F(f)}{\delta f}(x)\, h(x)\, dx$ poate fi interpretată ca fiind *derivata direcţională* a lui F în „punctul“ f pe direcţia lui h (care poate fi văzut ca un vector infinit, ale cărui componente sunt valorile $h(x)$, pentru $x \in \mathbb{R}$).

***Exemplu*:[146] Considerăm $F(f) \stackrel{def.}{=} \int f^2(x)\, dx$. Atunci,**

$$F(f+th) = \int (f+th)^2(x)\, dx = \int f^2(x)\, dx + 2t\int f(x)\, h(x)\, dx + t^2\int h^2(x)\, dx$$
$$\Rightarrow \frac{dF(f+th)}{dt} = \lim_{t\to 0}\frac{F(f+th)-F(f)}{t} = 2\int f(x)\, h(x)\, dx + 2t\int h^2(x)\, dx$$
$$\Rightarrow \frac{dF(f+th)}{dt}\Big|_{t=0} = 2\int f(x)\, h(x)\, dx = \int 2f(x)\, h(x)\, dx$$
$$\Rightarrow \frac{\delta F(f)}{\delta f}(x) = 2f(x).$$

Arătaţi că derivata funcţională a entropiei (H) unei variabile aleatoare continue X având p.d.f. p este

$$\frac{\delta H(p)}{\delta p}(x) = -\frac{1}{\ln 2}[1 + \ln p(x)].$$

Răspuns:

[145] Dacă X este variabilă aleatoare discretă, iar funcţia sa masă de probabilitate (p.m.f.) este p, atunci $H(X) = -\sum_{x\in Val,(X)} p(x)\, \log_2 p(x)$.

[146] Acest exemplu a fost preluat din cartea *A modern approach to functional integration*, 2010, de J. Klauder, pag. 38-39.

Pentru simplitate, vom considera funcţionala $F(p) \stackrel{def.}{=} \int p(x)\ln p(x)dx$**. Aşadar,** $H(p) = -\frac{1}{\ln 2}F(p)$**, de unde va rezulta** $\frac{\delta H(p)}{\delta p}(x) = -\frac{1}{\ln 2}\frac{\delta F(p)}{\delta p}(x)$**. Vom arăta că** $\frac{\delta F(p)}{\delta p}(x) = 1 + \ln p(x)$**. Pentru aceasta, vom porni de la definiţia** (87)**, mai precis de la expresia din dreapta egalităţii respective şi vom explicita mai întâi** $F(p+th)$**:**

$$\begin{aligned} F(p+th) &= \int (p+th)(x)\cdot \ln(p+th)(x)\,dx = \int (p(x)+th(x))\cdot \ln(p(x)+th(x))\,dx \\ &= \int p(x)\,\ln(p(x)+th(x))\,dx + t\int h(x)\,\ln(p(x)+th(x))\,dx. \end{aligned}$$

Prin urmare,

$$\begin{aligned} &\frac{dF(p+th)}{dt} \stackrel{def.}{=} \lim_{t\to 0}\frac{F(p+th)-F(p)}{t} \\ &= \lim_{t\to 0}\Big[\frac{1}{t}\Big\{\int p(x)\,\ln(p(x)+th(x))\,dx - \int p(x)\ln p(x)\,dx\Big\}\Big] \\ &\qquad + \int h(x)\,\ln(p(x)+th(x))\,dx\Big] \\ &= \lim_{t\to 0}\Big[\frac{1}{t}\Big\{\int p(x)(\ln(p(x)+th(x)) - \ln p(x))\,dx\Big\}\Big] + \\ &\quad \lim_{t\to 0}\int h(x)\,\ln(p(x)+th(x))\,dx. \end{aligned} \tag{89}$$

Vom calcula acum aceste (ultime) două limite. Vom începe cu cea de-a doua limită, fiindcă este mai simplă.

$$\begin{aligned} \lim_{t\to 0}\int h(x)\,\ln(p(x)+th(x))\,dx &= \int \lim_{t\to 0} h(x)\,\ln(p(x)+th(x))\,dx &(90)\\ &= \int h(x)\ln p(x)\,dx. &(91)\end{aligned}$$

Observaţi că am putut interschimba simbolii $\lim_{t\to 0}$ **şi** $\int$**, pentru că integrala aceasta este definită, deci reprezintă de fapt tot o limită. (Aceste două limite se calculează folosind variabile diferite,** t **şi** x**.)**
Revenim acum la prima limită de mai sus:

$$\begin{aligned} &\lim_{t\to 0}\Big[\frac{1}{t}\Big\{\int p(x)(\ln(p(x)+th(x)) - \ln p(x))\,dx\Big\}\Big] \\ &= \lim_{t\to 0}\int \frac{1}{t}\,p(x)(\ln(p(x)+th(x)) - \ln p(x))\,dx \\ &= \int \lim_{t\to 0}\frac{1}{t}\,p(x)(\ln(p(x)+th(x)) - \ln p(x))\,dx &(92)\\ &= \int \Big[\frac{d}{dt}p(x)\ln(p(x)+th(x))\Big]_{t=0}\,dx &(93)\\ &= \int \Big[p(x)\cdot\frac{1}{p(x)+th(x)}\cdot h(x)\Big]_{t=0}\,dx = \int h(x)\,dx. &(94)\end{aligned}$$

Observaţi că la egalitatea (92) **am folosit aceeaşi justificare ca şi pentru egalitatea** (90)**. La egalitatea** (93) **am folosit definiţia derivatei [numerice] pentru funcţia** $p(x)\ln(p(x)+th(x))$ **în raport cu** t**, calculată în punctul** $t = 0$**.**

Acum, înlocuind rezultatele (91) **şi** (94) **în relaţia** (89)**, obţinem:**

$$\frac{dF(p+th)}{dt} = \int h(x)dx + \int h(x)\ln p(x)\,dx = \int (1+\ln p(x))h(x)\,dx, \tag{95}$$

rezultat care nu depinde de t **(deci nu mai este nevoie să aplicăm operatorul** $[\,]_{t=0}$**). Aşadar, din relaţia** (95) **rezultă că** $\frac{\delta F(p)}{\delta p}(x) = 1+\ln p(x)$ **şi, în consecinţă,** $\frac{\delta H(p)}{\delta p}(x) = -\frac{1}{\ln 2}(1+\ln p(x))$**.**

***Observaţie*: Vom demonstra că formula din enunţ este valabilă şi pentru cazul discret.**[147] **Aşadar, vom folosi definiţia** $H[p(x)] = -\sum_x p(x)\log_2 p(x)$**, unde** x **ia valori într-o mulţime discretă. De data aceasta, este mai convenabil să pornim de la membrul stâng al egalităţii** (87)**, care a fost dată pentru a putea introduce noţiunea de derivată funcţională.**

$$\begin{aligned}
\Big[\frac{d}{dt}H[p(x)+th(x)]\Big]_{t=0} &= -\Big[\frac{d}{dt}\sum_x (p(x)+th(x))\log_2(p(x)+th(x))\Big]_{t=0} \\
&= -\frac{1}{\ln 2}\Big[\frac{d}{dt}\sum_x (p(x)+th(x))\ln(p(x)+th(x))\Big]_{t=0} \\
&= -\frac{1}{\ln 2}\Big[\sum_x \frac{d}{dt}(p(x)+th(x))\ln(p(x)+th(x))\Big]_{t=0} \\
&= -\frac{1}{\ln 2}\Big[\sum_x [h(x)\ln(p(x)+th(x)) + \cancel{(p(x)+th(x))}\frac{h(x)}{\cancel{p(x)+th(x)}}]\Big]_{t=0} \\
&= -\frac{1}{\ln 2}\sum_x [h(x)\ln p(x) + h(x)] = -\frac{1}{\ln 2}\sum_x [1+\ln p(x)]h(x) \\
&= \sum_x -\frac{1}{\ln 2}[1+\ln p(x)]h(x).
\end{aligned} \tag{96}$$

În concluzie, $\frac{\delta H(p)}{\delta p}(x) = -\frac{1}{\ln 2}(1+\ln p(x))$**.**[148] **Rezultatul acesta este identic (d.p.v. sintactic) cu cel din cazul continuu.**

[147] Prin urmare, vom arăta [şi] cum se poate defini — şi calcula — derivata funcţională în raport cu o funcţie care nu este continuă.

[148] Observaţi că semnului de integrare $\int dx$ din relaţia de definiţie (88) îi corespunde în relaţia (96) simbolul de sumare $\sum_x$, aşa cum este natural la trecerea de la cazul continuu la cazul discret. (Într-o versiune viitoare a culegerii, vom face probabil legătura în sens invers, adică vom prezenta definiţia noţiunii de derivată funcţională mai întâi pentru cazul discret şi apoi pentru cazul continuu.)

0.1.6 Funcţii-nucleu

67. **(Găsirea mapării care corespunde unei funcţii-nucleu polinomiale particulare)**

University of Utah, 2008 fall, Hal Daumé III, HW5, pr. 1.2

Considerăm $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ o funcţie polinomială de gradul al doilea: $K(x, z) = (1 + x \cdot z)^2$. Scrieţi forma detaliată a acestei funcţii pentru cazul $3-$dimensional (adică $x = (x_1, x_2, x_3)$ şi $z = (z_1, z_2, z_3)$).

Arătaţi că funcţia K satisface condiţiile din *definiţia funcţiei-nucleu*, adică există o funcţie $\phi : \mathbb{R}^3 \to \mathbb{R}^n$ cu n ales convenabil, astfel încât $K(x, z) = \phi(x) \cdot \phi(z)$.

Răspuns:

Produsul scalar $x \cdot z$ se scrie desfăşurat $x_1z_1 + x_2z_2 + x_3z_3$, deci

$$\begin{aligned} K(x,z) &= 1 + x_1^2z_1^2 + x_2^2z_2^2 + x_3^2z_3^2 + 2x_1z_1 + 2x_2z_2 + 2x_3z_3 + \\ &\quad + 2x_1x_2z_1z_2 + 2x_1x_3z_1z_3 + 2x_2x_3z_2z_3. \end{aligned}$$

Conform *definiţiei* funcţiei-nucleu, va trebui să obţinem forma analitică a mapării $\phi : \mathbb{R}^3 \to \mathbb{R}^n$ (unde n va fi stabilit îndată) astfel încât $K(x, z) = \phi(x) \cdot \phi(z)$. Din expresia lui $K(x, z)$ de mai sus este evident că dacă definim

$$\phi(x) = (1, x_1^2, x_2^2, x_3^2, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_3, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3)$$

atunci vom avea $K(x, z) = \phi(x) \cdot \phi(z)$. Din forma analitică a lui ϕ rezultă că $n = 10$.

68. **(Matrice-nucleu: teorema lui Mercer — condiţii necesare [şi suficiente] pentru ca o funcţie de două variabile din $\mathbb{R}^d$ să fie funcţie-nucleu; o proprietate de tip „construcţie“ de noi funcţii-nucleu)**

• ○ ***CMU, 2008 fall, Eric Xing, final exam, pr. 2***

a. Demonstraţi că orice funcţie-nucleu K este simetrică, adică pentru orice elemente x_1 şi x_2 din domeniul de definiţie al lui K, are loc egalitatea $K(x_1, x_2) = K(x_2, x_1)$.

b. Considerăm X un set de instanţe $x_1, \ldots, x_m \in \mathbb{R}^d$ şi o funcţie-nucleu $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Fie A matricea de tip $m \times m$ având elementele $A(i, j) \stackrel{def.}{=} K(x_i, x_j)$, pentru $i, j = 1, \ldots, m$. Matricea A astfel definită se numeşte *matricea-nucleu* asociată funcţiei-nucleu K şi datasetului X.[149]

Prin *definiţie*, se spune că o matrice M de tip $m \times m$ este *pozitiv semidefinită* (sau: nenegativ definită) dacă pentru orice vector m-dimensional f (văzut ca vector-coloană), are loc inegalitatea $f^\top Mf \geq 0$.[150]

[149] Matricea-nucleu mai este numită şi *matrice Gram*. Noţiunea de matrice Gram este definită de unii autori independent de noţiunea de funcţie-nucleu: date fiind elementele $z_1, \ldots, z_m \in \mathbb{R}^m$, matricea Gram este matricea constituită din produsele scalare $z_i \cdot z_j$, cu $i, j = \overline{1, m}$.

[150] M este *matrice pozitiv definită* dacă $f^\top Mf > 0$ pentru orice f diferit de vectorul nul. Similar se definesc şi noţiunile de matrice negativ semidefinită şi matrice negativ definită.

Demonstraţi că orice matrice-nucleu este pozitiv semidefinită.

***Indicaţie*: Dacă socotiţi că este mai uşor pentru dumneavoastră, demonstraţi această afirmaţie în cazul particular al funcţiei-nucleu $K(x_i, x_j) = (1 + x_i \cdot x_j)^2$, cu x_i şi x_j din $\mathbb{R}^2$, văzuţi ca vectori-coloană.**

***Observaţie*: Punctele a şi b de mai sus arată că *orice matrice-nucleu este simetrică şi pozitiv semidefinită*. Se poate arăta — dar nu demonstrăm aici efectiv — că este adevărată şi *reciproca* acestei afirmaţii, şi anume:**
Dată fiind funcţia $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, dacă pentru orice număr $m \in \mathbb{N}$ şi pentru orice elemente $x_1, \ldots, x_m \in \mathbb{R}^d$, matricea A de tip $m \times m$, având elementele $A(i,j) \stackrel{def.}{=} K(x_i, x_j)$ pentru $i, j \in \{1, \ldots, m\}$ este simetrică şi pozitiv semidefinită, atunci K este funcţie-nucleu.[151]

c. Folosind *observaţia* de mai sus, demonstraţi că *suma a două funcţii-nucleu* oarecare K_1 şi K_2 *este de asemenea o funcţie-nucleu*.

Răspuns:

a. Dată fiind o funcţie-nucleu oarecare $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, conform definiţiei există o altă funcţie $\phi : \mathbb{R}^d \to \mathbb{R}^n$ (numită în general [*funcţie de*] *mapare* a trăsăturilor) astfel încât $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$, pentru orice x_i şi x_j.[152] **Datorită comutativităţii produsului scalar vom avea $K(x_i, x_j) \stackrel{def.}{=} \phi(x_i) \cdot \phi(x_j) = \phi(x_j) \cdot \phi(x_i) \stackrel{def.}{=} K(x_j, x_i)$ pentru orice x_i şi x_j. Aşadar, funcţia-nucleu K este simetrică.**

b. Scriind vectorul-coloană f şi matricea-nucleu A pe componente,

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \ldots \\ f_m \end{bmatrix} \text{ şi } A = \begin{bmatrix} \phi(x_1) \cdot \phi(x_1) & \phi(x_1) \cdot \phi(x_2) & \ldots & \phi(x_1) \cdot \phi(x_m) \\ \phi(x_2) \cdot \phi(x_1) & \phi(x_2) \cdot \phi(x_2) & \ldots & \phi(x_2) \cdot \phi(x_m) \\ \ldots & \ldots & \ldots & \ldots \\ \phi(x_m) \cdot \phi(x_1) & \phi(x_m) \cdot \phi(x_2) & \ldots & \phi(x_m) \cdot \phi(x_m) \end{bmatrix},$$

vom avea:

$$\begin{aligned} f^\top A f &= (f^\top A) f = \left[\left(\sum_{i=1}^m f_i \, \phi(x_i) \right) \cdot \phi(x_1), \ldots, \left(\sum_{i=1}^m f_i \, \phi(x_i) \right) \cdot \phi(x_m) \right] \begin{bmatrix} f_1 \\ f_2 \\ \ldots \\ f_m \end{bmatrix} \\ &= \left(\sum_{i=1}^m f_i \, \phi(x_i) \right) \cdot (f_1 \, \phi(x_1)) + \ldots + \left(\sum_{i=1}^m f_i \, \phi(x_i) \right) \cdot (f_m \, \phi(x_m)) \\ &= \left(\sum_{i=1}^m f_i \, \phi(x_i) \right) \cdot \left(\sum_{j=1}^m f_j \, \phi(x_j) \right) = \left(\sum_{i=1}^m f_i \, \phi(x_i) \right)^2 \\ &\stackrel{def.}{=} || \sum_{i=1}^m f_i \, \phi(x_i) ||^2 \geq 0. \end{aligned}$$

[151] Afirmaţia directă şi reciproca ei constituie împreună *teorema lui Mercer* (în variantă discretă): $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ este funcţie-nucleu dacă şi numai dacă pentru $\forall m \in \mathbb{N}$ şi $\forall x_1, \ldots, x_m \in \mathbb{R}^d$, matricea A de tip $m \times m$ având elementele $A(i,j) \stackrel{def.}{=} K(x_i, x_j)$ este simetrică şi pozitiv semidefinită (adică, pentru orice $f \in \mathbb{R}^m$ văzut ca vector-coloană, avem $f^\top A f \geq 0$).

Acest rezultat a fost publicat de către matematicianul englez James Mercer (1883-1932) în anul 1909 în articolul *Functions of positive and negative type and their connection with the theory of integral equations*, în revista Philosophical Transactions of the Royal Society, London, series A (Mathematical, Physical and Engineering Sciences).

[152] În general, $n > d$ sau chiar $n >> d$, ultima notaţie însemnând că n este mult mai mare decât d.

***Observaţie*: Dacă vom nota cu Φ matricea care are coloanele $\phi(x_1), \ldots, \phi(x_m)$ — cu un uşor abuz de notaţie, putem scrie $\Phi = [\phi(x_1), \ldots, \phi(x_m)]$ —, matricea-nucleu A se va scrie simplu ca un produs, $A = \Phi^\top \Phi$, iar calculul de mai sus se poate exprima foarte succint astfel:**

$$f^\top A f = f^\top (\Phi^\top \Phi) f = (f^\top \Phi^\top)(\Phi f) = (\Phi f)^\top (\Phi f) = ||\Phi f||^2 \geq 0.$$

În acest calcul am ţinut cont de asociativitatea înmulţirii matricelor, precum şi de faptul că pentru orice două matrice înlănţuite[153] operaţia de transpunere are proprietatea $(XY)^\top = Y^\top X^\top$.

c. Fie K_1 şi K_2 două funcţii-nucleu oarecare definite pe $\mathbb{R}^d \times \mathbb{R}^d$, iar A_1 şi A_2 matricele-nucleu corespunzătoare lui K_1 şi respectiv K_2 pentru un set de elemente $x_1, \ldots, x_m$ arbitrar alese din $\mathbb{R}^d$.

Conform punctelor a şi b, rezultă că matricele-nucleu A_1 şi A_2 sunt simetrice şi pozitiv semidefinite. Conform *observaţiei* din enunţ, funcţia $K_1 + K_2$ (căreia, vom vedea mai jos, îi corespunde matricea-nucleu $A_1 + A_2$) este funcţie-nucleu dacă $A_1 + A_2$ este matrice simetrică şi $f^\top (A_1 + A_2) f \geq 0$ pentru orice vector-coloană $f \in \mathbb{R}^m$.

Faptul că $A_1 + A_2$ este matrice simetrică rezultă imediat din ipoteză şi din punctul a (aplicat pe rând funcţiilor nucleu K_1 şi K_2).

Întrucât înmulţirea matricelor este distributivă faţă de adunare, rezultă:

$$f^\top (A_1 + A_2) f = f^\top (A_1 f + A_2 f) = f^\top A_1 f + f^\top A_2 f \geq 0$$

fiindcă $f^\top A_1 f \geq 0$ şi $f^\top A_2 f \geq 0$, inegalităţi care derivă imediat din ipoteză, conform rezultatului de la punctul b.[154]

În concluzie, matricea $A_1 + A_2$ fiind simetrică şi pozitiv semidefinită, rezultă că funcţia $K_1 + K_2$ este funcţie-nucleu.

***Consecinţă*: Pentru orice număr finit de funcţii-nucleu $K_1, K_2, \ldots, K_l$, suma lor este de asemenea o funcţie-nucleu. (Demonstraţia se face imediat prin inducţie matematică.)**

***Observaţie*: Proprietatea demonstrată la punctul c priveşte „calitatea" de funcţie-nucleu a sumei $K_1 + K_2$. Implicit, este dovedită existenţa unei „mapări" ϕ care corespunde lui $K_1 + K_2$. Însă demonstraţia aceasta nu este constructivă, adică nu se dă efectiv expresia funcţiei ϕ. Totuşi, este foarte uşor de verificat următorul fapt: dacă luăm ca definiţie pentru $\phi(x)$ vectorul obţinut prin concatenarea vectorilor $\phi_1(x)$ şi $\phi_2(x)$ — unde ϕ_1 şi ϕ_2 sunt „mapări" corespunzătoare nucleelor K_1 şi respectiv K_2 —, se verifică egalitatea:**

$$\phi(x) \cdot \phi(x') = \underbrace{K_1(x, x')}_{\phi_1(x) \cdot \phi_1(x')} + \underbrace{K_2(x, x')}_{\phi_2(x) \cdot \phi_2(x')} \stackrel{def.}{=} (K_1 + K_2)(x, x') \text{ pentru orice } x \text{ şi } x' \in \mathbb{R}^d.$$

[153] Adică, de dimensiune $n_1 \times n_2$ şi respectiv $n_2 \times n_3$.

[154] Ambele egalităţi de mai sus au fost deduse pe baza proprietăţii de distributivitate a înmulţirii matricelor faţă de operaţia de adunare.

69. **(Funcţii-nucleu: [alte] câteva proprietăţi de „construcţie“)**

CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, final exam, pr. 7.a.1
MIT, 2009 fall, Tommi Jaakkola, lecture 3
Stanford, 2008 fall, Andrew Ng, HW2, pr. 1.ce

Fie

K_1 **şi** K_2 **două funcţii-nucleu definite pe** $\mathbb{R}^d \times \mathbb{R}^d$,

$\Phi_1 : \mathbb{R}^d \to \mathbb{R}^n$ **şi** $\Phi_2 : \mathbb{R}^d \to \mathbb{R}^n$ **funcţii de mapare a trăsăturilor, care corespund funcţiilor-nucleu** K_1 **şi respectiv** K_2**.**

a. Arătaţi cum se pot defini funcţiile de mapare ale următoarelor funcţii-nucleu, în raport cu Φ_1 **şi** Φ_2**:**

i. $K(x,z) = cK_1(x,z)$**, unde** c **este o constantă oarecare, pozitivă.**
ii. $K(x,z) = f(x)K_1(x,z)f(z)$**, unde** f **este o funcţie cu valori în** $\mathbb{R}$**.**
(Se observă uşor că punctul *ii* este o generalizare a punctului *i*.)
iii. $K(x,z) = K_1(x,z)\,K_2(x,z)$**.**

b. Arătaţi că funcţia $K(x,z) = p(K_1(x,z))$**, unde** p **este un polinom cu coeficienţi pozitivi este de asemenea funcţie-nucleu.**

Răspuns:

a. Se verifică imediat că, definind $\Phi(x) = \sqrt{c}\,\Phi_1(x)$ **în cazul *i*, şi** $\Phi(x) = f(x)\Phi_1(x)$ **în cazul *ii*, este satisfăcută relaţia din definiţia funcţiei-nucleu:** $K(x,z) = \Phi(x) \cdot \Phi(z)$ **pentru orice** x **şi** z **din** $\mathbb{R}^d$**.**

Pentru cazul *iii*, soluţia nu mai este chiar atât de simplă. Dată fiind o instanţă x **arbitrar aleasă din** $\mathbb{R}^d$**, vom nota componentele vectorului** $\Phi_1(x)$ **cu** $\phi_{11}(x)$**,** $\phi_{12}(x)$**, ...,** $\phi_{1n_1}(x)$ **şi, similar, componentele vectorului** $\Phi_2(x)$ **cu** $\phi_{21}(x)$**,** $\phi_{22}(x)$**, ...,** $\phi_{2n_2}(x)$**. Vom defini** $\Phi(x)$ **ca fiind un vector cu** $n_1 n_2$ **componente, fiecare componentă fiind un produs (de numere reale) de tipul** $\phi_{1i}(x)\phi_{2j}(x)$**:**

$$\begin{aligned}\Phi(x) \stackrel{def.}{=} [\ & \phi_{11}(x)\phi_{21}(x),\ \phi_{11}(x)\phi_{22}(x),\ \ldots,\ \phi_{11}(x)\phi_{2n_2}(x),\\ & \phi_{12}(x)\phi_{21}(x),\ \phi_{12}(x)\phi_{22}(x),\ \ldots,\ \phi_{12}(x)\phi_{2n_2}(x),\\ & \ldots\\ & \phi_{1n_1}(x)\phi_{21}(x),\ \phi_{1n_1}(x)\phi_{22}(x),\ \ldots,\ \phi_{1n_1}(x)\phi_{2n_2}(x)\ ]\end{aligned}$$

Considerând acum perechea de elemente x **şi** z **arbitrar alese din** $\mathbb{R}^d$**, va rezulta:**

$$\begin{aligned}\Phi(x)\cdot\Phi(z) &= \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\phi_{1i}(x)\phi_{2j}(x)\phi_{1i}(z)\phi_{2j}(z)\\ &= \Big(\sum_{i=1}^{n_1}\phi_{1i}(x)\phi_{1i}(z)\Big)\Big(\sum_{j=1}^{n_2}\phi_{2j}(x)\phi_{2j}(z)\Big)\\ &= (\Phi_1(x)\cdot\Phi_1(z))(\Phi_2(x)\cdot\Phi_2(z))\\ &= K_1(x,z)K_2(x,z) = (K_1K_2)(x,z).\end{aligned}$$

La cea de-a doua egalitate de mai sus am ţinut cont de distributivitatea înmulţirii numerelor reale faţă de adunare.

b. Faptul că funcţia compusă $p(K_1(x,z))$**, unde** p **este un polinom cu coeficienţi pozitivi este funcţie-nucleu decurge imediat din relaţiile *i* şi *iii* de mai sus şi punctul** c **de la problema 68.**

70. **(Funcţii-nucleu: [încă] o proprietate de „construcţie“)**

■ □ ● · ***Stanford, 2009 fall, Andrew Ng, practice midterm exam, pr. 3.a***

Considerăm K o funcţie-nucleu definită pe $\mathbb{R}^d \times \mathbb{R}^d$. Arătaţi că funcţia compusă K_e definită prin relaţia $K_e(x,z) = e^{K(x,z)}$ este de asemenea funcţie-nucleu.

***Sugestie*: Puteţi folosi dezvoltarea sub formă de *serie Taylor* a lui e^x:**

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots + \frac{x^n}{n!} + \ldots,$$

precum şi faptul că pentru orice şir de numere nenegative $\{a_n\}_{n\in\mathbb{N}}$, dacă există $a \overset{not.}{=} \lim_{n\to\infty} a_n$, atunci $a \geq 0$.

***Consecinţă*: Pentru orice număr real $a > 1$, avem $\ln a > 0$, deci funcţia $(\ln a)K(x,z)$ este şi ea funcţie-nucleu (vedeţi problema 69.a.i). Apoi,**

$$a^{K(x,z)} = (e^{\ln a})^{K(x,z)} = e^{(\ln a)K(x,z)}.$$

Prin urmare, va rezulta că şi funcţia $a^{K(x,z)}$ este funcţie-nucleu, pentru orice $a > 1$.

Răspuns:

***Observaţie*: În rezolvarea pe care o dăm mai jos, vom demonstra proprietatea din enunţ nu în forma dată (generală, adică $e^{K(x,z)}$), ci într-un caz particular care este simplu, dar semnificativ. În mod concret, vom arăta că funcţia $e^{x\cdot z}$ de variabile x şi z din $\mathbb{R}^d$ este funcţie-nucleu. (Evident, funcţia $x\cdot z$ este funcţie-nucleu.) După aceea, a arăta că $e^{K(x,z)}$ este funcţie-nucleu revine la a urma exact firul demonstraţiei de mai jos — înlocuind $x \cdot z$ cu $K(x,z) = \phi(x) \cdot \phi(x)$ şi respectiv $x_i \cdot x_j$ cu $K(x_i, x_j)$ —, fiindcă justificările în baza cărora se asigură validitatea raţionamentului în cazul simplu pe care l-am ales mai sus rămân valabile şi pentru cazul general.**

Vom demonstra că $e^{x\cdot z}$ este funcţie-nucleu nu în mod direct — adică, dovedind existenţa unei funcţii de mapare ϕ astfel încât $e^{x\cdot z} = \phi(x) \cdot \phi(z)$ —, ci folosind teorema de „caracterizare“ a lui Mercer (a se vedea problema 68.ab şi nota de subsol 151).

Conform acestei teoreme, pentru a dovedi că $e^{x\cdot z}$ este funcţie-nucleu este suficient să arătăm că pentru orice $m \in \mathbb{N}$ şi pentru orice elemente $x_1, \ldots, x_m \in \mathbb{R}^d$, matricea G (Gram), care este de tip $m \times m$ şi are elementul generic de forma $G(i,j) \overset{not.}{=} e^{x_i \cdot x_j}$ pentru $i, j = \overline{1,m}$, este simetrică şi pozitiv semidefinită.

Fie deci un număr natural $m \in \mathbb{N}$, fixat, precum şi numerele $x_1, \ldots, x_m \in \mathbb{R}^d$, de asemenea fixate. Folosind dezvoltarea lui $e^{x_i \cdot x_j}$ ca serie Taylor, avem:

$$e^{x_i \cdot x_j} = \sum_{n=0}^{\infty} \frac{(x_i \cdot x_j)^n}{n!} \overset{def.}{=} \lim_{n\to\infty} \left(\sum_{l=0}^{n} \frac{(x_i \cdot x_j)^l}{l!} \right)$$

Cu aceasta, devine evident că putem scrie matricea G ca fiind limita unui şir de matrice, $\{G_n\}_{n\in\mathbb{N}}$, elementul generic al matricei G_n fiind

$$G_n(i,j) \overset{not.}{=} \sum_{l=0}^{n} \frac{(x_i \cdot x_j)^l}{l!} \text{ pentru } i, j = \overline{1,m}.$$

Matricea G_n astfel definită este simetrică, datorită proprietăţii de simetrie a produsului scalar. Evident, în urma trecerii la limită ($n \to \infty$) pe componente, se păstrează simetria matricei.[155] **Aşadar, rezultă că matricea G este simetrică.**

Matricea G_n este şi pozitiv semidefinită. Justificarea provine din faptul că funcţia $x \cdot z$ de variabile x şi z din $\mathbb{R}^d$ este funcţie-nucleu şi din aplicarea proprietăţilor de „construcţie" de noi nuclee: K_1+K_2, K_1K_2 şi cK_1 cu $c > 0$ sunt funcţii-nucleu dacă K_1 şi K_2 sunt funcţii-nucleu (a se vedea problemele 68.c şi 69).[156] **Rămâne de arătat că prin trecere la limită ($G_n \to G$) se păstrează proprietatea de „pozitiv semidefinire" a matricelor.**

Fie $f \in \mathbb{R}^m$, văzut ca vector-coloană. Urmează:

$$f^\top G f = f^\top (\lim_{n\to\infty} G_n) f = \lim_{n\to\infty} (\underbrace{f^\top G_n f}_{\geq 0}) \geq 0.$$

Cea de-a doua egalitate de mai sus are loc datorită proprietăţii de liniaritate a limitei de şiruri: limita combinaţiei liniare a $m \times m$ şiruri este combinaţia liniară a limitelor şirurilor respective. Inegalitatea $\lim_{n\to\infty}(f^\top G_n f) \geq 0$ are loc fiindcă $f^\top G_n f \geq 0$ pentru orice n (datorită faptului că matricea G_n este pozitiv semidefinită, după cum am arătat mai sus) şi datorită proprietăţii de păstrare a pozitivităţii prin trecerea la limită (vedeţi cea de-a doua *sugestie* din enunţ).

Aşadar, matricea G este simetrică şi pozitiv semidefinită. În concluzie, funcţia $e^{x\cdot z}$ este funcţie-nucleu.

***Observaţie*: Din demonstraţia de mai sus rezultă imediat că printr-un raţionament similar se poate dovedi următoarea *proprietate de tip „construcţie"*: dacă avem un şir de funcţii-nucleu $K_1, K_2, \ldots, K_n, \ldots$ definite pe $\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ şi există $\lim_{n\to\infty} K_n \overset{not.}{=} K$, atunci K este şi ea funcţie-nucleu.**[157]

71. (Calculul distanţei euclidiene dintre imaginile a două instanţe în spaţiul de trăsături, cu ajutorul funcţiei-nucleu)

University of Utah, 2008 spring, Hal Daumé III, HW1C, pr. 3

Presupunem că pentru o funcţie $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ există o funcţie $\phi : \mathbb{R}^d \to \mathbb{R}^n$ care satisface proprietatea $K(x, z) = \phi(x) \cdot \phi(z)$ pentru orice $x, z \in \mathbb{R}^d$. (Operatorul $\cdot$ reprezintă produsul scalar în $\mathbb{R}^n$.) Fie distanţa euclidiană în spaţiul n-dimensional,

$$|| \phi(x) - \phi(z) || \overset{def.}{=} \sqrt{\sum_{i=1}^{n} (\phi(x)_i - \phi(z)_i)^2}.$$

Arătaţi cum se poate calcula această distanţă folosind doar funcţia K, adică fără a apela (în mod explicit) la valorile funcţiei ϕ.

[155] Altfel spus, $G_n(i,j) = G_n(j,i) \to G(i,j) = G(j,i)$, unde $G(i,j) = \lim_{n\to\infty} G_n(i,j)$ şi $G(j,i) = \lim_{n\to\infty} G_n(j,i)$.

[156] Se ţine cont de relaţia de recurenţă între matricele-nucleu $G_n = G_{n-1} + \frac{1}{n!}\left[(x_i \cdot x_j)^n\right]_{i,j=\overline{1,m}}$, cu $G_1 = I$, matricea identitate de ordin d. Este imediat că $(x \cdot z)^n$, unde $n \geq 2$, este funcţie-nucleu, întrucât $x \cdot z$ este funcţie-nucleu.

[157] Cf. https://en.wikipedia.org/wiki/Positive-definite_kernel, accesat la data 30.03.2022.

***Observaţie*: Rezultatul acesta este util în cazul în care se aplică algoritmul de clasificare k-NN sau algoritmi de clusterizare (precum K-means) după ce s-a făcut „maparea“ datelor de antrenament într-un nou „spaţiu de trăsături“.**[158]

Răspuns:

$$||\phi(x)-\phi(z)|| = \sqrt{\sum_{i=1}^{n}(\phi(x)_i-\phi(z)_i)^2} = \sqrt{\sum_{i=1}^{n}(\phi(x)_i^2+\phi(z)_i^2-2\phi(x)_i\phi(z)_i)}$$

$$= \sqrt{\sum_{i=1}^{n}\phi(x)_i\phi(x)_i+\sum_{i=1}^{n}\phi(z)_i\phi(z)_i-2\sum_{i=1}^{n}\phi(x)_i\phi(z)_i}$$

$$= \sqrt{\phi(x)\cdot\phi(x)+\phi(z)\cdot\phi(z)-2\phi(x)\cdot\phi(z)} = \sqrt{K(x,x)+K(z,z)-2K(x,z)}.$$

72. **(Un exemplu de funcţie-nucleu care exprimă / măsoară similaritatea dintre două imagini oarecare)**

• ○ ***CMU, 2014 fall, E. Xing+B. Poczos, HW2, pr. 3.1***

Considerăm X o mulţime formată din imagini dreptunghiulare de dimensiuni arbitrare, în care fiecare pixel este reprezentat sub forma unui număr întreg din mulţimea $\{0,\ldots,255\}$. Fie $k_1 : X\times X\to\mathbb{R}$ o funcţie de similaritate a imaginilor, definită astfel: $k_1(x,x')$ = numărul de zone pătratice de pixeli (engl., pixel patches), de dimensiune 16×16, care apar atât în imaginea x cât şi în imaginea x'.[159]

a. Demonstraţi că funcţia k_1 este funcţie-nucleu.

b. Demonstraţi că funcţia de similaritate definită mai jos *nu* este funcţie-nucleu.

$$k_2(x,x') = \begin{cases} 1 & \text{dacă } k_1(x,x')\geq 1, \text{ adică, există cel puţin un “patch” (zonă pătratică) comun(ă) pentru } x \text{ şi } x' \\ 0 & \text{în caz contrar.}\end{cases}$$

***Sugestie*: Arătaţi că există [un anumit set de] instanţe pentru care matricea Gram generată folosind funcţia k_2 *nu* este pozitiv semidefinită. Conform teoremei a lui Mercer (vedeţi problema 68) va rezulta că funcţia k_2 nu este funcţie-nucleu.**

Răspuns:

a. Vom demonstra că există o funcţie de „mapare“ ϕ definită convenabil, astfel încât $k_1(x,x')=\phi(x)\cdot\phi(x')$ pentru oricare două imagini dreptunghiulare, x şi x'.

[158] Vedeţi de exemplu problema 15 de la capitolul *Învăţare bazată pe memorare* şi — mai ales! — problema 51 de la capitolul *Clusterizare*.

[159] *Precizare*: Nu se va lua în considerare poziţia în care apare o astfel de *zonă pătratică*, ci doar *imaginea* / coloritul ei ca atare. O astfel de *zonă* poate să apară în mai multe poziţii într-o aceeaşi imagine, însă funcţia $k_1(x,x')$ va lua valoarea „contoriza“ o dată o anumită *zonă pătratică* dacă ea apare [nu are importanţă de câte ori anume] atât în x cât şi în x'.

Notăm cu d numărul tuturor zonelor pătratice constituite din pixeli (patch-uri), de dimesiune 16×16. Întrucât fiecare pixel poate avea 256 de valori, urmează că

$$d = 256^{16\times 16} = (2^8)^{256} = 2^{2048}.$$

Fie funcţia ϕ definită pe mulţimea tuturor imaginilor dreptunghiulare posibile care ia valori în mulţimea $\{0,1\}^d$, astfel: $\phi(x) \overset{not.}{=} (\phi(x)_1, \ldots, \phi(x)_i, \ldots, \phi(x)_d)$, unde $\phi(x)_i$ ia valoarea 1 dacă patch-ul i este prezent în imaginea x, şi 0 în caz contrar. Se poate constata uşor că în baza acestei definiţii rezultă $k_1(x, x') = \phi(x) \cdot \phi(x')$. Prin urmare, funcţia k_1 este funcţie-nucleu.

b. Fie A şi B două patch-uri având toţi pixelii 0 şi, respectiv, toţi pixelii 1. De asemenea, fie x_1, x_2 şi x_3 trei imagini definite astfel: $x_1 = A$, $x_2 = B$, iar $x_3 = [AB]$, unde prin $[AB]$ am notat imaginea obţinută prin concatenarea imaginilor A şi B pe orizontală. Vom demonstra că matricea Gram care se obţine pe acest set de imagini folosind funcţia-nucleu k_2 nu este matrice pozitiv semidefinită.

Într-adevăr, respectiva matrice Gram este

$$G \overset{not.}{=} \begin{bmatrix} k_2(x_1,x_1) & k_2(x_1,x_2) & k_2(x_1,x_3) \\ k_2(x_2,x_1) & k_2(x_2,x_2) & k_2(x_2,x_3) \\ k_2(x_3,x_1) & k_2(x_3,x_2) & k_2(x_3,x_3) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

Conform definiţiei, G este matrice pozitiv semidefinită dacă pentru orice $f \in \mathbb{R}^3$ (văzut ca vector-coloană) urmează că $f^\top G f \geq 0$.

Considerând $f = (1,1,-1)^\top$, rezultă că $f^\top G f = (0,0,1)(1,1,-1)^\top = -1 < 0$. Prin urmare, matricea G nu este pozitiv semidefinită şi, în consecinţă, conform teoremei lui Mercer, funcţia k_2 nu este funcţie-nucleu.

73. **(O metodă particulară de mapare a atributelor, care asigură separare liniară)**

University of Utah, 2008 fall, Hal Daumé III, HW3, pr. 2

Considerăm N puncte într-un spaţiu $D-$dimensional, *nu* neapărat separabile liniar. În acest exerciţiu vom arăta că aceste puncte se pot mapa într-un spaţiu cu $D+N$ dimensiuni, în aşa fel încât să devină separabile liniar. Acest lucru se realizează transformând punctele $x_n \in \{x_1, x_2, \ldots, x_D\}$ în puncte de forma $(x_1, x_2, \ldots, x_D, 1_{\{n=1\}}, 1_{\{n=2\}}, 1_{\{n=3\}}, \ldots, 1_{\{n=N-1\}}, 1_{\{n=N\}})$, unde de exemplu prin notaţia $1_{\{n=3\}}$ se înţelege valoarea 1 dacă $n = 3$ şi 0 în cazul $n \neq 3$. Arătaţi că punctele obţinute în urma acestei mapări *sunt* într-adevăr separabile liniar.

Răspuns:

Considerând $x_1, \ldots, x_N$ instanţele date şi $X_1, \ldots, X_N$ imaginile lor din spaţiul $(D+N)$-dimensional, vom arăta că există $w \in \mathbb{R}^{D+N}$ astfel încât semnul expresiei $w \cdot X_i$ este pozitiv pentru instanţele etichetate pozitiv şi negativ în rest.

Dacă luăm $w = (0, 0, \ldots, 0, y_1, y_2, \ldots, y_N) \in \mathbb{R}^{D+N}$, unde

$$y_i = \begin{cases} 1 & \text{dacă punctul } X_i \text{ este etichetat cu } + \\ -1 & \text{dacă punctul } X_i \text{ este etichetat cu } -, \end{cases}$$

rezultă că

$$w \cdot X_i = \sum_{i=1}^{N} y_i \cdot X_i^{(D+i)} = y_i,\ \forall i \in \{1, \ldots, N\}$$

adică $w \cdot X_i = +1$ pentru toate instanţele x_i pozitive şi $w \cdot X_i = -1$ pentru toate instanţele x_i negative.

Aşadar, hiperplanul $w \cdot X = 0$ este un separator liniar pentru cele N puncte din spaţiul $(D+N)$-dimensional.

***Observaţie*: Deşi are o anumită importanţă teoretică,[160] maparea indicată în enunţul acestei probleme nu este eficientă din punct de vedere practic, în special atunci când N şi / sau D sunt numere foarte mari. De asemenea, ea produce overfitting, datorită incapacităţii ei intrinseci de a generaliza.**

74. **(RBF este funcţie-nucleu: demonstraţie)**

■ □ • · ***Stanford, 2009 fall, Andrew Ng, practice midterm exam, pr. 3.b***

Arătaţi că funcţia cu baza radială $e^{-\frac{1}{2\sigma^2}\|x-z\|^2}$, de variabile $x, z \in \mathbb{R}^d$, este într-adevăr funcţie-nucleu.

***Observaţie*: În acest exerciţiu nu se cere să se găsească efectiv funcţia de mapare ϕ pentru nucleul RBF. Este suficient să se dovedească existenţa ei, folosind de exemplu teorema lui Mercer (a se vedea problema 68.ab şi nota de subsol 151) sau proprietăţile de „construcţie" de noi nuclee (problemele 68.c, 69, 70 şi 155). Problema 75 va arăta că există o funcţie de „mapare" ϕ pentru nucleul RBF care asociază fiecărei instanţe x un „vector" $\phi(x)$ a cărui dimensiune este infinită![161]**

Răspuns:

Mai întâi vom pune expresia de definiţie pentru funcţia cu baza radială sub o formă mai convenabilă:

[160] În afară de asigurarea separabilităţii în spaţiul de trăsături (ceea ce am demonstrat în problema de faţă), această mapare — sau, chiar mai bine, o mapare similară cu aceasta — serveşte la determinarea numărului maxim de actualizări pe care le face perceptronul [Rosenblatt] cu margine (engl., margin perceptron) pe seturi de date neliniar-separabile, lucrând în spaţiul de trăsături corespunzător. Pentru detalii, vedeţi la capitolul *Reţele neuronale artificiale* problema 40.c (în conjuncţie cu problema 18).

[161] Prof. Tommi Jaakkola de la MIT precizează (Machine Learning course, 2009 fall, lecture notes 3, pages 3-4) că

$$\exp(-\frac{\beta}{2} \parallel x - x' \parallel^2) = \int \phi(z;x)\phi(z;x')dz \qquad (\star)$$

unde $\phi(z;x) \stackrel{def.}{=} c(\beta,d)\,\mathcal{N}(z|x, \frac{1}{2\beta})$, cu $c(\beta,d)$ o constantă, şi $\mathcal{N}(z|x, \frac{1}{2\beta})$ distribuţia normală de medie x şi varianţă $\frac{1}{2\beta}$. $\phi(x)$ poate fi definit ca fiind „vectorul" infinit $\{\phi(z;x)\}_{z\in\mathbb{R}^d}$. Aşadar, indexarea trăsăturilor se face după $z \in \mathbb{R}^d$, nu după un indice întreg aşa cum este cazul alteori. *Semnificaţia* relaţiei $(*)$ este următoarea: nucleul RBF măsoară probabilitatea ca o instanţă z să fie generată de două gaussiene de medii x şi respectiv x' şi varianţă comună, $\frac{1}{2\beta}$. Funcţiile-nucleu sunt adeseori definite dintr-o astfel de perspectivă.

$$e^{-\frac{\| x-z \|^2}{2\sigma^2}} = e^{-\frac{(x-z)\cdot(x-z)}{2\sigma^2}} = e^{-\frac{(x^2-2x\cdot z+z^2)}{2\sigma^2}} = e^{-\frac{x^2}{2\sigma^2}}\, e^{\frac{x\cdot z}{\sigma^2}}\, e^{-\frac{z^2}{2\sigma^2}}$$
$$= f(x)\, e^{\frac{x\cdot z}{\sigma^2}}\, f(z),$$

unde prin $f(x)$ am notat expresia $e^{-\frac{x^2}{2\sigma^2}}$.

Acum putem aplica diverse proprietăţi de „construcţie“ a nucleelor, pornind de la un nivel simplu spre un nivel din ce în ce mai complex. Evident, funcţia $x\cdot z$ este funcţie-nucleu. La fel şi funcţia $\frac{x\cdot z}{\sigma^2}$ (vedeţi problema 69 punctul $a.i$). A demonstra faptul că funcţia $e^{\frac{x\cdot z}{\sigma^2}}$ este funcţie-nucleu revine la a reproduce linia demonstraţiei de la problema 70 (până la un factor constant). În sfârşit, datorită proprietăţii de la problema 69 punctul $a.ii$, rezultă că $f(x)\, e^{\frac{x\cdot z}{\sigma^2}}\, f(z)$ — deci şi $e^{-\frac{1}{2\sigma^2}\|x-z\|^2}$ — este funcţie-nucleu.

75. **(Spaţii de „trăsături“ infinite)**

■ □ • ○ ***CMU, 2011 fall, T. Mitchell, A. Singh, HW6, pr. 2.2***

Pentru date reale de tipul $x\in\mathbb{R}$ putem să ne gândim la o transformare („mapare“) a trăsăturilor $\phi_n : \mathbb{R}^1 \to \mathbb{R}^{n+1}$ definită într-o manieră ceva mai complicată decât în mod obişnuit:

$$\phi_n(x) = \left\{ e^{-x^2/2},\, e^{-x^2/2}\, x,\, e^{-x^2/2}\, \frac{x^2}{\sqrt{2}},\, \ldots,\, e^{-x^2/2}\, \frac{x^i}{\sqrt{i!}},\, \ldots,\, e^{-x^2/2}\, \frac{x^n}{\sqrt{n!}} \right\}.$$

Presupunem acum că $n\to\infty$ şi definim o nouă „mapare“ a trăsăturilor, care produce un vector infinit:

$$\phi_\infty(x) = \left\{ e^{-x^2/2},\, e^{-x^2/2} x,\, e^{-x^2/2}\, \frac{x^2}{\sqrt{2}},\, \ldots,\, e^{-x^2/2}\, \frac{x^i}{\sqrt{i!}},\, \ldots \right\}. \tag{97}$$

Se poate arăta — vedeţi de exemplu problemele 9.c şi 12.f de la capitolul *Maşini cu vectori-suport* — că în multe situaţii putem să exprimăm ecuaţia separatorului produs de un clasificator liniar folosind doar produse scalare de vectori (anumite instanţe de antrenament) într-un spaţiu oarecare (sau în aşa-numitul spaţiu de „trăsături“).[162] Ne punem problema dacă n-am putea să folosim cumva spaţiul de trăsături obţinut prin „maparea“ ϕ_∞. Totuşi, pentru aceasta ar trebui să putem calcula produsul scalar de [imagini de] instanţe în acest spaţiu de „trăsături“ infinit. Definim produsul scalar dintre doi vectori infiniţi $a=\{a_1,\ldots,a_i,\ldots\}$ şi $b=\{b_1,\ldots,b_i,\ldots\}$ ca fiind suma infinită

$$a\cdot b = \sum_{i=1}^{\infty} a_i b_i \overset{def.}{=} \lim_{n\to\infty} \sum_{i=1}^{n} a_i b_i. \tag{98}$$

[162] Pentru *caracterizări teoretice* ale unor astfel de clasificatori (aşa-numitele „teoreme de reprezentare“), vedeţi problemele 88 şi 182.

a. Putem oare să calculăm în mod explicit $\phi_\infty(a)\cdot\phi_\infty(b)$? Altfel spus, care este expresia explicită pentru funcţia-nucleu $K(a,b) \overset{not.}{=} \phi_\infty(a)\cdot\phi_\infty(b)$?
***Sugestie*: Folosiţi dezvoltarea în *serie Taylor* a lui e^x:**

$$e^x = \lim_{n\to\infty} \sum_{i=0}^{n} \frac{x^i}{i!}. \tag{99}$$

b. Ar trebui oare ca într-un asemenea spaţiu de dimensiune infinită să ne punem problema *complexităţii* clasificatorului [care foloseşte această „mapare"]?

Răspuns:

a. Putem scrie imediat:

$$K(a,b) \overset{not.}{=} \phi_\infty(a)\cdot\phi_\infty(b) \overset{(98)(97)}{=} \sum_{i=0}^{\infty} \frac{e^{-a^2/2}\,a^i}{\sqrt{i!}} \cdot \frac{e^{-b^2/2}\,b^i}{\sqrt{i!}}$$
$$= \exp\left(-\frac{a^2+b^2}{2}\right)\cdot\sum_{i=0}^{\infty}\frac{(ab)^i}{i!} \overset{(99)}{=} \exp\left(-\frac{a^2+b^2}{2}\right)\cdot\exp(ab) = \exp\left(-\frac{(a-b)^2}{2}\right).$$

***Observaţie*: Ultima expresie care a fost obţinută aici corespunde unei funcţii-nucleu cu baza radială, ale cărei argumente (a şi b) sunt numere reale. Demonstraţia pentru cazul *ceva mai* general al nucleului RBF $K(a,b) \overset{def.}{=} \exp(-\gamma(a-b)^2)$, unde $\gamma \in \mathbb{R}_+^*$, iar $a,b \in \mathbb{R}$, urmează îndeaproape linia acestei demonstraţii.[163] Aşadar, pentru funcţiile-nucleu de tip RBF cu argumente reale, „maparea" ϕ ia valori într-un spaţiu de dimensiune infinită. Pentru cazul nucleelor RBF cu argumente din $\mathbb{R}^d$, unde $d \geq 2$, este util să vedeţi nota de subsol 161.**

b. În general, ar trebui ca un model [de clasificare] bazat pe astfel de vectori de „trăsături" să ne îngrijoreze (din perspectiva *overfitting*-ului), însă la clasificatorii care folosesc funcţii-nucleu, calculele nu se fac în spaţiul de „trăsături", ci în spaţiul de intrare (adică, $\mathbb{R}$ în cazul de faţă, fiindcă $a,b \in \mathbb{R}$ şi $\phi(a)\cdot\phi(b) = K(a,b)$.) În cazul maşinilor cu vectori-suport, complexitatea modelului este dată de numărul de vectori care intră *de facto* în definiţia separatorului optimal (aşa-numiţii „vectori-suport"), nu de dimensiunea spaţiului de „trăsături".[164]

76. (O proprietate simplă a nucleului de tip RBF)

CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, final exam, pr. 7.a.2

Aşa-numita funcţie cu baza radială (RBF), $K(x_i,x_j) = e^{-\frac{||x_i-x_j||^2}{2\sigma^2}}$ este una dintre cele mai folosite funcţii-nucleu. Considerăm punctele x, z_1 şi z_2 situate geometric astfel: z_1 foarte aproape de x, iar z_2 situat foarte departe de x.

Cum vor fi valorile lui $K(z_1,x)$ şi $K(z_2,x)$? Alegeţi una dintre următoarele variante:

[163] Vedeţi CMU, 2014 fall, Eric Xing, Barnabas Poczos, HW2, pr. 3.2.

[164] Vedeţi, mai concret, formulele (314) şi (319) de la problemele 9 şi respectiv 12 de la capitolul *Maşini cu vectori-suport*. Vectorii-suport sunt acele (în general, puţine!) instanţe de antrenament x_i pentru care coeficienţii α_i sunt nenuli.

i. **$K(z_1, x)$ va fi foarte aproape de 1, iar $K(z_2, x)$ va fi foarte aproape de 0.**

ii. **$K(z_1, x)$ va fi foarte aproape de 0, iar $K(z_2, x)$ va fi foarte aproape de 1.**

iii. **$K(z_1, x)$ va avea o valoare mult mai mare decât 1, iar $K(z_2, x)$ va avea o valoare mult mai mică decât 0.**

iv. **$K(z_1, x)$ va avea o valoare mult mai mică decât 0, iar $K(z_2, x)$ va avea o valoare mult mai mare decât 1.**

Răspuns:

Funcţia e^{-y} are valoarea 1 pentru $y = 0$. Fiind o funcţie continuă, limita ei pentru $y \to 0$ este 1. Pentru $y \to +\infty$ limita acestei funcţii este 0.

Prin urmare, pentru $z_1 \to x$ vom avea $\|z_1 - x\| \to 0$ şi $K(z_1, x) \to 1$, iar pentru $\|z_2 - x\| \to +\infty$ vom avea $K(z_2, x) \to 0$. Aşadar, doar afirmaţia de la punctul *i.* este adevărată; celelalte afirmaţii sunt false.

77. **(Funcţii-nucleu: da sau nu?)**

□ • ○ ***CMU, 2017 fall, Nina Balcan, midterm, pr. 1.1***

Care dintre următoarele funcţii nu constituie funcţii-nucleu valide?

a. **$c_1 K_1(x_1, x_2) + c_2 K_2(x_1, x_2)$, unde K_1 şi K_2 sunt funcţii-nucleu, iar c_1 şi c_2 sunt constante reale;**

b. **$K(x_1, x_2) \stackrel{def.}{=} x_1^\top A x_2$, unde $A \in \mathbb{R}^{d \times d}$ este o matrice simetrică şi pozitiv semidefinită;**[165]

c. **$K(x_1, x_2) \stackrel{def.}{=} x_1^\top A x_2$, unde $A \in \mathbb{R}^{d \times d}$ este o matrice simetrică.**

Răspuns:

a. **Conform problemelor 69.a şi 68.c, ştim că atunci când c_1 şi c_2 sunt pozitive, suma $c_1 K_1(x_1, x_2) + c_2 K_2(x_1, x_2)$ reprezintă o funcţie-nucleu. Vom arăta că atunci când se renunţă la condiţia $c_1, c_2 \geq 0$ concluzia nu se mai menţine (în general). Într-adevăr, dacă luăm $c_1 = 0$ şi $c_2 = -1$, funcţia $c_1 K_1(x_1, x_2) + c_2 K_2(x_1, x_2) = -K_2(x_1, x_2)$ va avea, pentru orice set de instanţe $\{x_i\}_{i=1}^m$, matricea Gram $-G_2$, unde prin G_2 am notat matricea Gram corespunzătoare pentru funcţia-nucleu K_2. Conform teoremei lui Mercer (vedeţi problema 68.ab), G_2 este matrice pozitiv semidefinită. În cazul (particular) în care *o* astfel de matrice G_2 este chiar pozitiv definită rezultă că pentru orice vector-coloană nenul f din $\mathbb{R}^m$ vom avea $f^\top G_2 f > 0$. Dacă presupunem prin *reducere la absurd că* funcţia $-K_2$ este funcţie-nucleu, conform aceleaşi teoreme ar rezulta că într-un astfel de context vom avea $f^\top(-G_2)f > 0 \Leftrightarrow -f^\top G_2 f > 0 \Leftrightarrow f^\top G_2 f < 0$ pentru orice vector-coloană nenul $f \in \mathbb{R}^m$, ceea ce evident intră în *contradicţie* cu relaţia $f^\top G_2 f > 0$. Aşadar, într-un astfel de caz funcţia $-K_2$ *nu* este funcţie-nucleu.**[166]

[165] Problema 163 indică două modalităţi de obţinere de *noi* matrice pozitiv semidefinite pornind de la alte asemenea matrice.

[166] *Observaţie*: În cazul (complementar) în care *orice* astfel de matrice G_2 este pozitiv semidefinită (nu pozitiv definită, ca mai sus) rezultă — presupunând în continuare (prin reducere la absurd) că funcţia K_2 este funcţie-nucleu, ceea ce implică faptul că matricea $-G_2$ este pozitiv semidefinită — că $f^\top G_2 f = 0$ pentru orice $f \in \mathbb{R}^m$. În particular, atunci când luăm drept f vectorii de tip „one-hot“ din $\mathbb{R}^m$, rezultă imediat că vom avea $K_2(x_i, x_i) = 0$ pentru orice x_i. (Această înseamnă că funcţia-nucleu K_2 este funcţia constantă 0.) În consecinţă, $\Phi_2(x_i) = 0$ pentru orice x_i, ceea ce nu este deloc util din punctul de vedere al task-urilor principale din învăţarea automată (clasificare, regresie ori clusterizare).

b. **În acest caz, răspunsul este pozitiv: funcţia $x_1^\top A x_2$, unde A este matrice de numere reale, simetrică şi pozitiv semidefinită, este funcţie-nucleu. Justificarea se face pe baza proprietăţii de *factorizare* care a fost demonstrată la problema 36.a (vedeţi *Observaţia* (4) de la pagina 81): există o matrice B astfel încât $A = BB^\top$. Prin urmare,**

$$x_1^\top A x_2 = x_1^\top B B^\top x_2 = (B^\top x_1)^\top (B^\top x_2).$$

Definind funcţia de „mapare“ $\phi(x) = B^\top x$, egalitatea precedentă implică imediat faptul că funcţia $K(x_1, x_2) \stackrel{def.}{=} x_1^\top A x_2 = \phi(x_1)^\top \phi(x_2)$ este funcţie-nucleu.

c. **În acest caz, chestiunea revine la a determina dacă renunţarea (în raport cu punctul *b*) la condiţia de pozitiv semidefinire a matricei A (păstrând însă condiţia de simetrie) are sau nu vreun efect asupra „nucleicităţii“ funcţiei $x_1^\top A x_2$. Răspunsul este negativ.**

Într-adevăr, dacă am presupune (prin *reducere la absurd*) că funcţia $K(x_1, x_2) \stackrel{def.}{=} x_1^\top A x_2$, unde A este o matrice de numere reale, simetrică dar fără a fi pozitiv semidefinită este funcţie-nucleu, atunci în baza unui raţionament absolut similar cu cel din rezolvarea problemei 68.b[167] ar rezulta cu necesitate că A este matrice pozitiv semidefinită, ceea ce reprezintă o *contradicţie*. Prin urmare, atunci când A nu este pozitiv semidefinită nu rezultă cu necesitate că funcţia $K(x_1, x_2) \stackrel{def.}{=} x_1^\top A x_2$ este funcţie-nucleu.

0.1.7 Metode de optimizare în învăţarea automată

78. **(Definiţii şi proprietăţi de bază ale funcţiilor convexe)**

□ • ○ ***CMU, 2015 fall, A. Smola, B. Poczos, HW1, pr. 3.1***

În anumiţi algoritmi de optimizare, cum este de exemplu *metoda gradientului descendent*, convexitatea funcţiei "target" joacă un rol important în a stabili dacă algoritmul va converge (sau nu), cât de mare este rata / rapiditatea convergenţei ş.a.m.d. În această problemă, pornind de la *definiţii*, vom pune în evidenţă câteva *proprietăţi* de care ne vom putea servi atunci când va trebui să examinăm convexitatea unei funcţii.[168]

> ***Definiţie* (*mulţime convexă*):**
> **Spunem că o mulţime $C \subseteq \mathbb{R}^d$ este convexă dacă pentru orice pereche de puncte $x, y \in C$ şi pentru orice $t \in [0, 1]$ urmează că şi punctul $tx + (1-t)y \in C$.[169]**

[167] Veţi ţine cont că ipoteza „K este funcţie-nucleu“ implică faptul că există o „mapare“ ϕ astfel încât $K(x_1, x_2) = \phi(x_1) \cdot \phi(x_2)$ pentru orice x_1 şi x_2 din domeniul de definiţie al lui K. Apoi, G, matricea Gram pentru instanţele $x_1, \ldots, x_m$ se va scrie generic $[x_i^\top A x_j]_{i,j} = [\phi(x_i)^\top \phi(x_j)]_{i,j}$, după care inegalitatea $f^\top G f \geq 0$ pentru orice $f \in \mathbb{R}^m$ va fi demonstrată exact ca la problema 68.b.

[168] De *exemplu*, problema 166 examinează convexitatea unora dintre cele mai des folosite funcţii obiectiv din domeniul învăţării automate [profunde].

[169] Această proprietate înseamnă că pentru orice pereche de puncte x şi y din mulţimea convexă C, orice punct care este situat pe segmentul de dreaptă care uneşte perechea de puncte x, y aparţine de asemenea mulţimii C. Este foarte util ca o mulţime să aibă această proprietate, deoarece putem ajunge la orice punct din mulţime pornind de la un punct dat [din aceeaşi mulţime], parcurgând o linie dreaptă, fără să atingem „marginea“ mulţimii respective. Adeseori, a atinge „marginea“ mulţimii înseamnă că poţi ajunge să fii blocat într-un punct de optim local şi să nu mai ai niciodată şansa să găseşti soluţia optimă globală.

***Definiţie* (*funcţie convexă*):**
Spunem că o funcţie $f : \mathbb{R}^d \to \mathbb{R}$ este convexă dacă domeniul ei de definiţie (notat de aici încolo cu $dom(f)$) este o mulţime convexă, şi pentru orice $x, y \in dom(f)$ şi orice $t \in [0,1]$ urmează că:[170]

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y). \tag{100}$$

Dacă în inegalitatea (100) înlocuim relaţia $\leq$ cu $<$, vom spune că funcţia f este *strict convexă*.

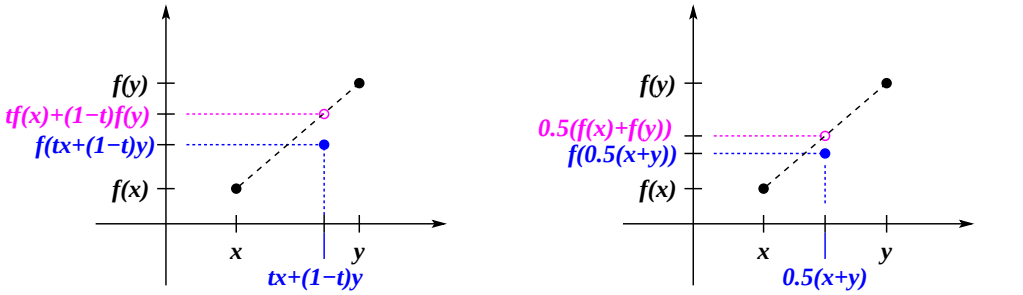
a. Uneori nu este uşor să demonstrăm convexitatea unei funcţii în mod direct, adică bazându-ne doar pe definiţia de mai sus. De aceea, în cele ce urmează vom formula câteva *proprietăţi ale funcţiilor convexe*, fiecare dintre aceste proprietăţi fiind echivalente cu definiţia.[171] **Pentru simplitate, vom presupune că $dom(f) = \mathbb{R}$.**

1. **Funcţia f este convexă dacă şi numai dacă**

$$f\Big(\frac{x+y}{2}\Big) \leq \frac{f(x)+f(y)}{2}, \text{ pentru orice } x, y \in \mathbb{R}.$$

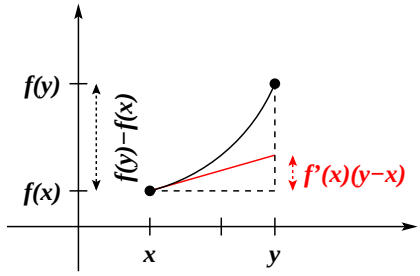
Aceasta este aşa-numita „proprietate a punctului de mijloc“ (engl., *mid-point property*).

Figura următoare oferă o reprezentare grafică a relaţiei din definiţia noţiunii de funcţie convexă (relaţia (100)), precum şi a relaţiei din condiţia 1.



2. **Fie f o funcţie derivabilă. f este convexă dacă şi numai dacă inegalitatea $f(x) \geq f(y)+f'(y)(x-y)$ este satisfăcută, pentru orice $x, y \in \mathbb{R}$.**

***Observaţie*: Interschimbând rolurile variabilelor x şi y, ajungem la următoarea *formulare echivalentă*:**



$$f(y) \geq f(x) + f'(x)(y-x), \text{ pentru orice } x, y \in \mathbb{R}.$$

[170] Inegalitatea (100) spune că segmentul de dreaptă care uneşte două puncte situate pe graficul funcţiei f este întotdeauna situat deasupra [graficului] funcţiei f. O astfel de proprietate este foarte utilă atunci când urmărim să identificăm valoarea minimă a funcţiei, întrucât pentru orice două puncte, x şi y, alese în mod arbitrar din $dom(f)$, există un punct z situat între x şi y astfel încât $f(z)$ coincide cu minimul funcţiei f pe intervalul $[x, y]$.
[171] Astfel de proprietăţi se numesc *caracterizări* ale definiţiei.

***Semnificaţia geometrică* a acestor relaţii este următoarea: dacă f este funcţie convexă, tangenta la graficul lui f (în orice punct din domeniul de definiţie) este situată sub graficul lui f.**[172]

3. **Fie f o funcţie dublu derivabilă (adică, există atât f' cât şi f''). Funcţia f este convexă dacă şi numai dacă $f''(x) \geq 0$, pentru orice $x \in \mathbb{R}$.**

Vă cerem ca pentru fiecare dintre punctele 1, 2 şi 3 să faceţi demonstraţia echivalenţei cu definiţia noţiunii de funcţie convexă, în ambele direcţii.

b. Folosiţi definiţia sau proprietăţile formulate mai sus pentru a demonstra următoarele afirmaţii:

4. **Dacă f şi g sunt funcţii convexe având acelaşi domeniu de definiţie, atunci funcţia h definită prin relaţia $h(x) = \max(f(x), g(x))$ este de asemenea o funcţie convexă.**[173]

5. **Dacă f şi g sunt funcţii convexe având acelaşi domeniu de definiţie, atunci funcţia h definită prin relaţia $h(x) = f(x) + g(x)$ este de asemenea o funcţie convexă.**[174]

6. **Dacă f şi g sunt funcţii convexe, iar $dom(f) \supseteq image(g)$, atunci funcţia compusă $f \circ g$ este oare în mod necesar o funcţie convexă? Dacă nu, atunci ce tip de restricţii ar trebui să mai adăugăm pentru ca funcţia $f \circ g$ să fie convexă? (Justificaţi răspunsul în mod riguros.)**

Răspuns:

a. Vom lua pe rând afirmaţiile 1, 2 şi 3 şi vom demonstra că fiecare dintre ele este *echivalentă* cu *definiţia* noţiunii de funcţie convexă. (Aşa cum s-a cerut în enunţ, în fiecare din cele trei cazuri, vom demonstra echivalenţa în ambele sensuri.) Aşadar, cele trei afirmaţii constituie *caracterizări* ale noţiunii de funcţie convexă.

• Definiţia implică afirmaţia 1:

Luând $t = 1/2$ în definiţia noţiunii de funcţie convexă, obţinem imediat afirmaţia 1.

• Afirmaţia 1 implică definiţia:

Este [mai] uşor să demonstrăm această implicaţie prin *reducere la absurd*. Presupunem că există o funcţie f care satisface afirmaţia 1 dar nu este convexă. Prin urmare, există o pereche de puncte x_0, y_0 astfel încât

$$f(t_0x_0 + (1 - t_0)y_0) > t_0f(x_0) + (1 - t_0)f(y_0) \tag{101}$$

pentru un anumit $t_0 \in [0, 1]$. Vom defini funcţia g cu ajutorul relaţiei

$$g(t) = f(tx_0 + (1 - t)y_0) - tf(x_0) - (1 - t)f(y_0),$$

[172] Se poate spune că această „condiţie“ (relaţie echivalentă cu definiţia) este *de ordinul întâi*, spre deosebire de condiţia 1, care este *de ordinul zero* (la fel ca şi condiţia din *definiţia* (100)), dar şi de următoarea condiţie (3), care este *de ordinul al doilea*.

[173] Aşadar, clasa funcţiilor convexe definite pe un acelaşi domeniu este „închisă“ la operatorul max.

[174] Mai general, dacă $f_1, f_2, \ldots, f_n$ sunt funcţii convexe definite pe un acelaşi domeniu şi $a_1, a_2, \ldots, a_n$ sunt numere reale nenegative, atunci $a_1f_1 + a_2f_2 + \ldots + a_nf_n$ este funcţie convexă. (Aşadar, clasa funcţiilor convexe definite pe un acelaşi domeniu este „închisă“ la *combinaţii canonice*. Vedeţi nota de subsol 183.)

pentru $t \in [0,1]$. Se constată imediat că $g(0) = 0$ şi $g(1) = 0$. Evident, $g(t_0) > 0$. În continuare, vom folosi notaţiile $M = \max_{t\in(0,1)} g(t)$ şi $t^* = \min\{t \in (0,1) | g(t) = M\}$. Aceasta înseamnă că t^* indentifică cea mai mică valoare a lui t pentru care se atinge maximul funcţiei g pe intervalul $(0,1)$.[175] Mai întâi vom arăta că şi funcţia g satisface „proprietatea punctului de mijloc", după care, folosind această proprietate a lui g vom arăta că se obţine o contradicţie.

Pentru orice a şi b din $[0,1]$,

$$g\Big(\frac{a+b}{2}\Big) \stackrel{def.}{=} f\Big(\frac{a+b}{2}x_0 + \Big(1 - \frac{a+b}{2}\Big)y_0\Big) - \frac{a+b}{2}f(x_0) - \Big(1 - \frac{a+b}{2}\Big)f(y_0).$$

Întrucât

$$\frac{a+b}{2}x_0 + \Big(1 - \frac{a+b}{2}\Big)y_0 = \frac{1}{2}[ax_0 + (1-a)y_0] + \frac{1}{2}[bx_0 + (1-b)y_0],$$

iar f satisface „proprietatea punctului de mijloc" (afirmaţia 1), rezultă

$$\begin{aligned} g\Big(\frac{a+b}{2}\Big) &\leq \frac{1}{2}f(ax_0 + (1-a)y_0) + \frac{1}{2}f(bx_0 + (1-b)y_0) \\ &\quad -\frac{1}{2}af(x_0) - \frac{1}{2}bf(x_0) - \frac{1}{2}(1-a)f(y_0) - \frac{1}{2}(1-b)f(y_0) \\ &\stackrel{def.}{=} \frac{1}{2}g(a) + \frac{1}{2}g(b). \end{aligned} \tag{102}$$

Aşadar, am arătat că funcţia g satisface (şi ea) „proprietatea punctului de mijloc". Fie acum un număr $\varepsilon > 0$ suficient de mic pentru ca $(t^* + \varepsilon)$ şi $(t^* - \varepsilon)$ să aparţină [încă] intervalului $(0,1)$.[176] În fine, folosind aceste proprietăţi obţinem:

$$\begin{aligned} g(t^*) &= g\Big(\frac{(t^*+\varepsilon) + (t^*-\varepsilon)}{2}\Big) \\ &\stackrel{(102)}{\leq} \frac{1}{2}g(t^*+\varepsilon) + \frac{1}{2}g(t^*-\varepsilon) < \frac{M+M}{2} = M. \end{aligned}$$

Ultima inegalitate de mai sus este justificată de faptul că $g(t^* - \varepsilon) < M$ şi $g(t^* + \varepsilon) \leq M$. Inegalitatea obţinută, $g(t^*) < M$, contrazice presupunerea pe care am făcut-o iniţial, şi anume că $g(t^*) = M$. Prin urmare, presupunerea făcută etse falsă. Aşadar, concluzionăm că orice funcţie f care satisface afirmaţia 1 este cu necesitate [funcţie] convexă.

• Definiţia implică afirmaţia 2:

Definiţia noţiunii de funcţie convexă spune că pentru orice x şi y din $\mathbb{R}$ şi pentru orice $t \in [0,1]$, are loc inegalitatea

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y),$$

ceea ce (pentru $t \neq 0$) conduce la următoarea inegalitate:

$$\frac{f(y + t(x-y)) - f(y)}{t} \leq f(x) - f(y).$$

[175] Conform presupunerii noastre, rezultă că există cel puţin un t astfel încat $g(t) > 0$. (Vedeţi relaţia (101).)
[176] Este imediat că există un astfel de ε.

Prin împărţirea şi înmulţirea termenului stâng cu $(x-y)$, obţinem:

$$\frac{f(y+t(x-y))-f(y)}{t(x-y)}(x-y) \leq f(x)-f(y).$$

Prin urmare, făcând $t \to 0$, va rezulta că

$$f'(y)(x-y) \leq f(x)-f(y),$$

ceea ce este echivalent cu

$$f(y)+f'(y)(x-y) \leq f(x).$$

Aşadar, a rezultat că afirmaţia 2 este adevărată.

• Afirmaţia 2 implică definiţia:

Trebuie să demonstrăm că pentru orice x şi y din $\mathbb{R}$ şi pentru orice $t \in [0,1]$ are loc inegalitatea $f(tx+(1-t)y) \leq tf(x)+(1-t)f(y)$.

Considerând un z arbitrar ales din $\mathbb{R}$ şi aplicând afirmaţia 2 mai întâi pentru x şi z (în locul lui x şi y) şi apoi pentru y şi z (în locul lui x şi y), vom obţine:

$$f(x) \geq f(z)+f'(z)(x-z) \tag{103}$$
$$f(y) \geq f(z)+f'(z)(y-z) \tag{104}$$

Înmulţind prima dintre aceste relaţii cu t şi pe cea de-a doua cu $(1-t)$,[177] va rezulta:

$$tf(x)+(1-t)f(y) \geq f(z)+f'(z)(t(x-z)+(1-t)(y-z)) = f(z)+f'(z)(tx+(1-t)y-z).$$

În particular, pentru $z=tx+(1-t)y$, rezultă:

$$tf(x)+(1-t)f(y) \geq f(z) = f(tx+(1-t)y).$$

***Observaţie*: Atunci când $f:\mathbb{R}^d \to \mathbb{R}$, afirmaţia 2 devine: dacă f este funcţie derivabilă (adică există toate derivatele sale parţiale de ordinul întâi), atunci funcţia f este convexă dacă şi numai dacă pentru orice $x,y \in \mathbb{R}^d$ are loc inegalitatea $f(x) \geq f(y)+\nabla f(y)^\top (x-y)$, unde $\nabla f(y)$ este vectorul gradient (adică vectorul de derivate parţiale) al lui f calculat în punctul y.**

• Definiţia implică afirmaţia 3:

Am arătat mai sus că definiţia implică afirmaţia 2, deci pentru orice x şi y are loc inegalitatea

$$f(y) \geq f(x)+f'(x)(y-x).$$

Conform teoremei lui Taylor,[178] într-o vecinătate suficient de mică a lui x îl putem înlocui pe $f(y)$ cu suma dintre aproximarea sa ca polinom Taylor de ordinul al doilea (notat cu $P(y)$) şi un termen care reprezintă eroarea la aproximare:

$$f(y) = \underbrace{f(x)+f'(x)(y-x)+\frac{1}{2}f''(x)(y-x)^2}_{P(y)} + h_2(y)(y-x)^2,$$

[177] Atât t cât şi $1-t$ sunt ≥ 0.
[178] Vedeţi https://en.wikipedia.org/wiki/Taylor's_theorem.

unde $h_2(y) \to 0$ pentru $y \to x$. Prin urmare, [atunci când y este în respectiva vecinătate a lui x] inegalitatea precedentă implică

$$f''(x)(y-x)^2 + h_2(y)(y-x)^2 \geq 0.$$

Împărţind ambii membri ai acestei inegalităţi cu cantitatea $(y-x)^2$, care este strict pozitivă pentru orice $y \neq x$, şi făcând apoi $y \to x$, rezultă că $f''(x) \geq 0$.

• Afirmaţia 3 implică definiţia:

Întrucât funcţia $f(x)$ este dublu derivabilă, putem scrie din nou aproximarea lui $f(y)$ ca polinom Taylor de ordinul al doilea:

$$f(y) = f(x) + f'(x)(y-x) + \frac{1}{2}f''(x)(y-x)^2 + h_2(y)(y-x)^2,$$

unde, ca şi mai sus, $h_2(y) \to 0$ pentru $y \to x$. Afirmaţia 3 spune că $f''(x) \geq 0$ pentru orice x. Teorema lui Taylor afirmă că $|h_2(y)(y-x)^2|$ creşte mult mai lent decât $f''(x)(y-x)^2$. Prin urmare, [putem considera că] suma ultimilor doi termeni din membrul drept al egalităţii precedente este mai mare sau egală cu zero. Aşadar, vom avea

$$f(y) = f(x) + f'(x)(y-x) + \frac{1}{2}f''(x)(y-x)^2 + h_2(y)(y-x)^2 \geq f(x) + f'(x)(y-x).$$

Deci este satisfăcută afirmaţia 2 şi, în concluzie, f este funcţie convexă.

***Observaţie*: Atunci când $f : \mathbb{R}^d \to \mathbb{R}$, afirmaţia 3 devine: dacă f este funcţie dublu derivabilă (adică există toate derivatele sale parţiale de ordinul al doilea), atunci funcţia f este convexă dacă şi numai dacă pentru orice $x \in \mathbb{R}^d$ matricea sa hessiană[179] $\nabla^2 f(x)$ (adică, matricea derivatelor parţiale de ordinul al doilea ale lui f calculate in punctul x) este *pozitiv semidefinită*, ceea ce înseamnă că pentru orice $v \in \mathbb{R}^d$ urmează că $v^\top \nabla^2 f(x) v \geq 0$.[180]**

b.4 Vom face demonstraţia pornind de la definiţia noţiunii de funcţie convexă. Pentru orice x şi y, putem scrie

$$\begin{aligned}
h(tx+(1-t)y) &\overset{def.}{=} \max\{f(tx+(1-t)y),\ g(tx+(1-t)y)\} \\
&\leq \max\{tf(x)+(1-t)f(y),\ tg(x)+(1-t)g(y)\} \\
&\leq \max\{tf(x),\ tg(x)\} + \max\{(1-t)f(y),\ (1-t)g(y)\} \\
&= t\max\{f(x),\ g(x)\} + (1-t)\max\{f(y),\ g(y)\} \\
&\overset{def.}{=} th(x) + (1-t)h(y).
\end{aligned}$$

Aşadar, funcţia h este convexă.

b.5 Şi de data aceasta vom face demonstraţia pornind de la definiţie. Pentru orice x şi y, putem scrie

$$\begin{aligned}
h(tx+(1-t)y) &\overset{def.}{=} f(tx+(1-t)y) + g(tx+(1-t)y) \\
&\leq tf(x)+(1-t)f(y)+tg(x)+(1-t)g(y) \\
&= t(f(x)+g(x)) + (1-t)(f(y)+g(y)) \\
&\overset{def.}{=} th(x) + (1-t)h(y).
\end{aligned}$$

[179] După numele lui Ludwig Otto Hesse (1811 – 1874), matematician german.

[180] Se poate constata uşor că pentru cazul $d = 1$ relaţia $v^\top \nabla^2 f(x) v \geq 0$ este echivalentă cu condiţia $f''(x) \geq 0$.

Prin urmare, funcţia h este convexă.

***Observaţie*: Proprietăţile $b.4$ şi $b.5$ sunt valabile şi pentru cazul general $f, g : \mathbb{R}^d \to \mathbb{R}$, nu doar pentru cazul $d = 1$.**

b.6 Afirmaţia din enunţ nu este adevărată întotdeauna. Să luăm de exemplu funcţiile $g(x) = x^2$ şi $f(x) = -x$. Ambele funcţii sunt convexe, dar $f(g(x)) = -x^2$ este o funcţie concavă!

În cele ce urmează, pentru a identifica un set de condiţii suficiente astfel încât funcţia $f \circ g$ să fie convexă, vom porni de la afirmaţia $a.3$. Conform acestei afirmaţii, atunci când f şi g sunt funcţii dublu derivabile, compunerea lor (notată $f \circ g$) este funcţie convexă dacă $[f(g(x))]'' \geq 0$ pentru orice x. Derivata de ordinul al doilea al funcţiei $f \circ g$ se calculează asftel:

$$\begin{aligned}[f(g(x))]'' &= [f'(g(x))\, g'(x)]' = (f'(g(x)))'g'(x) + f'(g(x))g''(x) \\ &= f''(g(x))(g'(x))^2 + f'(g(x))g''(x).\end{aligned}$$

Întrucât funcţiile f şi g sunt convexe şi dublu derivabile, rezultă că $f''(x) \geq 0$ şi $g''(x) \geq 0$ pentru orice x. Aşadar, primul termen din expresia (105) este mai mare sau egal cu zero. Pentru a ne asigura că şi al doilea termen din această expresie este mai mare sau egal cu zero, trebuie să impunem condiţia ca $f'(x) \geq 0$, ceea ce însemnă că f este o funcţie nedecrescătoare.

În concluzie, putem formula următoarea *proprietate*:
Dacă f şi g sunt funcţii convexe şi dublu derivabile, iar f este funcţie nedecrescătoare, rezultă că $f \circ g$ este de asemenea funcţie convexă.

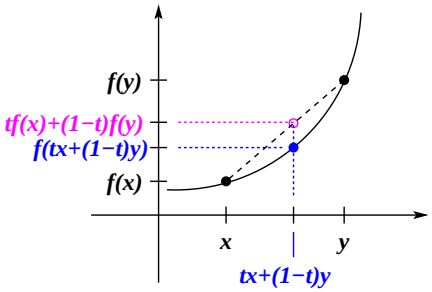
79. **(Inegalitatea lui Jensen şi câteva consecinţe ale ei)**

■ ***Liviu Ciortuz, 2020***

Dacă $f : \mathbb{R} \to \mathbb{R}$ este o *funcţie convexă*, atunci, conform *definiţiei*,[181] pentru orice $t \in [0, 1]$ şi orice $x_1, x_2 \in \mathbb{R}$ urmează

$$f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2). \quad (105)$$

Dacă f este funcţie strict convexă, atunci egalitatea are loc doar dacă $x_1 = x_2$.



a. Folosind definiţia de mai sus, demonstraţi *inegalitatea lui Jensen*:[182]
Pentru orice $a_i \geq 0,\ i = 1, \ldots, n$ cu $\sum_i a_i = 1$ şi orice $x_i \in \mathbb{R},\ i = 1, \ldots, n$, dacă f este funcţie convexă,[183] **atunci**

$$f\Big(\sum_i a_i x_i\Big) \leq \sum_i a_i f(x_i). \quad (106)$$

[181] Vedeţi problema 78.

[182] Johan Jensen, inginer şi matematician danez (1859–1925).

[183] Dacă numerele a_i, cu $i = 1, \ldots, n$, satisfac cele două proprietăţi indicate, spunem că suma $\sum_i a_i x_i$ este o *combinaţie convexă* a numerelor $x_1, \ldots, x_n$. Atunci când se foloseşte doar prima condiţie ($a_i \geq 0,\ i = 1, \ldots, n$), suma respectivă se numeşte *combinaţie canonică*. Când se foloseşte doar cea de-a doua condiţie ($\sum_i a_i = 1$), suma se numeşte *combinaţie afină*. În fine, dacă se renunţă la ambele condiţii, suma se numeşte *combinaţie liniară*. (Cf. https://en.wikipedia.org/wiki/Linear_combination.)

Mai general, pentru orice $a_i' \geq 0$, cu $i = 1, \ldots, n$ şi $\sum_i a_i' \neq 0$ avem

$$f\left(\frac{\sum_i a_i' x_i}{\sum_j a_j'}\right) \leq \frac{\sum_i a_i' f(x_i)}{\sum_j a_j'}. \tag{107}$$

***Observaţii*:**
1. Dacă f este strict convexă, atunci în relaţiile de mai sus egalitatea are loc doar dacă $x_1 = \ldots = x_n$.
2. Evident, rezultate similare cu cele de mai sus pot fi formulate şi pentru funcţii concave, înlocuind în relaţiile (106) şi (107) semnul $\leq$ cu $\geq$.

b. Demonstraţi *inegalitatea mediilor* folosind inegalitatea lui Jensen:[184]

$$\frac{x_1 + x_2 + \ldots + x_n}{n} \geq \sqrt[n]{x_1 \cdot x_2 \cdot \ldots \cdot x_n} \quad \textbf{pentru orice } x_i \geq 0,\ i = 1, \ldots, n.$$

c. În contextul teoriei probabilităţilor, inegalitatea lui Jensen este exprimată astfel: dacă X este o variabilă aleatoare şi f este o funcţie convexă, atunci $f(E[X]) \leq E[f(X)]$. (Similar, dacă f este funcţie concavă, atunci $f(E[X]) \geq E[f(X)]$.)
Demonstraţi această inegalitate în cazul în care X este variabilă aleatoare discretă cu un număr finit de valori (adică, $|Val\,(X)| < \infty$).

Răspuns:

a. Vom privi inegalitatea (106) ca pe o *ipoteză inductivă*, pe care o vom desemna cu $P(n)$, unde $n \in \mathbb{N}$, $n \geq 2$.[185] **Vom demonstra că $P(n)$ este adevărată, folosind *principiul inducţiei matematice*:**

$P(2)$: Dacă $x_1, x_2 \in \mathbb{R}$ şi $a_1, a_2 \in [0, 1]$ astfel încât $a_1 + a_2 = 1$, iar f este funcţie convexă, atunci inegalitatea $f(a_1x_1 + a_2x_2) \leq a_1f(x_1) + a_2f(x_2)$ se rescrie echivalent ca $f(a_1x_1 + (1 - a_1)x_2) \leq a_1f(x_1) + (1 - a_1)f(x_2)$. Această ultimă inegalitate coincide practic cu relaţia (105) din definiţia funcţiei convexe, deci este adevărată.

Presupunem că proprietatea $P(n)$ este adevărată şi vom demonstra $P(n + 1)$, adică: dacă $x_1, \ldots, x_n, x_{n+1} \in \mathbb{R}$ şi $a_1, \ldots, a_n, a_{n+1} \in [0, 1]$ astfel încât $\sum_{i=1}^{n+1} a_i = 1$, iar f o funcţie convexă, atunci rezultă că

$$f\left(\sum_{i=1}^{n+1} a_i x_i\right) \leq \sum_{i=1}^{n+1} a_i f(x_i). \tag{108}$$

Vom rescrie acum membrul stâng al acestei inegalităţi într-o formă convenabilă:

$$\begin{aligned} f\left(\sum_{i=1}^{n+1} a_i x_i\right) &= f\left(\sum_{i=1}^{n} a_i x_i + a_{n+1}x_{n+1}\right) \\ &= f\left(\Big(\sum_{i=1}^{n} a_i\Big) \cdot \sum_{i=1}^{n} \frac{a_i}{\sum_{i=1}^{n} a_i} x_i + a_{n+1}x_{n+1}\right) \\ &= f\left((1 - a_{n+1}) \cdot \sum_{i=1}^{n} \frac{a_i}{1 - a_{n+1}} x_i + a_{n+1}x_{n+1}\right) \end{aligned} \tag{109}$$

[184] Primul membru al *inegalităţii mediilor* este media aritmetică, iar cel de-al doilea este media geometrică.
[185] Inegalitatea (106) se verifică şi pentru $n = 1$, fiindcă $f(x_1) = f(x_1)$.

Este imediat că dacă vom considera $A_i \overset{not.}{=} \dfrac{a_i}{1-a_{n+1}}$**, pentru** $i=1,\ldots,n$**, atunci rezultă că** $A_i \geq 0$ **pentru** $i=1,\ldots,n$ **şi** $\sum_{i=1}^{n} A_i = \dfrac{\sum_{i=1}^{n} a_i}{1-a_{n+1}} = \dfrac{1-a_{n+1}}{1-a_{n+1}} = 1$.[186]

Prin urmare,

$$\begin{aligned}
f\Big(\sum_{i=1}^{n+1} a_i x_i\Big) &\overset{(109)}{=} f\Big((1-a_{n+1})\cdot \underbrace{\sum_{i=1}^{n} A_i x_i}_{x_1'} + a_{n+1}\underbrace{x_{n+1}}_{x_2'}\Big)\\
&\overset{f\ convexă}{\leq} (1-a_{n+1})\cdot f\Big(\sum_{i=1}^{n} A_i x_i\Big) + a_{n+1}\cdot f(x_{n+1})\\
&\overset{P(n)}{\leq} (1-a_{n+1})\cdot \sum_{i=1}^{n} A_i f(x_i) + a_{n+1}\cdot f(x_{n+1})\\
&= (1-a_{n+1})\cdot\left(\sum_{i=1}^{n} \frac{a_i}{1-a_{n+1}} f(x_i)\right) + a_{n+1}\cdot f(x_{n+1})\\
&= \sum_{i=1}^{n} a_i f(x_i) + a_{n+1} f(x_{n+1}) = \sum_{i=1}^{n+1} a_i f(x_i).
\end{aligned}$$

Aşadar, $f\Big(\sum_{i=1}^{n+1} a_i x_i\Big) \leq \sum_{i=1}^{n+1} a_i f(x_i)$**, deci proprietatea** $P(n+1)$ **este adevărată.**

Sumarizând, din faptul că $P(2)$ **este adevărată, iar implicaţia** $P(n) \Rightarrow P(n+1)$ **este adevărată, conform principiul inducţiei complete rezultă că proprietatea** $P(n)$ **este adevărată pentru orice** $n \in \mathbb{N}^* \setminus \{1\}$**.**

b. Fie numerele $x_1, x_2, \ldots, x_n$**, toate mai mari sau egale cu** 0**. Considerăm** $a_1 = a_2 = \ldots = a_n = \dfrac{1}{n}$ **(ceea ce implică** $\sum_{i=1}^{n} a_i = 1$**). Conform inegalităţii lui Jensen, în care vom alege pe postul funcţiei** f **funcţia** $\ln$ **(logaritmul având ca bază numărul** e**) care este concavă, putem scrie:**

$$\ln\Big(\sum_{i=1}^{n} a_i x_i\Big) \geq \sum_{i=1}^{n} a_i \ln(x_i) \Leftrightarrow \ln\Big(\frac{1}{n}\sum_{i=1}^{n} x_i\Big) \geq \frac{1}{n}\sum_{i=1}^{n} \ln(x_i) \Leftrightarrow$$

$$\ln\Big(\frac{1}{n}\sum_{i=1}^{n} x_i\Big) \geq \frac{1}{n}\ln\Big(\prod_{i=1}^{n} x_i\Big) \Leftrightarrow \ln\Big(\frac{1}{n}\sum_{i=1}^{n} x_i\Big) \geq \ln\Big(\prod_{i=1}^{n} x_i\Big)^{\frac{1}{n}} \Leftrightarrow$$

$$\ln\Big(\frac{1}{n}\sum_{i=1}^{n} x_i\Big) \geq \ln(\sqrt[n]{x_1 x_2 \ldots x_n}) \Leftrightarrow \frac{\sum_{i=1}^{n} x_i}{n} \geq \sqrt[n]{x_1 x_2 \ldots x_n}.$$

Ultima echivalenţă are loc întrucât funcţia $\ln$ **este strict crescătoare.**

c. Fie $f : \mathbb{R} \to \mathbb{R}$ **o funcţie convexă şi** X **o variabilă aleatoare discretă având următorul *tablou de repartiţie*:**

$$X : \begin{pmatrix} x_1 & x_2 & \ldots & x_n \\ p_1 & p_2 & \ldots & p_n \end{pmatrix}.$$

[186] Evident, A_i este bine definit dacă $a_{n+1} \neq 1$. Dacă $a_{n+1} = 1$, atunci $a_1 = \ldots = a_n = 0$, iar inegalitatea (108) devine în acest caz $f(x_{n+1}) \leq f(x_{n+1})$, care este adevărată în mod trivial.

Conform inegalităţii lui Jensen (pe care o putem aplica luând în locul „ponderilor“ a_i probabilităţile p_i, întrucât $p_i \geq 0$ şi $\sum_{i=1}^{n} p_i = 1$), rezultă:

$$f\Big(\sum_{i=1}^{n} p_i x_i\Big) \leq \sum_{i=1}^{n} p_i f(x_i) \Leftrightarrow f(E_P[X]) \leq E_P[f(X)].$$

În mod similar, dacă f este funcţie concavă, rezultă că $f(E_P[X]) \geq E_P[f(X)]$.

80. **(Găsirea optimului unei funcţii reale de gradul al doilea, folosind metoda analitică, metoda gradientului descendent şi metoda lui Newton)**

prelucrare de Liviu Ciortuz, după
■ • ***University of Utah, 2008 fall, Hal Daumé III, HW4, pr. 1***

Să presupunem că dorim să găsim minimul funcţiei $f(x) = 3x^2 - 2x + 1$.

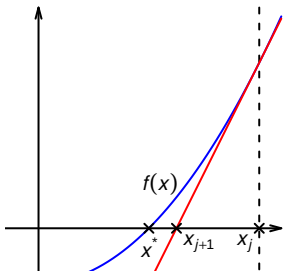
a. Verificaţi că această funcţie este convexă.

b. Ca urmare a punctului precedent, rezultă că funcţia f are un minim global. Găsiţi acest minim, folosind cunoştinţe de analiză matematică.

c. La acest punct vom căuta optimul funcţiei f aplicând algoritmul gradientului descendent.[187] Efectuaţi trei paşi ai acestui algoritm pornind de la punctul iniţial $x_0 = 1$ şi folosind rata de învăţare $\eta = 0.1$. Cât de aproape a ajuns algoritmul de soluţia reală?

d. În sfârşit, căutaţi din nou optimul funcţiei f aplicând de această dată metoda lui Newton,[188] pornind tot de la punctul $x_0 = 1$. Ce observaţi? (Câte iteraţii sunt necesare pentru ca algoritmul să conveargă la soluţia optimă?)

***Comentariu* [Metoda lui Newton]:**
În forma sa cea mai simplă, metoda lui Newton — cunoscută în literatura de specialitate şi sub numele de *metoda tangentei* — are ca obiectiv identificarea rădăcinii (sau, a rădăcinilor) unei funcţii f care este convexă şi derivabilă. (Rădăcinile unei funcţii sunt acele valori x^* ale argumentului funcţiei pentru care se anulează funcţia respectivă, adică $f(x^*) = 0$.) În acest scop, adică pentru aflarea rădăcinilor unei funcţii, metoda lui Newton, care este o metodă iterativă, foloseşte următoarea relaţie de „actualizare“:



$$x_{j+1} = x_j - \frac{f(x_j)}{f'(x_j)},$$

cu x_0 ales în mod arbitrar. Facem *observaţia* că metoda lui Newton poate fi aplicată (modificată uşor) şi în scopul găsirii optimului unei funcţii. În acest caz, se caută valorile argumentului x pentru care se anulează prima derivată a funcţiei obiectiv f. Prin urmare, pentru a putea aplica metoda lui Newton

[187] Pentru o prezentare formală a acestui algoritm, vedeţi enunţul problemei 167.
[188] Isaac Newton (1642-1727) a fost un matematician, fizician, astronom, alchimist şi teolog englez.

se cere ca funcţia f să fie dublu derivabilă (adică să existe atât prima cât şi cea de-a doua derivată a lui f). Relaţia de actualizare acum devine:

$$x_{j+1} = x_j - \frac{f'(x_j)}{f''(x_j)}. \tag{110}$$

La problema de faţă, se foloseşte metoda lui Newton în varianta aceasta, adică pentru aflarea optimului (mai precis, a minimului) unei funcţii.

Facem *observaţia* că formula (110) **se foloseşte exact în aceeaşi formă şi atunci când — în locul minimului — se caută maximul unei funcţii, folosind *metoda lui Newton*. (Pentru a vă convinge, este suficient să înlocuiţi f cu $-f$ în relaţia** (110)**.) În schimb, relaţia** (111)**, care este folosită de metoda *gradientului descendent* (pentru căutarea unui punct de minim) trebuie înlocuită cu $\Delta x = +\eta \cdot f'(x)$ în cazul folosirii metodei *gradientului ascendent* (pentru căutarea unui punct de maxim).**

Generalizarea* metodei lui Newton la cazul multidimensional, numită şi *metoda Newton-Raphson*,[189] lucrează cu *relaţia de actualizare

$$x_{j+1} = x_j - (H(x_j))^{-1} \nabla f(x_j).$$

Aici, $\nabla f(x)$ este, ca de obicei, vectorul gradient, format din derivatele parţiale ale lui $f(x)$, calculate în punctul x, iar $H(x)$ este *matricea hessiană* a lui f, formată din derivatele parţiale de ordinul al doilea, calculate în punctul x: $H_{ik}(x) = \frac{\partial^2 f(x)}{\partial x_i \, \partial x_k}$.

În mod obişnuit, metoda lui Newton are o rată / viteză de convergenţă mai mare decât metoda gradientului (varianta "batch") şi necesită mult mai puţine iteraţii pentru a ajunge foarte aproape de punctul de optim. Totuşi, trebuie reţinut că o [singură] iteraţie a algoritmului lui Newton poate fi mai costisitoare decât o iteraţie a metodei gradientului, fiindcă metoda lui Newton necesită atât calcularea cât şi inversarea matricei hessiene. Cu toate acestea, pentru valori ale lui d (numărul de dimensiuni) nu prea mari, metoda lui Newton este în mod obişnuit mult mai rapidă decât metoda gradientului.

Răspuns:

a. Pentru a studia convexitatea funcţiei $f(x)$ se calculează derivata a doua:

$$f''(x) = 6 > 0, \forall x \in \mathbb{R}.$$

Conform proprietăţii a.3 de la problema 78, rezultă că funcţia f este convexă (de fapt, ea este *strict* convexă) pe întreg domeniul ei de definiţie. Aşadar, ea are un (singur) punct de minim.

[189] Joseph Raphson (c.1668 - c.1715) a fost un matematician englez, contemporan cu Isaac Newton.

b. Pentru a calcula minimul funcţiei $f(x) = 3x^2 - 2x + 1$ **utilizăm derivata de ordinul întâi:**

$$f'(x) = 6x - 2.$$

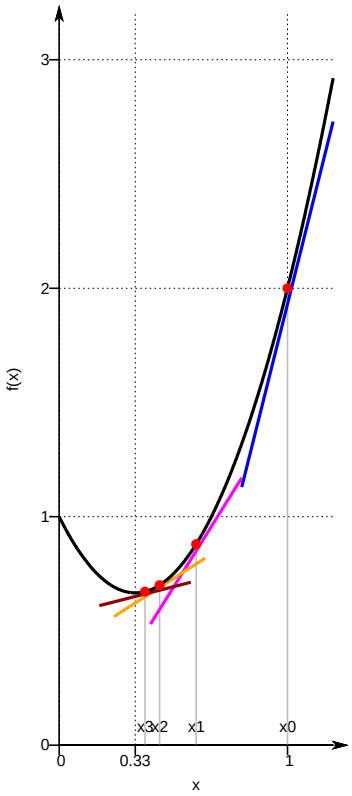
Punctul de minim este dat de soluţia ecuaţiei $f'(x) = 0$**, şi anume:** $x = \frac{1}{3} \approx 0.33$**.**

c. Având punctul iniţial $x_0 = 1$ **şi** $\eta = 0.1$**, se poate aplica metoda gradientului descendent conform formulelor:**

$$\begin{aligned} x_{n+1} &\leftarrow x_n + \Delta x \\ \Delta x &= -\eta \cdot f'(x) \end{aligned} \qquad (111)$$

Primii trei paşi ai algoritmului sunt următorii:

$$\begin{aligned} x_1 &= x_0 - \eta \cdot f'(x_0) = 1 - 0.1(6-2) \\ &= 1 - 0.4 = 0.6 \\ x_2 &= x_1 - \eta \cdot f'(x_1) = 0.6 - 0.1(6 \cdot 0.6 - 2) \\ &= 0.6 - 0.1 \cdot 1.6 = 0.6 - 0.16 = 0.44 \\ x_3 &= x_2 - \eta \cdot f'(x_2) = 0.44 - 0.1(6 \cdot 0.44 - 2) \\ &= 0.44 - 0.1 \cdot 0.64 = 0.44 - 0.064 = 0.376 \end{aligned}$$



Diferenţa dintre valoarea obţinută după efectuarea a trei paşi de gradient descendent şi valoarea pre-calculată a punctului de minim este: $0.376 - 0.333 = 0.043$**.**

***Observaţii*: Se constată uşor că**
i. **apropierea de punctul de optim este [mai] rapidă atâta timp cât valoarea primei derivate (i.e., panta tangentei la graficul funcţiei) este mare în valoare absolută;**
ii. **dacă rata de învăţare** η **a fost fixată la o valoare prea mare, atunci este posibil ca la un moment dat să depăşim punctul de optim şi apoi să „pendulăm“ în jurul lui.[190] Acest *punct slab* al metodei gradientului poate fi contracarat reducând în mod dinamic mărimea lui** η**. Altminteri, metoda gradientului are *avantajul* de a fi o *tehnică de optimizare* foarte simplă din punct de vedere conceptual şi uşor de implementat.[191]**

d. La aplicarea metodei lui Newton pentru funcţia dată în enunţ, la prima iteraţie vom proceda astfel:

$$x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)} = 1 - \frac{6 \cdot 1 - 2}{6} = 1 - \frac{2}{3} = \frac{1}{3}.$$

Se observă că $f'(x_1) = 0$**, iar dacă am mai continua să facem şi alte iteraţii, am obţine** $x_2 = x_3 = \ldots = x_1$**. Aşadar, cu metoda lui Newton, soluţia optimă se obţine (pentru această funcţie) într-o singură iteraţie.**

[190] Vedeţi ca exemplu problema 167.b.*ii*.
[191] Alte două puncte slabe ale metodei gradientului descendent sunt: imposibilitatea de a garanta găsirea optimului global, şi numărul mare de iteraţii care trebuie executate pe unele seturi de date reale.

***Observaţie* (1): Se poate constata uşor că soluţia optimă pentru funcţia f se obţine într-o singură iteraţie, indiferent de valoarea atribuită lui x_0:**

$$x_1 = x_0 - \frac{f'(x_0)}{f''(x_0)} = x_0 - \frac{6x_0 - 2}{6} = \frac{1}{3}.$$

Este interesant de reţinut faptul că această proprietate, care este valabilă de fapt pentru orice funcţie polinomială de gradul al doilea,[192] se întâlneşte şi în cazul rezolvării *regresiei liniare* cu ajutorul metodei lui Newton (vedeţi problema 7.b de la capitolul *Metode de regresie*).

***Observaţie* (2): La problema 45 se arată cum se folosesc metoda gradientului şi metoda lui Newton pentru a estima parametrul unei distribuţii probabiliste în sensul MLE.**

81. (Noţiunile de *subderivată*, *subdiferenţială* şi *subgradient*: definiţii, exemplificare)

prelucrare de Liviu Ciortuz, după
□ • ○ *CMU, 2009 fall, Carlos Guestrin, HW2, pr. 2.b*
CMU, 2016 fall, N. Balcan, M. Gormley, HW4, pr. 3.a

a. Fie $C \subseteq \mathbb{R}$ o mulţime convexă şi $f : C \to \mathbb{R}$ o funcţie convexă.[193] Numim *subderivată* lui f într-un punct $x_0 \in C$ un număr c astfel încât proprietatea $f(x) - f(x_0) \geq c(x - x_0)$ este satisfăcută pentru orice $x \in C$.[194] Cu alte cuvinte, un număr c este subderivată a funcţiei f în punctul x_0 dacă dreapta care trece prin punctul $(x_0, f(x_0))$ şi are panta c este situată sub graficul funcţiei f.

***Observaţie*: Se poate demonstra că în cazul în care funcţia f este derivabilă la stânga şi la dreapta în punctul $x_0 \in C$, mulţimea subderivatelor funcţiei f în punctul x_0 este un interval închis, nevid, de forma $[c_1, c_2]$, unde:**

$$c_1 = \lim_{x \nearrow x_0} \frac{f(x) - f(x_0)}{x - x_0} \text{ şi } c_2 = \lim_{x \searrow x_0} \frac{f(x) - f(x_0)}{x - x_0}.$$

Mulţimea tuturor subderivatelor lui f în punctul x_0 (în cazul de mai sus, intervalul $[c_1, c_2]$) se numeşte *subdiferenţiala* lui f în x_0 şi se notează cu $\partial f(x_0)$.[195]

***i*. Fie funcţia $f(x) = |x|$.[196] Care este mulţimea subderivatelor lui f în punctul 0?**

[192] Considerând $f(x) = ax^2 + bx + c$, rezultă $f'(x) = 2ax + b$ şi $f''(x) = 2a$. Aşadar, la aplicarea regulii [de actualizare] din metoda lui Newton, vom obţine:

$$x_{j+1} = x_j - \frac{f'(x_j)}{f''(x_j)} = x_j - \frac{2ax_j + b}{2a} = x_j - x_j - \frac{b}{2a} = -\frac{b}{2a},$$

adică exact abscisa punctului de optim al funcţiei de gradul al doilea.

Proprietatea aceasta — adică faptul că metoda lui Newton converge într-o singură iteraţie — se *generalizează* la funcţii de gradul al doilea cu un număr oarecare de variabile (nu doar una singură, cum a fost cazul mai sus); vedeţi problema 172.

[193] Pentru definiţiile noţiunilor de *mulţime convexă* şi *funcţie convexă*, vedeţi problema 78.

[194] Vedeţi proprietatea similară de la problema 78.a.2.

[195] În cazul funcţiilor derivabile, subdiferenţiala practic coincide cu derivata. Pentru alte câteva proprietăţi ale subdiferenţialei, vedeţi https://en.wikipedia.org/wiki/Subderivative.

[196] În legătură cu această funcţie, la problema 165.b se cere să se arate că este funcţie convexă.

ii. **Folosind următoarea**

***Teoremă*: Punctul x_0 este punct de minim global al funcţiei f dacă şi numai dacă $0 \in \partial f(x_0)$.**

arătaţi că $x_0 = 0$ este punctul [unic] de minim global al funcţiei $f(x) = |x|$.

b. Funcţia de cost / pierdere *hinge* este definită astfel:[197]

$$L(y, y') = \max(0, 1 - yy') \text{ pentru orice } y, y' \in \mathbb{R}.$$

Fie $x \in \mathbb{R}^d$ şi $y \in \mathbb{R}$. Cu ajutorul lui L, definim o nouă funcţie: $F(w) = L(y, w \cdot x)$, pentru orice $w \in \mathbb{R}^d$. Calculaţi subdiferenţiala lui F în w (considerând de data aceasta că w este fixat în $\mathbb{R}^d$). *Indicaţie*: Trebuie să trataţi 3 cazuri.

***Notă*: Noţiunile de subderivată şi subdiferenţială pot fi generalizate în mod natural la cazul funcţiilor de mai multe variabile. Dacă $f : U \to \mathbb{R}$ este o funcţie convexă cu valori reale definită pe o mulţime convexă deschisă din spaţiul euclidian $\mathbb{R}^n$, un vector v din acest spaţiu este un *subgradient* al funcţiei f în punctul $x_0 \in U$ dacă pentru orice $x \in U$ are loc inegalitatea**

$$f(x) - f(x_0) \geq v \cdot (x - x_0),$$

unde operatorul $\cdot$ desemnează produsul scalar. Mulţimea tuturor subgradienţilor lui f în punctul x_0 se numeşte *subdiferenţiala* lui f în x_0 şi se notează cu $\partial f(x_0)$.

***Observaţie*:** Pentru folosirea noţiunii / metodei subgradientului la conceperea [şi implementarea] unor algoritmi de învăţare automată, vedeţi problema 171 de la prezentul capitol, problema 28 de la capitolul *Metode de regresie* şi problemele 20, 21 şi 51 de la capitolul *Maşini cu vectori-suport*.

Răspuns:

a.*i*. Calculăm cele două limite indicate în enunţ:

$$\lim_{x \nearrow 0} \frac{f(x) - f(x_0)}{x - 0} = \lim_{x \nearrow 0} \frac{|x| - 0}{x} = \lim_{x \nearrow 0} \frac{-x}{x} = -1$$
$$\lim_{x \searrow 0} \frac{f(x) - f(x_0)}{x - 0} = \lim_{x \searrow 0} \frac{|x| - 0}{x} = \lim_{x \searrow 0} \frac{x}{x} = +1.$$

Aşadar, subdiferenţiala funcţiei $|x|$ în punctul 0 este intervalul $[-1, +1]$.

a.*ii*. Este imediat că

$$\partial f(x_0) = \begin{cases} -1 & \text{dacă } x_0 < 0 \\ 1 & \text{dacă } x_0 > 0. \end{cases}$$

Conform rezultatului de la punctul *a.i*, rezultă că $0 \in \partial f(0)$. Prin urmare, $x_0 = 0$ este punctul de minim global al funcţiei $f(x) = |x|$.

b. Fie $y, y' \in \mathbb{R}$, ca în enunţ. Pentru a uşura calculele, vom nota $z = yy'$ şi vom defini funcţia $L_1(z) = L(y, y') = \max(0, 1 - z)$.[198] **Este imediat că**

$$\frac{\partial}{\partial z} L_1(z) = \begin{cases} -1 & \text{dacă } z < 1 \\ 0 & \text{dacă } z > 1. \end{cases}$$

[197] Pentru exemplificarea modului în care funcţia de cost *hinge* este folosită în clasificarea automată, vedeţi problema 20 de la capitolul *Maşini cu vectori-suport*.
[198] Puteţi vedea graficul acestei funcţii la problema 88.

Mulţimea subderivatelor lui L_1 în punctul $z = 1$ este intervalul $[-1, 0]$.

Acum, renotând $z \stackrel{def.}{=} yw \cdot x$, putem scrie:[199]

$$\frac{\partial}{\partial w} F(w) = \frac{\partial}{\partial w} L(y, w \cdot x) = \frac{\partial}{\partial w} L_1(yw \cdot x) = \frac{\partial}{\partial z} L_1(z) \frac{\partial}{\partial w} z,$$

în cazul în care $z \neq 1$. Prin urmare,

$$\frac{\partial}{\partial w} F(w) = \frac{\partial}{\partial w} \max(0, 1 - yw \cdot x) = \begin{cases} -yx & \textbf{dacă } yw \cdot x < 1 \\ 0 & \textbf{dacă } yw \cdot x > 1. \end{cases}$$

În cazul vectorilor w_0 pentru care $yw_0 \cdot x = 1$, subdiferenţiala lui F în w_0 este formată din toţi vectorii $v \in \mathbb{R}^d$ care au proprietatea

$$\begin{aligned} F(w) - F(w_0) \geq v \cdot (w - w_0) &\Leftrightarrow L_1(yw \cdot x) - L_1(yw_0 \cdot x) \geq v \cdot (w - w_0) \\ &\Leftrightarrow L_1(yw \cdot x) - \underbrace{L_1(1)}_{0} \geq v \cdot (w - w_0) \Leftrightarrow L_1(yw \cdot x) \geq v \cdot (w - w_0) \\ &\Leftrightarrow \max(0, 1 - yw \cdot x) \geq v \cdot (w - w_0). \end{aligned}$$

Se observă că vectorul $v = 0$ din $\mathbb{R}^d$ satisface inegalitatea de mai sus. Aşadar, pentru aplicarea metodei subgradientului în conjuncţie cu funcţia de cost *hinge* definită prin $\max(0, yw \cdot x)$, este *suficient*(!) să considerăm

$$\frac{\partial}{\partial w} \max(0, 1 - yw \cdot x) = \begin{cases} -yx & \textbf{dacă } yw \cdot x < 1 \\ 0 & \textbf{dacă } yw \cdot x \geq 1. \end{cases}$$

În cazul $d = 1$, adică atunci când $w, x \in \mathbb{R}$, subdiferenţiala acestei funcţii *hinge* în punctul w_0 care satisface proprietatea $yw_0x = 1$ este intervalul $[-yx, 0]$.

82. (Problema de optimizare [convexă] cu restricţii — o variantă uşor simplificată: demonstrarea proprietăţii de dualitate „slabă“)

■ □ • ○ ***prelucrare de Liviu Ciortuz, după CMU, 2014 fall, E. Xing, B. Poczos, HW1, pr. 3.1***

***Introducere*:** În *cazul general*, o *problemă de optimizare convexă cu restricţii* este definită sub *forma primală* astfel:

$$\begin{aligned} & \min_x f(x) \\ \text{a. î.} \quad & g_i(x) \leq 0 \text{ pentru } i = 1, \ldots, m \\ & h_j(x) = 0 \text{ pentru } j = 1, \ldots, k, \end{aligned}$$

unde f (*funcţia obiectiv*) şi g_i sunt funcţii convexe, iar h_j sunt *funcţii afine* (adică funcţii liniare augmentate cu termen liber), cu $m \geq 0$ şi $k \geq 0$. Condiţiile $g_i(x) \leq 0$ şi $h_j(x) = 0$ se numesc *condiţii de fezabilitate primală*.

Pentru problema de optimizare definită mai sus, *lagrangeanul generalizat* se defineşte astfel:[200]

$$L(x, \alpha, \beta) \stackrel{def.}{=} f(x) + \sum_i \alpha_i g_i(x) + \sum_j \beta_j h_j(x),$$

[199] Este uşor de arătat că $\frac{\partial}{\partial w} z = \frac{\partial}{\partial w} yw \cdot x = yx$.

[200] După numele matematicianlui şi astronomului Joseph-Louis Lagrange (în italiană, Giuseppe Lodovico Lagrangia), 1736-1813. El a lucrat mai întâi în Italia (la Torino, oraşul său natal), apoi în Germania (la Berlin) şi în final în Franţa (la Paris).

unde $\alpha_i \geq 0$ pentru $i = 1, \ldots, m$ şi $\beta_j \in \mathbb{R}$ pentru $j = 1, \ldots, k$. Condiţiile $\alpha_i \geq 0$ se numesc *condiţii de fezabilitate duală.* Unii autori numesc funcţia L *lagrangeanul primal* şi o (re)notează cu L_P.

***Observaţie* (1):** Condiţia de convexitate asupra funcţiilor f şi g_i, precum şi condiţia ca funcţiile h_j să fie afine nu sunt necesare în demonstraţiile care urmează. Le-am scris aici pentru a asigura un cadru comun cu problema care urmează (83).

Fie următoarea problemă de optimizare cu restricţii în *formă primală*, unde f, h_1, $h_2 : \mathbb{R}^d \to \mathbb{R}$:

$$\begin{array}{rl} & \min_x f(x) \\ \textbf{a. î.} & h_1(x) \leq 0, \\ & h_2(x) = 0. \end{array}$$

a. Scrieţi *lagrangeanul generalizat* $L(x, \lambda, u)$, unde $\lambda \geq 0$ şi $u \in \mathbb{R}$ sunt *variabilele duale* care corespund inegalităţii şi respectiv egalităţii din cadrul problemei de optimizare.

b. Vom nota cu P aşa-numita *regiune fezabilă* a problemei de optimizare în forma primală, adică mulţimea formată din acele puncte $x \in \mathbb{R}^d$ pentru care $h_1(x) \leq 0$ şi $h_2(x) = 0$.

Demonstraţi că pentru orice $x \in P$ are loc egalitatea

$$f(x) = \max_{\lambda \geq 0, u} L(x, \lambda, u).$$

***Observaţie* (2):** Se poate demonstra că pentru orice problemă de optimizare convexă (vedeţi *Introducerea* de mai sus), adică în care funcţiile g_i sunt convexe, iar funcţiile h_j sunt afine, regiunea fezabilă este mulţime convexă.

c. Datorită rezultatului de la punctul b, problema de optimizare dată în enunţ (în forma primală) se poate rescrie echivalent (în mod compact) astfel:

$$\min_{x \in P} \max_{\lambda \geq 0, u} L(x, \lambda, u).$$

Mai departe, renunţând la restricţia $x \in P$ — adică, lucrând cu $x \in \mathbb{R}^d$, nerestricţionat[201] **— şi *inversând* cei doi operatori, $\min$ şi $\max$, vom obţine problema de optimizare convexă în *forma duală*:**[202]

$$\max_{\lambda \geq 0, u} \min_{x \in \mathbb{R}^d} L(x, \lambda, u).$$

Putem spune că funcţia obiectiv a problemei duale este (în manieră „conceptuală“) $g(\lambda, u) \overset{not.}{=} \min_{x \in \mathbb{R}^d} L(x, \lambda, u)$.

Arătaţi că

$$\min_{x \in P} f(x) \geq \max_{\lambda \geq 0, u} g(\lambda, u).$$

Atfel spus, valoarea optimă [a funcţiei obiectiv] din forma primală a problemei de optimizare este mai mare sau egală cu valoarea optimă [a funcţiei obiectiv]

[201] De fapt, se poate constata — pentru justificare (chiar pentru cazul general al problemei de optimizare cu restricţii), vedeţi documentul *Support Vector Machines* de Andrew Ng, Stanford University, CS229 Lecture Notes, Part V, pag. 8-9 — că problema de optimizare $\min_{x \in P} \max_{\lambda \geq 0, u} L(x, \lambda, u)$ este echivalentă cu problema de optimizare $\min_{x \in \mathbb{R}^d} \max_{\lambda \geq 0, u} L(x, \lambda, u)$.

[202] Observaţi că restricţiile asociate formei duale a problemei de optimizare (şi anume,$\lambda \geq 0$) sunt mult mai simple decât cele din forma primală!

din problema dală. Veţi putea constata uşor că acest rezultat se menţiune şi atunci când în problema de optimizare avem mai multe restricţii de tip inegalitate sau egalitate. Proprietatea aceasta se numeşte *dualitate slabă* (engl., weak duality).

***Observaţie* (3):** Se poate demonstra că pentru orice problemă de optimizare convexă (vedeţi *Introducerea* de mai sus), adică în care funcţiile f şi g_i sunt convexe, iar funcţiile h_j sunt afine, funcţia $g(\lambda, u) \stackrel{not.}{=} \min_{x\in\mathbb{R}^d} L(x, \lambda, u)$ este concavă,[203] deci putem să-i calculăm maximul aplicând o metodă de optimizare convexă asupra funcţiei $-g(\lambda, u)$.

Răspuns:

a. Lagrangeanul generalizat se obţine combinând (într-o singură funcţie) *funcţia obiectiv* şi funcţiile cu ajutorul cărora se scriu *restricţiile* din problema de optimizare convexă:

$$L(x, \lambda, u) \stackrel{def.}{=} f(x) + \lambda h_1(x) + u h_2(x), \text{ unde } \lambda \in \mathbb{R}_+ \text{ şi } u \in \mathbb{R}.$$

b. Dacă $x \in P$, regiunea fezabilă a problemei de optimizare în forma primală, atunci urmează că

$$f(x) \geq f(x) + \underbrace{\lambda}_{\geq 0}\underbrace{h_1(x)}_{\leq 0} + u\underbrace{h_2(x)}_{=0} \stackrel{def.}{=} L(x, \lambda, u) \text{ pentru orice } \lambda \geq 0 \text{ şi orice } u \in \mathbb{R}.$$

Mai departe, se observă că inegalitatea $f(x) \geq L(x, \lambda, u)$, pe care tocmai am obţinut-o, este satisfăcută cu egalitate dacă se ia $\lambda = 0$. Prin urmare, putem scrie $f(x) = \max_{\lambda\geq 0, u} L(x, \lambda, u)$.

c. Vom relua acum inegalitatea

$$f(x) \geq L(x, \lambda, u), \text{ pentru orice } x \in P,\ \lambda \geq 0,\ u \in \mathbb{R},$$

pe care am obţinut-o la punctul b. Considerând acum că $\lambda \geq 0$ şi $u \in \mathbb{R}$ sunt [aleşi în mod arbitrar dar] fixaţi, şi aplicând apoi operatorul $\min_{x\in P}$ la ambii membri ai acestei inegalităţi, vom obţine

$$\min_{x\in P} f(x) \geq \min_{x\in P} L(x, \lambda, u) \geq \min_{x\in\mathbb{R}^d} L(x, \lambda, u) \stackrel{not.}{=} g(\lambda, u).$$

Ultima inegalitate se justifică prin faptul că $P \subseteq \mathbb{R}^d$. Aşadar, sumarizând, ceea ce am obţinut până acum este inegalitatea

$$\min_{x\in P} f(x) \geq g(\lambda, u) \text{ pentru orice } \lambda \geq 0 \text{ şi orice } u \in \mathbb{R}.$$

În particular, inegalitatea are loc pentru acei $\lambda \geq 0$ şi $u \in \mathbb{R}$ pentru care se atinge maximul termenului drept al inegalităţii. Aşadar,

$$\min_{x\in P} f(x) \geq \max_{\lambda\geq 0, u} g(\lambda, u).$$

***Comentariu*:** Folosind notaţia din documentul *Support Vector Machines* de Andrew Ng, putem rescrie într-o manieră foarte sugestivă inegalitatea pe care tocmai am obţinut-o (adică, *proprietatea de dualitate slabă*):

$$p^* \geq d^*,$$

[203] Vedeţi documentul *Convex Optimization Overview (cont'd)* de Chuong B. Do, 2009.

unde prin p^* şi d^* am notat *valorile optime* pentru problema primală, respectiv problema duală. Este imediat că proprietatea care a fost demonstrată în această problemă are loc pentru *orice* problemă de optimizare cu restricţii.

83. **(Demonstrarea unei părţi din *teorema Karush-Kuhn-Tucker*, folosind puncte-şa: dacă sunt îndeplinite condiţiile Karush-Kuhn-Tucker, atunci dispunem de o soluţie pentru problema primală)**

□ • ○ ***prelucrare de Liviu Ciortuz, 2019, după CMU, 2015 fall, A. Smola, B. Poczos, HW3, pr. 4***

Fie problema de optimizare convexă cu restricţii

$$\begin{array}{ll} \min_x & f(x) \\ \textbf{a. î.} & g_i(x) \leq 0 \textbf{ pentru } i = 1, \ldots, m \\ & h_j(x) = 0 \textbf{ pentru } j = 1, \ldots, k, \end{array}$$

unde $x \in \mathbb{R}^d$, funcţiile $f, g_1, \ldots, g_m$ sunt convexe şi derivabile în raport cu x, iar $h_1, h_2, \ldots, h_k$ sunt funcţii afine (adică funcţii liniare augmentate cu termeni liberi). În cele ce urmează, această problemă de optimizare convexă va fi desemnată cu (P).[204]

***Comentariu*:**
Pentru o *problemă de optimizare convexă* cu restricţii, care este definită în *forma primală* (P), *condiţiile Karush-Kuhn-Tucker* sunt următoarele:

- fezabilitate $\begin{cases} \text{primală} & g_i(x^*) \leq 0 \text{ pentru } i = 1, \ldots, m \text{ şi} \\ & h_j(x^*) = 0 \text{ pentru } j = 1, \ldots, k \\ \text{duală} & \alpha_i^* \geq 0 \text{ pentru } i = 1, \ldots, m \end{cases}$
- complementaritate [duală]: $\alpha_i^* g_i(x^*) = 0$ pentru $i = 1, \ldots, m$
- staţionaritate (sau, optimalitate): $\dfrac{\partial}{\partial x} L(x^*, \alpha^*, \beta^*) = 0,$

unde $L(x, \alpha, \beta)$ este lagrangeanul generalizat pentru problema de optimizare convexă cu restricţii (P). (Vedeţi *Introducerea* la problema 82.) Precizăm că orice punct în care se anulează $\dfrac{\partial}{\partial x} L$ se numeşte *punct de staţionaritate*.

Teorema Karush-Kuhn-Tucker[205] afirmă următoarele:

(1) Dacă este îndeplinită condiţia de *dualitate tare* pentru problemele de optimizare (P) şi duala sa (D) — adică valorile lor optime satisfac relaţia $p^* = d^*$ —, atunci soluţiile acestor două probleme satisfac condiţiile Karush-Kuhn-Tucker.[206]

[204] Vă readucem aminte că această *formă* a problemei de optimizare convexă se numeşte [forma] *primală*.

[205] Teorema Karush-Kuhn-Tucker (abreviat, KKT) are o istorie interesantă. Ea a fost descoperită în 1939 de către studentul american William Karush, care a inclus-o în teza sa de master, dar n-a fost cunoscută până când a fost redescoperită în 1950 de către matematicienii Harold Kuhn (american) şi Albert Tucker (canadian). O variantă a acestei teoreme fusese descoperită şi de către Fritz John (german) în 1948, însă *Duke Mathematical Journal* i-a refuzat publicarea. Vedeţi Chuong B. Do, *Convex Optimization Overview (cont'd)*, 2009, pag. 7.

[206] Pentru modul cum este definită / obţinută forma duală (D) a problemei de optimizare convexă (P), vedeţi problema 82.c.

(2) Dacă x^*, α^* şi β^* satisfac condiţiile Karush-Kuhn-Tucker, atunci x^* este soluţie a problemei (P), iar α^*, β^* constituie o soluţie pentru problema (D).[207]

***Observaţie importantă*:** Sunt cunoscute anumite *condiţii suficiente* pentru ca să aibă loc dualitatea tare ($p^* = d^*$) şi, în consecinţă să fie satisfăcute condiţiile Karush-Kuhn-Tucker. Aceste condiţii suficiente sunt numite „condiţie de regularitate“ (engl., *regularity constraint*, sau *constraint qualification*). Una dintre ele, pe care o vom folosi adeseori, se enunţă după cum urmează.
Condiţia lui Slater pentru o problemă de optimizare convexă cu restricţii (P): dacă există x astfel încât $g_i(x) < 0$ pentru orice i şi $h_j(x) = 0$ pentru orice j, atunci $p^* = d^*$, adică are loc dualitatea tare.[208]

În acest exerciţiu vom demonstra prima parte a punctului (2) din Teorema Karush-Kuhn-Tucker: dacă x^*, α^* şi β^* satisfac condiţiile Karush-Kuhn-Tucker, atunci x^* este soluţie a problemei (P).

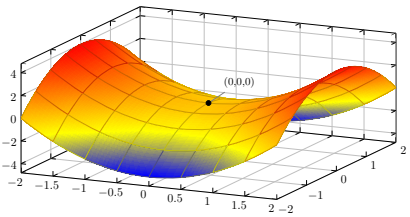
Pentru convenienţă, în cele ce urmează, vom renota cu λ ansamblul variabilelor duale α şi β.

A. Mai întâi vom demonstra că atunci când există $x^* \in \mathbb{R}^d$ şi $\lambda^* \in \mathbb{R}^m_+ \times \mathbb{R}^k$ care statisfac condiţiile Karush-Kuhn-Tucker pentru problema (P), urmează că (x^*, λ^*) este un punct-şa (engl., saddle point) pentru lagrangeanul generalizat $L(x, \lambda)$.

Un punct (x^*, λ^*) este numit *punct-şa* al funcţiei $L(x, \lambda)$ dacă are loc dubla inegalitate

$$L(x^*, \lambda) \leq L(x^*, \lambda^*) \leq L(x, \lambda^*) \text{ pentru } \forall x \in \mathbb{R}^d \text{ şi } \forall \lambda \in \mathbb{R}^m_+ \times \mathbb{R}^k. \tag{112}$$

***Exemplu*: Fie funcţia $g(x, y) = x^2 - y^2$. Se poate verifica imediat că originea sistemului de coordonate, adică punctul $(0, 0)$ este punct-şa pentru funcţia g: $-y^2 \leq x^2 - y^2 \leq x^2$ pentru orice x şi y din $\mathbb{R}$.**



a. Introduceţi multiplicatorii Lagrange $\lambda_1, \lambda_2, \ldots, \lambda_{m+k}$ şi scrieţi funcţia Lagrange generalizată care corespunde problemei (P). Scrieţi apoi *condiţiile Karush-Kuhn-Tucker* pentru problema de optimizare (P).

b. Folosiţi condiţiile de *fezabilitate primală* şi *fezabilitate duală*, precum şi condiţiile de *complementaritate* (engl., complementary slackness) pentru a demonstra că — atunci când există $x^* \in \mathbb{R}^d$ şi $\lambda^* \in \mathbb{R}^m_+ \times \mathbb{R}^k$ care statisfac aceste condiţii — prima jumătate a condiţiei din definiţia noţiunii de punct-şa este satisfăcută.

c. Folosiţi *condiţia de fezabilitate duală* pentru a demonstra că $L(x, \lambda^*)$ — care este ultimul membru din dubla inegalitate (112) — este o funcţie *convexă* în raport cu variabila x (considerând λ^* fixat) şi, prin urmare, *condiţia de*

[207] În condiţiile [mai restrictive] în care are loc dualitatea tare ($p^* = d^*$), teorema Karush-Kuhn-Tucker capătă forma unei echivalenţe (i.e., o condiţie necesară şi suficientă):
Dacă $p^* = d^*$, atunci x^* este soluţie a problemei primare, iar α^* şi β^* sunt soluţii ale problemei duale dacă şi numai dacă x^*, α^* şi β^* satisfac sistemul de condiţii Karush-Kuhn-Tucker.
[208] Cf. Morton Slater, *Lagrange multipliers revisited*, 1950.

staţionaritate **(engl., the stationary condition) implică faptul că cea de-a doua jumătate a condiţiei din definiţia noţiunii de punct-şa trebuie să fie satisfăcută. (*Sugestie*: Folosiţi proprietăţile funcţiilor convexe, pe care le-am demonstrat la problema 78.)**

B. În cele ce urmează vom arăta că pentru orice $x^* \in \mathbb{R}^d$ pentru care există un $\lambda_i^* \in \mathbb{R}_+^m \times \mathbb{R}^k$ astfel încât (x^*, λ^*) este punct-şa al funcţiei Lagrange generalizate $L(x, \lambda)$, rezultă că x^* este soluţie optimă pentru problema (P).

d. Arătaţi că atunci când prima jumătate a condiţiei (112) din definiţia noţiunii de punct-şa este satisfăcută, rezultă cu necesitate că $h_j(x^*) = 0$ pentru $j = 1, \ldots, k$, apoi $g_i(x^*) \leq 0$ pentru $i = 1, \ldots, m$ şi $\sum_{i=1}^m \lambda_i^* g_i(x^*) = 0$, după care putem conchide că $f(x^*) = L(x^*, \lambda^*)$.

e. Completaţi demonstraţia [de la punctul precedent], arătând că $L(x^*, \lambda^*)$ — adică, membrul din mijloc din dubla inegalitate care apare în definiţia noţiunii de punct-şa — este mărginit superior de p^*, valoarea optimă a funcţiei obiectiv pentru problema (P), şi apoi că $f(x^*) = p^*$, deci x^* este soluţie optimă a problemei (P).

Răspuns:

a. Folosind multiplicatorii Lagrange $\lambda_1, \lambda_2, \ldots, \lambda_{m+k}$, scriem lagrangeanul generalizat pentru problema de optimizare (P) care a fost dată în enunţ în modul următor:

$$L(x, \lambda) = f(x) + \sum_{i=1}^{m} \lambda_i g_i(x) + \sum_{j=1}^{k} \lambda_{m+j} h_j(x), \tag{113}$$

cu $\lambda_i \geq 0$ pentru $i \in [m]$, unde notaţia $[m]$ desemnează mulţimea $\{1, \ldots, m\}$.

Condiţiile Karush-Kuhn-Tucker pentru problema (P) se scriu astfel:

$$\begin{aligned}
\textbf{fezabilitate primală:}\quad & g_i(x^*) \leq 0, \forall i \in [m],\ h_j(x^*) = 0,\ \forall j \in [k];\\
\textbf{fezabilitate duală:}\quad & \lambda_i^* \geq 0, \forall i \in [m];\\
\textbf{complementaritate:}\quad & \lambda_i^* g_i(x^*) = 0, \forall i \in [m];\\
\textbf{staţionaritate / optimalitate:}\quad & \nabla f(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla g(x^*) + \sum_{j=1}^{k} \lambda_{m+j}^* \nabla h_j(x^*) = 0,
\end{aligned}$$

unde operatorul / simbolul ∇ desemnează vectorul de derivate parţiale $\dfrac{\partial}{\partial x}$.

b. Pentru orice $\lambda \in \mathbb{R}^{m+k}$ astfel încât $\lambda_i \geq 0$ pentru $i \in [m]$, vom avea:

$$\begin{aligned}
L(x^*, \lambda) &\overset{(113)}{=} f(x^*) + \sum_{i=1}^{m} \underbrace{\lambda_i}_{\geq 0} \underbrace{g_i(x^*)}_{\leq 0} + \sum_{j=1}^{k} \lambda_{m+j} \underbrace{h_j(x^*)}_{=0}\\
&\leq f(x^*)\\
&= f(x^*) + \sum_{i=1}^{m} \underbrace{\lambda_i^* g_i(x^*)}_{=0} + \sum_{j=1}^{k} \lambda_{m+j}^* \underbrace{h_j(x^*)}_{=0}\\
&\overset{(113)}{=} L(x^*, \lambda^*).
\end{aligned}$$

c. Întrucât f şi $g_1, g_2, \ldots, g_m$ sunt funcţii convexe, $\lambda_i \geq 0$ pentru $i = 1, \ldots, m$, iar funcţiile $h_1, \ldots, h_k$ sunt afine, deci convexe, ţinând cont de proprietăţile care au fost demonstrate la problema 78 — şi anume, că produsul unei constante nenegative cu o funcţie convexă este de asemenea funcţie convexă, iar suma a două (sau mai multe) funcţii convexe este tot o funcţie convexă — rezultă că expresia

$$L(x, \lambda^*) \overset{(113)}{=} f(x) + \sum_{i=1}^{m} \lambda_i^* g_i(x) + \sum_{j=1}^{k} \lambda_{m+j}^* h_j(x)$$

reprezintă şi ea o funcţie convexă în raport cu variabila x.

Ca o *consecinţă* a faptului că $L(x, \lambda^*)$ este funcţie convexă în raport cu variabila x, rezultă că valorile ei extreme sunt realizate în puncte pentru care vectorul gradient în raport cu x ia valoarea zero (mai exact, vectorul care are toate componentele 0). Condiţia de *stationaritate* Karush-Kuhn-Tucker afirmă că soluţia (x^*, λ^*) satisface această proprietate, aşadar $L(x^*, \lambda^*) \leq L(x, \lambda^*)$ pentru orice x.[209] Altfel spus, $L(x^*, \lambda^*)$ este o margine inferioară (engl., lower bound) pentru $L(x, \lambda^*)$.

$L(x, \lambda^*)$ este cel de-al treilea termen din dubla inegalitate (112), care constituie definiţia punctului-şa. Înseamnă că, în condiţiile date, şi a doua jumătate a acestei dublei inegalităţi este satisfăcută.

***Sumarizând* această primă parte a exerciţiului nostru, putem formula următoarea**
***Lemă* (1): Dacă perechea (x^*, λ^*) satisface condiţiile Karush-Kuhn-Tucker pentru problema (P), atunci (x^*, λ^*) este punct-şa pentru lagrangeanul generalizat $L(x, \lambda)$ asociat lui (P).**

d. Din prima jumătate a dublei inegalităţi (112), adică relaţia de definiţie pentru noţiunea de punct-şa, rezultă imediat că

$$\sup_{\lambda_i \geq 0,\ i \in [m]} L(x^*, \lambda) \leq L(x^*, \lambda^*). \tag{114}$$

Dacă prin reducere la absurd ar exista un $j \in [k]$ astfel încât $h_j(x^*) \neq 0$, atunci în expresia

$$L(x^*, \lambda) \overset{(113)}{=} f(x^*) + \sum_{i=1}^{m} \lambda_i g_i(x^*) + \sum_{j=1}^{k} \lambda_{m+j} h_j(x^*)$$

am putea lua $\lambda_j = sign(h_j(x^*))(+\infty)$, iar marginea inferioară din relaţia (114) ar deveni $+\infty$, ceea ce ar contrazice relaţia de definiţie pentru noţiunea de punct-şa (112). Aşadar, $h_j(x^*) = 0$ pentru orice $j \in [k]$.

În ceea ce priveşte $g_i(x^*)$, dacă prin reducere la absurd ar exista vreun $i \in [m]$ astfel încât $g_i(x) > 0$, atunci marginea inferioară din relaţia (114) ar deveni $+\infty$ dacă se consideră $\lambda_i = +\infty$. Prin urmare, $g_i(x^*) \leq 0$ pentru orice $i \in [m]$.

Până aici am demonstrat că, în condiţiile date, cele două condiţii de fezabilitate primală din problema de optimizare convexă (P) sunt îndeplinite.

Acum, dacă $\lambda_i \geq 0$ pentru $i \in [m]$, rezultă $\sum_{i=1}^{m} \lambda_i g_i(x) \leq 0$. Mai mult, se observă că există o soluţie trivială, şi anume $\lambda_i = 0$ pentru $i \in [m]$, pentru ca această inegalitate să fie satifăcută cu egalitate, adică $\sum_{i=1}^{m} \lambda_i g_i(x) = 0$. Aşadar,

[209] Vedeţi *proprietatea* 2 de la problema 78.a.

$$\sup_{\lambda_i\geq 0, i\in[m]} L(x^*,\lambda) = \sup_{\lambda_i\geq 0, i\in[m]} \Big(f(x^*) + \sum_{i=1}^{m} \lambda_i \underbrace{g_i(x^*)}_{\leq 0} + \sum_{i=1}^{k} \lambda_{m+i} \underbrace{h_i(x^*)}_{=0} \Big) = f(x^*).$$

Mai mult, relaţia (114) **devine** $\sup_{\lambda_i\geq 0,\, i\in[m]} L(x^*,\lambda) = L(x^*,\lambda^*)$ **întrucât prin ipoteză avem** $\lambda_i^* \geq 0$ **pentru** $i \in [m]$**. Prin urmare, ajungem la concluzia care a fost formulată în enunţ:**

$$f(x^*) = \sup_{\lambda_i\geq 0, i\in[m]} L(x^*,\lambda) = L(x^*,\lambda^*).$$

e. La punctul precedent am demonstrat că x^***, prima componentă din punctul-şa** (x^*,λ^*)**, satisface toate restricţiile din problema de optimizare dată şi, de asemenea, că** $L(x^*,\lambda^*) = f(x^*)$**. Aici vom arăta că, în plus,** $f(x^*) \leq \inf_{x\in X} f(x)$ **(vedeţi mai jos relaţia** (115)**), de unde va rezulta că** x^* **este soluţie optimă pentru problema (P), deci** $f(x^*) = p^* = L(x^*,\lambda^*)$**.**

Fie X **spaţiul / mulţimea tuturor soluţiilor fezabile pentru problema de optimizare dată. Are loc următoarea inegalitate multiplă:**

$$L(x^*,\lambda^*) \leq \inf_x L(x,\lambda^*) \leq \inf_{x\in X} L(x,\lambda^*) \leq \inf_{x\in X} f(x). \tag{115}$$

Într-adevăr, prima dintre aceste inegalităţi este dată de cea de-a doua jumătate a relaţiei (112)**. A doua inegalitate din** (115) **are loc pur şi simplu fiindcă mulţimea** X **este o submulţime din** $\mathbb{R}^d$**. În fine, pentru a justifica ultima inegalitate din** (115)**, vom demonstra următoarea *proprietate* (chiar dacă este uşor mai generală decât este nevoie aici):**

[*Propoziţie*]: Pentru orice soluţie fezabilă $\bar{x}$ **pentru problema (P) şi pentru orice** $\lambda \in \mathbb{R}^{m+k}$ **astfel încât** $\lambda_i \geq 0$ **pentru orice** $i \in [m]$ **rezultă că următoarea inegalitate este satisfăcută:**

$$L(\bar{x},\lambda) \leq f(\bar{x}).$$

Într-adevăr, dacă x **o soluţie fezabilă pentru problema (P), ea satisface toate restricţiile, adică** $g_i(\bar{x}) \leq 0$ **pentru** $i \in [m]$ **şi** $h_{m+j}(\bar{x}) = 0$ **pentru** $j \in [k]$**. Întrucât** $\lambda_i \geq 0$ **pentru orice** $i \in [m]$**, urmează că** $\lambda_i g_i(\bar{x}) \leq 0$ **pentru orice** $i \in [m]$**. Aşadar, rezultă că**

$$L(\bar{x},\lambda) = f(\bar{x}) + \sum_{i=1}^{m} \underbrace{\lambda_i}_{\geq 0} \underbrace{g_i(\bar{x})}_{\leq 0} + \sum_{i=1}^{k} \lambda_{m+i} \underbrace{h_i(\bar{x})}_{=0} \leq f(\bar{x}). \tag{116}$$

***Observaţie*: Ultimul termen din relaţia** (115) **este exact** p^***, valoarea optimă a funcţiei obiectiv pentru problema (P), problemă care se poate scrie echivalent sub forma** $\inf_{x\in X} f(x)$**. Aşadar,** $L(x^*,\lambda^*) \leq p^*$**. Întrucât la punctul** d **am obţinut că** $L(x^*,\lambda^*) = f(x^*)$**, rezultă că** $L(x^*,\lambda^*) = p^*$**.**

***Sumarizând* cea de-a doua parte a exerciţiului nostru, putem formula următoarea**
***Lemă* (2): Dacă** (x^*,λ^*) **este punct-şa pentru lagrangeanul generalizat** $L(x,\lambda)$ **asociat problemei (P), atunci** x^* **este soluţie optimă pentru (P).**

Punând impreună cele două părţi ale exerciţiului nostru (altfel spus, combinând *Lema* (1) şi *Lema* (2)), rezultă că putem formula următoarea

***Propoziţie*: Dacă perechea (x^*, λ^*) satisface condiţiile Karush-Kuhn-Tucker pentru problema de optimizare convexă cu restricţii (P), atunci x^* este soluţie optimă pentru (P).**[210]

84. **(Aplicarea metodei dualităţii Lagrange, folosind condiţiile Karush-Kuhn-Tucker, la rezolvarea unei probleme de optimizare în care atât funcţia obiectiv, cât şi restricţiile sunt funcţii pătratice; punerea în evidenţă a punctului-şa pentru lagrangeanul generalizat, şi a legăturii sale cu soluţiile problemelor dual şi primală)**

prelucrare de Liviu Ciortuz, 2019, după
• ◦ University of Helsinki, 2014 spring, Jyrki Kivinen, HW5, pr. 1

Considerăm funcţiile $f(x) = x^2$ şi $g(x) = (x-2)^2$. Fie *problema de optimizare* convexă

$$\begin{array}{l} \min_x \ f(x) \\ \textbf{a. î.} \quad g(x) \leq 1. \end{array} \qquad \textbf{(P)}$$

a. Scrieţi lagrangeanul generalizat $L_P(x, \alpha)$, unde $\alpha \geq 0$ este multiplicatorul Lagrange asociat *restricţiei* din problema (P). Folosind Maple (sau un alt soft matematic), faceţi *graficul* acestei funcţii. (Veţi obţine un grafic tridimensional.)

b. Rezolvaţi *problema* dată, folosind sistemul reprezentat de *condiţiile Karush-Kuhn-Tucker*.

c. Daţi expresia analitică a funcţiilor $x \mapsto \max_{\alpha \geq 0} L_P(x, \alpha)$ şi $\alpha \mapsto \min_x L_P(x, \alpha)$. Indicaţi punctul (x^*, α^*) în care cele două funcţii au aceeaşi valoare. Am folosit următoarele notaţii: $x^* = \arg\min_x \max_{\alpha \geq 0} L_P(x, \alpha)$ şi $\alpha^* = \arg\max_{\alpha \geq 0} \min_x L_P(x, \alpha)$. (Vedeţi problema 82.c.)

***Sugestie*: Ar fi util să reprezentaţi aceste două funcţii pe un acelaşi sistem [de două axe] de coordonate, considerând atât variabila x cât şi variabila α reprezentate pe axa orizontală.**[211]

d. Scrieţi *forma duală* a problemei (P), iar apoi rezolvaţi problema duală (D).

e. Înlocuiţi *restricţia* din *problema* de optimizare dată mai sus cu $g(x) \leq 8$ şi apoi rezolvaţi noua problemă (notată cu (P′)), urmând aceleaşi cerinţe ca mai sus.[212]

Răspuns:

a. Expresia lagrangeanului generalizat al problemei de optimizare (P) se scrie astfel:

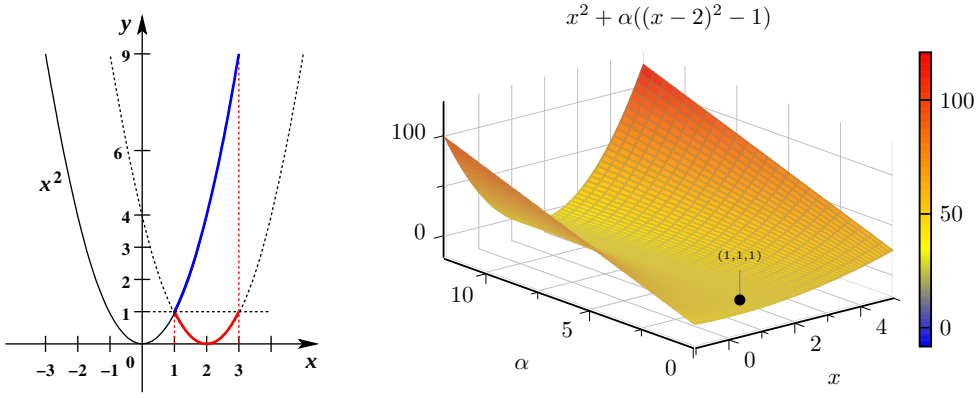
$$L_P(x, \alpha) = x^2 + \alpha((x-2)^2 - 1).$$

[210] Teorema Karush-Kuhn-Tucker — vedeţi *Comentariul* de la pagina 187 — afirmă că în aceleaşi condiţii (ca în această ultimă *Propoziţie*) rezultă că λ^* este soluţie optimă pentru duala problemei (P).

[211] *Atenţie*: Pentru prima funcţie, $x \mapsto \max_{\alpha \geq 0} L_P(x, \alpha)$, nu veţi reprezenta punctele în care ea ia valoarea $+\infty$.

[212] Veţi observa că spre deosebire de punctele $a - d$, unde restricţia $g(x) \leq 1$ era „activă“ (fiindcă optimul global al funcţiei $f(x)$ nu este în interiorul mulţimii fezabile), aici restricţia $g(x) \leq 8$ este superfluă.

Am reprezentat funcţia obiectiv, precum şi funcţia corespunzătoare restricţiei din cadrul acestei probleme de optimizare în graficul de mai jos, partea stângă, iar pentru lagrangeanul generalizat L_P am obţinut graficul din partea dreaptă.[213] Pe acest al doilea grafic am marcat (cu simbolul •) punctul-şa al acestui lagrangean, şi anume punctul $(1,1,1)$, care corespunde lui $x^* = 1$, $\alpha^* = 1$, $L_P(x^*, \alpha^*) = 1$ (vedeţi rezolvarea punctului c al problemei 83).



b. Condiţiile Karush-Kuhn-Tucker corespunzătoare problemei (P) sunt următoarele:

$$\begin{aligned} (x-2)^2 - 1 &\leq 0 && \textbf{fezabilitate primală} \\ \alpha &\geq 0 && \textbf{fezabilitate duală} \\ \alpha((x-2)^2 - 1) &= 0 && \textbf{complementaritate} \\ \frac{\partial L_P}{\partial x} = 2x + \alpha(2x-4) &= 0 && \textbf{staţionaritate (optimalitate)} \end{aligned}$$

Condiţia de *staţionaritate* se poate scrie echivalent astfel:

$$\alpha = \frac{x}{2-x}, \text{ dacă } x \neq 2 \tag{117}$$

sau, altfel:

$$x = \frac{2\alpha}{1+\alpha} = 2 - \frac{2}{1+\alpha} \text{ (remarcaţi că } \alpha \geq 0 \Rightarrow \alpha \neq -1). \tag{118}$$

Vom începe rezolvarea sistemului de condiţii Karush-Kuhn-Tucker pornind de la condiţia de *complementaritate*. Ca de obicei, vom avea de tratat două cazuri.

În primul caz, vom considera că $\alpha = 0$. (Observaţi că, în mod implicit, este satisfăcută condiţia de *fezabilitate duală*.) Conform relaţiei (118), rezultă $x = 0$, însă constatăm imediat că această valoare nu satisface condiţia de *fezabilitate primală*.

În al doilea caz, vom considera că $\alpha \neq 0$ sau, mai precis (datorită condiţiei de *fezabilitate duală*) $\alpha > 0$. Din egalitatea $\alpha((x-2)^2 - 1) = 0$ rezultă că

[213] Analizând graficul din partea stângă se poate constata că *regiunea fezabilă* pentru problema (P), adică mulţimea $\{x | g(x) \leq 1\}$ este intervalul $[1,3]$. La efectuarea graficului lagrangeanului generalizat $L_P(x,\alpha)$ (şi, de fapt, chiar mai înainte, la stabilirea domeniului de definiţie pentru L_P), nu trebuie să se considere că x aparţine doar intervalului $[1,3]$. Corect este să considerăm $L_P : \mathbb{R} \times [0, +\infty)$. Vă readucem aminte că restricţia $x \in [1,3]$ a fost deja înglobată în expresia lagrangeanului generalizat $L_P(x,\alpha) \stackrel{def.}{=} f(x) + \alpha(g(x) - 1)$.

$(x-2)^2-1=0$. **(Observaţi că, în mod implicit, este satisfăcută condiţia de *fezabilitate primală*.) Ecuaţia** $(x-2)^2-1=0$ **are două soluţii:** $x_1=1$ **şi** $x_2=3$. **Conform relaţiei** (117), **rezultă** $\alpha_1=\dfrac{1}{2-1}=1\geq 0$ **şi respectiv** $\alpha_2=\dfrac{3}{2-3}=-3\leq 0$. **Aşadar, numai** α_1 **satisface condiţia de *fezabilitate duală*.**

***Sumarizând*, rezultă că singura soluţie a sistemului de condiţii Karush-Kuhn-Tucker este perechea** $(x^*=1,\ \alpha^*=1)$. **Conform *teoremei Karush-Kuhn-Tucker* (mai exact, vedeţi proprietatea demonstrată la problema 83.B) rezultă că *soluţia optimă* a problemei (P) este** $x=1$. **Valorea optimă a funcţiei obiectiv pentru această problemă este** $p^*\overset{not.}{=}1$.

c. Vom calcula mai întâi expresia analitică a funcţiei $x\mapsto \max_{\alpha\geq 0} L_P(x,\alpha)$.

Aşadar, pentru un $x\in\mathbb{R}$ **oarecare, dar fixat, vom avea:**

$$\max_{\alpha\geq 0} L_P(x,\alpha)=\max_{\alpha\geq 0}(x^2+\alpha((x-2)^2-1))=x^2+\max_{\alpha\geq 0}(\alpha((x-2)^2-1)). \qquad (119)$$

Observaţi că

$$\begin{array}{ll}(x-2)^2-1=0 & \textbf{pentru } x\in\{1,3\}\\ (x-2)^2-1<0 & \textbf{pentru } x\in(1,3)\\ (x-2)^2-1>0 & \textbf{pentru } x\in(-\infty,1)\cup(3,+\infty),\end{array}$$

ceea ce implică

$$\max_{\alpha\geq 0}\alpha((x-2)^2-1)=\begin{cases}0 & \textbf{pentru } x\in[1,3]\\ +\infty & \textbf{în caz contrar}.\end{cases}$$

Ţinând cont de acest rezultat intermediar, relaţia (119) **devine:**

$$\max_{\alpha\geq 0} L_P(x,\alpha)=\begin{cases}x^2 & \textbf{pentru } x\in[1,3]\\ +\infty & \textbf{în rest}.\end{cases}$$

Este imediat că valoarea minimă a acestei funcţii — vă reamintim că este vorba despre funcţia $x\mapsto\max_{\alpha\geq 0} L_P(x,\alpha)$ **— este** $p^*=1$, **obţinută pentru** $x^*=1$.

***Observaţie importantă* (1):**
Mai general, se poate demonstra,[214] elaborând un raţionament similar cu cel de mai sus, că pentru orice problemă de optimizare convexă, pentru care funcţia obiectiv este f, dacă notăm cu X mulţimea punctelor fezabile ale acestei probleme şi cu α şi β *multiplicatorii Lagrange* asociaţi restricţiilor (de tip inegalitate şi respectiv egalitate) din problema de optimizare, rezultă că funcţia $x\mapsto\max_{\alpha\geq 0,\ \beta} L_P(x,\alpha,\beta)$ este egală cu

$$\begin{cases}f(x) & \text{pentru orice } x\in X,\\ +\infty & \text{în rest.}\end{cases}$$

Mai mult, avem că

$$\min_{x\in X} f(x)=\min_{x}\ \max_{\alpha\geq 0,\ \beta} L_P(x,\alpha,\beta).$$

Observaţi că x apare nerestricţionat în partea dreaptă a acestei ultime egalităţi, iar restricţiile asupra lui α sunt foarte simple. Aşadar, noua problemă de optimizare (cea din partea dreaptă a ultimei egalităţi) se exprimă mai simplu decât problema iniţială.

[214] Vedeţi Chuong B. Do, *Convex Optimization Overview (cont'd)*, 2009, pag. 4-5.

Acum vom calcula expresia analitică a funcţiei $\alpha \mapsto \min_x L_P(x,\alpha)$. Pentru un $\alpha \geq 0$ oarecare, dar fixat, vom avea:[215]

$$\begin{aligned}\min_x L_P(x,\alpha) &= \min_x(x^2+\alpha((x-2)^2-1)) = \min_x((1+\alpha)x^2-4\alpha x+3\alpha)\\ &= -\frac{4\alpha^2-3\alpha(1+\alpha)}{1+\alpha} = \frac{\alpha(3-\alpha)}{1+\alpha}\\ &= -\alpha+\frac{4\alpha}{1+\alpha} = -\alpha+4-\frac{4}{1+\alpha}.\end{aligned} \tag{120}$$

Vom nota această funcţie cu $L_D(\alpha)$.[216]

Folosind mijloace clasice de analiză matematică din liceu, vom demonstra că valoarea maximă a acestei funcţii este $d^* = 1$ şi ea se obţine pentru $\alpha^* = 1$.

$$L_D(\alpha) = -\alpha+4-\frac{4}{1+\alpha}$$

$$\Rightarrow L'_D(\alpha) = -1+\frac{4}{(1+\alpha)^2} \tag{121}$$

$$= \frac{4-(1+\alpha)^2}{(1+\alpha)^2} = \frac{(2-(1+\alpha))(2+(1+\alpha))}{(1+\alpha)^2} = \frac{(1-\alpha)(3+\alpha)}{(1+\alpha)^2}$$

$$\overset{(121)}{\Rightarrow} L''_D(\alpha) = -\frac{8}{(1+\alpha)^3}. \tag{122}$$

Rezultă că $L''_D(\alpha) < 0$ pentru orice $\alpha \geq 0$ (deci lagrangeanul dual este funcţie *concavă*!), în vreme ce $L'_D(\alpha) = 0$ pentru $\alpha = 1$, $L'_D(\alpha) < 0$ pentru $\alpha = (1,+\infty)$, iar $L'_D(\alpha) > 0$ pentru $\alpha = [0,1)$. Prin urmare, funcţia $L_D(\alpha)$ este strict crescătoare pe intervalul $[0,1]$ şi strict descrescătoare pe intervalul $[1,+\infty)$. Rezultă că, într-adevăr, valoarea maximă a funcţiei $L_D(\alpha)$ este 1, iar această valoare maximă se obţine pentru $\alpha = 1$.

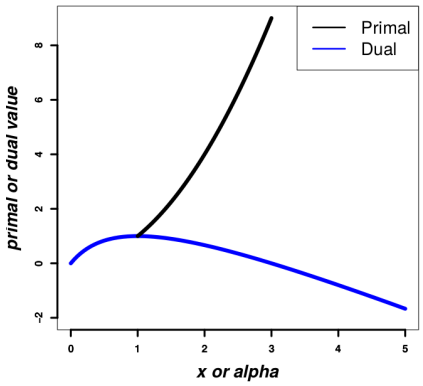
[215] La prima egalitate de pe rândul al doilea (120) de la calculul expresiei lui $\min_x L_P(x,\alpha)$ am aplicat formula care ne dă optimul funcţiei de gradul al doilea ax^2+bx+c, şi anume $\frac{-\Delta}{4a}$, cu $\Delta = b^2-4ac$. De fapt, atunci când $b = 2b'$ (aşa cum se întâmplă în cazul de faţă), optimul se calculează mai simplu astfel: $-\frac{\Delta'}{a}$, unde $\Delta' = b'^2-ac$. Vă readucem aminte că $\sqrt{\Delta}$ intervine în calculul rădăcinilor ecuaţiei $ax^2+bx+c=0$: $x_{1,2} = -\frac{-b\pm\sqrt{\Delta}}{2a} = -\frac{-b'\pm\sqrt{\Delta'}}{a}$. Această ultimă variantă este întâlnită uneori în manuale sub numele de *formula* [cu Δ] *pe jumătate*, fiindcă $\sqrt{\Delta} = 2\sqrt{\Delta'}$.

Remarcaţi de asemenea faptul că expresia (118) reprezintă chiar abscisa punctului de optim pentru funcţia $L_P(x,\alpha) = (1+\alpha)x^2-4\alpha x+3\alpha$, atunci când parametrul α se consideră fixat. Valoarea pentru această abscisă se poate calcula folosind formula $x_{min} = \frac{-b}{2a} = \frac{b'}{a}$. Mai târziu (vedeţi punctul *d*), vom nota această abscisă cu $\arg\min_x L_P(x,\alpha)$.

[216] O astfel de funcţie se numeşte îndeobşte *lagrangeanul dual* corespunzător problemei (P). El ve fi funcţia obiectiv a *formei duale* a problemei de optimizare (P). Vedeţi punctul *d*.

Întrucât $p^* = d^*$, conform primei părţi a teoremei Karush-Kuhn-Tucker rezultă că sistemul de condiţii Karush-Kuhn-Tucker este satisfăcut şi putem conchide (vedeţi rezultatul teoretic demonstrat la problema 83.A) că punctul $(x^* = 1, \alpha^* = 1)$ este punct-şa (ba mai mult, chiar unicul punct-şa) pentru lagrangeanul generalizat $L_P(x, \alpha)$.[217]

În imaginea alăturată am reprezentat într-un singur grafic cele două funcţii calculate la acest punct.



d. Funcţia obiectiv pentru *duala unei probleme de optimizare convexă oarecare* se defineşte ca $L_D(\alpha) \overset{not.}{=} \min_x L_P(x, \alpha, \beta)$, unde $L_P(x, \alpha, \beta)$ este lagrageanul generalizat asociat problemei (P), cu α şi β multiplicatorii Lagrange corespunzători restricţiilor de tip inegalitate şi, respectiv, egalitate.

În cazul problemei noastre de optimizare (P), ţinând cont de rezolvarea punctului c, rezultă că forma duală corespunzătoare lui (P) este următoarea:

$$\max_{\alpha \geq 0} L_D(\alpha) = \max_{\alpha \geq 0} \Big(-\alpha + 4 - \frac{4}{1+\alpha} \Big).$$

Calculele de la punctul c arată că soluţia acestei probleme duale este $\alpha^* = 1$, iar valoarea optimă pentru funcţia obiectiv a acestei probleme este $d^* = 1$.

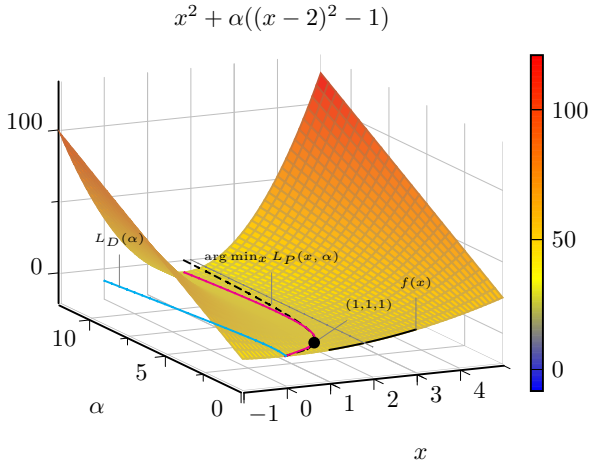
***Observaţie importantă* (2):**
Era de aşteptat să obţinem $p^* = d^*$ (vedeţi rezolvarea punctului c). Justificarea ţine de faptul că pentru problema (P) este satisfăcută *condiţia lui Slater*: $\exists x : g(x) < 0$, adică $\exists x : (x-2)^2 < 1$, deci $p^* = d^*$ (adică are loc *dualitatea tare*).[218] Prima parte a *teoremei Karush-Kuhn-Tucker* afirmă că în aceste condiţii soluţiile problemelor (P) şi (D) (duala lui (P)) satisfac sistemul de condiţii Karush-Kuhn-Tucker.
Pe noi însă la acest exerciţiu nu ne-a interesat această chestiune / legătură, ci (în special la punctul b) cealaltă parte a teoremei Karush-Kuhn-Tucker: dacă sistemul de condiţii Karush-Kuhn-Tucker este satisfăcut, atunci dispunem în mod natural de o soluţie pentru problemele (P) şi (D). Mai mult, la punctele c şi d am ilustrat rezultatele teoretice care au fost demonstrate la problema 83.

[217] Faptul că punctul $(x^* = 1, \alpha^* = 1)$ satisface condiţia de punct-şa (vedeţi relaţia (112)) se poate verifica şi în mod direct:

$$\begin{aligned} & L_P(x^*, \alpha) \leq L_P(x^*, \alpha^*) \leq L_P(x, \alpha^*) \\ \overset{x^*=1, \alpha^*=1}{\Leftrightarrow} \; & 1 + \alpha(1-1) \leq 1 + 1(1-1) \leq x^2 + (x-2)^2 - 1 \\ \Leftrightarrow \; & 1 \leq 1 \leq 2(x-1)^2 + 1, \; \forall x, \; \forall \alpha \geq 0. \end{aligned}$$

[218] Vedeţi *Observaţia importantă* de la pagina 188.

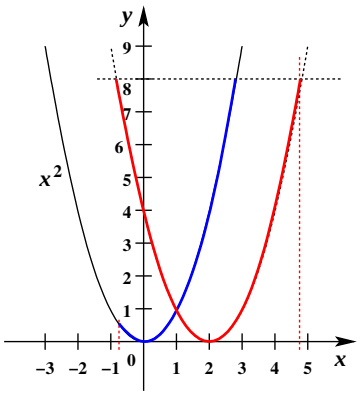
Figura alăturată încearcă să ofere o imagine de *sinteză* (din punctul de vedere al reprezentărilor grafice) pentru raţionamentele pe care le-am dezvoltat în cursul rezolvării problemei de optimizare convexă (P).
Cele două funcţii care au fost calculate la punctul c sunt reprezentate aici astfel: prima (în negru) în planul de ecuaţie $\alpha = 0$, iar cea de-a doua (în bleu) în planul de ecuaţie $x = 0$.

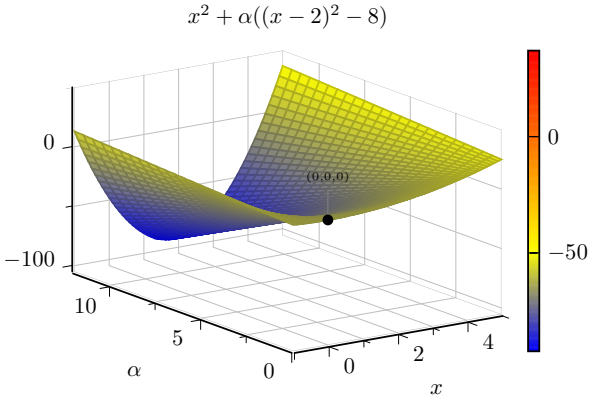


Pe lângă acestea, am reprezentat funcţia $\alpha \mapsto \arg\min_x L_P(x, \alpha)$ sub forma unei curbe punctate, în planul de ecuaţie $z = 0$, unde z notează cea de-a treia coordonată. Punctele de pe curba de culoare magenta (roz) au coordonatele $(\arg\min_x L_P(x, \alpha), \alpha, L_D(\alpha))$. Aşadar, linia punctată reprezintă proiecţia acestei curbe pe planul $z = 0$, în vreme ce curba în bleu este proiecţia ei pe planul $x = 0$.

e. În cazul $g(x) \leq 8$, noul lagrangean generalizat este $L_{P'}(x, \alpha) = x^2 - \alpha((x - 2)^2 - 8)$.

Am reprezentat funcţia obiectiv şi funcţia corespunzătoare restricţiei din cadrul aceste noi probleme de optimizare în graficul de mai jos, partea stângă, iar pentru lagrangeanul generalizat $L_{P'}$ am obţinut graficul din partea dreaptă:





Condiţiile de *fezabilitate primală* şi respectiv *complementaritate* devin:

$$(x-2)^2 - 8 \leq 0$$
$$\alpha((x-2)^2 - 8) = 0,$$

iar celelalte condiţii Karush-Kuhn-Tucker rămân neschimbate. Ca o consecinţă, se poate constata imediat că relaţiile (117) şi (118) sunt valabile şi aici.

La rezolvarea noului sistem de restricţii Karush-Kuhn-Tucker se constată, analizând condiţia de *complementaritate*, că în cazul $\alpha > 0$ — care implică

$(x-2)^2-8=0$ **şi mai departe** $x_{1,2}=2\pm 2\sqrt{2}$ **şi** $\alpha_1=-(1+\sqrt{2})/\sqrt{2}<0$, $\alpha_2=(1-\sqrt{2})/\sqrt{2}<0$ — **condiţiile de *fezabilitate dual*ă sunt încălcate, în vreme ce în cazul** $\alpha=0$ **rezultă imediat că** $x=0$ **şi toate condiţiile Karush-Kuhn-Tucker sunt satisfăcute.**

Trecem acum la obţinerea şi analiza funcţiilor $x\mapsto \max_{\alpha\geq 0} L_{P'}(x,\alpha)$ **şi** $\alpha\mapsto \min_x L_{P'}(x,\alpha)$**.**

Pentru $x\in\mathbb{R}$ **oarecare, dar fixat, vom obţine (fie în urma efectuării calculelor, fie ţinând cont de *Observaţia importantă* (1) de la pag. 194):**

$$\max_{\alpha\geq 0} L_{P'}(x,\alpha)=\begin{cases} x^2 & \textbf{pentru } x\in[2-\sqrt{2},2+\sqrt{2}] \\ +\infty & \textbf{în rest}. \end{cases}$$

Valoarea minimă a acestei funcţii este 0**, obţinută pentru** $x^*=0$**.**

Pentru $\alpha\geq 0$ **oarecare, dar fixat, vom avea:**

$$\begin{aligned}\min_x L_{P'}(x,\alpha) &= \min_x(x^2+\alpha((x-2)^2-8))=\min_x((1+\alpha)x^2-4\alpha x-4\alpha)\\ &= -\frac{(2\alpha)^2+4\alpha(1+\alpha)}{1+\alpha}=-\frac{4\alpha(2\alpha+1)}{1+\alpha}=-\Big(8\alpha-4+\frac{4}{1+\alpha}\Big).\end{aligned}$$

Aşadar, lagrangeanul dual este $L_{D'}(\alpha)=-\dfrac{4\alpha(2\alpha+1)}{1+\alpha}$**, iar problema dual**ă **se scrie ca** $\max_{\alpha\geq 0} L_{D'}(\alpha)$**. Folosind din nou cunoştinţele de analiză matematică de liceu, se constată că**

$$L'_{D'}(\alpha)=-8+\frac{4}{(1+\alpha)^2}=\frac{4[1-\sqrt{2}(1+\alpha)][1+\sqrt{2}(1+\alpha)]}{(1+\alpha)^2}<0,\ \forall\alpha\geq 0$$

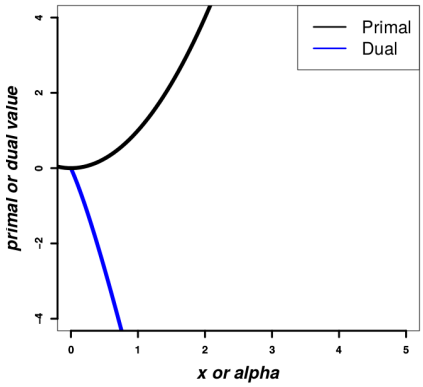
şi

$$L''_{D'}(\alpha)=-\frac{8}{(1+\alpha)^3}<0,\ \forall\alpha\geq 0,$$

deci funcţia $L_{D'}(\alpha)$ **este concavă, iar valoarea maximă a sa este** $d^*=0$**, fiind obţinută pentru** $\alpha^*=0$**.**[219]

Prin urmare, punctul $(x^*=0,\alpha^*=0)$ **este punct-şa (din nou, unicul punct-şa!) pentru lagrangeanul generalizat** $L_{P'}(x,\alpha)$**.**[220]

În imaginea alăturată am reprezentat într-un singur grafic cele două funcţii, $x\mapsto \max_{\alpha\geq 0} L_{P'}(x,\alpha)$ **şi** $\alpha\mapsto \min_x L_{P'}(x,\alpha)$**.**



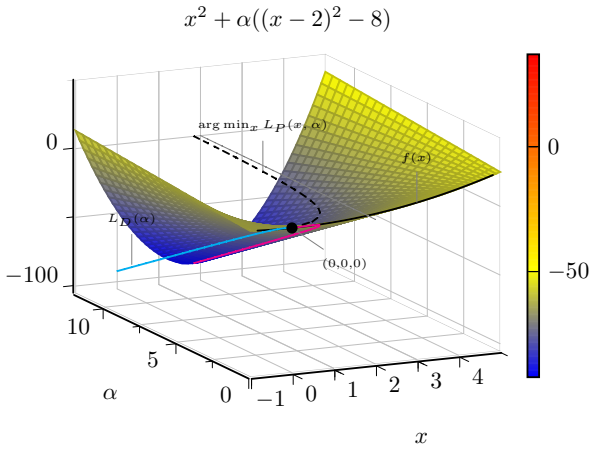
[219] Se poate constata [din grafice] că în cazul restricţiei $g(x)\leq 8$ (deci pentru cea de-a doua problemă de optimizare dată), *regiunea fezabilă* include punctul $x=0$ pentru care se atinge minimul funcţiei $f(x)=x^2$, deci în acest caz restricţia nu [mai] afectează de fapt optimizarea funcţiei f. În cazul restricţiei $g(x)\leq 1$ (deci pentru prima problemă de optimizare dată), punctul $x=0$ era situat în afara regiunii fezabile.

[220] Şi aici, verificăm în mod direct faptul că punctul $(x^*=1,\alpha^*=1)$ satisface condiţia de punct-şa:

$$L_{P'}(x^*,\alpha)\leq L_{P'}(x^*,\alpha^*)\leq L_{P'}(x,\alpha^*)$$

$$\overset{x^*=0,\alpha^*=0}{\Leftrightarrow}\ \alpha(4-8)\leq 0\leq x^2\Leftrightarrow -4\alpha\leq 0\leq x^2,\ \forall x,\ \forall\alpha\geq 0.$$

Ca şi în cazul problemei (P), prezentăm alăturat o reprezentare grafică de *sinteză* pentru componentele-cheie folosite la rezolvarea problemei de optimizare convexă (P′). Semnificaţiile culorilor diferitelor curbe din acest grafic sunt aceleaşi cu cele din figura corespunzătoare de la punctul *d*.



85. **(Metoda dualităţii Lagrange: aplicare în $\mathbb{R}^2$, în două moduri: mai întâi folosind condiţiile Karush-Kuhn-Tucker, iar apoi folosind corespondenţa cu soluţiile problemei duale)**

prelucrare de L. Ciortuz şi S. Ciobanu, după
□ ***University of Helsinki, 2014 spring, Jyrki Kivinen, Example of convex optimisation with Lagrange coefficients***

Fie următoarea problemă de optimizare cu restricţii:

$$\min_{(x,y)\in\mathbb{R}^2} x^2 + y^2$$
$$\text{a. î. } x + y \leq 1$$

a. Într-un sistem de coordonate bidimensional, indicaţi (prin haşurare) care este mulţimea punctelor din plan care satisfac restricţia inclusă în cadrul problemei de optimizare din enunţ.[221]

b. Indicaţi soluţia acestei probleme de optimizare (însă fără a face calcule).

c. Stabiliţi dacă problema de optimizare dată este convexă.

d. Este oare condiţia lui Slater satisfăcută?[222] **Are loc proprietatea de *dualitate tare*? Justificaţi.**

e. Scrieţi problema dală (D) asociată problemei iniţiale (P).
Soluţiile acestor două probleme satisfac condiţiile Karush-Kuhn-Tucker? (*Atenţie*! Nu este nevoie să rezolvaţi cele două probleme ca să puteţi răspunde la această întrebare.)

f. Rezolvaţi problema (P) folosindu-vă doar de sistemul reprezentat de condiţiile Karush-Kuhn-Tucker.[223]

g. Acum, rezolvaţi problema (P) în alt mod: aflaţi mai întâi soluţia problemei duale (D) şi apoi găsiţi soluţiile problemei primale (P) folosindu-vă doar de

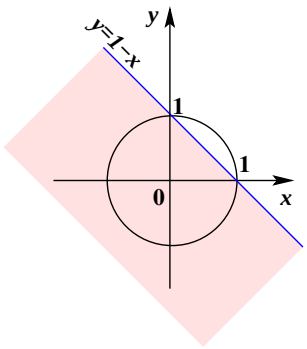
[221] *Observaţie*: Vă readucem aminte că această mulţime se mai numeşte mulţimea soluţiilor *fezabile* (engl., feasible) sau, mai simplu, *regiunea fezabilă* pentru problema de optimizare dată.
[222] Vedeţi *Observaţia importantă* de la pagina 188.
[223] Teorema Karush-Kuhn-Tucker este formulată la problema 83, mai precis în *Comentariul* de la pagina 187.

condiţia de staţionaritate **(sau, de *optimalitate*) din sistemul dat de condiţiile Karush-Kuhn-Tucker, întrucât această condiţie poate fi văzută ca o *relaţie de legătură* între soluţiile celor două probleme, (P) şi (D).**

h. Rezolvaţi din nou punctele $a-g$, de data aceasta pentru problema de optimizare care se obţine din problema iniţială înlocuind restricţia $x+y \leq 1$ cu $x+y \leq -1$. (Se poate observa că restricţia iniţială era de fapt irelevantă, pe când noua restricţie este „activă“.)

Răspuns:

a. Figura alăturată prezintă regiunea fezabilă pentru problema de optimizare dată — problemă pe care de acum încolo o vom desemna cu (P) —, adică mulţimea punctelor din plan care satisfac restricţia $x+y \leq 1$.



b. Dacă nu ţinem cont de restricţia inclusă în problema (P), rezultă imediat că

$$\min_{x,y\in\mathbb{R}^2}(x^2+y^2)=0,$$

iar această valoare optimă se obţine pentru $x^*=y^*=0$. Întrucât această soluţie satisface restricţia $x+y\leq 1$ (altfel spus, punctul (x^*,y^*) aparţine regiunii fezabile pe care am determinat-o la punctul precedent), rezultă că $x^*=0, y^*=0$ este soluţia optimă a problemei (P).

c. Da, problema (P) este o problemă de optimizare convexă. Justificarea constă în faptul că pe de o parte funcţia obiectiv x^2+y^2 este convexă, iar pe de altă parte restricţia $x+y\leq 1$ se poate scrie în mod echivalent $x+y-1\leq 0$, funcţia $x+y-1$ fiind liniară în ambele argumente, deci convexă.[224]

d. Da, condiţia lui Slater este satisfăcută pentru problema de optimizare convexă (P), întrucât $\exists x,y$ astfel încât $x+y<1$, de exemplu $x=0$, $y=0$. Condiţia lui Slater implică proprietatea de *dualitate tare*, adică $p^*=d^*$, unde p^* este soluţia optimă a problemei (P), iar d^* este soluţia optimă a dualei sale, (D).

e. Lagrangeanul generalizat al problemei de optimizare (P) este

$$L_P(x,y,\alpha)=x^2+y^2+\alpha(x+y-1),$$

[224]Funcţia x^2+y^2 este convexă întrucât ea are matricea hessiană

$$H \overset{not.}{=} \nabla^2(x^2+y^2)=\begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix},$$

iar această matrice este pozitiv semidefinită: este imediat că $v^\top Hv\geq 0$ pentru orice $v\in\mathbb{R}^2$. (Vedeţi *Observaţia* de la finalul rezolvării punctului a.3 de la problema 78.)

unde $\alpha \in \mathbb{R}_+$ este *variabila duală* (sau, multiplicatorul Lagrange) care corespunde restricţiei $x + y \leq 1$ din problema (P).[225] Calculând derivata parţială a lui L_P în raport cu variabila x şi egalând-o apoi cu 0, vom avea:

$$\frac{\partial}{\partial x} L_P(x, y, \alpha) = 0 \Leftrightarrow 2x + \alpha = 0 \Leftrightarrow x = -\frac{\alpha}{2}.$$

Similar, calculând derivata parţială a lui L_P în raport cu variabila y şi egalând-o apoi cu 0, va rezultă $y = -\frac{\alpha}{2}$.

Înlocuind x şi y cu $-\frac{\alpha}{2}$ în expresia lui $L_P(x, y, \alpha)$, vom obţine *lagrangeanul dual*, care va fi funcţia obiectiv a problemei de optimizare (D), forma duală a problemei (P):[226]

$$L_D(\alpha) = 2 \cdot \frac{\alpha^2}{4} + \alpha(-\alpha - 1) = -\frac{\alpha^2}{2} - \alpha.$$

Aşadar, duala problemei (P) este:

$$\max_{\alpha \geq 0} \Big(-\frac{\alpha^2}{2} - \alpha \Big).$$

Soluţiile problemelor (P) şi (D) satisfac condiţiile Karush-Kuhn-Tucker pentru că, aşa cum am arătat la punctul precedent, are loc dualitatea tare.[227]

f. Sistemul reprezentat de condiţiile Karush-Kuhn-Tucker pentru problema (P) este următorul:

$$\begin{cases} x + y \leq 1 & \text{fezabilitate primală} \\ \alpha \geq 0 & \text{fezabilitate duală} \\ \alpha(x + y - 1) = 0 & \text{complementaritate} \\ x = -\frac{\alpha}{2} \text{ şi } y = -\frac{\alpha}{2} & \text{staţionaritate (optimalitate)} \end{cases}$$

Remarcaţi faptul că forma aceasta a condiţiilor de staţionaritate / optimalitate a fost dedusă la punctul e, acolo unde am obţinut soluţiile derivatelor parţiale ale lagrangeanului generalizat L_P în raport cu x şi respectiv y.

Ştim că, atunci când există, soluţiile x^*, y^* şi α^* ale sistemului de condiţii Karush-Kuhn-Tucker sunt totodată soluţii optime ale problemelor (P) şi respectiv (D).[228]

Pornind de la *condiţia de complementaritate*, putem rezolva sistemul de mai sus în felul următor:

***Cazul I*: $\alpha = 0$.**
Conform *condiţiei de staţionaritate*, rezultă $x = y = 0$. Condiţiile de fezabilitate ($x + y \leq 1$ şi $\alpha \geq 0$) sunt satisfăcute în acest caz, deci $\alpha^* = x^* = y^* = 0$ este

[225] Ţinând cont de condiţia de staţionaritate / optimalitate din sistemul Karush-Kuhn-Tucker pe de o parte, şi de definiţia formei duale a problemei de optimizare convexă cu restricţii (vedeţi problema 82.b) pe de altă parte, vom putea deduce lagrangeanul dual pe o altă cale (ceva mai directă!) decât am făcut-o la problema 84.c.

[226] Observaţi că spre deosebire de modul în care am procedat la problema precedentă, unde L_D a fost calculat apelând la definiţia $\alpha \mapsto \min_{x,y} L_P(x, y, \alpha)$, aici obţinem L_D din L_P pur şi simplu particularizând L_P pentru acele valori ale lui x şi respectiv y (exprimate în funcţie de α) care au fost deduse din condiţia de staţionaritate / optimalitate.

[227] Vedeţi *Comentariul* de la pag. 187, teorema Karush-Kuhn-Tucker, prima parte.

[228] Vedeţi *Comentariul* de la pag. 187, teorema Karush-Kuhn-Tucker, a doua parte.

soluţie a sistemului constituit de condiţiile Karush-Kuhn-Tucker. Urmează că $x^* = y^* = 0$ este soluţie optimă a problemei (P), iar $\alpha^* = 0$ este soluţie optimă a problemei (D).

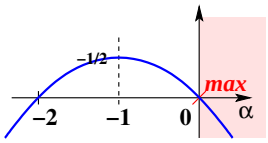
***Cazul II*: $\alpha \neq 0$, deci $\alpha > 0$.**
Din condiţia de complementaritate, rezultă $x+y-1 = 0$, deci $x+y = 1$. Urmează că este satisfăcută şi condiţia de fezabilitate primală, $x + y \leq 1$. Din relaţia $x + y - 1 = 0$, ţinând cont de condiţia de staţionaritate ($x = y = -\frac{\alpha}{2}$), rezultă că $-\alpha = 1$, adică $\alpha = -1$, ceea ce contrazice condiţia de fezabilitate duală ($\alpha \geq 0$). Prin urmare, în acest caz ($\alpha > 0$), sistemul Karush-Kuhn-Tucker nu are soluţie.

În concluzie, singura soluţie a sistemului Karush-Kuhn-Tucker este $\alpha^* = x^* = y^* = 0$, iar singura soluţie optimă a problemei (P) este $x^* = y^* = 0$.

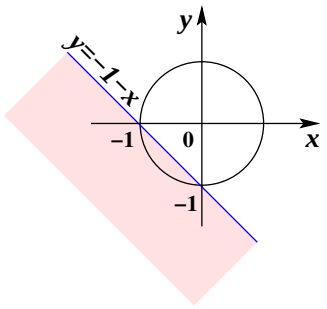
g. Soluţia optimă a problemei (D), a cărei formă a fost dedusă la punctul d,

$$\max_{\alpha \geq 0} \Big(-\frac{\alpha^2}{2} - \alpha \Big) \Leftrightarrow \max_{\alpha \geq 0} \Big(-\frac{\alpha}{2}(\alpha + 2) \Big),$$

este $\alpha^* = 0$. Justificarea este imediată, dacă ţinem cont de graficul funcţiei $-\frac{\alpha}{2}(\alpha + 2)$, din figura alăturată.
Din relaţiile $x = y = -\frac{\alpha}{2}$, care au fost deduse la punctul e, rezultă că soluţia optimă a problemei (P) este $x^* = y^* = 0$.

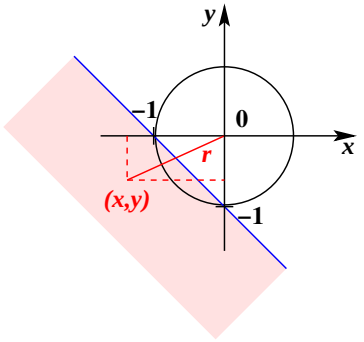


h. Figura alăturată prezintă regiunea fezabilă pentru noua problemă de optimizare convexă (pe care o vom nota cu (P′)), adică mulţimea punctelor din plan care satisfac restricţia $x + y \leq -1$.



Se poate observa că problema (P′) este echivalentă cu o altă problemă de optimizare (care nu este însă convexă, fiindcă restricţia de tip egalitate nu este afină):

$$\begin{array}{ll} \min_{r \in \mathbb{R}} & r^2 \\ \textbf{a. î.} & r^2 = x^2 + y^2 \\ & x + y \leq -1 \end{array}$$



şi că valoarea minimă pentru r este corespunzătoare punctului $(x^* = -1/2, y^* = -1/2)$, deci $r^* = \frac{1}{\sqrt{2}}$.
Evident, noua restricţie ($x + y \leq -1$) nu schimbă tipul problemei (P′), ea rămânând una de optimizare convexă. De asemenea, condiţia lui Slater este satisfăcută şi aici: $\exists\ x, y$ astfel încât $x + y < -1$ (consideraţi, de exemplu,

$x = y = -1$). **În consecinţă, şi în acest caz are loc *dualitatea tare*. Aşadar, putem rezolva problema (P′) rezolvând mai întâi duala sa (D′) şi folosind apoi relaţia de corespondenţă dintre soluţiile celor două probleme.**

Deocamdată însă, vom rezolva problema (P′) pornind de la sistemul de condiţii Karush-Kuhn-Tucker:

$$\begin{cases} x + y \leq -1 & \text{fezabilitate primală} \\ \alpha \geq 0 & \text{fezabilitate duală} \\ \alpha(x + y + 1) = 0 & \text{complementaritate} \\ x = -\dfrac{\alpha}{2} \text{ şi } y = -\dfrac{\alpha}{2} & \text{staţionaritate (optimalitate)} \end{cases}$$

Remarcaţi faptul că forma aceasta a condiţiilor de staţionaritate / optimalitate a fost obţinută calculând în prealabil soluţiile derivatelor parţiale ale lagrangeanului generalizat $L_{P'}$ în raport cu x şi respectiv y.

Rezolvarea sistemului de condiţii Karush-Kuhn-Tucker este următoarea:

***Cazul I*: $\alpha = 0$.**
Conform condiţiei de staţionaritate, rezultă $x = y = 0$, dar această soluţie nu este fezabilă ($x + y \not\leq -1$).

***Cazul II*: $\alpha > 0$.**
Din condiţia de complementaritate rezultă $x + y = -1$, iar condiţia de staţionaritate va implica $x = y = -1/2$ şi $\alpha = 1$. Aceste valori pentru x, y şi α satisfac condiţiile de fezabilitate.

Prin urmare, unica soluţie a sistemului Karush-Kuhn-Tucker este $x^* = y^* = -1/2$ şi $\alpha^* = 1$, de unde rezultă că unica soluţie a problemei (P′) este $x^* = y^* = -1/2$.

***Alternativ*, aşa cum de fapt am precizat mai sus, am fi putut găsi soluţia problemei (P′) rezolvând duala sa (D′) şi folosind apoi relaţia de corespondenţă dintre soluţii. Este exact ceea ce vom face acum:**

$$L_{P'}(x, y, \alpha) = x^2 + y^2 + \alpha(x + y + 1)$$
$$\Rightarrow \frac{\partial}{\partial x} L_{P'}(x, y, \alpha) = 0 \Leftrightarrow 2x + \alpha = 0 \Leftrightarrow x = -\frac{\alpha}{2}$$
$$\text{şi, similar, } \frac{\partial}{\partial y} L_{P'}(x, y, \alpha) = 0 \Leftrightarrow y = -\frac{\alpha}{2}$$
$$\Rightarrow L_{D'}(\alpha) = \frac{\alpha^2}{2} + \alpha(-\alpha + 1) = -\frac{\alpha^2}{2} + \alpha = -\frac{\alpha}{2}(\alpha - 2).$$

Prin urmare, problema duală (D′) este

$$\max_{\alpha \geq 0} \Big(-\frac{\alpha}{2}(\alpha - 2) \Big).$$

Vă puteţi convinge singuri (de exemplu, făcând graficul funcţiei $-\dfrac{\alpha}{2}(\alpha - 2)$, sau folosind formula generală de calcul pentru abscisa punctului de optim al funcţiei de gradul al doilea) că soluţia problemei (D′) este $\alpha^* = 1 \geq 0$. În concluzie, soluţia problemei (P′) este $x^* = y^* = -1/2$, exact ceea ce am găsit şi anterior, însă pe altă cale (şi anume, folosind doar condiţiile Karush-Kuhn-Tucker).

86. **(Metoda dualităţii Lagrange: aplicare pentru găsirea codificării optimale pentru [literele din] alfabetul unui limbaj)**

□ • ○ ***CMU, 2014 fall, E. Xing, B. Poczos, HW1, pr. 3.2***

În acest exerciţiu vom rezolva o problemă [relativ simplă] de optimizare convexă.

Alfabetul unui limbaj constă din litere, pe care le vom nota cu α_i, pentru $i \in [n]$, unde $[n] \overset{not.}{=} \{1, 2, \ldots, n\}$. Presupunem că aceste litere apar în decursul folosirii [limbajului respectiv, în mod natural] cu probabilităţile p_i. Ne propunem să găsim pentru aceste litere o *codificare optimală* sub formă de biţi, în aşa fel încât să minimizăm *numărul mediu de biţi* (engl., the expected number of bits) folosiţi de litere, păstrând însă capacitatea de a decodifica secvenţa de biţi transmişi pe un canal (engl., stream) de comunicare, fără a avea nevoie să folosim delimitatori (engl., markers) între litere.

De *exemplu*, dacă alfabetul considerat are $n = 32$ de litere şi toate literele apar cu probabilităţi egale (deci $p_i = 1/32$), este normal să folosim pentru codificarea optimă 5 biţi pentru fiecare literă.

Se poate arăta că o codificare optimă pentru alfabetul unui limbaj poate fi găsită rezolvând următoarea *problemă de optimizare cu restricţii*:[229]

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \sum_{i=1}^{n} p_i x_i \\ \textbf{a. î.} \quad &\sum_{i=1}^{n} 2^{-x_i} \leq 1 \\ &x_i \geq 0 \ \forall i \in [n]. \end{aligned} \tag{123}$$

În această problemă, *funcţia obiectiv* este numărul mediu de biţi per literă, iar prima restricţie este aşa-numita *inegalitate a lui Kraft*, care asigură faptul că literele vor putea fi *decodificate* în mod unic.[230]

***Observaţie*: Deşi numerele x_i ar trebui să fie întregi şi pozitive, noi vom căuta soluţia problemei de optimizare în mulţimea numerelor reale pozitive.**

a. Demonstraţi că *regiunea fezabilă* (engl., feasible region) corespunzătoare problemei de optimizare (123) — adică, mulţimea formată din toţi acei $x \overset{not.}{=} (x_1, \ldots, x_n) \in \mathbb{R}^n$ pentru care sunt satisfăcute cele două restricţii din problema de optimizare dată — este convexă.

b. Arătaţi că prima restricţie din problema (123) este satisfăcută cu egalitate în cazul soluţiei optime [a acestei probleme].

c. Datorită rezultatului de la punctul precedent, putem rezolva problema (123) înlocuind în prima restricţie semnul $\leq$ cu $=$. Rezolvaţi această nouă problemă de optimizare convexă.

***Sugestie*: Demonstraţi în prealabil că funcţia din membrul stâng din prima**

[229] *Atenţie*: Observaţi că nu am afirmat că problema (123) este o problemă de optimizare convexă! Rezolvarea punctelor *a* şi *b* de mai jos nu depinde de această chestiune. Doar la punctul *c* vă vom cere să arătaţi (între altele) că într-adevăr aceasta este o problemă de optimizare convexă (cu restricţii).

[230] Vedeţi https://en.wikipedia.org/wiki/Kraft-McMillan_inequality.

restricţie din cadrul problemei de optimizare (123) **este într-adevăr convexă, pentru a justifica aplicarea [ulterioară a] metodei dualităţii Lagrange.**

Răspuns:

a. Notăm cu P **regiunea fezabilă a problemei** (123)**. Pentru a demonstra convexitatea mulţimii** P**, vom folosi *definiţia* din enunţul problemei 78. Aşadar, vom arăta că** $\alpha x + (1-\alpha)y \in P$ **pentru orice** $x, y \in P$ **şi orice** $\alpha \in (0,1)$**. Notând** $x \overset{not.}{=} (x_1, \ldots, x_n)$ **şi** $y \overset{not.}{=} (y_1, \ldots, y_n)$**, rezultă că** $\alpha x + (1-\alpha)y = (\alpha x_1 + (1-\alpha)y_1, \ldots, \alpha x_n + (1-\alpha)y_n)$**. Folosind *inegalitatea ponderată a mediilor*[231] precum şi faptul că** x **şi** y **aparţin regiunii fezabile** (P)**, putem scrie:**[232]

$$\begin{aligned} \sum_{i=1}^{n} 2^{-(\alpha x_i + (1-\alpha)y_i)} &\leq \sum_{i=1}^{n} \left[\alpha\, 2^{-x_i} + (1-\alpha)\, 2^{-y_i}\right] \\ &= \alpha \underbrace{\sum_{i=1}^{n} 2^{-x_i}}_{\leq 1} + (1-\alpha) \underbrace{\sum_{i=1}^{n} 2^{-y_i}}_{\leq 1} \\ &\leq \alpha + (1-\alpha) = 1. \end{aligned} \tag{124}$$

Aşadar, $\alpha x + (1-\alpha)y$ **satisface prima restricţie (adică inegalitatea) din problema de optimizare** (123)**. Apoi, este clar că** $x_i,\ y_i \geq 0 \Rightarrow \alpha x_i + (1-\alpha)y_i \geq 0$ **pentru orice** $\alpha \in [0,1]$**. Prin urmare,** $\alpha x + (1-\alpha)y \in P$**. Cu aceasta, demonstraţia pentru convexitatea mulţimii** P **este încheiată.**

b. Presupunem (prin *reducere la absurd*) că o soluţie optimă $x^* \overset{not.}{=} (x_1^*, \ldots, x_n^*)$ **a problemei** (123) **satisface restricţia de tip inegalitate în sens strict, adică** $\sum_{i=1}^{n} 2^{-x_i^*} < 1$**. Este imediat că pentru orice** $j \in [n]$**, avem** $x_j^* > 0$**, fiindcă altfel**

[231] În forma cea mai simplă (cazul netrivial), inegalitatea care leagă media aritmetică de media geometrică (engl., *inequality of arithmetic and geometric means*, abbr. *AM–GM inequality*), se formulează astfel: pentru orice două numere nenegative x şi y, se poate arăta că

$$\frac{x+y}{2} \geq \sqrt{xy},$$

egalitatea având loc dacă şi numai dacă $x = y$.

Varianta *ponderată* a acestei inegalităţi (engl., *weighted AM–GM inequality*) este următoarea: Fie numerele nenegative $x_1, x_2, \ldots, x_m$, precum şi ponderile nenegative $w_1, w_2, \ldots, w_m$. Notăm $w = w_1 + w_2 + \ldots + w_m$. Dacă $w > 0$, atunci este satisfăcută inegalitatea

$$\begin{aligned} &\frac{w_1 x_1 + w_2 x_2 + \ldots + w_m x_m}{w} \geq \sqrt[w]{x_1^{w_1} x_2^{w_2} \ldots x_m^{w_m}} \\ \Leftrightarrow\ &\frac{w_1}{w} x_1 + \frac{w_2}{w} x_2 + \ldots + \frac{w_m}{w} x_m \geq x_1^{\frac{w_1}{w}} \cdot \ldots \cdot x_m^{\frac{w_m}{w}}, \end{aligned}$$

egalitatea având loc dacă şi numai dacă toţi acei x_k pentru care $w_k > 0$ sunt egali. În acest context se foloseşte convenţia $0^0 = 1$. Observaţi că atunci când toate ponderile w_k au valoarea 1, obţinem inegalitatea mediilor care a fost prezentată mai sus.

(*Menţiune*: Conţinutul acestei note de subsol a fost preluat de pe site-ul https://en.wikipedia.org/wiki/Inequality_of_arithmetic_and_geometric_means#Weighted_AM-GM_inequality.)

[232] Pentru fiecare valoare a lui $i \in [n]$, fixată, dacă în inegalitatea ponderată a mediilor luăm $m = 2$, $x_1 = 2^{-x_i}$, $x_2 = 2^{-y_i}$, $w_1 = \alpha$ şi, $w_2 = 1 - \alpha$, obţinem

$$2^{-\alpha x_i} \cdot 2^{-(1-\alpha)y_i} = \left(2^{-x_i}\right)^{\alpha} \cdot \left(2^{-y_i}\right)^{(1-\alpha)} \leq \alpha\, 2^{-x_i} + (1-\alpha)\, 2^{-y_i}.$$

Însumând aceste inegalităţi membru cu membru pentru $i = 1, \ldots, n$, vom obţine inegalitatea (124).

am avea $\sum_{i=1}^{n} 2^{-x_i^*} > 1$.[233] Prin urmare, putem să micşorăm valoarea [măcar] pentru unul dintre aceşti x_j^*, fără a afecta condiţiile de fezabilitate ale problemei, dar reuşind în schimb să micşorăm valoarea funcţiei obiectiv. Aşadar, x^* nu este soluţie optimă a problemei (123)**, ceea ce intră în *contradicţie* cu presupunerea pe care am făcut-o anterior.**

În *concluzie*, orice soluţie optimă a problemei (123) **satisface prima restricţie cu egalitate. Ca o *consecinţă* directă a acestui fapt, a rezolva problema de optimizare** (123) **revine la a rezolva următoarea problemă de optimizare:**[234]

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} \sum_{i=1}^{n} p_i x_i \\ \textbf{a. î.}\ &\sum_{i=1}^{n} 2^{-x_i} = 1 \\ &x_i \geq 0\ \forall i \in [n]. \end{aligned} \tag{125}$$

c. Vom arăta mai întâi că funcţia $\Big(\sum_{i=1}^{n} 2^{-x_i}\Big) - 1$, care corespunde primei restricţii din problema de optimizare (123)**, este convexă. Pentru aceasta, vom demonstra că matricea hessiană a acestei funcţii este pozitiv semidefinită.**

Vectorul gradient pentru această funcţie este

$$\nabla\Big(\sum_{i=1}^{n} 2^{-x_i}\Big) = (\ln 2)\ [2^{-x_1}, \ldots, 2^{-x_n}]^\top.$$

Prin urmare, matricea hessiană pentru aceeaşi funcţie $\Big(\sum_{i=1}^{n} 2^{-x_i}\Big) - 1$ este

$$\nabla^2\Big(\sum_{i=1}^{n} 2^{-x_i}\Big) = (\ln 2)^2 \begin{bmatrix} 2^{-x_1} & & 0 \\ & \ddots & \\ 0 & & 2^{-x_n} \end{bmatrix} \overset{not.}{=} (\ln 2)^2\ diag\Big(2^{-x_i}\Big)_{i=\overline{1,n}}.$$

Pentru orice vector $v \in \mathbb{R}^n$, avem:

$$\begin{aligned} v^\top \nabla^2\Big(\sum_{i=1}^{n} 2^{-x_i}\Big) v &= [v_1, \ldots, v_n]^\top\ (\ln 2)^2\, diag\Big(2^{-x_i}\Big)_{i=\overline{1,n}}\ [v_1, \ldots, v_n] \\ &= (\ln 2)^2\ [v_1 2^{-x_1}, \ldots, v_n 2^{-x_n}]\ [v_1, \ldots, v_n] = (\ln 2)^2 \sum_{i=1}^{n} v_i^2\ 2^{-x_i} \geq 0, \end{aligned}$$

ceea ce înseamnă că matricea hessiană $\nabla^2\Big(\sum_{i=1}^{n} 2^{-x_i} - 1\Big)$ este pozitiv semidefinită.

Prin urmare, problema de optimizare (123) **este convexă şi, pentru rezolvarea ei, putem aplica metoda dualităţii Lagrange.**

[233] LC: S-a considerat $n \geq 2$, fiindcă pentru $n = 1$ problema este trivială. (De fapt, chiar şi pentru $n = 2$ problema este trivială: cele două litere ale alfabetului vor fi codificate prin biţii 0 şi 1.)

[234] *Observaţie* (LC): Deşi restricţia $\sum_{i=1}^{n} 2^{-x_i} = 1$ nu este de tip afin — vedeţi definiţia neţiunii de *problemă de optimizare convexă cu restricţii* care a fost dată la ex. 83 —, pentru rezolvarea problemei (125) nu este greşit să folosim metoda multiplicatorilor lui Lagrange, întrucât schimbarea de variabilă $2^{-x_i} = y_i$ pentru $i = 1, \ldots, n$ conduce la o reformulare a acestei probleme care satisface cerinţele din definiţia neţiunii de problemă de optimizare convexă cu restricţii. (Puteţi verifica singuri detaliile.)

Considerând variabilele duale $\lambda \in \mathbb{R}$ şi $u_i \geq 0$ pentru $i \in [n]$, funcţia lagrangeană generalizată pentru problema de optimizare convexă (125) este definită astfel:

$$L(x, \lambda, u) = \sum_{i=1}^{n} p_i x_i + \lambda\Big(\sum_{i=1}^{n} 2^{-x_i} - 1\Big) - \sum_{i=1}^{n} u_i x_i. \tag{126}$$

Fie (x, λ, u) un tuplu care satisface condiţiile Karush-Kuhn-Tucker pentru problema (123). Una dintre aceste condiţii, şi anume *condiţia de complementaritate* (engl., *complementary slackness*), afirmă că pentru orice $i \in [n]$, avem $u_i x_i = 0$. Ştim deja de la punctul b că $x_i > 0$, ceea ce conduce la concluzia $u_i = 0$.

Pentru orice $i \in [n]$, întrucât $u_i = 0$,

$$\frac{\partial}{\partial x_i} L(x, \lambda, u) \overset{(126)}{=} p_i - \lambda 2^{-x_i} \ln 2 - u_i = p_i - \lambda 2^{-x_i} \ln 2,$$

$$\textbf{deci } \frac{\partial L}{\partial x_i} = 0 \Leftrightarrow p_i = (\lambda \ln 2)\, 2^{-x_i}. \tag{127}$$

Însumând egalităţile $p_i = (\lambda \ln 2)\, 2^{-x_i}$ membru cu membru, după toate valorile lui i, obţinem

$$\sum_{i=1}^{n} p_i = \lambda \ln 2 \sum_{i=1}^{n} 2^{-x_i} \Rightarrow 1 = \lambda \ln 2,$$

întrucât p_i sunt probabilităţi, iar $\sum_{i=1}^{n} 2^{-x_i} = 1$ (vedeţi punctul b). Aşadar, $\lambda = 1/\ln 2$, iar din relaţia (127) obţinem $p_i = 2^{-x_i}$, de unde prin logaritmare rezultă $x_i = -\log_2 p_i$. Este uşor de verificat că tuplul definit prin $x_i = -\log_2 p_i$, $\lambda = 1/\ln 2$, $u_i = 0$ satisface toate condiţiile Karush-Kuhn-Tucker. Prin urmare, (conform proprietăţii care a fost demonstrată la problema 83) aceasta este soluţie optimă pentru problema (125), deci şi pentru problema (123).

***Observaţie*: Rezultă că funcţia obiectiv a problemei (123) este $-\sum_{i=1}^{n} p_i \log_2 p_i$, adică exact *entropia* variabilei aleatoare a cărei distribuţie probabilistă este reprezentată de p_i, $i \in [n]$.**

87. (O variantă a algoritmului Perceptron, pentru care relaţia de actualizare a ponderilor se obţine rezolvând o problemă de optimizare convexă cu restricţii)

■ □ • ○ ***University of Helsinki, 2014 spring, Jyrki Kivinen, HW5, pr. 2***

În această problemă vom lucra cu o variantă a algoritmului Perceptron,[235] în care vectorul „actualizat" de ponderi w_{t+1} este definit ca fiind acel vector w care constituie soluţia problemei de optimizare următoare:

$$\begin{aligned} \min_w \quad & \|w - w_t\|^2 \\ \text{a. î.} \quad & y_t\, w \cdot x_t \geq 1. \end{aligned}$$

unde $w, w_t, x_t \in \mathbb{R}^d$ şi $y_t \in \{-1, +1\}$.

[235] Pentru o prezentare a algoritmului Perceptron [neciclic, nestochastic], vedeţi problema 16 de la capitolul *Reţele neuronale artificiale*. Pentru varianta kernel-izată a acestui algoritm vedeţi problema 19, tot de la capitolul *Reţele neuronale artificiale*. Pentru o variantă stochastică a algoritmului Perceptron, vedeţi problema 57.A de la capitolul *Maşini cu vectori-suport*.

a. Determinaţi cum anume se poate scrie vectorul w_{t+1} în funcţie de w_t, x_t şi y_t. (Cu alte cuvinte, rezolvaţi problema de optimizare dată).

b. Presupunând că iniţial vectorul de ponderi w_1 are toate componentele zero, arătaţi că w_{t+1} este o combinaţie liniară de instanţele $x_1, \ldots, x_t$. Scrieţi versiunea kernel-izată a acestui algoritm.

Răspuns:

a. Fie funcţiile $f : \mathbb{R}^d \to \mathbb{R}$ şi $p : \mathbb{R}^d \to \mathbb{R}$, definite prin prin relaţiile $f(w) = \|w - w_t\|^2 = (w - w_t) \cdot (w - w_t) = w \cdot w - 2w \cdot w_t + w_t \cdot w_t$ şi respectiv $p(w) = 1 - y_t w \cdot x_t$. Vom demonstra mai întâi că f este funcţie convexă.

Ştim că o funcţie care este definită pe o mulţime convexă şi este dublu derivabilă (adică, admite toate derivatele parţiale de ordin secund) este convexă dacă şi numai dacă matricea sa hessiană (notată cu $\nabla^2 f(w)$) este pozitiv semidefinită în interiorul mulţimii pe care este definită funcţia respectivă.[236] Fiindcă $f(w) = \sum_{i=1}^{d} (w_i - w_{t,i})^2$, este uşor de observat că

$$\frac{\partial f(w)}{\partial w_i} = 2(w_i - w_{t,i}) \quad \textbf{pentru orice } i \in \{1, 2, \ldots, d\},$$

iar pentru orice $i, j \in \{1, 2, \ldots, d\}$

$$\frac{\partial^2 f(w)}{\partial w_i \partial w_j} = 2 \textbf{ dacă } i = j \quad \textbf{şi} \quad \frac{\partial^2 f(w)}{\partial w_i \partial w_j} = 0 \textbf{ dacă } i \neq j.$$

Aşadar, matricea hessiană a lui f este $\nabla^2 f(w) = 2I_d$ pentru orice $w \in \mathbb{R}^d$. Este imediat că pentru orice vector-coloană $z \in \mathbb{R}^d$ avem $z^\top (2I_d) z = 2\|z\|^2 \geq 0$, ceea ce înseamnă că matricea $\nabla^2 f(w)$ este pozitiv semidefinită în orice punct $w \in \mathbb{R}^d$.

Problema de optimizare din enunţ constă în a minimiza $f(w)$ ţinând cont de restricţia $p(w) \leq 0$. Vom rezolva această problemă determinând soluţiile sistemului reprezentat de condiţiile Karush-Kuhn-Tucker:[237]

$$\begin{cases} p(w) \leq 0 & \textbf{(fezabilitate primală)} \\ \lambda \geq 0 & \textbf{(fezabilitate duală)} \\ \lambda p(w) = 0 & \textbf{(complementaritate)} \\ \nabla f(w) + \lambda \nabla p(w) = 0 & \textbf{(staţionaritate / optimalitate)} \end{cases}$$

sau, echivalent (făcând uz de regulile de derivare vectorială (5b) şi (5a) de la problema 88.c),

$$\begin{cases} 1 - y_t w \cdot x_t \leq 0 & (i) \\ \lambda \geq 0 & (ii) \\ \lambda(1 - y_t w \cdot x_t) = 0 & (iii) \\ 2w - 2w_t - \lambda y_t x_t = 0 & (iv) \end{cases}$$

Egalitatea (iv) implică

$$w = w_t + (1/2)\lambda y_t x_t. \tag{128}$$

Datorită egalităţii (iii), vom trata două cazuri.
În primul caz, $\lambda = 0$, deci inegalitatea (ii) este adevărată. De asemenea, din

[236] Vedeţi *Observaţia* de la finalul rezolvării punctului *a*.3 de la problema 78.

[237] În cele ce urmează, simbolul ∇ desemnează vectorul gradient / derivata vectorială calculat(ă) în raport cu w.

relaţia (128) **rezultă** $w = w_t$ **şi, datorită condiţiei** (i)**, va trebui ca inegalitatea** $1 - y_t w_t \cdot x_t \leq 0$ **să fie satisfăcută.**

În cazul al doilea, $\lambda > 0$**, din egalitatea** (iii) **rezultă**

$$\begin{aligned} 1 - y_t w \cdot x_t &\overset{(128)}{=} 1 - y_t\Big(w_t + \frac{1}{2}\lambda y_t x_t\Big) \cdot x_t \\ &= 1 - y_t w_t \cdot x_t - \frac{1}{2}\lambda x_t \cdot x_t = 0. \end{aligned} \tag{129}$$

Am ţinut cont de faptul că $y_t^2 = 1$**. Egalitatea** $1 - y_t w \cdot x_t = 0$ **implică faptul că relaţia** (i) **este satisfăcută cu egalitate. Scoţându-l pe** λ **din egalitatea** (129)**, obţinem**

$$\lambda = 2 \cdot \frac{1 - y_t w_t \cdot x_t}{\|x_t\|^2}. \tag{130}$$

Observaţi că datorită relaţiei (ii)**, din relaţia** (130) **rezultă că este necesar ca inegalitatea** $1 - y_t w_t \cdot x_t > 0$ **să fie satisfăcută.**

Înlocuind în relaţia (128) **expresia lui** λ **obţinută la** (130)**, vom obţine:**

$$\begin{aligned} w &= w_t + \frac{1}{2} \cdot 2 \cdot \frac{1 - y_t w_t \cdot x_t}{\|x_t\|^2} y_t x_t \\ &= w_t + y_t \frac{1 - y_t w_t \cdot x_t}{\|x_t\|^2} x_t. \end{aligned}$$

În sfârşit, punând toate acestea împreună, vom obţine *regula de actualizare* a ponderilor:

$$w_{t+1} = w_t + \sigma_t y_t \frac{1 - y_t w_t \cdot x_t}{\|x_t\|^2} x_t,$$

unde

$$\sigma_t = \begin{cases} 0 & \textbf{dacă } y_t w_t \cdot x_t \geq 1 \\ 1 & \textbf{dacă } y_t w_t \cdot x_t < 1. \end{cases}$$

b. Ni s-a cerut să arătăm că dacă se ia $w_1 = 0$**, atunci rezultă că** w_{t+1} **este o combinaţie liniară de vectorii** $x_1, x_2, \ldots, x_t$**. Cu alte cuvinte, există** $\alpha_1, \alpha_2, \ldots, \alpha_t \in \mathbb{R}$ **astfel încât** $w_{t+1} = \alpha_1 x_1 + \alpha_2 x_2 + \ldots + \alpha_t x_t$**. Pentru** $t = 0$**, suma este vidă, deci afirmaţia** ($w_1 = 0 = \sum_{i=1}^{0} \alpha_i x_i$) **este adevărată. Fie acum** $t \in \{1, 2, \ldots\}$ **şi să presupunem că** $w_t = \sum_{i=1}^{t-1} \alpha_i x_i$**. Folosind regula de actualizare a ponderilor, vom putea scrie:**

$$w_{t+1} = \sum_{i=1}^{t-1} \alpha_i x_i + \sigma_t y_t \frac{1 - y_t w_t \cdot x_t}{\|x_t\|^2} x_t = \sum_{i=1}^{t} \alpha_i x_i,$$

unde am notat $\alpha_t = \sigma_t y_t (1 - y_t w_t \cdot x_t)/\|x_t\|^2$**. Folosind pricipiul inducţiei complete, rezultă că proprietatea de liniaritate este adevărată pentru orice** $t \in \mathbb{N}$**.**

În algoritmul Perceptron stochastic kernel-izat, inlocuim vectorii / instanţele x_i **cu** $\phi(x_i)$**, imaginile lor în spaţiul de „trăsături". Acest algoritm poate fi formalizat astfel:**

For $t = 1, 2, \ldots, T$ do
1. Get the instance x_t.
2. Let $p_t = w_t \cdot \phi(x_t) = \Big(\sum_{i=1}^{t-1} \alpha_i \phi(x_i)\Big) \cdot \phi(x_t) = \sum_{i=1}^{t-1} \alpha_i K(x_i, x_t)$.
   Predict $\hat{y}_t = \mathsf{sign}(p_t)$.

3. Get the correct answer y_t.
4. If $y_t p_t < 1$,
   set $\alpha_t \leftarrow y_t(1 - y_t w_t \cdot \phi(x_t))/\|\phi(x_t)\|^2 = y_t - \sum_{i=1}^{t-1} \alpha_i K(x_i, x_t)/K(x_t, x_t)$,
   otherwise
   set $\alpha_t \leftarrow 0$ (and x_t can be discarded).
5. Let $\|w_{t+1}\|^2 = \sum_{i=1}^{t} \alpha_i \phi(x_i) \cdot \sum_{j=1}^{t} \alpha_j \phi(x_j) = \sum_{i=1}^{t} \sum_{j=1}^{t} \alpha_i \alpha_j K(x_i, x_j)$.
   If $\|w_{t+1}\| > 1$, set $\alpha_i \leftarrow \alpha_i / \|w_t\|$ for all $i \in \{1, 2, \ldots, t\}$.

Remarcaţi faptul că operaţia de la punctul 5 obligă ponderile să rămână mici (în modul), ceea ce serveşte la contracararea overfitting-ului.

88. **(Teorema de reprezentare: un rezultat foarte util pentru kernel-izarea mai multor algoritmi de învăţare automată care produc separatori liniari)**

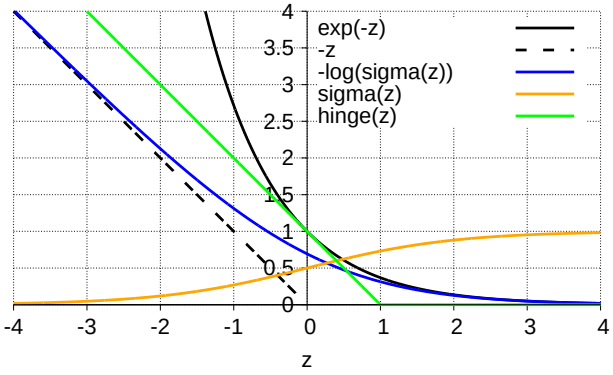
prelucrare de Liviu Ciortuz după
■ • ***Stanford, John Duchi, Supplemental lecture notes***
CMU, 2015 spring, Alex Smola, midterm, pr. 5

Fie $L : \mathbb{R}^2 \to \mathbb{R}$. Vom spune că L este *funcţie de cost* sau *funcţie de pierdere* (engl., loss function) dacă ea este i. nenegativă şi ii. convexă în raport cu primul argument, pentru orice valoare fixată a celui de-al doilea argument.

Spre *exemplu*,

– pentru regresia liniară şi pentru perceptronul liniar se foloseşte – cu rol de funcţie de cost – pătratul erorii, adică $L(r, y) = (y - r)^2$,

– pentru regresia logistică se foloseşte *funcţia de cost logistică* $L(r, y) = \ln(1 + e^{-yr})$,
– pentru maşini cu vectori-suport (SVM) se foloseşte *funcţia de cost hinge* $L(r, y) = \max\{0, 1 - yr\}$, iar
– pentru AdaBoost, se foloseşte *funcţia de cost* [*negativ*] *exponenţială* $L(r, y) = e^{-yr}$.



(În figura alăturată, am folosit notaţia $z = yr$.)

Fie de asemenea instanţele $x_i \in \mathbb{R}^d$ (cu $d \in \mathbb{N}^*$), precum şi $y_i \in \mathbb{R}$ (desemnând fie clase, fie — ca în cazul regresiei liniare şi al perceptronului liniar — valori reale oarecare pentru funcţia de învăţat), pentru $i = 1, \ldots, n$.

a. Demonstraţi că orice soluţie w^* a următoarei *probleme de optimizare* (cu termen de regularizare de normă L_2)

$$\min_{w \in \mathbb{R}^d} \Big(\frac{1}{n} \sum_{i=1}^{n} L(w \cdot x_i, y_i) + \frac{\lambda}{2} \|w\|^2 \Big), \tag{131}$$

unde $\lambda \geq 0$, iar $\|\cdot\|$ desemnează norma euclidiană, poate fi scrisă sub forma următoare:

$$w^* = \sum_{i=1}^{n} \alpha_i x_i \text{ pentru anumite valori } \alpha_i \in \mathbb{R}. \tag{132}$$

***Sugestie*: Puteţi rezolva problema în *cazul particular* când $\lambda > 0$, iar funcţia de cost $L : \mathbb{R}^2 \to \mathbb{R}$ este derivabilă în raport cu primul argument, pentru orice valoare fixată a celui de-al doilea argument. Puteţi, eventual, să folosiţi următoarele formule de calcul cu derivate vectoriale:**[238]

(5a) $\dfrac{\partial}{\partial X} a^\top X = \dfrac{\partial}{\partial X} X^\top a = a$

(5b) $\dfrac{\partial}{\partial X} X^\top A X = (A + A^\top) X$

***Observaţie*:** Acest rezultat se numeşte *teorema de reprezentare* şi se datorează statisticienilor Grace Wahba şi George Kimeldorf.[239] Conform relaţiei (132), vectorul-soluţie w^* pentru problema de optimizare (131) se „reprezintă" ca o combinaţie liniară de instanţele de antrenament $x_1, \ldots, x_n$.

b. Se dă o funcţie $\phi : \mathbb{R}^d \to \mathbb{R}^m$, unde $m > d$ (de obicei m este chiar mult mai mare decât d). În continuare vom considera „imaginile" instanţelor x_i prin funcţia ϕ, adică $\phi(x_i)$, pentru $i = 1, \ldots, n$. Dintre toate operaţiile care pot fi (sau, sunt deja) definite între aceste „imagini", singura care va fi permisă (pentru a fi folosită) aici este produsul scalar: $\phi(x_i) \cdot \phi(x_j) \overset{not.}{=} K(x_i, x_j)$. Funcţia K astfel definită — şi, de fapt, extinsă în mod natural la întreg spaţiul $\mathbb{R}^d \times \mathbb{R}^d$ — se numeşte *funcţie-nucleu*.[240]

Aplicaţi aşa-numitul "kernel-trick" problemei de optimizare

$$\tilde{w}^* = \arg\min_{\tilde{w} \in \mathbb{R}^m} \underbrace{\left(\frac{1}{n} \sum_{i=1}^{n} L(\tilde{w} \cdot \phi(x_i), y_i) + \frac{\lambda}{2} \|\tilde{w}\|^2 \right)}_{not.: J(\tilde{w})} \tag{133}$$

adică:

i. **Înlocuiţi în expresia $J(\tilde{w})$ din relaţia (133) pe $\tilde{w}$ cu expresia lui**

$$\tilde{w}^* = \sum_{j=1}^{n} \alpha_j \phi(x_j) \in \mathbb{R}^m, \tag{134}$$

existenţa coeficienţilor $\alpha_j \in \mathbb{R}$ care satisfac această egalitate fiind garantată de rezultatul de la punctul a. Veţi nota această nouă expresie cu $J(\alpha)$, unde $\alpha \overset{not.}{=} (\alpha_1, \ldots, \alpha_n) \in \mathbb{R}^n$.

ii. **Transformaţi *forma* expresiei $J(\alpha)$ pe care aţi obţinut-o la punctul $b.i$ într-una echivalentă, în care nu mai apare funcţia $\phi(\,)$ ci (doar) funcţia-nucleu $K(\,,\,)$.**

[238] Cf. *Matrix Identities*, Sam Roweis, 1999, http://www.cs.nyu.edu/∼roweis/notes/matrixid.pdf.

[239] Vedeţi articolul *A correspondence between Bayesian estimation on stochastic processes and smoothing by splines* din Annals of Mathematical Statistics, 41 (2): 495–502, 1970.

[240] Pentru mai multe informaţii despre acest subiect, vedeţi secţiunea *Funcţii-nucleu* din prezentul capitol.

iii. **Similar, scrieţi ecuaţia *separatorului decizional* $\tilde{w}^* \cdot \phi(x) = 0$ — unde x este o instanţă oarecare de test din $\mathbb{R}^d$ —, înlocuind $\tilde{w}^*$ conform relaţiei (134) şi, în final, eliminaţi $\phi(\,)$, punând $K(x_i, x)$ în locul produselor scalare $\phi(x_i) \cdot \phi(x)$.**

c. Fie problema de optimizare

$$\alpha^* = \arg\min_{\alpha \in \mathbb{R}^n} J(\alpha),$$

unde expresia $J(\alpha)$ a fost obţinută la punctul *b.ii*. Presupunând că se doreşte rezolvarea acestei probleme de optimizare folosind metoda gradientului descendent, deduceţi *regula de actualizare* pentru vectorul de coeficienţi α.

***Sugestie*: Pentru a vă facilita deducerea gradientului funcţiei $J(\alpha)$, vă sugerăm ca în prealabil să rescrieţi expresia funcţiei $J(\alpha)$, pe care aţi obţinut-o la punctul *b.ii*, sub formă matriceală. Concret, notând cu K_i vectorul-coloană $(K(x_i, x_1), \ldots, K(x_i, x_n))^\top$, pentru $i = 1, \ldots, n$, şi apoi cu K matricea de dimensiune $n \times n$ obţinută prin „alăturarea" vectorilor K_i, adică $K = (K_1, \ldots, K_n)$, veţi putea demonstra egalitatea**

$$J(\alpha) = \frac{1}{n} \sum_{i=1}^{n} L(K_i^\top \alpha, y_i) + \frac{\lambda}{2} \alpha^\top K \alpha.$$

Apoi veţi putea, eventual, să folosiţi formulele de calcul cu derivate vectoriale care au fost menţionate în *Sugestia* de la punctul a.

Răspuns:

a. Conform *Sugestiei* din enunţ, vom presupune că $\lambda > 0$ şi, de asemenea, că *funcţia de cost* $L : \mathbb{R}^2 \to \mathbb{R}$ este derivabilă în raport cu primul argument, pentru orice valoare fixată a celui de-al doilea argument. Pentru convenienţă, vom nota costul (sau *riscul*) empiric mediu cu regularizare L_2 astfel:

$$J(w) = \frac{1}{n} \sum_{i=1}^{n} L(w \cdot x_i, y_i) + \frac{\lambda}{2} \|w\|^2.$$

De asemenea, în cele ce urmează vom folosi notaţia $L'(z, y) \overset{not.}{=} \dfrac{\partial}{\partial z} L(z, y)$. Gradientul funcţiei $J(w)$ în raport cu vectorul w este

$$\begin{aligned} \nabla J(w) &= \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial w} L(w \cdot x_i, y_i) + \frac{\lambda}{2} \frac{\partial}{\partial w} \|w\|^2 \\ &= \frac{1}{n} \sum_{i=1}^{n} L'(w \cdot x_i, y_i) \frac{\partial}{\partial w} (w \cdot x_i) + \frac{\lambda}{2} \frac{\partial}{\partial w} w^2 = \frac{1}{n} \sum_{i=1}^{n} L'(w \cdot x_i, y_i)\, x_i + \lambda w. \end{aligned}$$

La a doua egalitate de mai sus am folosit o generalizare a regulii de derivare pentru compunere de funcţii, iar la ultima egalitate am folosit regulile de derivare vectorială (5a) şi (5b), care au fost menţionate la finalul enunţului acestei probleme.

Întrucât în punctul de minim al funcţiei J gradientul ei se anulează (adică, $\nabla J(w^*) = 0 \in \mathbb{R}^d$),[241] rezultă

[241] Proprietatea pe care am aplicat-o aici generalizează ceea ce ştim deja din liceu despre funcţiile reale f de o singură variabilă, x, care sunt derivabile, şi anume: $f'(x^*) = 0$, pentru orice punct de optim x^*.

$$w^* = -\frac{1}{n\lambda}\sum_{i=1}^{n}\underbrace{\frac{\partial}{\partial w}L(w^*\cdot x_i, y_i)}_{\in\mathbb{R}}\,x_i.$$

Considerând $\alpha_i \overset{not.}{=} -\frac{1}{n\lambda}\frac{\partial}{\partial w}L(w^*\cdot x_i, y_i)$**, relaţia precedentă va putea fi rescrisă imediat sub forma egalităţii** (132) **din enunţ.**

b. Înlocuind în expresia funcţiei $J(\tilde{w})$**, care a fost definită în relaţia** (133)**, argumentul** $\tilde{w}$ **cu expresia lui** $\tilde{w}^*$ **din relaţia** (134)**, vom obţine funcţia** J **exprimată de această dată în raport cu vectorul de coeficienţi** α**:**

$$\begin{aligned} J(\alpha) &= \frac{1}{n}\sum_{i=1}^{n}L\Big(\Big(\sum_{j=1}^{n}\alpha_j\phi(x_j)\Big)\cdot\phi(x_i),\, y_i\Big) + \frac{\lambda}{2}\Big(\sum_{j=1}^{n}\alpha_j\phi(x_j)\Big)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}L\Big(\sum_{j=1}^{n}\alpha_j\big(\phi(x_j)\cdot\phi(x_i)\big),\, y_i\Big) + \frac{\lambda}{2}\Big(\sum_{i=1}^{n}\alpha_i\phi(x_i)\Big)\cdot\Big(\sum_{j=1}^{n}\alpha_j\phi(x_j)\Big) \\ &= \frac{1}{n}\sum_{i=1}^{n}L\Big(\sum_{j=1}^{n}\alpha_j K(x_i,x_j),\, y_i\Big) + \frac{\lambda}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j K(x_i,x_j). \end{aligned} \tag{135}$$

Am ţinut cont de faptul că funcţia K **este simetrică, întrucât produsul scalar al vectorilor este simetric:** $K(x_i,x_j) \overset{def.}{=} \phi(x_i)\cdot\phi(x_j) = \phi(x_j)\cdot\phi(x_i) \overset{def.}{=} K(x_j,x_i)$**.**

Ecuaţia ***separatorului decizional*** $\tilde{w}^*\cdot\phi(x) = 0$**, unde** x **este o instanţă oarecare de test din** $\mathbb{R}^d$**, se scrie astfel:**

$$\tilde{w}^*\cdot\phi(x) = 0 \Leftrightarrow \Big(\sum_{i=1}^{n}\alpha_i\phi(x_i)\Big)\cdot\phi(x) = 0 \Leftrightarrow \sum_{i=1}^{n}\alpha_i\Big(\phi(x_i)\cdot\phi(x)\Big) = 0 \Leftrightarrow$$

$$\sum_{i=1}^{n}\alpha_i K(x_i,x) = 0. \tag{136}$$

Observaţi că expresia ***separatorului decizional*** **este liniară în** ***spaţiul de trăsături*** $\mathbb{R}^m$ **(şi anume,** $\tilde{w}^*\cdot\phi(x)$**), însă în general ea este neliniară în spaţiul iniţial,** $\mathbb{R}^d$**. Pentru exemplificare — şi pentru a înţelege mai bine afirmaţia aceasta —, vă sugerăm să-l înlocuiţi în relaţia** (136) **pe** $K(x_i,x)$ **cu expresia nucleului gaussian,** $e^{-\frac{\|x-x_i\|^2}{2\sigma^2}}$**, unde** $\sigma\in\mathbb{R}_+$ **este fixat.**[242]

c. Folosind notaţiile matriceale introduse în ***Sugestia*** **din enunţ, se verifică imediat că relaţia** (135) **se rescrie într-adevăr sub forma**

$$J(\alpha) = \frac{1}{n}\sum_{i=1}^{n}L(K_i^\top\alpha, y_i) + \frac{\lambda}{2}\alpha^\top K\alpha.$$

Gradientul funcţiei $J(\alpha)$ **se obţine uşor folosind regulile de derivare vectorială indicate în** ***Sugestia*** **de la punctul** a **(vedeţi enunţul):**

$$\nabla_\alpha J(\alpha) = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\alpha}L(K_i^\top\alpha, y_i) + \frac{\partial}{\partial\alpha}\frac{\lambda}{2}\alpha^\top K\alpha$$

[242] Această funcţie-nucleu mai este cunoscută şi sub numele de "Radial Basis Function", RBF. Vedeţi problemele 74 şi 75.

$$\overset{(5a)}{=} \frac{1}{n}\sum_{i=1}^{n} L'(K_i^\top \alpha, y_i)\,\frac{\partial}{\partial \alpha}\big(K_i^\top \alpha\big) + \frac{\lambda}{2}(K + K^\top)\alpha$$

$$\overset{(5b)}{=} \frac{1}{n}\sum_{i=1}^{n} L'(K_i^\top \alpha, y_i) K_i + \lambda K \alpha. \tag{137}$$

Am ţinut cont de faptul că matricea K este simetrică (deci $K = K^\top$), întrucât funcţia-nucleu K este simetrică, după cum am justificat mai sus.

Acum putem scrie regula gradientului descendent, astfel:

$$\alpha \leftarrow \alpha - \eta\Big[\frac{1}{n}\sum_{i=1}^{n} L'(K_i^\top \alpha, y_i) K_i + \lambda K \alpha\Big],$$

unde $\eta > 0$ este *rata de învăţare*.

***Observaţie*: Dacă ne propunem să obţinem regula de actualizare corespunzătoare gradientului descendent *stochastic*, atunci este util (şi chiar necesar!) de observat că produsl $K\alpha$ se poate scrie în mod echivalent astfel:**

$$K\alpha = \sum_{i=1}^{n} \alpha_i K_i.$$

Ţinând cont de această relaţie, precum şi de relaţia (137)**, rezultă că putem scrie regula gradientului descendent stochastic astfel:**[243]

$$\alpha \leftarrow \alpha - \eta\Big[\frac{1}{n} L'(K_i^\top \alpha, y_i) K_i + \lambda \alpha_i K_i\Big].$$

***Comentariu*:**

La problemele 9 şi 18 de la capitolul *Metode de regresie*, mai precis de la secţiunea Regresia liniară şi respectiv de la secţiunea Regresia logistică, am arătat cum anume se pot kernel-iza aceste metode, în vederea obţinerii unor separatori neliniari. Pentru aceasta, în ambele cazuri a fost esenţială *proprietatea* — care a fost argumentată acolo —, că ponderile w, care se combină liniar cu atributele de intrare, se pot rescrie ca o combinaţie liniară de instanţele de antrenament x_i.

Pentru regresia logistică, echivalenţa dintre maximizarea funcţiei de log-verosimilitate condiţională şi minimizarea costului mediu folosind funcţia de cost logistică este demonstrată la problema 13.c de la capitolul *Metode de regresie*. Pentru regresia liniară, echivalenţa dintre maximizarea funcţiei de log-verosimilitate condiţională şi minimizarea sumei pătratelor erorilor a fost demonstrată pentru cazul unidimensional la problema 1.a de la capitolul *Metode de regresie*, iar pentru cazul general la problema 3.d de la acelaşi capitol.

***Sugestie*: Vă recomandăm ca, după ce aţi înţeles această problemă, să parcurgeţi — pe lângă problemele 9 şi 18 de la capitolul *Metode de regresie* — şi problema 19 de la capitolul *Reţele neuronale artificiale*, problema 51 de la**

[243] *Observaţie*: În documentul *Supplemental lecture notes*, John Duchi recomandă următoarea variantă a acestei reguli:

$$\alpha \leftarrow \alpha - \eta[L'(K_i^\top \alpha, y_i) K_i + n\lambda\alpha_i K_i],$$

cu scopul de a evita ca estimatorul α calculat prin aplicarea metodei gradientului să fie *deplasat* (engl., biased). Se recomandă de asemenea ca valoarea lui λ să fie mică, iar η să se modifice în raport cu iteraţia. De exemplu, pentru iteraţia t se poate lua $\eta_t = 1/\sqrt{t}$ sau un multiplu al acestei valori.

capitolul *Clusterizare* şi problema 15 de la capitolul *Învăţare bazată pe memorare*. Este foarte instructiv să analizaţi elementele pe care aceste probleme le au în comun cu problema de faţă. De asemenea, menţionăm că această problemă poate constitui o foarte bună „poartă“ de intrare în capitolul de *Maşini cu vectori-suport*.

0.2 Fundamente — Probleme propuse

0.2.1 Evenimente aleatoare şi formula lui Bayes

89. **(Calcul de probabilităţi elementare)**

• ○ ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW1, pr. 1.3***

Doi soldaţi A şi B trag la ţintă. Probabilitatea ca soldatul A să greşească ţinta este de $1/5$. Probabilitatea ca soldatul B să greşească ţinta este de $1/2$. Probabilitatea ca ambii soldaţi să greşească simultan ţinta este de $1/10$.

a. Care este probabilitatea ca cel puţin unul dintre soldaţi să greşească ţinta?

b. Care este probabilitatea ca exact unul dintre cei doi soldaţi să greşească ţinta?

90. **(Aplicarea formulei probabilităţii totale)**

□ • · ***CMU, 2011 fall, T. Mitchell, A. Singh, HW1, pr. 1.e***

O urnă conţine w bile albe şi b bile negre. Extragem o bilă dintre acestea, în mod aleatoriu. Apoi repunem bila în urnă, împreună cu alte d bile de aceeaşi culoare (ca a bilei extrase). După aceasta, extragem din urnă încă o bilă, în mod aleatoriu.

a. Demonstraţi că probabilitatea ca a doua bilă extrasă să fie albă nu depinde de d.

b. Particularizaţi [raţionamentul dumneavoastră] pentru cazul $w = 2$, $b = 3$ şi $d = 7$.

91. **(Probabilităţi condiţionate, evenimente aleatoare independente: câteva proprietăţi simple)**

• ○ ***CMU, 2005 fall, T. Mitchell, A. Moore, HW1, pr. 3.1***
CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW1, pr. 1.1
CMU, (?) spring, 10-701, HW1, pr. 1.2

Fie A şi B două evenimente aleatoare.

a. Arătaţi că $P(A \mid A, B) = 1$.

b. Arătaţi că dacă $P(A) = 0$ atunci A şi B sunt independente.

c. Arătaţi că dacă $P(B) = 1$ atunci $P(A \mid B) = P(A)$.

***Observaţie*: În general, dacă $P(B) \neq 0$, din $P(A \mid B) = P(A)$ rezultă imediat $P(A \cap B) = P(A)P(B)$, ceea ce, conform definiţiei, înseamnă că A şi B sunt independente. Din acest motiv, se poate spune că $P(A \mid B) = P(A)$ este o formă mai restrictivă (deşi, mai aproape de intuiţie!) sau mai „tare“ pentru condiţia de independenţă a două evenimente aleatoare.**

92. **(Legătura dintre forma „slabă“ şi forma „tare“ a definiţiei pentru evenimente aleatoare independente condiţional)**

• ***CMU, 2005 fall, T. Mitchell, A. Moore, HW1, pr. 3.2***

Folosind doar definiţia probabilităţii condiţionate arătaţi că dacă $P(A \mid B, C) = P(A \mid C)$ sau $P(B \mid A, C) = P(B \mid C)$ atunci $P(A, B \mid C) = P(A \mid C) \cdot P(B \mid C)$.

93. **(Proprietăţi ale funcţiei de probabilitate)**

• ○ ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, midterm, pr. 1.d***

Presupunem că evenimentele $B_1, B_2, \ldots, B_k$ formează o *partiţie* a *spaţiului de eşantionare* (engl., sample space), Ω.[244] Considerăm o funcţie de probabilitate P definită pe 2^Ω şi un eveniment A pentru care $P(A) > 0$.

Arătaţi că dacă $P(B_1 \mid A) < P(B_1)$, atunci $P(B_i \mid A) > P(B_i)$ pentru cel puţin o valoare a lui i din mulţimea $\{2, 3, \ldots, k\}$.

***Indicaţie*: Una sau mai multe dintre următoarele proprietăţi vă pot fi de folos:**

a. $\sum_{i=1}^{k} P(B_i) = 1$

b. $\sum_{i=1}^{k} P(B_i \cap A) = P(A)$ — **o variantă a *formulei probabilităţii totale***

c. $P(B_i \mid A) \cdot P(A) = P(B_i \cap A)$ — ***regula de înmulţire***

d. $\sum_{i=1}^{k} P(B_i \mid A) = 1$

e. $P(B_i \cap A) + P(B_i \cap \bar{A}) = P(B_i)$.

Demonstraţi în prealabil proprietăţile de mai sus, în ordinea în care au fost date.

94. **(Formula lui Bayes)**

• ○ ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW1, pr. 4***

Într-o grupă de copii de la o creşă, 30% dintre ei au ochi căprui, 50% au ochi albaştri, iar restul de 20% au ochi de alte culori. Într-o zi, educatoarea organizează cu aceşti copii un joc. Pentru prima rundă a jocului, ea selecţionează 65% dintre copiii cu ochi căprui, 82% dintre copiii cu ochi albaştri şi 50% dintre copiii cu ochi de alte culori.

Care este probabilitatea ca un copil ales în mod aleatoriu din această grupă să aibă ochi albaştri, dacă ştim că el n-a fost selecţionat pentru prima rundă a jocului?

[244] Faptul că $B_1, B_2, \ldots, B_k$ (văzute ca *mulţimi*) formează o partiţie a lui Ω revine la: $\bigcup_{i=1}^{k} B_i = \Omega$ şi $B_i \cap B_j = \emptyset$ pentru orice $i \neq j$.

95. **(Formula lui Bayes)**

• ***CMU, 2005 fall, T. Mitchell, A. Moore, HW1, pr. 3.3***

Avem două urne. Prima urnă conţine 11 bile albe şi 4 bile roşii. Cea de-a doua urnă conţine 8 bile albe şi 5 bile roşii. Se alege în mod aleatoriu cu probabilitate uniformă una din cele două urne. Apoi se extrage o bilă din urna aleasă. Dacă bila extrasă este albă, care este probabilitatea ca ea să provină din prima urnă?

96. **(Probabilităţi elementare şi probabilităţi condiţionate: Adevărat sau Fals?)**

• ∗ ***prelucrare de Liviu Ciortuz, după***
CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, HW1, pr. 4.a
CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, midterm, pr. 1

Marcaţi cu *adevărat* sau *fals* fiecare dintre afirmaţiile următoare:

a. $P(A \cup B \cup C) \geq P(A) + P(B) + P(C)$**.**

b. $P(A|B,C) = \dfrac{P(B|A,C)\,P(A|C)}{P(B|C)}$**.**

c. $P(A|C) = \sum_{i=1}^{n} P(A, B_i|C)$**, unde** $B_i \cap B_j = \emptyset$ **pentru orice** $i \neq j$ **şi** $\bigcup_{i=1}^{n} B_i = \Omega$ **(evenimentul sigur).**[245]

d. $P(A|C) = \sum_{i=1}^{n} P(A|B_i, C)\,P(B_i|C)$**, evenimentele** B_i **satisfac aceleaşi proprietăţi ca la punctul precedent.**

Justificaţi în mod riguros fiecare răspuns. Pentru afirmaţiile *false*, puteţi da un contraexemplu. Pentru afirmaţiile *adevărate*, exprimaţi în câteva cuvinte, dacă este posibil, *tipul* proprietăţii respective.

0.2.2 Variabile aleatoare

97. **(Variabile aleatoare: medii şi varianţe; exemplificări ale unor proprietăţi)**

• ∗ ***CMU, 2016 fall, N. Balcan, M. Gormley, HW1, pr. 6.3.2***

Fie X **o variabilă aleatoare având media** $E[X] = 1$ **şi varianţa** $Var\,(X) = 1$**. Calculaţi:**

$$i. E[3X]; \qquad ii. Var\,(3X); \qquad iii. Var\,(X+3).$$

[245] În locul condiţiei $\bigcup_{i=1}^{n} B_i = \Omega$ se poate considera fie proprietatea $P(\bigcup_{i=1}^{n} B_i) = 1$ fie $A \subseteq \bigcup_i B_{i=1}^{n}$, deşi ambele sunt mai laxe (i.e., mai puţin restrictive) decât condiţia din enunţ.

98. **(Variabila *indicator* pentru un eveniment aleator: calculul mediei)**

• ○ ***CMU, 2015 spring, T. Mitchell, N. Balcan, HW2, pr. 1.c***

Fie un eveniment aleatoriu oarecare A, iar X o variabilă aleatoare definită astfel:

$$X = \begin{cases} 1 & \text{dacă evenimentul } A \text{ se realizează} \\ 0 & \text{în caz contrar.} \end{cases}$$

Uneori, X este numită variabila aleatoare *indicator* pentru evenimentul A. Arătaţi că $E[X] = P(A)$, unde $E[X]$ reprezintă *media* variabilei X.

99. **(Variabile aleatoare discrete: distribuţii comune, distribuţii marginale; medii, independenţă)**

○ ***CMU, 2009 fall, Carlos Guestrin, HW1, pr. 1.2***

Fie două variabile aleatoare discrete cu distribuţia (i.e., funcţia masă de probabilitate) comună dată în tabelul de mai jos (partea dreaptă).

a. Calculaţi valorile medii ale variabilelor X şi Y în funcţie de α, β, γ şi δ.

b. Găsiţi condiţiile necesare şi suficiente astfel încât variabilele X şi Y să fie independente.

X	Y	$P(X,Y)$
0	0	α
0	1	β
1	0	γ
1	1	δ

100. **(Covarianţa nulă vs. independenţa variabilelor aleatoare binare)**

prelucrare de L. Ciortuz, după

• ∗ ***CMU, 2009 fall, Geoff Gordon, HW1, pr. 3.2***

Se ştie că atunci când covarianţa a două variabile aleatoare este nulă nu rezultă în mod neapărat că variabilele respective sunt independente (vedeţi pr. 11.a). Însă, dacă X şi Y sunt variabile aleatoare binare luând valori în mulţimea $\{0,1\}$ şi covarianţa lor este nulă, rezultă că X şi Y sunt independente (vedeţi pr. 11.b).

Arătaţi că, în cazul în care doar variabila aleatoare X este binară, nu rezultă în mod neapărat că X şi Y sunt independente, deşi covarianţa lor este nulă.

***Indicaţie*: Folosiţi ca exemplu variabilele aleatoare ale căror distribuţii sunt date în tabelul alăturat.**

X	Y	$P(X,Y)$
0	−1	0.1
0	0	0.4
0	1	0.1
1	−1	0
1	0	0.4
1	1	0

101. **(Variabile aleatoare discrete: independenţă, independenţă condiţională)**

■ • ○ ***CMU, 2015 spring, T. Mitchell, N. Balcan, HW2, pr. 1.d***

Fie X, Y şi Z variabile aleatoare luând valori în mulţimea $\{0,1\}$. Tabelul de mai jos conţine probabilităţile corespunzătoare fiecărei asignări posibile (0 sau 1) pentru variabilele X, Y şi Z:

	$Z=0$		$Z=1$	
	$X=0$	$X=1$	$X=0$	$X=1$
$Y=0$	1/15	1/15	4/15	2/15
$Y=1$	1/10	1/10	8/45	4/45

De exemplu, $P(X=0,Y=1,Z=0)=1/10$ şi $P(X=1,Y=1,Z=1)=4/45$.

a. Este oare variabila X *independentă* de variabila Y? De ce da, sau de ce nu?

b. Este X *independentă condiţional* de Y în raport cu variabila Z? De ce da, sau de ce nu?

c. Calculaţi probabilitatea $P(X=0 \mid X+Y>0)$.

102. **(Variabile aleatoare: independenţa condiţională; Formula lui Bayes)**

◦ ***CMU, 2005 fall, T. Mitchell, A. Moore, midterm, pr. 1.1***

a. Fie variabilele aleatoare H, E_1 şi E_2. Presupunem că vrem să calculăm $P(H \mid E_1, E_2)$, dar nu avem informaţii referitoare la independenţa condiţională. Care din următoarele seturi de numere sunt suficiente pentru acest scop?

i. $P(E_1,E_2)$, $P(H)$, $P(E_1 \mid H)$, $P(E_2 \mid H)$
ii. $P(E_1,E_2)$, $P(H)$, $P(E_1,E_2 \mid H)$
iii. $P(H)$, $P(E_1 \mid H)$, $P(E_2 \mid H)$.

b. Presupunem acum că $P(E_1|E_2,H)=P(E_1|H)$ pentru toate valorile posibile ale lui H, E_1 şi E_2. (Aşadar, variabila E_1 este independentă condiţional de E_2, în raport cu variabila H.) În acest caz, care dintre seturile de numere de la punctul a sunt suficiente pentru a calcula $P(H \mid E_1, E_2)$?

103. **(Independenţa condiţională a variabilelor aleatoare: o condiţie suficientă)**

prelucrare de Liviu Ciortuz, după
∗ ***CMU, 2009 fall, Geoff Gordon, HW2, pr. 2.2***
CMU, 2008 fall, Eric Xing, final exam, pr. 9.3

Arătaţi că dacă distribuţia de probabilitate comună a lui X şi Y este independentă de W în raport cu Z, atunci variabilele X şi W sunt independente în raport cu Z. Similar, variabila Y este independentă de W în raport cu Z.

În notaţie simplificată: $(X,Y) \perp W \mid Z$ implică $X \perp W \mid Z$ (şi similar $Y \perp W \mid Z$).

104. **(Variabile aleatoare continue: funcţia densitate de probabilitate)**

∗ ***CMU, 2008 spring, Eric Xing, HW1, pr. 1.1.a***

Fie funcţia $f(x)=ce^{-|x|}, -\infty<x<\infty$.

Ce valoare / valori poate avea constanta reală c în aşa fel ca f să poată reprezenta o funcţie densitate de probabilitate?

105. **(Variabile aleatoare continue: calculul unei probabilităţi, folosind funcţia densitate de probabilitate)**

∗ ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, midterm, pr. 1.b***

Presupunem că funcţia densitate de probabilitate a unei variabile aleatoare continue X este definită astfel:

$$p(x) = \begin{cases} \frac{4}{3}(1-x^3) & \textbf{pentru } 0 \leq x \leq 1 \\ 0 & \textbf{în caz contrar.} \end{cases}$$

Cât este $P(X < 0)$?

106. **(Variabile aleatoare uniforme continue: p.d.f. condiţională; independenţă)**

∗ ***CMU, 2003 fall, T. Mitchell, A. Moore, midterm, pr. 2***

Fie X şi Y variabile aleatoare continue având funcţia densitate de probabilitate comună definită astfel:

$$p(x,y) = \begin{cases} 1 & \textbf{pentru } 0 < x < 1, \mid y \mid < x \\ 0 & \textbf{în caz contrar.} \end{cases}$$

a. Cât este $p(y \mid x = 0.5)$?

b. Este variabila X independentă de variabila Y?

107. **(Variabile aleatoare discrete vs. variabile aleatoare continue; distincţia dintre p.m.f. şi p.d.f.)**

□ • ○ ***CMU, 2012 spring, Ziv Bar-Joseph, HW1, pr. 1.1***

Mickey nu este încă bine iniţiat în teoria probabilităţilor şi se confruntă cu chestiunea următoare, despre care el este înclinat să creadă că este un paradox:

Fie X o variabilă aleatoare cu distribuţia de probabilitate uniformă, definită pe intervalul $[0, \frac{1}{2}]$, şi anume $f(x) = 2$ pentru $x \in [0, \frac{1}{2}]$. Întrucât suma probabilităţilor care corespund unei distribuţii aleatoare trebuie să fie 1, Mickey este nedumerit de ce valoarea lui $f(x)$ este mai mare decât suma totală.

Explicaţi acest paradox. Altfel spus, arătaţi ce anume nu ştie Mickey.

108. **(Variabile aleatoare uniform continue, variabile aleatoare discrete; independenţă, variabile aleatoare condiţionate, medii)**

○ ***CMU, 2009 spring, Ziv Bar-Joseph, final exam, pr. 1.1***

Fie $X : \Omega \to [0,1]$ o variabilă aleatoare continuă având distribuţia uniformă (notaţie: $X \sim Uniform(0,1)$). Fie a şi b două numere reale, astfel încât $0 < a <$

$b < 1$. Definim (tot pe Ω) variabilele aleatoare discrete Y şi Z în funcţie de valorile lui X, astfel:

$$Y(\omega) = \begin{cases} 1 & \text{dacă } 0 \le X(\omega) \le a, \\ 0 & \text{altfel} \end{cases} \qquad Z(\omega) = \begin{cases} 1 & \text{dacă } b \le X(\omega) \le 1, \\ 0 & \text{altfel.} \end{cases}$$

a. Sunt variabilele Y şi Z independente? Justificaţi răspunsul.

***Indicaţie*: Pentru a găsi răspunsul corect vă sugerăm că ar fi util să completaţi tabelul de mai jos.**

y	z	$P(Y=y)$	$P(Z=z)$	$P(Y=y)P(Z=z)$	$P(Y=y, Z=z)$
0	0				
0	1				
1	0				
1	1				

b. Pentru fiecare dintre valorile z ale variabilei Z, calculaţi media variabilei condiţionate $Y \mid Z = z$. (Notaţie: $E_Y[Y \mid Z = z]$) .

109. **(Matricea de cross-covarianţă pentru doi vectori de variabile aleatoare: o proprietate)**

prelucrare de Liviu Ciortuz, după
□ • ○ ***MIT, 2006 fall, Tommi Jaakkola, HW1, pr. 5.b***

Fie A şi B două matrice de numere reale de dimensiune $p \times q$, iar x un vector aleatoriu de dimensiune $q \times 1$. Demonstraţi egalitatea următoare:

$$Cov(Ax, Bx) = A\ Cov(x)\ B^\top, \tag{138}$$

unde prin $Cov(u, v) \stackrel{def.}{=} E[(u - E[u])(v - E[v])^\top]$ am notat matricea de *cross-covarianţă* pentru doi vectori aleatori oarecare u şi v, iar prin $Cov(u) \stackrel{def.}{=} E[(u - E[u])(u - E[u])^\top]$ am notat matricea de covarianţă pentru vectorul u.[246]

***Observaţii*:**

0. Media unui vector (sau a unei matrice) de variabile aleatoare este vectorul (respectiv matricea) formată din mediile respectivelor variabile aleatoare, presupunând că aceste medii există şi sunt finite.

1. Proprietatea de *liniaritate a mediilor* (vedeţi ex. 9.a) se extinde în mod natural atunci când vectorii (sau matricele) de variabile aleatoare se înmulţesc cu matrice de numere reale.

2. Este imediat că definiţia pentru noţiunea de matrice de cross-covarianţă generalizează noţiunea de matrice de covarianţă: $Cov(u) = Cov(u, v)$ atunci când $u = v$.

[246] Definiţia matricei de covarianţă de aici coincide cu cea care a fost dată (în manieră descriptivă) la problema 20.

3. Proprietatea (138) **reprezintă o generalizare a proprietăţii**

$$Cov(aX, bY) = ab\, Cov(X, Y), \text{ pentru } \forall a, b \in \mathbb{R} \text{ şi } \forall X, Y \text{ variabile aleatoare},$$

proprietate care a fost introdusă la problema 9.c.

4. Ştim de la ex. 9.c că pentru oarice două variabile aleatoare X şi Y are loc egalitatea $Cov\,(X, Y) = E[XY] - E[X] \cdot E[Y]$. Se poate demonstra că această relaţie se extinde în mod natural la matricea de cross-covarianţă pentru vectori de variabile aleatoare: $Cov\,(u, v) = E[uv^\top] - E[u](E[v])^\top$.

110. **(Variabile aleatoare: Adevărat sau Fals?)**

□ • ○ ***CMU, 2005 fall, T. Mitchell, A. Moore, midterm, pr. 1.2***

Fie X şi Y două variabile aleatoare. Care dintre următoarele afirmaţii sunt adevărate?

a. Dacă X şi Y sunt independente, atunci $E[2XY] = 2E[X]\,E[Y]$ şi $Var[X+2Y] = Var\,[X] + Var\,[Y]$.

b. Dacă X şi Y sunt independente, iar $X > 1$, atunci $Var\,[X + 2Y^2] = Var\,[X] + 4\,Var\,[Y^2]$ şi $E[X^2 - X] \geq Var\,[X]$.

c. Dacă X şi Y nu sunt independente, atunci $Var\,[X + Y] = Var\,[X] + Var\,[Y]$.

d. Dacă X şi Y sunt independente, atunci $E[XY^2] = E[X]\,E[Y]^2$ şi $Var[X+Y] = Var\,[X] + Var\,[Y]$.

e. Dacă X şi Y nu sunt independente şi $f(X) = X^2$, atunci $E[f(X)\,Y] = E[f(X)]\,E[Y]$ şi $Var\,[X + 2Y] = Var\,[X] + 4\,Var\,[Y]$.

0.2.3 Distribuţii probabiliste uzuale

111. **(Distribuţia binomială: exemple de aplicare, calcularea unor valori pentru funcţia masă de probabilitate)**

• ○ ***CMU, 2009 spring, Ziv Bar-Joseph, HW1, pr. 1.4***

Un marinar încearcă să meargă pe o punte alunecoasă, însă datorită mişcărilor navei, el poate face exact un pas la fiecare interval de timp egal cu unitatea, şi anume: fie un pas înainte (cu probabilitatea p), fie un pas înapoi (cu probabilitatea $1 - p$). Marinarul nu poate rămâne imobil.

Poziţia marinarului la momentul de timp $i \in \{0, 1, \ldots\}$ va fi exprimată cu ajutorul unei variabile aleatoare $X_i \in \{-\infty, \ldots, -2, -1, 0, 1, 2, \ldots, +\infty\}$, astfel:

$X_0 = 0$ **cu probabilitate** 1;
$P(X_{t+1} = x_i + 1 \mid X_t = x_i) = p$, **pentru** $t \geq 0$;
$P(X_{t+1} = x_i - 1 \mid X_t = x_i) = 1 - p$, **pentru** $t \geq 0$.

a. Cât este $P(X_{16} = 8)$, adică probabilitatea ca marinarul să se afle în poziţia $+8$ după exact 16 unităţi de timp?

b. Generalizare: Cât este $P(X_n = r)$, adică probabilitatea ca marinarul să se afle în poziţia r după exact n unităţi de timp (considerând $n \geq r$ şi $n - r$ număr par)?

c. Bazat pe formula de la punctul precedent, calculaţi $P(X_{32} = 16)$.

d. Este probabilitatea de la punctul c aceeaşi cu $P(X_{32} = 16 \mid X_{16} = 8)$, adică probabilitatea ca marinarul să se afle la poziţia $+16$ după exact 32 de unităţi de timp ştiind că la momentul de timp 16 se afla la poziţia $+8$? Justificaţi riguros.

112. **(Distribuţia categorială: calcul de medii şi probabilităţi)**

■ • ○ ***CMU, 2009 fall, Geoff Gordon, HW1, pr. 4***

Presupunem că avem n coşuri şi m mingi. Aruncăm mingile în coşuri în mod independent şi aleatoriu, aşa încât fiecare minge este la fel de probabil să cadă în oricare dintre coşuri. (Pentru simplitate, vom presupune că la orice aruncare mingea cade într-un coş oarecare.)

a. Care este probabilitatea ca prima minge să cadă în primul coş?

b. Care este, în medie, numărul de mingi care au căzut în primul coş?

***Sugestii*:**
1. Puteţi defini o variabilă aleatoare care să reprezinte (similar unei *funcţii-indicator*; vedeţi problema 98) faptul că mingea i a căzut în primul coş:

$$X_i = \begin{cases} 1 & \text{dacă mingea } i \text{ a căzut în primul coş,} \\ 0 & \text{altfel.} \end{cases}$$

2: Ţineţi cont de proprietatea de liniaritate a mediei (vedeţi problema 9.a).

c. Care este probabilitatea ca primul coş să rămână gol după aruncarea celor m mingi?

d. Care este, în medie, numărul de coşuri care rămân goale după aruncarea celor m mingi?

113. **(O mixtură de distribuţii Bernoulli; formula lui Bayes)**

• ○ ***CMU, 2009 fall, Carlos Guestrin, HW1, pr. 1.3***

Avem două monede dintre care una este perfectă, iar cealaltă nu. În cazul monedei imperfecte, probabilitatea de a obţine stema este de $1/20$. Închizând ochii, alegem cu probabilitate de $1/2$ una din cele două monede, după care o aruncăm de două ori.

Calculaţi:
a. probabilitatea să obţinem stema la prima aruncare;
b. probabilitatea să fi ales moneda perfectă ştiind că la ambele aruncări s-a obţinut stema.

114. **(O mixtură de distribuţii Bernoulli; formula lui Bayes)**

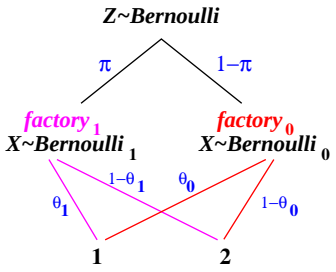
■ □ • ○ ***MIT, 2016 fall, R. Barzilay, S. Sra, Weekly exercises, week 2, pr. 3***



Să zicem că tocmai ai cumpărat un zar cu două feţe — ca în figura alăturată —, la aruncarea căruia se poate produce fie rezultatul 1 fie rezultatul 2.

Intenţionezi să foloseşti acest zar straniu la nişte jocuri cu prietenii tăi în seara aceasta, dar mai întâi ai dori să ştii care este probabilitatea ca zarul să producă rezultatul 1.

Cunoşti faptul că zarul provine fie de la fabrica 0 fie de la fabrica 1, dar nu ştii de la care anume dintre ele. Fabrica 0 livrează zaruri care produc rezultatul 1 cu probabilitatea θ_0, iar fabrica 1 livrează zaruri care produc rezultatul 1 cu probabilitatea θ_1. Iniţial, tu eşti de părere că zarul provine de la fabrica 1 cu probabilitatea π.



a. Fără să fi văzut încă vreo aruncare a acestui zar, ci doar ţinând cont de cele spuse mai sus, ai putea să calculezi cât este $P(X = 1)$, adică probabilitatea ca acest zar să producă rezultatul 1?

b. Dacă aruncăm zarul şi „observăm" rezultatul acestei aruncări, poţi să inferezi ceva despre fabrica la care a fost el realizat? (Dedu expresia de calcul pentru probabilităţile $P(Z = 1|X = x)$ şi $P(Z = 0|X = x)$; detaliază.)

c. În mod concret, să presupunem că

- **$\theta_0 = 1$, adică orice zar livrat de fabrica 0 produce întotdeuna rezultatul 1;**
- **$\theta_1 = 0.5$, adică orice zar livrat de fabrica 1 este perfect (produce rezultatul 1 cu probabilitatea de 0.5);**
- **$\pi = 0.7$, adică eşti de părere că zarul tău provine cu probabilitate de 0.7 de la fabrica 1.**

Acum aruncăm zarul, iar rezultatul este 1. Cât este $P(Z = 1|X = x)$, adică probabilitatea a posteriori ca el să provină de la fabrica 1?

d. Aruncăm zarul din nou, iar rezultatul este iarăşi 1! Cât este acum probabilitatea a posteriori ca el să provină de la fabrica 1, adică $P(Z = 1|X_1 = 1, X_2 = 1)$?

e. Răspundeţi la aceeaşi întrebare ca la punctul d, considerând însă că la cea de-a doua aruncare s-a obţiunut rezultatul 2.

f. Demonstrează în cazul general — adică, fără a folosi valorile numerice date mai sus — că dacă se „observă" rezultatele a două aruncări (notate cu x_a şi x_b), atunci

$$P(x_a, x_b) = P(x_b, x_a) \text{ şi, de asemenea, } P(Z = z|x_a, x_b) = P(Z = z|x_b, x_a),$$

adică ordinea acestor „observaţii" nu influenţează rezultatul cu privire la probabilitatea a posteriori ca el să provină de la fabrica z.

***Indicţie*: Prima egalitate se demonstrează aplicând mai întâi regula de înmulţire şi apoi independenţa condiţională. La demonstrarea celei de-a doua egalităţi se face uz de prima egalitate.**

115. **(Calculul mediei şi al varianţei unei distribuţii uniforme continue)**

□ • ○ ***CMU, 2012 spring, Ziv Bar-Joseph, HW1, pr. 1.2***

Fie X o variabilă aleatoare cu distribuţia de probabilitate uniformă, definită pe intervalul $[0, \frac{1}{2}]$, şi anume $f(x) = 2$ pentru $x \in [0, \frac{1}{2}]$. Calculaţi media şi varianţa acestei distribuţii folosind definiţia mediei şi respectiv a varianţei.

116. **(Exemple de distribuţii probabiliste continue care nu au medie finită)**

prelucrare de Liviu Ciortuz, 2019, după
□ • ○ ***CMU, 2017 fall, Nina Balcan, HW1, pr. 19***

a. Distribuţia Cauchy de parametru $\theta \in \mathbb{R}$ are p.d.f.-ul (adică, funcţia de densitate de probabilitate; engl. probability density function):

$$f(x) = \frac{1}{\pi} \cdot \frac{1}{1 + (x - \theta)^2}.$$

Arătaţi că media oricărei variabile aleatoare care urmează o distribuţie Cauchy este $+\infty$.

b. Procedaţi similar pentru distribuţia continuă având următoarea densitate de probabilitate:

$$p(x) = \begin{cases} \frac{1}{x^2} & \text{dacă } x \geq 1 \\ 0 & \text{altfel.} \end{cases}$$

***Indicaţie*: Veţi demonstra în prealabil că aceste două funcţii sunt într-adevăr funcţii de densitate de probabilitate.**

117. **(Distribuţia gaussiană bidimensională: eşantionare; o implementare simplă)**

□ • ○ ***CMU, 2018 spring, N. Balcan, HW0***

a. Generaţi 100 instanţe $x = (x_1, x_2)$ folosind o distribuţie gaussiană bidimensională având ca medie vectorul $(0, 0)^\top$ şi ca matrice de covarianţă matricea identitate, adică $p(x) = -\frac{1}{\sqrt{(2\pi)^2}} \exp\left(-\frac{||x||^2}{2}\right)$, iar apoi faceţi un "scatter plot" (x_1 vs. x_2).

b. Cum se schimbă scatter plot-ul atunci când înlocuim media cu vectorul $(-1, 1)^\top$?

c. Cum se schimbă scatter plot-ul atunci când în matricea de covarianţă dublăm fiecare componentă?

d. Cum se schimbă scatter plot-ul atunci când înlocuim matricea de covarianţă cu următoarea matrice?

$$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

e. Cum se schimbă scatter plot-ul atunci când înlocuim matricea de covarianţă cu următoarea matrice?

$$\begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$$

118. **(Distribuţia gaussiană multidimensională): două proprietăţi)**

□ ∗ ***Mihaela Hudişteanu, UAIC, Iaşi, 2026***

Fie X o variabilă aleatoare care urmează distribuţia gaussiană d-dimensională de medie μ şi matrice de covarianţă Σ (notaţie: $X \sim \mathcal{N}(\mu, \Sigma)$).
Demonstraţi că
i. **$E[X]$, media variabilei X, este egală cu μ;**
ii. **$Cov(X)$, matricea de covarianţă a variabilei X, este egală cu Σ.**

***Observaţie*: Pentru cazul unidimensional, demonstraţia a fost făcută la ex. 32.bc.**

119. **(Mixturi de distribuţii gaussiene multidimensionale: media (i.e., vectorul de medii) şi matricea de covarianţă)**

prelucrare de Liviu Ciortuz, după
□ • ○ ***CMU, 2015 fall, Z. Bar-Joseph, E. Xing, HW4, pr. 3***

Fie o mixtură de distribuţii gaussiene multidimensionale formată din K componente:

$$p(x) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(x|\mu_k, \Sigma_k). \tag{139}$$

a. Arătaţi că $E[x] = \sum_{k=1}^{K} \pi_k \, \mu_k$.

b. Arătaţi că

$$Cov[x] \overset{def.}{=} E[(x - E[x])(x - E[x])^\top] = \sum_{k=1}^{K} \pi_k [\Sigma_k + \mu_k (\mu_k)^\top] - E[x](E[x])^\top .$$

***Observaţie*: Ştim de la ex. 9.b că pentru orice variabilă aleatoare X are loc egalitatea $Var[X] = E[X^2] - (E[X])^2$. Se poate demonstra că această relaţie se extinde în mod natural la [matricea de covarianţă pentru] orice vector x de variabile aleatoare: $Cov(x) = E[xx^\top] - E[x](E[x])^\top$. Ca o *consecinţă*, folosind rezultatul de la ex. 118.*ii* obţinem imediat următorul rezultat: pentru orice distribuţie gaussiană multidimensională de densitate de probabilitate $\mathcal{N}(\mu, \Sigma)$ urmează că $E[xx^\top] = \Sigma + \mu\mu^\top$.**

120. **(Mixturi de distribuţii [oarecare]: calculul mediilor şi al varianţelor)**

□ • · ***CMU, 2010 fall, Aarti Singh, HW1, pr. 2.2.3-5***

Modelele de mixturi (engl., mixture models) sunt adeseori folosite în învăţarea automată şi în statistică.

Presupunem că pe un anumit spaţiu de eşantionare (engl., sample space) sunt definite câteva distribuţii de probabilitate: $P_i(X) = P(X|C = i), i = 1, \ldots, k$. (Ca exemplu, gândiţi-vă la cele două zaruri din problema 29; însă aici nu impunem restricţii asupra distribuţiilor P_i, deci ele pot fi şi distribuţii continue.)

Considerăm că dispunem şi de o distribuţie probabilistă definită peste aceste „componente“ ale mixturii, identificată prin probabilităţile $P(C = i)$, unde C este o variabilă aleatoare discretă luând k valori. (La problema 29, unde $k = 2$, variabila C este reprezentată de aruncarea monedei.)

a. Exprimaţi $P(X)$ în funcţie de $P_i(X)$ şi $P(C = i)$ pentru $i = 1, \ldots, k$.

b. Exprimaţi $E[X]$ în funcţie de $E[X|C]$. Justificaţi în detaliu.

c. Exprimaţi $Var[X]$ în funcţie de $Var[X|C]$ şi $E[X|C]$. Justificaţi în detaliu.

121. (Funcţia generatoare de *momente* pentru o variabilă aleatoare: definiţie; două exemplificări; o proprietate)

□ • · ***Liviu Ciortuz, pornind de la Stanford, Machine Learning course, John Duchi, Supplemental Lecture Notes – Hoeffding's inequality***

Fie Z o variabilă aleatoare. *Funcţia generatoare de momente* (engl., moment generating function) a lui Z este definită astfel:[247]

$$M_Z(\lambda) \stackrel{def.}{=} E[\exp(\lambda Z)],$$

care poate fi infinită pentru anumite valori ale lui λ.

a. Funcţia generatoare de momente are o proprietate interesantă în raport cu suma de variabile aleatoare. Fie $Z_1, \ldots, Z_n$ variabile independente. Demonstraţi că

$$M_{Z_1+\ldots+Z_n}(\lambda) = \prod_{i=1}^{n} M_{Z_i}(\lambda).$$

Aceasta înseamnă că funcţia generatoare de momente pentru suma unor variabile aleatoare independente este egală cu produsul funcţiilor generatoare de momente pentru respectivele variabile.

b. Demonstraţi că pentru $Z \sim \mathcal{N}(0, \sigma^2)$ avem

$$M_Z(\lambda) = \exp\Big(\frac{\lambda^2\sigma^2}{2}\Big).$$

c. Fie S o *variabilă* aleatoare *Rademacher*. Se mai spune că S este o *variabilă aleatoare semn* (engl., random sign variable). Prin definiţie, $S = 1$ cu probabilitate $1/2$ şi $S = -1$ cu probabilitate $1/2$. Demonstraţi că

$$M_s(\lambda) \leq \exp\Big(\frac{\lambda^2}{2}\Big) \text{ pentru orice } \lambda \in \mathbb{R}.$$

***Sugestie*: Vă recomandăm să folosiţi mai întâi dezvoltarea în serie Taylor pentru funcţia exponenţială, adică $e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!}$, şi apoi inegalitatea evidentă $(2k)! \geq 2^k$ pentru orice $k \in \mathbb{N}$.**

[247] Pentru o prezentare succintă a noţiunii statistice de *moment*, vedeţi intrarea *moments* de la pag. 282 din *The Cambridge Dictionary of Statistics*, B. S. Everitt, A. Skrondal, Cambridge University, 2010.

122. **(Distribuţii probabiliste discrete şi distribuţii probabiliste continue)**

□ • ○ ***CMU, 2015 spring, T. Mitchell, N. Balcan, HW1, pr. 2.2***

Faceţi corespondenţa dintre numele de distribuţii probabiliste din coloana din stânga cu funcţiile [masă, respectiv densitate] de probabilitate din coloana din dreapta.

a. **gaussiană multidimensională**
b. **exponenţială**
c. **uniformă**
d. **Bernoulli**
e. **binomială**

f. $p^{1-x}(1-p)^x$ **cu** $x \in \{0, 1\}$
g. $\frac{1}{b-a}$ **pentru** $a \leq x \leq b$; 0 **în caz contrar**
h. $C_n^x p^x (1-p)^{n-x}$ **pentru** $x \in \{0, \ldots, n\}$
i. $\lambda e^{-\lambda x}$ **pentru** $x \geq 0$; 0 **în caz contrar**
j. $\frac{1}{\sqrt{(2\pi)^d |\Sigma|}} \exp\left(-\frac{1}{2} - (x-\mu)^\top \Sigma^{-1} (x-\mu)\right)$

123. **(Familia de distribuţii exponenţiale: exemplificare)**

□ • · ∗ ***Liviu Ciortuz, pornind de la Stanford, 2007 fall, Andrew Ng, ML course, Lecture Notes, Part III – Generalized Liniar Models***

Spunem că o clasă de distribuţii face parte din familia exponenţială dacă ea poate fi scrisă sub forma următoare:

$$p(y|\eta) = b(y) \exp(\eta^\top T(y) - a(\eta))$$

În această expresie, η **se numeşte *parametrul natural* (sau, parametrul *canonic*) al distribuţiei;** $T(y)$ **este *statistica suficientă* (adeseori vom avea** $T(y) = y$**); iar** $a(\eta)$ **este numit *funcţia de log-partiţie*. Cantitatea** $e^{-a(\eta)}$ **joacă rolul de constantă de normalizare, datorită căreia valorile distribuţiei** $p(y|\eta)$ **– indexate în raport cu** y **– se sumează / integrează la valoarea** 1**.**

Atunci când se fixează T**,** a **şi** b**, se defineşte o familie (sau o mulţime) de distribuţii care este parametrizată prin intermediul lui** η**; variind valorile lui** η**, obţinem diferite distribuţii din cadrul acestei familii.**

Demonstraţi că distribuţia Bernoulli, distribuţia gaussiană (normală) standard (mai general, distribuţia gaussiană de medie μ **şi varianţă** 1**),**[248] **distribuţia exponenţială, precum şi distribuţia categorială sunt parte a familiei de distribuţii exponenţiale.**[249]

[248] Pentru distribuţia gaussiană $\mathcal{N}(\mu, \sigma^2)$ ca membru în familia exponenţială, vedeţi problema 43.a de la capitolul *Metode de regresie*.

[249] Similar, pentru distribuţia multinomială, distribuţia Dirichlet şi distribuţia gaussiană multivariată, vedeţi problema 41 de la acest capitol, iar pentru distribuţia geometrică şi distribuţia Poisson, vedeţi problema 42.a de la capitolul *Metode de regresie*.

0.2.4 Estimarea parametrilor unor distribuţii probabiliste

124. **(Distribuţia Bernoulli: câteva chestiuni de bază; estimarea parametrului)**

□ • ○ ***CMU, 2018 spring, Nina Balcan, HW0, pr. A.3.1-4***

Considerăm setul de date $S = \{1, 1, 0, 1, 0\}$, obţinut prin aruncarea unei monede X de cinci ori, unde 0 indică faptul că la aruncarea monedei a fost obţinută faţa cu banul, iar 1 semnifică faptul că la aruncarea monedei a fost obţinută faţa cu stema.

a. Calculaţi $\bar{x} = \frac{1}{|S|} \sum_{x_i \in S} x_i$, *media empirică* (sau, media la eşantionare; engl., the sample mean) pentru acest set de date.

b. Calculaţi $\sigma^2 = \frac{1}{|S|} \sum_{x_i \in S} (x_i - \bar{x})^2$, *varianţa empirică* (sau, varianţa la eşantionare; engl., the sample variance) pentru acest set de date.

c. Calculaţi probabilitatea de a „observa“ / obţine aceste date, presupunând că ele au fost generate prin aruncarea unei monede perfecte, adică având distribuţia de probabilitate $P(X = 1) = 0.5$ şi $P(X = 0) = 0.5$.

d. Remarcaţi faptul că probabilitatea [generării] acestui set de date ar fi putut fi mai mare dacă valoarea lui $P(X = 1)$ n-ar fi fost 0.5, ci alta. Care este valoarea care maximizează probabilitatea [generării] setului S? Justificaţi riguros.

125. **(Distribuţia Bernoulli, estimarea parametrului în sens MLE: bias-ul şi varianţa estimatorului)**

prelucrare de Liviu Ciortuz, după

■ □ • ○ ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW2, pr. 1***

Fie X o variabilă aleatoare binară, luând valoarea 0 cu probabilitatea p şi valoarea 1 cu probabilitatea $1 - p$. Fie instanţele $X_1, \ldots, X_n$ produse în mod independent, conform distribuţiei variabilei X.

a. Calculaţi o estimare în sens MLE pentru p; o veţi nota cu $\hat{p}$.

b. Este oare $\hat{p}$, văzut ca variabilă aleatoare, un estimator nedeplasat (engl., unbiased estimator) al lui p? Demonstraţi.

c. Calculaţi media erorii pătratice a lui $\hat{p}$ (engl., the expected square error of $\hat{p}$) în raport cu p, adică $E[(\hat{p} - p)^2]$.

***Observaţie*: Veţi vedea că, datorită rezultatului de la punctul precedent, această medie coincide cu varianţa lui $\hat{p}$.**

Cum evoluează valorile lui $Var[\hat{p}]$ pe măsură ce numărul de instanţe considerate (n) tinde la infinit?

d. Demonstraţi că atunci când ştim că p se află în intervalul $[1/4; 3/4]$ şi ne sunt date doar $n = 3$ „observări“ ale variabilei X, rezultă că $\hat{p}$ este un estimator *inadmisibil* pentru parametrul p relativ la minimizarea mediei pătratelor erorilor (engl., expected square error) produse prin estimare.

***Notă*: Se zice că un estimator δ pentru parametrul θ este *inadmisibil* dacă există un alt estimator δ' astfel încât**

$R(\theta, \delta') \leq R(\theta, \delta)$ pentru orice valoare a lui θ, şi

$R(\theta, \delta') < R(\theta, \delta)$ pentru o valoare oarecare a lui θ,

unde $R(\theta, \delta)$ este o *funcţie de risc* (engl., risk function), care în problema de faţă va fi considerată media pătratelor erorilor estimatorului, $E[(\hat{p} - p)^2]$.

126. **(Distribuţia Bernoulli: estimarea parametrului (MLE şi MAP), într-un caz particular)**

□ • ○ ***CMU, 2011 fall, T. Mitchell, A. Singh, midterm exam, pr. 2***

În acest exerciţiu veţi estima probabilitatea ca la aruncarea unei monede să apară faţa *stemă* (engl., head), făcând estimări în sensul verosimilităţii maxime (MLE) şi, respectiv, în sensul probabilităţii maxime a posteriori (MAP).

Presupunem că dispunem de o monedă pentru care probabilitatea de apariţie a feţei cu stema este $p = 0.5$, adică este o monedă perfectă (engl., fair coin). Totuşi, am dori să-i calculăm un estimator, $\hat{\theta}$. În mod obişnuit, un astfel de estimator — pentru o distribuţie de tip Bernoulli — se calculează presupunând că el poate lua orice valoare din intervalul $[0, 1]$ (vedeţi problema 43.c sau problema 125.a). Aici, în schimb, vom impune ca $\hat{\theta}$ să ia valori într-o mulţime finită, şi anume $\{0.3, 0.6\}$.

a. Presupunem că am aruncat moneda de 3 ori şi am obţinut de 2 ori *banul* şi o dată *stema*. Calculaţi estimarea de verosimilitate maximă $\hat{\theta}_{MLE}$ a lui p în raport cu setul de valori posibile, $\{0.3, 0.6\}$.[250]

b. Presupunem că folosim următoarea distribuţie de probabilitate a priori pentru valorile parametrului p:

$$P(p = 0.3) = 0.3 \text{ şi } P(p = 0.6) = 0.7.$$

Presupunem din nou că am aruncat moneda de 3 ori şi am obţinut de 2 ori *banul* şi o dată *stema*. Calculaţi estimarea de probabilitate maximă a posteriori $\hat{\theta}_{MAP}$ a lui p în raport cu mulţimea de valori posibile, $\{0.3, 0.6\}$, folosind această distribuţie a priori.

c. Presupunem că aruncăm moneda de un număr de ori care tinde la infinit. În acest caz, care va fi estimarea de verosimilitate maximă $\hat{\theta}_{MLE}$ a lui p în raport cu setul de valori posibile, $\{0.3, 0.6\}$? Justificaţi răspunsul. (Vă readucem aminte că moneda este perfectă, deci $p = 0.5$.)

d. Presupunem din nou că aruncăm moneda de un număr de ori care tinde la infinit. Calculaţi estimarea de probabilitate maximă a posteriori $\hat{\theta}_{MAP}$ a lui p în raport cu mulţimea de valori posibile, $\{0.3, 0.6\}$, folosind distribuţia de probabilitate a priori care a fost definită la punctul b.

[250] LC: Se va considera că aceste două valori sunt echiprobabile.

127. **(Estimare în sens MLE pentru distribuţia Bernoulli şi distribuţia binomială: exemplificare)**

□ • ***CMU, 2018 spring, Nina Balcan, HW2, pr. 2***

Presupunem că „observăm“ valorile a n variabile aleatoare i.i.d. $X_1, \ldots, X_n$, toate urmând o [aceeaşi] distribuţie Bernoulli de parametru θ.[251] Cu alte cuvinte, ştim că pentru orice $i = 1, \ldots, n$, au loc egalităţile

$$P(X_i = 1) = \theta \text{ şi } P(X_i = 0) = 1 - \theta.$$

***Scopul* nostru este să estimăm [în sensul verosimilităţii maxime, MLE] valoarea parametrului θ, pornind de la aceste valori „observate“, $X_1, \ldots, X_n$.**

a. Scrieţi formula funcţiei de log-verosimilitate, $\ell(\theta)$. Această funcţie va depinde de variabilele aleatoare $X_1, \ldots, X_n$ şi de parametrul θ. Aduceţi expresia acestei funcţii la cea mai simplă formă posibilă. (Aşadar, nu vă limitaţi la a scrie doar definiţia funcţiei de log-verosimilitate.)

Determinaţi o formulă analitică (engl., a closed form expression) pentru estimarea de verosimilitate maximă, $\hat{\theta}_{MLE}$.[252]

Folosiţi formula pe care aţi determinat-o mai sus pentru a calcula θ_{MLE} pentru următoarele 10 „observaţii“:

$$X = (0, 1, 1, 1, 1, 0, 1, 0, 1, 1).$$

b. Acum vom considera o altă distribuţie, înrudită cu cea de mai sus. Presupunem că „observăm“ valorile a m variabile aleatoare i.i.d. $Y_1, \ldots, Y_m$, toate urmând o aceeaşi *distribuţie binomială* $B(n, \theta)$. O distribuţie binomială modelează numărul de apariţii ale lui 1 dintr-o secvenţă de n variabile independente Bernoulli de parametru θ. Cu alte cuvinte,

$$P(Y_i = k) = C_n^k \theta^k (1 - \theta)^{n-k} = \frac{n!}{k!\,(n-k)!} \theta^k (1 - \theta)^{n-k}.$$

Scrieţi expresia funcţiei de log-verosimilitate $\ell(\theta)$ pentru $Y_1, \ldots, Y_m$, care au valorile $k_1, \ldots, k_m$, respectiv. Această funcţie depinde de m, n, $k_1, \ldots, k_m$ şi θ.

Determinaţi o formulă analitică pentru estimarea de verosimilitate maximă a parametrului θ.

Apoi folosiţi această formulă pentru a calcula θ_{MLE} în cazul a două variabile aleatoare binomiale Y_1 şi Y_2, ambele având parametrii $n = 5$ şi θ. Variabilele Bernoulli pentru Y_1 şi Y_2 au produs valorile $(0, 1, 1, 1, 1)$ şi respectiv $(0, 1, 0, 1, 1)$, prin urmare $Y_1 = 4$ şi $Y_2 = 3$.

c. Comparaţi estimările pe care le-aţi obţinut la punctele a şi b. Dacă aceste două estimări diferă, care credeţi că este motivul?

[251] Abrevierea i.i.d. înseamnă: independente şi identic distribuite.

[252] *Sugestie*: Vă reamintim că x^* este un *punct critic* pentru funcţia $f(x)$ dacă $f'(x^*) = 0$. Nu uitaţi să argumentaţi de ce a maximiza funcţia $lnf(x)$ este echivalent cu a maximiza funcţia $f(x)$.

128. **(Estimare în sens MLE pentru parametrul distribuţiei binomiale folosind metoda gradientului, metoda lui Newton şi metoda analitică (prin calcul direct, cu ajutorul derivatei))**

prelucrare de Liviu Ciortuz, după
□ • ***CMU, 2008 spring, T. Mitchell, W. Cohen, HW2, pr. 1.2***

Funcţia masă de probabilitate pentru distribuţia binomială (engl., probability mass function, p.m.f.) este definită prin expresia $f(x) = C_n^x p^x (1-p)^{n-x}$.

În cele ce urmează veţi considera $n = 100$ **şi** $x = 8$. **Găsiţi [cât este] estimarea de verosimilitate maximă (engl., maximum likelikood estimation, MLE) pentru parametrul** p **al distribuţiei binomiale folosind mai întâi metoda gradientului şi apoi metoda lui Newton. Faceţi o implementare a acestor două metode în Python / R / Matlab sau în limbajul de programare pe care îl preferaţi. Folosiţi condiţia de oprire următoare:** $|p_j - p_{j-1}| \leq 0.0001$.

Veţi începe aplicarea ambelor metode dând parametrului p **valoarea iniţială** $p_0 = 0.1$. **Indicaţi rezultatul** (p_j) **care se obţine la convergenţă. Ulterior veţi verifica acest rezultat cu [cel obţinut prin] metoda analitică.**

129. **(Distribuţia categorială: estimare în sens MAP, folosind distribuţia Dirichlet)**

□ • ○ ***CMU, 2014 spring, B. Poczos, A. Singh, HW1, pr. 3***

Presupunem că tocmai ai primit un zar cu 6 feţe de la prietenul tău care este statistician. Din păcate, el nu-şi aminteşte exact ce valori au parametrii distribuţiei categoriale asociate acestui zar. În schimb, îşi aminteşte că a generat aceşti parametri $(p_1, p_2, \ldots, p_6)$ **folosind următoarea distribuţie Dirichlet:**

$$P(p_1, p_2, \ldots, p_6 | \theta_1, \ldots, \theta_6) = \frac{\Gamma(\sum_{i=1}^6 \theta_i)}{\prod_{i=1}^6 \Gamma(\theta_i)} \cdot \prod_{i=1}^6 p_i^{\theta_i - 1} \cdot \delta\Big(\sum_{i=1}^6 p_i - 1\Big),$$

şi de asemenea că, în această formulă, el a ales $\theta_i = i$ **pentru toate valorile lui** $i = 1, \ldots, 6$. **În notaţia de mai sus,** Γ **desemnează funcţia Gamma a lui Euler,**[253] **iar** δ **este funcţia delta, adică** $\delta(a) = 1$ **dacă** $a = 0$ **şi** $\delta(a) = 0$ **în cazul contrar.**

Pentru a estima probabilităţile $p_1, p_2, \ldots, p_6$**, îţi propui să arunci zarul de 1000 de ori şi să „observi"/determini pentru fiecare faţă** $i \in \{1, \ldots, 6\}$ **care este numărul** (n_i) **de a apariţii ale feţei respective din totalul de 1000 de aruncări** $(\sum_{i=1}^6 n_i = 1000)$.

a. Arată că distribuţia Dirichlet este [o] conjugată a priori pentru distribuţia categorială.[254]

b. Care este distribuţia a posteriori pentru probabilităţile feţelor, adică $P(p_1, p_2, \ldots, p_6 | n_1, n_2, \ldots, n_6)$**?**

[253] Vedeţi problema 31.b.
[254] Pentru definiţia noţiunii de *distribuţii conjugate* vedeţi enunţul problemei 43.B.

130. **(Distribuţia geometrică: estimarea parametrului, în sens MLE şi respectiv în sens MAP)**

● · ***prelucrare de Liviu Ciortuz, după CMU, 2016 fall, N. Balcan, M. Gormley, HW2, pr. 2***

Considerăm $X_1, \ldots, X_n$ variabile aleatoare independente, toate urmând *distribuţia geometrică* (care este o distribuţie discretă) de parametru θ. Aceasta înseamnă că pentru oricare variabilă X_i şi pentru orice număr natural k avem $P(X_i = k) = (1-\theta)^k\theta$.[255]

a. Fie un set de date D, conţinând „observaţiile" $D = \{X_1 = k_1, X_2 = k_2, \ldots, X_n = k_n\}$. Scrieţi expresia funcţiei de log-verosimilitate $\ell_D(\theta)$, ca funcţie de D şi θ. Este oare valoarea acestei funcţii afectată de ordinea în care sunt „observate" cele n variabile?

b. Pornind de la funcţia $\ell_D(\theta)$ dedusă la punctul precedent, calculaţi θ_{MLE}, estimarea de verosimilate maximă (engl., Maximum Likelihood Estimation, MLE) pentru parametrul θ.

c. Fie următoarea secvenţă de 15 „observaţii":

$$X = (0, 21, 23, 8, 9, 2, 9, 0, 7, 8, 20, 9, 7, 4, 17).$$

Aplicând formula dedusă la punctul precedent, calculaţi valoarea lui θ_{MLE}, mai întâi pentru mulţimea formată din primele cinci „observaţii", adică (0, 21, 23, 8, 9), apoi pentru primele zece „observaţii" şi, în final, pentru toate cele cincisprezece „observaţii".

d. Pentru estimarea în sensul probabilităţii maxime a posteriori (engl., Maximum A posteriori Probability, MAP) a parametrului θ, vom folosi ca distribuţie *a priori* distribuţia (continuă) Beta.[256]

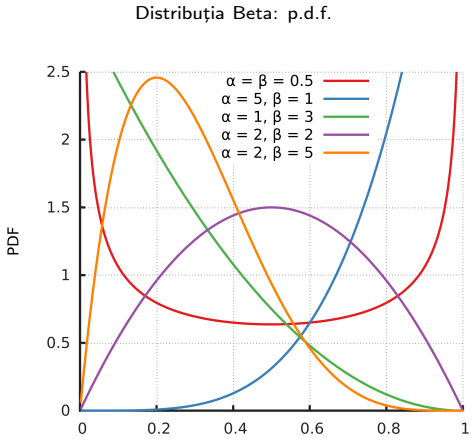
Funcţia de densitate de probabilitate pentru distribuţia Beta este

$$p(\theta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{B(\alpha, \beta)},$$

unde $B(\alpha, \beta)$ este funcţia Beta de argumente $\alpha, \beta \in \mathbb{R}_+$. Funcţia Beta se defineşte astfel:

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)},$$

cu $\Gamma(x) = (x-1)!$ pentru orice $x \in \mathbb{N}^*$.



Deduceţi formula de calcul pentru θ_{MAP}, estimarea de probabilitate maximă a posteriori pentru parametrul θ.

[255] Vedeţi nota de subsol 38.

[256] Distribuţia Beta este adeseori folosită ca „distribuţie conjugată" — în contextul estimării parametrilor în sens MAP — nu doar pentru distribuţia geometrică, ci şi pentru distribuţia Bernoulli (vedeţi ca exemplu pr. 43.B) şi pentru distribuţia binomială. Mai general, distribuţia Dirichlet, care este o generalizare a distribuţiei Beta, este distribuţie conjugată pentru distribuţia categorială (vedeţi pr. 129) şi pentru distribuţia multinomială. Pentru definiţia noţiunii de *distribuţii conjugate* vedeţi enunţul problemei 43.B.

e. Similar cu cerinţa de la punctul c, calculaţi valorile celor trei estimări în sens MAP pentru parametrul θ, folosind de fiecare dată următoarele valori pentru parametrii distribuţiei Beta: $\alpha = 1$ şi $\beta = 2$.

131. **(Estimarea parametrului unei distribuţii continue particulare, în sensul verosimilităţii maxime (MLE))**

□ • · ***CMU, 2014 fall, Z. Bar-Joseph, W. Cohen, midterm, pr. 2***

Considerăm următoarea distribuţie probabilistă, de parametru real nenul α:

$$p(x|\alpha) = \begin{cases} (\alpha^2+1)x^{\alpha^2} & x \in [0,1] \\ 0 & \text{în rest.} \end{cases}$$

Fiind dat un eşantion format din n puncte, notate $X_1, \ldots, X_n$, generate în mod independent conform distribuţiei de mai sus, stabiliţi care dintre expresiile de mai jos reprezintă valoarea lui α^2 care maximizează verosimilitatea datelor (engl., maximum likelihood estimator) în raport cu această distribuţie.

i. $\dfrac{\sum_i X_i}{n}$ *ii.* $\left(\dfrac{\sum_i X_i}{n}\right)^2$ *iii.* $-\dfrac{\sum_i \ln(X_i)}{n}$

iv. $\dfrac{-n - \sum_i \ln(X_i)}{\sum_i \ln(X_i)}$ *v.* $-\dfrac{\sum_i X_i - \sum_i \ln(X_i)}{n}$

***Indicaţie*: Folosiţi funcţia de log-verosimilitate.**

132. **(Estimare în sens MLE pentru parametrul unei distribuţii continue particulare)**

□ ○ ***CMU, 2001 fall, Andrew Moore, midterm, pr. 2***

Fie o distribuţie de probabilitate având următoarea funcţie de densitate (p.d.f.):

$$p(x) = \begin{cases} p(x) = 0 & \text{dacă } x < 0; \\ p(x) = \dfrac{2}{w} - \dfrac{2x}{w^2} & \text{dacă } 0 \le x \le w; \\ p(x) = 0 & \text{dacă } x > w. \end{cases}$$

a. Reprezentaţi grafic funcţia de densitate p dată mai sus.

b. Presupunând că variabila aleatoare X urmează distribuţia de probabilitate definită de funcţia p, care dintre expresiile următoare reprezintă media variabilei X?

i. $\int_{x=-\infty}^{\infty} \left(\frac{2}{w} - \frac{2x}{w^2}\right) dx$ *ii.* $\int_{x=-\infty}^{\infty} x \left(\frac{2}{w} - \frac{2x}{w^2}\right) dx$ *iii.* $\int_{x=-\infty}^{\infty} w \left(\frac{2}{w} - \frac{2x}{w^2}\right) dx$

iv. $\int_{x=0}^{w} \left(\frac{2}{w} - \frac{2x}{w^2}\right) dx$ *v.* $\int_{x=0}^{w} x \left(\frac{2}{w} - \frac{2x}{w^2}\right) dx$ *vi.* $\int_{x=0}^{w} w \left(\frac{2}{w} - \frac{2x}{w^2}\right) dx$

vii. $\int_{w=-\infty}^{\infty} \left(\frac{2}{w} - \frac{2x}{w^2}\right) dx$ *viii.* $\int_{w=-\infty}^{\infty} x \left(\frac{2}{w} - \frac{2x}{w^2}\right) dx$ *ix.* $\int_{w=-\infty}^{\infty} w \left(\frac{2}{w} - \frac{2x}{w^2}\right) dx$

x. $\int_{w=0}^{x} \left(\frac{2}{w} - \frac{2x}{w^2}\right) dx$ *xi.* $\int_{w=0}^{x} x \left(\frac{2}{w} - \frac{2x}{w^2}\right) dx$ *xii.* $\int_{w=0}^{x} w \left(\frac{2}{w} - \frac{2x}{w^2}\right) dx$

c. Calculaţi $P(x = 1|w = 2)$.

d. Cât este $p(x = 1|w = 2)$?

e. Cât este $p(x = 0|w = 1)$?

f. Presupunem că nu cunoaştem valoarea lui w, însă ni se furnizează o instanţă / „observaţie“ a variabilei X: $x = 3$. Care este estimarea de verosimilitate maximă a lui w?

133. **(O distribuţie uniformă continuă: estimarea parametrului, în sensul verosimilităţii maxime (MLE))**

□ • ○ ***CMU, 2015 fall, Z. Bar-Joseph, E. Xing, midterm, pr. 1.2***

Considerăm că datele $X_1, \ldots, X_n$ sunt generate în mod independent de către o distribuţie uniformă p, definită pe discul cu raza θ şi centrul în originea sistemului de coordonate din planul euclidian. Aşadar, $X_i \in \mathbb{R}^2$ şi

$$p(x|\theta) = \begin{cases} \dfrac{1}{\pi\theta^2} & \text{dacă } \|x\| \leq \theta \\ 0 & \text{în caz contrar,} \end{cases}$$

unde $\|x\| = \sqrt{x_1^2 + x_2^2}$.

Vă cerem să calculaţi estimarea de verosimilitate maximă pentru θ.

134. **(Distribuţia exponenţială: estimarea parametrilor în sens MLE şi respectiv în sens MAP, folosind ca distribuţie a priori distribuţia Gamma)**

• ○ ***CMU, 2015 fall, A. Smola, B. Poczos, HW1, pr. 1.1***

a. Distribuţia exponenţială de parameteru $\lambda > 0$ are funcţia densitate de probabilitate $\text{Exp}(x|\lambda) = \lambda e^{-\lambda x}$, definită pentru $x \geq 0$. Considerăm un set de date $\{x_i\}_{i=1}^n$ generate de această distribuţie (notaţie: $\{x_i\}_{i=1}^n \sim \text{Exp}(\lambda)$), independente şi identic distribuite (i.i.d.). Calculaţi estimarea de verosimilitate maximă (MLE) λ_{MLE}.

b. *Remember*: Distribuţia Gamma de parametri $r > 0$ şi $\alpha > 0$ are funcţia densitate de probabilitate (p.d.f.) următoare:[257]

$$\text{Gamma}(x|r, \alpha) = \frac{\alpha^r}{\Gamma(r)} x^{r-1} e^{-\alpha x} \text{ pentru } x \geq 0,$$

unde Γ desemnează funcţia gamma a lui Euler.[258]

Este oare estimatorul λ_{MLE} care a fost obţinut la punctul a un estimator deplasat (engl., biased)? Vă readucem aminte că λ_{MLE} este nedeplasat (engl., unbiased) dacă $E[\lambda_{\text{MLE}}] = \lambda$.

[257] Funcţia densitate de probabilitate (p.d.f.) pentru distribuţia Gamma $\left(r, \frac{1}{\beta}\right)$ a fost definită la problema 52.

[258] Vedeţi problema 31.b.

***Sugestie*: Pentru demonstraţie se pot folosi următoarele două proprietăţi:**
– distribuţia $\mathrm{Exp}(x|\lambda)$ **este identică cu distribuţia** $\mathrm{Gamma}(x|1,\lambda)$ **(vedeţi *Observaţia* de la problema 31.c);**
– însumând două variabile aleatoare independente care urmează distribuţiile $\mathrm{Gamma}(x|r_1,\alpha)$ **şi respectiv** $\mathrm{Gamma}(x|r_2,\alpha)$ **obţinem o variabilă care urmează distribuţia** $\mathrm{Gamma}(x|r_1+r_2,\alpha)$**.**

c. Demonstraţi că atunci când $X \overset{not.}{=} \{x_i\}_{i=1}^n$**, iar** $X \sim \mathrm{Exp}(\lambda)$ **şi** $\lambda \sim \mathrm{Gamma}(r,\alpha)$**, rezultă că** $P(\lambda|X) \sim \mathrm{Gamma}(r^*,\alpha^*)$ **pentru anumite valori** r^* **şi** α^***. Cu alte cuvinte, arătaţi că distribuţia Gamma este o distribuţie a priori *conjugată* pentru distribuţia exponenţială.**[259]

d. Calculaţi estimarea de probabilitate maximă a posteriori (MAP) λ_{MAP} **în funcţie de** r **şi** α**.**

e. Ce se întâmplă cu λ_{MLE} **şi** λ_{MAP} **atunci când** n **devine foarte mare [LC: adică, tinde la infinit]?**

135. **(Distribuţia gaussiană, cazul** $\mu = 0$**: estimarea varianţei în sensul MLE)**

■ • ○ ***CMU, 2009 spring, Ziv Bar-Joseph, HW1, pr. 2.1***

Fie X **o variabilă aleatoare de distribuţie normală (gaussiană) cu media** 0 **şi varianţa** σ^2**, adică** $X \sim \mathcal{N}(0,\sigma^2)$**.**

a. Găsiţi estimarea de verosimilitate maximă (engl., maximum likelihood estimate) pentru parametrul σ^2**, adică** σ^2_{MLE}**.**

b. Este oare estimatorul obţinut la punctul precedent *nedeplasat* (engl., unbiased)? Prin definiţie, aceasta revine la a stabili dacă are loc (sau nu) egalitatea $E[\sigma^2_{MLE}] = \sigma^2$**.**

136. **(Distribuţia gaussiană multidimensională: estimarea în sens MAP**[260] **a vectorului de medii** μ **şi a matricei de precizie** $\Lambda = \Sigma^{-1}$**)**

□ • ○ ***CMU, 2010 fall, Aarti Singh, HW1, pr. 3.2.2-3***

a. Să zicem că aveţi motive să credeţi că următoarea distribuţie[261]

$$gw(\mu,\Lambda) \overset{not.}{=} NW(\mu,\Lambda \mid \mu_0,s,V,\nu) \overset{def.}{=} \mathcal{N}(\mu \mid \mu_0,(s\Lambda)^{-1}) \cdot W(\Lambda \mid V,\nu)$$

este o distribuţie a priori convenabilă pentru modelarea parametrilor μ **şi** Λ **ai distribuţiei gaussiene** d**-dimensionale date la problema 53, cu**

$$W(\Lambda \mid V,\nu) \overset{def.}{=} \frac{|\Lambda|^{(\nu-d-1)/2}}{Z(V,\nu)} \exp\left(-\frac{\mathrm{Tr}(V^{-1}\Lambda)}{2}\right),$$

[259] Vedeţi definiţia *distribuţiilor conjugate* dată la problema 43.B.
[260] Pentru estimarea în sens MLE a parametrilor acestei distribuţii, vedeţi problema 53.a.
[261] Această distribuţie este numită uneori distribuţia [a priori] Gauss-Wishart (sau, distribuţia normală Wishart).

unde notaţia $\mathrm{Tr}(\cdot)$ desemnează *urma* (engl., trace) unei matrice diagonale, iar $1/Z(V,\nu)$ este *constanta de normalizare*. Presupunem de asemenea că ştiţi valorile hiper-parametrilor $\mu_0 \in \mathbb{R}^d$, $s > 0$, $\nu > d+1$, precum şi matricea $V \in \mathbb{R}^{d\times d}$, care este pozitiv definită. Derivaţi estimările în sens MAP (engl., Maximum A posteriori Probability) pentru parametrii μ and Λ. (Le veţi nota cu $\hat{\mu}_{MAP}$ şi respectiv $\hat{\Lambda}_{MAP}$).

b. Cum se comportă estimările $\hat{\mu}_{MAP}$ şi $\hat{\Lambda}_{MAP}$ atunci când n tinde la zero sau la infinit? Cum sunt ele în raport cu distribuţia a priori (gw) şi cu estimările în sens MLE ($\hat{\mu}_{MLE}$ şi $\hat{\Lambda}_{MLE}$)?

137. **(Distribuţia Gamma: estimarea parametrilor în sens MLE folosind metoda gradientului şi metoda lui Newton)**

prelucrare de Liviu Ciortuz, după
□ ● ○ ∗ ***CMU, 2015 fall, A. Smola, B. Poczos, HW1, pr. 1.2***

Este posibil ca pentru calcularea unor estimatori să nu existe soluţii analitice (engl., closed forms).

Am ilustrat deja această chestiune la problema 52, în raport cu distribuţia Gamma, şi anume pentru parametrul β al acestei distribuţii.

Aici vom arăta că într-o astfel de situaţie putem folosi — în locul metodelor analitice (exacte) — metode de optimizare cum sunt metoda gradientului şi metoda lui Newton.

a. Date fiind instanţele $\{x_i\}_{i=1}^n \sim \mathrm{Gamma}\left(r, \frac{1}{\beta}\right)$, elaboraţi paşii necesari pentru a calcula estimatorii $\hat{r}_{\mathrm{MLE}}$ şi $\hat{\beta}_{\mathrm{MLE}}$ pentru parametrii distribuţiei Gamma.

***Indicaţie*: Pentru $\hat{\beta}_{\mathrm{MLE}}$ veţi putea folosi expresia analitică care a fost dedusă la problema 52. Apoi, pentru a calcula valoarea $\hat{r}_{\mathrm{MLE}}$ a parametrului r pentru care se maximizează funcţia de log-verosimilitate $\ell(r, \hat{\beta}_{\mathrm{MLE}})$, veţi folosi metoda gradientului ascendent.[262] Elaboraţi *regula de actualizare* corespunzătoare.**

b. Funcţia de log-verosimilitate $\ell(r, \hat{\beta}_{\mathrm{MLE}})$ la care ne-am referit la punctul precedent este dublu derivabilă. Aşadar, pentru a optimiza această funcţie (adică, pentru a-i calcula maximul) în locul metodei gradientului, putem să folosim metoda lui Newton.[263]

Elaboraţi *regula de actualizare* corespunzătoare metodei lui Newton pentru a calcula estimatorul $\hat{r}_{\mathrm{MLE}}$ pentru distribuţia Gamma.

***Observaţie*: Regulile de actualizare obţinute la rezolvarea acestui exerciţiu vor putea face uz de funcţiile digamma ($\psi(r) \overset{def.}{=} \frac{\Gamma'(r)}{\Gamma(r)}$) şi trigamma ($\psi'(r)$), unde $\Gamma(r)$ desemnează funcţia a lui Euler.[264]**

c. Pe site-ul acestei culegeri, în directorul corespunzător prezentei probleme,[265] veţi găsi fişierul estimators.mat**, care conţine un set de „observaţii“ / instanţe**

[262] Metoda gradientului descendent a fost prezentată (prin intermediul unui exemplu) la problema 80.c.
[263] Pentru o introducere la metoda lui Newton, vedeţi *Comentariul* din enunţul problemei 80.
[264] Pentru calculul acestor funcţii există implementări disponibile. Pentru funcţia Γ a lui Euler, vedeţi problema 31.b.
[265] https://profs.info.uaic.ro/∼ciortuz/ML.ex-book/implementation-exercises/CMU.2015f.ASmola+BPoczos.HW1.pr.1.3.Gamma-distribution.grad-desc+Newton-method-for-MLE.data+sol/

care au fost generate de către o distribuţie Gamma, cu parametrii r şi β fixaţi la anumite valori. Elaboraţi o implementare a metodei gradientului ascendent (conform punctului b) şi a metodei lui Newton (conform punctului b) şi rulaţi programul respectiv pentru a obţine estimările de verosimilitate maximă (MLE) ale parametrilor acestei distribuţii. Creaţi un grafic care să pună în evidenţă convergenţa acestor două metode. Care dintre ele a necesitat mai multe iteraţii? Indicaţi cele două estimări pe care le-aţi obţinut.

138. **(Distribuţia gaussiană multidimensională): un estimator nedeplasat pentru matricea de covarianţă)**

□ • ***CMU, 2017 fall, Nina Balcan, midterm, pr. 4.4***

Fie setul de date $S = \{x_1, \ldots, x_n\}$, cu $x_i \in \mathbb{R}^d$ pentru orice $i \in \{1, \ldots, n\}$. Folosind acest set de date, vrem să facem estimare în sensul verosimilităţii maxime (MLE) pentru parametrii μ şi Σ ai unei distribuţii gaussiane multidimensionale. Demonstraţi că un *estimator nedeplasat* (engl., unbiased) pentru matricea de covarianţă Σ este

$$\hat{\Sigma} = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \hat{\mu})(x_i - \hat{\mu})^\top,$$

unde $\hat{\mu}$ este estimarea în sens MLE pentru parametrul μ.

***Sugestie*: Vedeţi problemele 135.a şi 51.**

139. **(MLE şi parametrizarea alternativă)**

□ • ***CMU, 2009 fall, Carlos Guestrin, HW1, pr. 3.1***

Estimarea în sensul verosimilităţii maxime (engl., maximum likelihood estimation, MLE) este o tehnică des folosită pentru estimarea parametrilor unui model. Totuşi, trebuie să spunem că parametrizarea unui model nu este în mod neapărat unică. Să luăm, de exemplu, distribuţia gaussiană unidimensională. Pentru această distribuţie, parametrizarea cea mai des folosită este (μ, σ), unde μ este media (engl., expected value) distribuţiei, iar σ este deviaţia standard (engl., standard deviation). În acest caz, funcţia densitate de probabilitate (engl., probability density function, p.d.f.) a distribuţiei gaussiene este:

$$f(x; \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{140}$$

O *alternativă* la această parametrizare este „parametrizarea naturală“ (numele acesta ţine de la modul în care este parametrizată familia de distribuţii exponenţiale — vedeţi problema 123 —, din care face parte şi distribuţia gaussiană). În acest caz, parametrii sunt (θ, η), iar funcţia densitate de probabilitate este:

$$f(x; \theta, \eta) = \exp\left(-\ln\sqrt{2\pi} + \frac{1}{2}\ln\eta - \eta\frac{x^2}{2} + \theta x - \frac{1}{2}\theta^2/\eta\right) \tag{141}$$

În acest exerciţiu vom vedea dacă un estimator în sensul verosimilităţii maxime pentru *o anumită parametrizare* ne poate spune ceva despre estimarea aceluiaşi model atunci când folosim *o altă parametrizare*.

a. Găsiţi o funcţie / transformare $S(\mu, \sigma) = (\theta, \eta)$ care să facă legătura între reprezentarea obişnuită pentru distribuţia gaussiană (dată de formula (140)) şi reprezentarea bazată pe parametrizarea naturală (dată de formula (141)). Este funcţia aceasta o bijecţie? (Aduceţi-vă aminte că $\sigma > 0$.)

***Indicaţie*: Două polinoame $p(x) = \sum_{i=0}^{n} a_i x^i$ şi $q(x) = \sum_{i=0}^{n} b_i x^i$ sunt identice pentru orice $x \in \mathbb{R}$ dacă şi doar dacă $a_i = b_i$ pentru orice $i = 0, \ldots, n$.**

b. Fie $f(x; \alpha)$ o funcţie densitate de probabilitate. Considerăm $S(\alpha) = \beta$ o reparametrizare a lui α, astfel încât S să fie o bijecţie, şi $g(x; \beta)$ funcţia densitate de probabilitate pentru varianta reparametrizată. Demonstraţi că dacă $\hat{\alpha}$ este estimarea de verosimilitate maximă (MLE) pentru $\{x_1, \ldots, x_n\}$, un set de instanţe / „observaţii" i.i.d., folosind parametrizarea α, atunci $\hat{\beta} = S(\hat{\alpha})$ este o estimare de verosimilitate maximă pentru funcţia de densitate reparametrizată.

***Indicaţia 1*: Scrieţi în mod explicit estimările în sens MLE folosind p.d.f.-urile g şi respectiv f. Apoi arătaţi cum anume se poate trece de la una la cealaltă folosind bijecţia S.**

***Indicaţia 2*: Reparametrizarea înseamnă că $f(x; \alpha) = g(x; S(\alpha))$ pentru orice x şi orice α.**

c. Considerăm $\{x_1, \ldots, x_n\}$ un set de n instanţe i.i.d. ale unei distribuţii gaussiene unidimensionale. Cât este estimarea de verosimilitate maximă (MLE) pentru parametrii (θ, η) care au fost introduşi la punctul a? Justificaţi riguros răspunsul dumneavoastră, pe scurt.

***Indicaţie*: Aduceţi-vă aminte cât este estimarea în sens MLE pentru (μ, σ).**

d. Considerăm o funcţie densitate de probabilitate, $f(x; \alpha)$. Fie $T(\alpha) = \beta$ o reparametrizare a lui α, iar $g(x; \beta)$ o funcţie densitate de probabilitate pentru forma reparametrizată a respectivei familii de distribuţii. Notăm $\Theta_\beta = \{\alpha | T(\alpha) = \beta\}$ şi definim *funcţia de verosimilitate indusă*, $M(\beta)$, prin

$$M(\beta) = \sup_{\alpha \in \Theta(\beta)} L(\alpha),$$

unde $L(\alpha)$ desemnează funcţia de verosimilitate corespunzătoare lui f (adică, $L(\alpha) = \prod_{i=0}^{n} f(x_i; \alpha)$ pentru o mulţime de instanţe $\{x_1, \ldots, x_n\}$).

Arătaţi că dacă $\hat{\alpha}$ este estimarea de verosimilitate maximă (MLE) corespunzătoare lui $L(\alpha)$, atunci $\hat{\beta} = T(\hat{\alpha})$ maximizează $M(\beta)$. (*Indicaţie*: Când anume va conţine mulţimea $\Theta(\beta)$ elementul $\hat{\alpha}$?)

Vom numi $\hat{\beta}$ *estimarea de verosimilitate maximă indusă*. Observaţi faptul că aceasta nu este identică cu *estimarea de verosimilitate maximă*, fiindcă $M(\beta)$ nu este în mod necesar totuna cu funcţia de verosimilitate corespunzătoare noii parametrizări. Ce se întâmplă oare atunci când T este o bijecţie? Arătaţi că în acest caz $M(\beta)$ coincide cu funcţia de verosimilitate corespunzătoare reparametrizării.

0.2.5 Elemente de teoria informaţiei

140. **(Calculul entropiei unei variabile aleatoare discrete)**

□ • ○ ***CMU, (?) 15-781, midterm example questions, pr. 1.b***

Cât este entropia următoarei distribuţii de probabilitate (discretă): [0.0625, 0.0625, 0.125, 0.25, 0.5]**?**

141. **(Câştigul de informaţie: câteva proprietăţi şi o exemplificare)**

prelucrare de Liviu Ciortuz, după
□ · ***CMU, 2011 fall, T. Mitchell, A. Singh, HW1, pr. 2***
CMU, 2012 spring, Roni Rosenfeld, HW2, pr. 9

La problema 55 am definit câştigul de informaţie astfel:

$$IG(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X).$$

De asemenea, am arătat că entropia condiţonală medie a unei variabile aleatoare X în raport cu o altă variabilă aleatoare Y se poate calcula cu formula

$$H(X|Y) = -\sum_{x\in Val(X)}\sum_{y\in Val(Y)} P(X=x,Y=y)\log_2 P(X=x|Y=y).$$

Următoarea *demonstraţie* ne arată că putem calcula câştigul de informaţie şi în alt mod:

$$\begin{aligned}
IG(X;Y) &= H(X) - H(X|Y) && (142)\\
&= -\sum_{x\in Val(X)} P(X=x)\log_2 P(X=x)\\
&\quad -\left(-\sum_{x\in Val(X)}\sum_{y\in Val(Y)} P(X=x,Y=y)\log_2 P(X=x|Y=y)\right) && (143)\\
&= -\sum_{x\in Val(X)}\sum_{y\in Val(Y)} P(X=x,Y=y)\log_2 P(X=x)\\
&\quad +\sum_{x\in Val(X)}\sum_{y\in Val(Y)} P(X=x,Y=y)\log_2 P(X=x|Y=y) && (144)\\
&= -\sum_{x\in Val(X)}\sum_{y\in Val(Y)} P(X=x,Y=y)(\log_2 P(X=x) - \log_2 P(X=x|Y=y)) && (145)\\
&= -\sum_{x\in Val(X)}\sum_{y\in Val(Y)} P(X=x,Y=y)\log_2 \frac{P(X=x)}{P(X=x|Y=y)} && (146)\\
&= -\sum_{x\in Val(X)}\sum_{y\in Val(Y)} P(X=x,Y=y)\log_2 \frac{P(X=x)P(Y=y)}{P(X=x,Y=y)}. && (147)
\end{aligned}$$

a. Justificaţi *de ce* anume au loc fiecare dintre egalităţile care intervin in demonstraţia de mai sus.

b. Definiţi independenţa a două variabile aleatoare X şi Y.
Apoi, folosind rezultatul demonstrat mai sus, şi anume

$$IG(X;Y) = \sum_{x \in Val(X)} \sum_{y \in Val(Y)} P(X=x, Y=y) \log_2 \frac{P(X=x, Y=y)}{P(X=x)P(Y=y)},$$

arătaţi că dacă X şi Y sunt independente, atunci $IG(X;Y) = 0$.

c. Demonstraţi că $IG(X;X) = H(X)$.

d. Fie X şi Y două variabile aleatoare independente, care iau valorile 0 şi 1 cu probabilităţi egale, adică $P(X=0) = P(X=1) = P(Y=0) = P(Y=1) = 1/2$. Considerăm încă o variabilă aleatoare Z, tot cu valori în mulţimea $\{0,1\}$. Distribuţia de probabilitate a variabilei condiţionate $Z|X,Y$ este definită în tabelul următor:

X	Y	$P(Z=0\|X,Y)$	$P(Z=1\|X,Y)$
0	0	0.8	0.2
0	1	0.2	0.8
1	0	0.2	0.8
1	1	0.8	0.2

i. **Folosind regula de înmulţire, precum şi independenţa variabilelor X şi Y, completaţi în tabelul următor distribuţia de probabilitate comună $P(X,Y,Z)$.**

X	Y	Z	$P(X,Y,Z)$
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

ii. **Calculaţi apoi distribuţiile marginale $P(X,Z)$, $P(Y,Z)$ şi $P(Z)$.**

X	Z	$P(X,Z)$
0	0	
0	1	
1	0	
1	1	

Y	Z	$P(Y,Z)$
0	0	
0	1	
1	0	
1	1	

Z	$P(Z)$
0	
1	

iii. **În final, arătaţi că $IG(X;Z) = IG(Y;Z) = 0$. (*Sugestie*: Folosiţi rezultatul de la punctul *b*.)**

142. **(Probabilităţi marginale, entropii, entropii condiţionale medii, câştiguri de informaţie)**

prelucrare de Liviu Ciortuz, după
• ○ CMU, 2011 spring, Roni Rosenfeld, HW2, pr. 1.d

Echipa de fotbal american The Steelers (din Pittsburgh) va juca în cupa Superbowl *XLV* contra echipei The Green Bay Packers. Pregătindu-şi meciul, ei (fotbaliştii echipei The Steelers) se gândesc să-şi definească strategia de joc în funcţie de doi factori majori:

- **dacă jucătorul Ben Roethlisberger va fi (sau nu) accidentat la vremea meciului (*Injured* = *yes* / *no*), şi**
- **cum anume va fi vremea (*Weather* = *foggy* / *rainy* / *clear sky*).**

Iată distribuţia comună a acestor două tipuri de evenimente:

	Weather = *foggy*	*rainy*	*clear sky*	$P(Injured)$
Injured = *no*	0.1	0.25	0.35	
Injured = *yes*	0.05	0.1	0.15	
$P(Weather)$				

a. Pornind de la distribuţia comună dată, completaţi ultima linie şi ultima coloană din tabelul de mai sus cu valorile corespunzătoare distribuţiilor marginale $P(Weather)$ şi $P(Injured)$.

b. Calculaţi entropiile $H(Weather)$ şi $H(Injured)$.

c. Calculaţi entropiile condiţionale medii $H(Injured|Weather)$ şi $H(Weather|Injured)$.

d. Calculaţi în entropia comună $H(Injured, Weather)$ folosind definiţia (vedeţi problema 55). Verificaţi apoi că

$$\begin{aligned} H(Injured, Weather) &= H(Injured) + H(Weather|Injured) \\ &= H(Weather) + H(Injured|Weather). \end{aligned}$$

***Indicaţie*: Vedeţi problema 55.c.**

e. Calculaţi câştigurile de informaţie $IG(Injured, Weather)$ şi $IG(Weather, Injured)$ folosind definiţia (vedeţi problema 55). Verificaţi apoi că

$$IG(Injured, Weather) = KL(P_{Injured,\ Weather} \,||\, (P_{Injured} \cdot P_{Weather})).$$

***Indicaţie*: Vedeţi problema 141.a şi/sau problema 65.b.**

***Sugestie*: Veţi putea folosi următoarele aproximări: $\log_2 3 = 1.585$, $\log_2 5 = 2.322$, $\log_2 7 = 2.807$, $\log_2 11 = 3.459$ şi $\log_2 13 = 3.700$.**

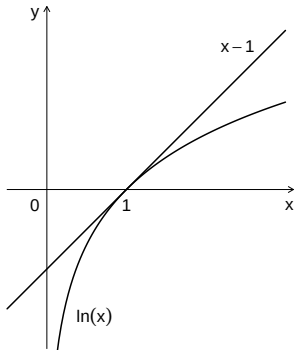
143. **(O margine superioară pentru valorea entropiei unei variabile aleatoare discrete)**

■ □ ∗ ***CMU, 2003 fall, T. Mitchell, A. Moore, HW1, pr. 1.1***

***Comentariu*:** La problema 55.a am demonstrat că entropia oricărei variabile aleatoare discrete este nenegativă ($H(X) \geq 0$).[266] La acest exerciţiu veţi demonstra — tot pentru cazul discret — că există şi o margine superioară pentru $H(X)$.

Aşadar, fie X o variabilă aleatoare discretă care ia n valori şi urmează distribuţia probabilistă P. Conform definiţiei, entropia lui X este

$$H(X) = -\sum_{i=1}^{n} P(X = x_i) \log_2 P(X = x_i).$$



Arătaţi că $H(X) \leq \log_2 n$.[267]

***Sugestie* (1): Puteţi folosi inegalitatea $\ln x \leq x - 1$, care are loc pentru orice $x > 0$.**

***Sugestie* (2): Puteţi folosi inegalitatea lui Jensen (vedeţi problema 79.a).**

***Sugestie* (3): Puteţi folosi metoda multiplicatorilor lui Lagrange.**

144. **(Entropia comună: forma particulară a relaţiei de „înlănţuire“ în cazul variabilelor aleatoare independente)**

■ □ • ○ ∗ ***prelucrare de Liviu Ciortuz, după CMU, 2012 spring, Roni Rosenfeld, HW2, pr. 7.b***

La problema 59 s-a demonstrat că dacă X şi Y sunt variabile aleatoare independente, atunci are loc egalitatea $H(X,Y) = H(X) + H(Y)$ şi reciproc. În demonstraţie s-au folosit rezultate de la problema 146 (sau, echivalent / alternativ, problema 65).

Implicaţia directă din echivalenţa de mai sus, şi anume X şi Y independente $\Rightarrow H(X,Y) = H(X)+H(Y)$ (respectiv X şi Y independente $\Rightarrow H(X) = H(X \mid Y)$ şi $H(Y) = H(Y \mid X)$) se poate obţine însă şi în mod direct, pornind de la definiţia independenţei variabilelor aleatoare. Vă cerem să faceţi astfel demonstraţia acestei implicaţii. Veţi trata mai întâi cazul variabilelor aleatoare discrete şi apoi cazul variabilelor aleatoare continue.

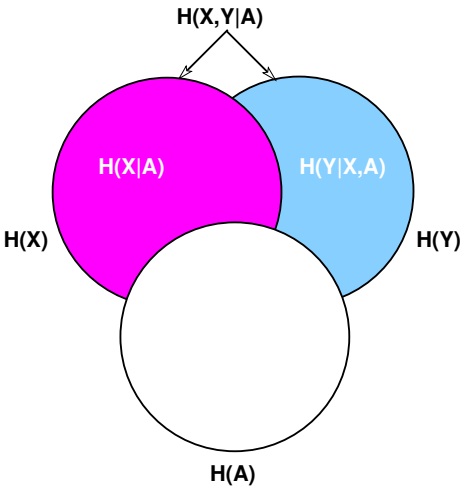
***Observaţie*: Conform problemei 55.c, unde am demonstrat că $H(X,Y) = H(X) + H(Y \mid X) = H(Y) + H(X \mid Y)$ (indiferent dacă X şi Y sunt sau nu independente), rezultă că egalitatea $H(X,Y) = H(X) + H(Y)$ este echivalentă cu egalităţile $H(X) = H(X \mid Y)$ şi $H(Y) = H(Y \mid X)$ (şi, de asemenea, cu egalitatea $IG(X;Y) = 0$).**

[266] Extensia acestei proprietăţi la cazul variabilelor aleatoare continue este imediată.

[267] *Observaţie*: Se poate demonstra că egalitatea $H(X) = \log_2 n$ are loc atunci şi numai atunci când distribuţia variabilei X este uniformă, adică $P(X = x_i) = \frac{1}{n}$ pentru $i = 1, \ldots, n$, unde $n = |Val(X)|$.

145. **(Entropie comună şi condiţională: formula de „înlănţuire“ condiţională)**

prelucrare de Liviu Ciortuz, după
■ □ • ○ ∗ *CMU, 2012 spring, Roni Rosenfeld, HW2, pr. 4.b*



a. Demonstraţi că proprietatea

$$\begin{aligned} H(X,Y|A) &= H(Y|A) + H(X|Y,A) \\ &= H(X|A) + H(Y|X,A) \end{aligned}$$

este adevărată pentru oricare 3 variabile aleatoare discrete X, Y şi A. Explicaţi în mod intuitiv, într-o singură frază, care este semnificaţia proprietăţii de mai sus.

b. Arătaţi că varianta generalizată a *regulii de înlănţuire* pentru entropii (care a fost demonstrată deja la problema 55.c)

$$H(X_1, \ldots, X_n) = H(X_1) + H(X_2|X_1) + \ldots + H(X_n|X_1, \ldots, X_{n-1})$$

poate fi dedusă [în manieră alternativă] pornind de la relaţia $H(X,Y) = H(X) + H(Y \mid X)$ — varianta de bază a regulii de înlănţuire pentru entropii, care a fost demonstrată tot la problema 55.c — şi ţinând cont de proprietatea care a fost demonstrată la punctul a.

c. Demostraţi varianta generalizată a formulei de „înlănţuire“ condiţională:

$$H(X_1, \ldots, X_n|A) = H(X_1|A) + H(X_2|X_1, A) + \ldots + H(X_n|X_1, \ldots, X_{n-1}, A).$$

146. **(Nenegativitatea câştigului de informaţie; o condiţie necesară şi suficientă pentru anularea lui)**

■ □ ∗ *Liviu Ciortuz, 2017*

Definiţia *câştigului de informaţie* (sau: a *informaţiei mutuale*) al unei variabile aleatoare X în raport cu o altă variabilă aleatoare Y este $IG(X,Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X)$.[268] La problema 65 s-a demonstrat — pentru cazul în care X şi Y sunt discrete — că $IG(X,Y) = KL(P_{X,Y}||P_X\, P_Y)$, unde KL desemnează *entropia relativă* (sau: *divergenţa Kullback-Leibler*), P_X şi P_Y sunt distribuţiile variabilelor X şi, respectiv, Y, iar $P_{X,Y}$ este distribuţia comună a acestor variabile. Tot la problema 65 s-a arătat că divergenţa KL este întotdeauna nenegativă. În consecinţă, $IG(X,Y) \geq 0$ pentru orice X şi Y.

a. Aici vă cerem să demonstraţi inegalitatea $IG(X,Y) \geq 0$ în manieră directă, plecând de la prima definiţie dată mai sus, fără a mai apela la divergenţa Kullback-Leibler.

[268] Vedeţi problema 55.

b. Arătaţi tot într-o manieră directă că $IG(X, Y) = 0$ dacă şi numai dacă X şi Y sunt independente. (Într-o manieră indirectă, acest rezultat a fost demonstrat la problema 65.c.)

***Sugestie*: Puteţi folosi următoarea formă [particulară] a inegalităţii lui Jensen:**[269]

$$\sum_{i=1}^{n} a_i \log x_i \leq \log \left(\sum_{i=1}^{n} a_i x_i \right)$$

unde baza logaritmului se consideră supraunitară, $a_i \geq 0$ pentru $i = 1, \ldots, n$ şi $\sum_{i=1}^{n} a_i = 1$.[270] **Egalitatea are loc dacă şi numai dacă $x_1 = x_2 = \ldots = x_n$.**

147. **(Cross-entropia — o aplicaţie: selecţia modelelor probabiliste)**

□ • ○ ***CMU, 2012 spring, Roni Rosenfeld, HW2, pr. 10***

Avem un zar măsluit (engl., unfair die). Probabilităţile [reale] de apariţie pentru fiecare dintre feţele de la 1 la 6 sunt date de distribuţia

$$P_{true} = (0.08, 0.55, 0.15, 0.12, 0.05, 0.05).$$

Fără să cunoască acest fapt, două persoane, identificate cu A şi respectiv B, ne sugerează următoarele *modele* de probabilitate pentru zarul măsluit:

$$P_A = (0.07, 0.14, 0.24, 0.24, 0.05, 0.26)$$
$$P_B = (0.25, 0.13, 0.21, 0.03, 0.11, 0.27)$$

a. Elaboraţi câteva idei relativ la cum am putea măsura / determina care dintre aceste două modele este mai bun.

b. Calculaţi cross-entropiile $CH(P_{true}, P_A)$, $CH(P_{true}, P_B)$ şi $CH(P_{true}, P_{true})$. Presupunând că alegem ca măsură / mijloc de evaluare a modelelor cross-entropia, care dintre cele două modele (P_A şi P_B) credeţi că este mai bun?

148. **(Varianta condiţională a divergenţei KL; regula de înlănţuire pentru divergenţa KL; divergenţa condiţională KL pentru un vector de variabile mutual independente în raport cu o altă variabilă)**

prelucrare de Liviu Ciortuz, după

■ □ • ○ ***Stanford, 2015 fall, Andrew Ng, HW3, pr. 5.b***

Vă readucem aminte — vedeţi problema 65 — că entropia relativă sau divergenţa Kullback-Leibler (KL) dintre două distribuţii discrete $P(X)$ şi $Q(X)$ este definită astfel:[271]

$$KL(P||Q) = \sum_{x} P(x) \log \frac{P(x)}{Q(x)}.$$

[269] Vedeţi problema 79.

[270] Avantajul la această problemă, comparativ cu problema 65.a, este că aici se lucrează cu o singură distribuţie (p), nu cu două distribuţii (p şi q). Totuşi, demonstraţia de aici va fi mai laborioasă.

[271] Atunci când P şi Q sunt p.d.f.-uri (funcţii de densitate de probabilitate) pentru variabile aleatoare continue, dacă în formula de mai sus înlocuim simbolul de sumare cu integrala, proprietatea enunţată în acest exerciţiu rămâne valabilă. Totuşi, din motive de simplitate, aici vom lucra doar cu funcţii masă de probabilitate / distribuţii discrete.

Pentru convenienţă, vom presupune că $P(x) > 0$ şi $Q(x) > 0$ pentru orice x. Uneori în cele ce urmează vom scrie $KL(P||Q)$ sub forma $KL(P(X)||Q(X))$.

Entropia relativă (sau, divergenţa KL) dintre două distribuţii condiţionale $P(X|Y)$ şi $Q(X|Y)$ se defineşte în felul următor:[272]

$$KL(P(X|Y)\,||\,Q(X|Y)) = \sum_y P(y)\Big(\sum_x P(x|y)\log\frac{P(x|y)}{Q(x|y)}\Big).$$

Aceasta poate fi văzută ca fiind media [probabilistă a] divergenţei KL dintre distribuţiile condiţionale corespunzătoare pentru x (adică, dintre $P(X|Y=y)$ şi $Q(X|Y=y)$), media fiind calculată în raport cu valorile y ale variabilei aleatoare Y.

a. Demonstraţi că are loc următoarea proprietate (de tip „regulă de înlănţuire“) pentru divergenţa KL:

$$\begin{aligned} KL(P(X,Y)\,||\,Q(X,Y)) &= KL(P(X)\,||\,Q(X)) + KL(P(Y|X)\,||\,Q(Y|X)) \\ &= KL(P(Y)\,||\,Q(Y)) + KL(P(X|Y)\,||Q(X|Y)). \end{aligned} \quad (148)$$

b. Demonstraţi că dacă $X = (X_1, \ldots, X_n)$ şi pentru orice $i \neq j$ avem că variabila X_i este independentă condiţional de X_j în raport cu variabila Y, atunci

$$KL(P(X|Y)\,||\,Q(X|Y)) = \sum_{i=1}^{n} KL(P(X_i|Y)\,||\,Q(X_i|Y)). \quad (149)$$

149. **(Echivalenţa dintre maximizarea verosmilităţii unui set de date şi minimizarea divergenţei KL faţă de distribuţia uniformă)**

■ □ • ○ ***Stanford, 2015 fall, Andrew Ng, HW3, pr. 5.c***

Presupunem că ni se dă o problemă de estimare a parametrilor unei distribuţii probabiliste şi că dispunem de setul de date „de antrenament“ $D = \{x^{(i)}; i = 1, \ldots, m\}$. Considerăm distribuţia empirică $\hat{P}(x) = \frac{1}{m}$, adică distribuţia uniformă peste setul de date D. Aşadar, a face eşantionare (engl., sampling) cu distribuţia empirică înseamnă a alege în mod aleatoriu un exemplu din setul de date D.

Presupunem că avem o anumită familie de distribuţii P_θ, indexată după parametrul θ. (Dacă doriţi, puteţi considera $P_\theta(x)$ ca notaţie alternativă pentru $P(x;\theta)$.)

Demonstraţi că a găsi valoarea de verosimilitate maximă (engl., maximum likelihood estimate) pentru parametrul θ este echivalent cu a găsi acea distribuţie P_θ pentru care entropia relativă (divergenţa KL) faţă de distribuţia empirică $\hat{P}$ este minimă. Aşadar, vă cerem să demonstraţi relaţia următoare:

$$\arg\min_\theta KL(\hat{P}||P_\theta) = \operatorname*{argmax}_\theta \sum_{i=1}^{m} \ln P_\theta(x^{(i)}).$$

***Comentariu*:** Ne vom referi acum la relaţia dintre proprietatea enunţată în acest exerciţiu pe de o parte şi estimarea parametrilor făcută de algoritmul de clasificare binară

[272] Această noţiune se numeşte *divergenţa condiţională KL*.

Bayes Naiv cu atribute de tip Bernoulli de cealaltă parte (vedeţi capitolul de *Clasificare bayesiană*). În modelul de clasificare Bayes Naiv se presupune că P_θ este de forma următoare: $P_\theta(x, y) = p(y) \prod_{i=1}^{n} p(x_i|y)$. Conform regulii de înlănţuire pentru divergenţa KL (vedeţi relaţiile (148) şi (149) de la ex. 148), avem:

$$KL(\hat{P}||P_\theta) = KL(\hat{P}(y)||p(y)) + \sum_{i=1}^{n} KL(\hat{P}(x_i|y)||p(x_i|y)).$$

Această relaţie arată că problema găsirii maximului verosimilităţii / minimumului divergenţei KL (pentru a estima valorile parametrilor) se descompune în $2n + 1$ probleme de optimizare independente: una pentru distribuţia a priori a clasei, $p(y)$, şi câte una pentru fiecare distribuţie condiţională $p(x_i|y)$, corespunzătoare feacărei trăsături x_i în raport cu fiecare din cele două valori posibile pentru eticheta y. Concret, găsirea estimării de verosimilitate maximă pentru fiecare dintre aceste probleme în mod individual rezultă în maximizarea verosimilităţii distribuţiei comune. (O observaţie similară se poate formula şi în legătură cu estimarea parametrilor reţelelor bayesiene (engl., Bayesian networks.))

***Observaţie*:** Pentru clasificatorul Bayes Naiv cu variabile booleene, legătura dintre maximizarea verosimilităţii datelor de antrenament şi estimarea parametrilor în sens MLE este arătată la problema 38.ab de la capitolul *Clasificare bayesiană*.

150. **(Determinarea distribuţiilor unidimensionale continue care — în anumite condiţii — au entropii maxime, folosind metoda dualităţii Lagrange)**

prelucrare de Andi Munteanu şi Liviu Ciortuz, 2019, după
□ • ∗ ***CMU, 2013 spring, A. Smola, B. Poczos, HW2, pr. 1.bc***

a. Demonstraţi că dintre toate distribuţiile continue care sunt definite pe un interval $[a, b] \subset \mathbb{R}$, cu a şi b fixaţi ($a < b$), cea mai mare entropie o are distribuţia continuă uniformă.[273]

b. Demonstraţi că dintre toate distribuţiile continue care sunt definite pe intervalul $[0, +\infty)$, pentru care media este egală cu μ (un anumit număr real, pozitiv, fixat), cea mai mare entropie o are distribuţia exponenţială.

c. Demonstraţi că dintre toate distribuţiile continue care sunt definite pe intervalul $\mathbb{R}$, pentru care media este este egală cu μ (un anumit număr real, fixat) şi varianţa este egală cu σ^2 (un anumit număr real, pozitiv, fixat), cea mai mare entropie o are distribuţia gaussiană.

***Sugestii*:**

1. Folosiţi metoda multiplicatorilor lui Lagrange.

2. Puteţi folosi următoarele *proprietăţi* ale *derivatelor funcţionale*:[274]

[273] Vă reamintim că la problema 143 se demonstrează că, dată fiind o variabilă aleatoare $X : \Omega \to \mathbb{R}$ cu funcţia de probabilitate P şi $Val(X) \overset{not.}{=} \{x_1, \dots x_n\}$, entropia variabilei X este maximă dacă $P(X = x_1) = \dots P(X = x_n) = 1/n$. Acest rezultat este varianta discretă a proprietăţii care a fost enunţată la punctul a al problemei de faţă.

[274] Pentru definiţia noţiunii de derivată funcţională şi pentru un exemplu de calculare a unei astfel de derivate funcţionale, vedeţi problema 66.

În formulele care vor urma, F şi G sunt *funcţionale* (adică, funcţii definite peste mulţimi de funcţii care au anumite proprietăţi, pe care însă nu le menţionăm aici), ρ este o funcţie de argument real x, iar g este o funcţie care se aplică lui [sau, echivalent, se compune cu] ρ.

- ***liniaritatea*: $\frac{\delta\left(\lambda F+\mu G\right)(\rho)}{\delta\rho(x)}=\lambda\frac{\delta F(\rho)}{\delta\rho(x)}+\mu\frac{\delta G(\rho)}{\delta\rho(x)}$, unde λ şi μ sunt constante reale;**
- ***regula produsului*: $\frac{\delta(FG)(\rho)}{\delta\rho(x)}=\frac{\delta F(\rho)}{\delta\rho(x)}\,G(\rho)+F(\rho)\,\frac{\delta G(\rho)}{\delta\rho(x)}$;**
- ***regula „înlănţuirii" la derivarea compunerii unei funcţionale în raport cu o funcţie derivabilă*: $\frac{\delta F(g(\rho))}{\delta\rho(x)}=\frac{\delta F(g(\rho))}{\delta g(\rho(x))}\cdot\frac{dg(\rho)}{d\rho(x)}$;**
- $\frac{\delta}{\delta\rho}\int_a^b F(\rho)dx=\frac{\delta F(\rho)}{\delta\rho}$.

151. **(Proprietăţi ale entropiei: Adevărat sau Fals?)**

□ ∗ ***CMU, 2011 spring, Roni Rosenfeld, HW2, pr. 2.a.1***
CMU, 2008 fall, Eric Xing, final exam, pr. 1.4
CMU, 2012 spring, Roni Rosenfeld, HW2, pr. 7

Stabiliţi dacă următoarele propoziţii sunt adevărate sau false.

a. Entropia nu este negativă.

b. $H(X,Y)\geq H(X)+H(Y)$ pentru orice două variabile aleatoare X şi Y.

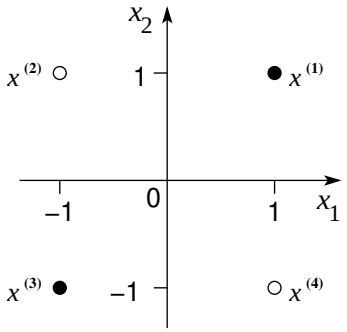
c. Dacă X şi Y sunt variabile aleatoare independente, atunci $H(X,Y)=H(X)+H(Y)$.

0.2.6 Funcţii-nucleu

152. **(Conceptul ¬XOR: neseparabilitate vs. separabilitate liniară folosind o funcţie de „mapare“ particulară)**

prelucrare de Liviu Ciortuz, după
□ • ○ ***CMU, 2020 fall, Pat Virtue, recitation 10, pr. 3***

Conceptul ¬XOR corespunde unei probleme de clasificare neliniară; el poate fi reprezentat grafic ca în figura alăturată.
Vom considera funcţia de „mapare“ (engl., feature transformation) ϕ definită prin relaţia $\phi((x_1, x_2)^\top) \stackrel{def.}{=} (x_1, x_1x_2)^\top$, unde x_1, $x_2 \in \mathbb{R}$.



a. Calculaţi expresia funcţiei-nucleu $K(x, z)$ care corespunde „mapării“ ϕ.

b. Într-un nou sistem de coordonate, corespunzător spaţiului de „trăsături“ determinat de „maparea“ ϕ, reprezentaţi grafic „imaginile“ instanţelor din setul / conceptul ¬XOR prin funcţia ϕ.

c. Este oare setul de date corespunzător conceptului ¬XOR liniar separabil în spaţiul de „trăsături“?
Dacă da, indicaţi care este *separatorul optimal* (adică, separatorul cu *margine geometrică maximă*) în spaţiul de „trăsături“.
Apoi scrieţi ecuaţia *graniţei de decizie* (engl., decision boundary) din spaţiul original care corespunde separatorului optimal pe care l-aţi identificat în spaţiul de „trăsături“.
Desenaţi această graniţă de decizie pe figura de mai sus. Haşuraţi *zonele de decizie* corespunzătoare instanţelor pozitive.

153. **(Găsirea mapării care corespunde unei funcţii-nucleu polinomiale particulare)**

○ ***CMU, 2010 fall, Aarti Singh, HW3, pr. 3.3.c***

Se consideră funcţia $K(u, v) = u \cdot v + 4(u \cdot v)^2$, unde u şi v sunt vectori din $\mathbb{R}^2$. Arătaţi că există o funcţie ϕ astfel încât $K(u, v) = \phi(u) \cdot \phi(v)$.

***Indicaţie*: Veţi determina efectiv $\phi(x)$, expresia funcţiei ϕ pentru argumentul $x = (x_1, x_2) \in \mathbb{R}^2$.**

154. **(Funcţii-nucleu: exemplificarea unor chestiuni de bază)**

• ***CMU, 2016 fall, N. Balcan, M. Gormley, HW4, pr. 2.1***

Presupunem că lucrăm cu instanţe de forma $\mathbf{x} = (x_1, x_2, x_3)$ din $\mathbb{R}^3$. Definim funcţia de „mapare“ a trăsăturilor $\phi(\mathbf{x}) = (x_1^2, x_2^2, x_3^2, \sqrt{2}x_1x_2, \sqrt{2}x_1x_3, \sqrt{2}x_2x_3)$.

a. Deduceţi expresia funcţiei-nucleu care corespunde acestei mapări, $K(\mathbf{x}, \mathbf{z})$. Exprimaţi răspunsul dumneavoastră mai întâi în funcţie de $x_1, x_2, x_3, z_1, z_2, z_3$, iar apoi aduceţi expresia lui $K(\mathbf{x}, \mathbf{z})$ la forma cea mai simplă.

b. Presupunem că vrem să calculăm valoarea funcţiei-nucleu $K(\mathbf{x}, \mathbf{z})$ pentru două instanţe oarecare $\mathbf{x}, \mathbf{z} \in \mathbb{R}^3$. Câte adunări şi câte înmulţiri sunt necesare dacă

***i.* „mapăm" vectorii de intrare (x şi z) în spaţiul de trăsături şi apoi aplicăm produsul scalar?**

***ii.* folosim expresia funcţiei-nucleu K care a fost obţinută la punctul a?**

155. **(Normalizarea funcţiilor-nucleu)**

prelucrare de Liviu Ciortuz, după

∗ ***University of Helsinki, 2005 spring, Jyrki Kivinen, HW9, pr. 1.b***

Fie $K : \mathbb{R}^{d \times d} \to \mathbb{R}$ o funcţie-nucleu, iar ϕ „maparea" corespunzătoare. Definim funcţia $\bar{K}$ printr-o procedură de „*normalizare*":[275]

$$\bar{K}(x, z) = \frac{K(x, z)}{\sqrt{K(x, x) K(z, z)}}.$$

Arătaţi că funcţia astfel obţinută este de asemenea funcţie-nucleu. Precizaţi care este expresia funcţiei de „mapare" (notată cu $\bar{\phi}$) corespunzătoare lui $\bar{K}$.

Ce puteţi spune despre valoarea numerică a lui $\|\bar{\phi}(x)\|$, pentru x arbitrar din $\mathbb{R}^d$? Ca o consecinţă, ce interpretare geometrică („vizualizare") puteţi să-i asociaţi lui $\bar{\phi}(x)$ în raport cu $\phi(x)$?

Credeţi că are vreo importanţă practică operaţia de „normalizare" a instanţelor de antrenament în învăţarea automată (în general, nu doar în contextul funcţiilor-nucleu)?

***Sugestie*: Pentru un exemplu de folosire a unei funcţii-nucleu normalizate pentru rezolvarea unei probleme de clasificare, vedeţi exerciţiul 41 de la capitolul *Maşini cu vectori-suport*.**

156. **(Funcţii-nucleu obţinute prin aplicarea proprietăţilor de „construcţie": câteva exemple)**

• · ***CMU, 2015 spring, T. Mitchell, N. Balcan, HW6, pr. 4.2***
CMU, 2017 fall, Nina Balcan, HW3, pr. 1.2.a
Stanford, 2020 summer, Andrew Ng, HW2, pr. 3.g

La problemele 68.c, 69 şi 70 am arătat că putem folosi funcţii-nucleu care au fost deja definite, ca să construim funcţii-nucleu noi.

Argumentaţi de ce funcţiile propuse mai jos sunt sau nu sunt funcţii-nucleu valide. Veţi presupune că $x, z \in \mathbb{R}^d$, cu $d \in \mathbb{N}^*$.

a. $K(x, z) = 5(x \cdot z)$.

[275] În cele ce urmează vom lucra doar în condiţiile în care $\phi(x) \neq 0$ şi $\phi(z) \neq 0$.

b. $K(x,z) = (x \cdot z)^3 + (x \cdot z + 1)^2$.

c. $K(x,z) = (x \cdot z)^2 + \exp(-\|x\|^2)\exp(-\|z\|^2)$.

d. $K(x,z) = ||x||^2\ ||z||^2\ e^{(||x||^2+||z||^2)}$.

e. funcţia $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ definită prin relaţia $K(x,z) \stackrel{def.}{=} K'(\psi(x), \psi(z))$, ştiind că $\psi : \mathbb{R}^d \to \mathbb{R}^p$, $p \in \mathbb{N}^*$, iar funcţia $K' : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ este funcţie-nucleu.

157. **(Funcţii-nucleu: aplicarea definiţiei, şi / sau a proprietăţilor de tip „construcţie“)**

• ***MIT / NYU, 2016 spring, David Sontag, HW3, pr. 1***
CMU, 2017 fall, Nina Balcan, HW3, pr. 1.2.b

a. Pentru oricare două documente x şi z, definim $K(x,z)$ ca fiind numărul de cuvinte de dicţionar care apar atât în documentul x cât şi în documentul z (deci cardinalul intersecţiei mulţimilor de cuvinte din cele două documente). Este oare această funcţie un nucleu? Justificaţi în mod riguros răspunsul dumneavoastră.

b. Presupunând că $x = (x_1, x_2)^\top$, $z = (z_1, z_2)^\top$ (adică, atât x cât şi z sunt vectori-coloană cu câte două componente), iar $\beta > 0$ este o constantă reală, demonstraţi că funcţia $K_\beta : \mathbb{R}^2 \to \mathbb{R}$ definită prin relaţia

$$K_\beta(x,z) = (1 + \beta x \cdot z)^2 - 1$$

este funcţie-nucleu. Veţi face demonstraţia arătând că există o „mapare“ $\phi(x)$ astfel încât $K_\beta(x,z) = \phi(x) \cdot \phi(z)$ pentru orice x şi $z \in \mathbb{R}^2$.

c. Folosind regulile de „construcţie“ de noi nuclee care au fost prezentate în diverse exerciţii, precum şi faptul că funcţia $K(x,z) = x \cdot z$ este nucleu (aşa-numitul *nucleu liniar*), demonstraţi că următoarea funcţie este şi ea un nucleu:

$$\left(1 + \frac{x}{\|x\|_2} \cdot \frac{z}{\|z\|_2}\right)^3.$$

d. Demonstraţi că funcţia $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ definită prin relaţia $K(x,z) = sinh\,(e^{x \cdot z})$ este funcţie-nucleu.

***Sugestie*: Puteţi eventual să folosiţi faptul că funcţia $sinh\,(a)$, numită *sinus hiperbolic*, se poate scrie ca limita următoarei serii infinite:**

$$sinh\,(a) = \sum_{n=0}^{\infty} \frac{1}{(2n+1)!} a^{2n+1} \text{ pentru orice } a \in \mathbb{R}.$$

158. **(Funcţii-nucleu: exemple de operaţii [cu funcţii] care *nu* conduc la „construirea“ de noi funcţii-nucleu)**

□ • ∗ ***CMU, 2013 spring, A. Smola, B. Poczos, HW2, pr. 4.2***

a. Considerăm $K_1(x,x')$ şi $K_2(x,x')$ două funcţii-nucleu. Demonstraţi că diferenţa lor, $K_1 - K_2$, *nu* este în mod neapărat funcţie-nucleu.

***Sugestie*: Vă reamintim că pentru orice funcţie-nucleu $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ şi pentru orice set de instanţe $x_1, \ldots, x_m$ din $\mathbb{R}^d$, matricea-nucleu corespunzătoare (numită şi matrice Gram), este pozitiv semidefinită.**[276]

[276] Vedeţi problema 68.b.

b. Ştim că $\exp(-\|x-y\|^2)$ este funcţie-nucleu.[277] Demonstraţi că $\exp(\|x-y\|^2)$ *nu* este funcţie-nucleu validă.

***Sugestie*: Construiţi o matrice Gram care nu este pozitiv semidefinită.**

159. **(Funcţii-nucleu: o inegalitate / o margine superioară pentru valoarea absolută a unei funcţii-nucleu oarecare)**

□ • ○ ***CMU, 2015 spring, Alex Smola, midterm, pr. 8.2***

Demonstraţi că pentru orice funcţie-nucleu are loc inegalitatea

$$K^2(x,x') \leq K(x,x)\,K(x',x').$$

***Sugestie*: Folosiţi inegalitatea Cauchy-Bunyakovsky-Schwarz: $|x \cdot y| \leq \|x\|\,\|y\|$, unde $\cdot$ desemnează produsul scalar al vectorilor, iar $\|\ \|$ reprezintă norma euclidiană.[278]**

***Consecinţă*: Din inegalitatea de mai sus, folosind notaţia de la problema 155, rezultă $|\bar{K}(x,x')| \leq 1$, ceea ce înseamnă că valorile funcţiior-nucleu normalizate, în modul, sunt întotdeuna mărginite superior de 1.**

160. **(O funcţie-nucleu particulară, care asigură separabilitate liniară [în spaţiul de „trăsături"] pentru orice set de date de antrenament)**

• · ***CMU, 2015 spring, T. Mitchell, N. Balcan, HW6, pr. 4.1***

Fie $X \overset{not.}{=} \{x_1,\ldots,x_m\} \subset \mathbb{R}^d$. Considerăm funcţia-nucleu $K : X \times X \to \mathbb{R}$ definită prin relaţia următoare:

$$K(x,x') = \begin{cases} 1, \text{ dacă } x = x' \\ 0, \text{ în caz contrar.} \end{cases}$$

a. Demonstraţi că aceasta este o funcţie-nucleu validă, folosind teorema lui Mercer. Apoi, găsiţi o funcţie de „mapare" a trăsăturilor $\phi : X \to \mathbb{R}^m$ astfel încât $K(x,x') = \phi(x) \cdot \phi(x')$ pentru orice $x, x' \in X$.

b. Demonstraţi că mulţimea $\phi(X) \overset{not.}{=} \{\phi(x) | x \in X\}$ — unde ϕ este maparea găsită la punctul a — este *liniar separabilă*, indiferent de modul cum s-ar face *etichetarea* instanţelor din mulţimea X (şi, corespunzător, din mulţimea $\phi(X)$).

c. Întrucât cu ajutorul acestei funcţii-nucleu obţinem [în noul spaţiul de trăsături] separabilitate liniară indiferent de cum a fost făcută etichetarea instanţelor date, suntem tentaţi să credem că funcţia aceasta ne poate servi în mod perfect pentru a învăţa orice concept-target. Este totuşi posibil ca în practică să se ajungă la concluzia că ideea aceasta nu este atât de bună precum pare la prima vedere?

[277] Vedeţi problema 74.

[278] Un exemplu simplu: $(ax+by)^2 \leq (a^2+b^2)(x^2+y^2)$. În teoria probabilităţilor, inegalitatea Cauchy-Bunyakovsky-Schwarz este cunoscută sub forma următoare: $(E[XY])^2 \leq E[X^2]\,E[Y^2]$.

161. **(O proprietate simplă a nucleului de tip RBF)**

CMU, 2011 spring, Tom Mitchell, HW6, pr. 1.1.b

Oricărei funcţii-nucleu îi este asociată în mod implicit o anumită funcţie („mapare“) ϕ care transformă instanţele de antrenament $x \in \mathbb{R}^d$ într-un spaţiu Q de dimensiune [de obicei] mult mai mare decât d, şi care satisface proprietatea următoare: $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$. În continuare vom lua drept nucleu funcţia cu baza radială (RBF)

$$K(x_i, x_j) = e^{-\frac{1}{2\sigma^2}||x_i - x_j||^2}$$

unde notaţia $||\ ||$ corespunde normei euclidiene.

Demonstraţi că $||\phi(x_i) - \phi(x_j)||^2 < 2$ unde ϕ este maparea asociată nucleului RBF definit mai sus.

***Sugestie*: Vă puteţi inspira din rezultatul / rezolvarea problemei 71.**

162. **(Crearea de trăsături noi, folosind funcţii cu baza radială)**

• ∗ ***MIT, 2018 spring, T. Jaakkola, R. Barzilay, midterm, pr. 4.a-f***

În acest exerciţiu vom pune în evidenţă cum anume, folosind funcţii cu baza radială, putem crea noi [spaţii de] trăsături asociate instanţelor de antrenament sau de test. Pentru a defini astfel trăsături noi, avem nevoie de o mulţime de *instanţe* $E = (e_1, \ldots, e_k)$,[279] unde fiecare e_i este un punct din $\mathbb{R}^d$, iar d este dimensiunea spaţiului [de trăsături] original. Funcţia pe care o vom folosi aici pentru crearea de trăsături este următoarea:

$$\phi(x) = \begin{bmatrix} \exp(-\beta\|e_1 - x\|^2) \\ \exp(-\beta\|e_2 - x\|^2) \\ \cdots \\ \exp(-\beta\|e_k - x\|^2) \end{bmatrix}.$$

unde $\beta > 0$ este o constantă reală.

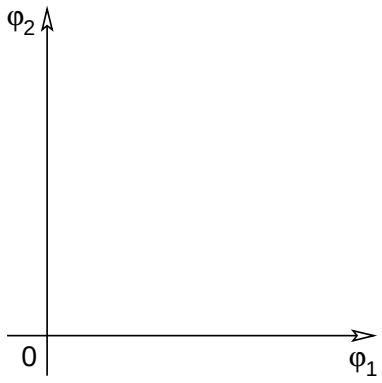
a. Cât este dimensiunea vectorului $\phi(x)$?

b. Considerăm următorul caz particular: $d = 1$, $k = 2$, $E = (1, 2)$, $\beta = 1$, iar setul de exemple de antrenament este format din instanţele etichetate $(0, +1), (1, -1), (2, +1)$.

Este acest set de exemple de antrenament liniar separabil în spaţiul original? Da sau Nu? (Justificaţi răspunsul.) În cazul afirmativ, veţi preciza ecuaţia unui separator liniar.

c. Pe sistemul de axe de coordonate alăturat, desenaţi punctele $\phi(0)$, $\phi(1)$ şi $\phi(2)$.[280] Etichetaţi-le folosind *convenţia* noastră obişnuită de notare: simbolul • desemnează instanţe pozitive, iar simbolul ∘ instanţe negative. Următorul tabel de valori vă poate fi de folos:

v	$\exp(-v)$
0	1.0
1	0.37
2	0.13
4	0.02
8	0.0003



Este setul de exemple de antrenament care a fost dat la punctul b liniar separabil în noul spaţiu de trăsături? (În acest nou spaţiu, instanţele etichetate sunt obţinute prin „maparea“ / transformarea $(x_i, y_i) \to (\phi(x_i), y_i)$.) Răspundeţi cu Da sau Nu şi justificaţi. În cazul afirmativ, veţi preciza ecuaţia unui separator liniar.

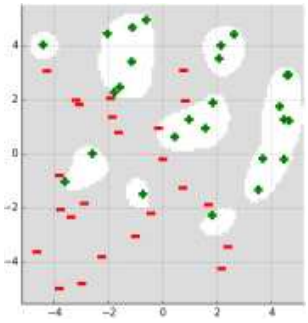
d. Pentru a alcătui mulţimea E, o modalitate des întâlnită constă în a folosi instanţe x_i din setul de date de antrenament.

Considerăm acum că se lucrează în spaţiul euclidian bidimensional. Cele trei imagini de mai jos ne prezintă separatorii obţinuţi pentru cazul în care se

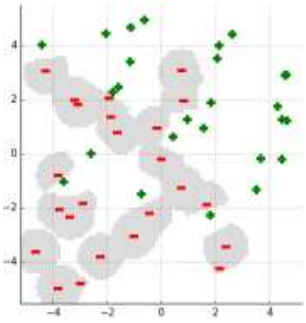
[279] Ordinea acestor instanţe contează, de aceea am folosit notaţia cu paranteze în loc de acolade.

[280] În imagine am notat cu ϕ_1 şi ϕ_2 cele două componente ale funcţiei ϕ, adică $\phi(x) = (\phi_1(x), \phi_2(x))$, unde $\phi_1(x) = \exp(-\beta\|e_1 - x\|^2)$ şi $\phi_2(x) = \exp(-\beta\|e_2 - x\|^2)$.

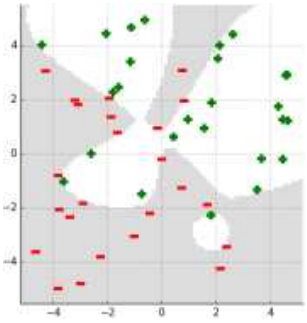
rulează algoritmul perceptron [în varianta Rosenblatt] kernelizat,[281] parametrul β luând valori în mulţimea $\{0.1, 1.0, 1000\}$. Zonele gri corespund regiunilor din spaţiu care, după antrenare, au fost clasificate ca fiind negative. Pentru fiecare dintre aceste 3 imagini indicaţi ce valoare a lui β (dintre cele 3 valori menţionate mai sus) corespunde respectivei imagini.



i.



ii.



iii.

163. **(Matrice pozitiv semidefinite: câteva proprietăţi)**

□ • ***CMU, 2013 spring, A. Smola, B. Poczos, HW2, pr. 3***

a. [Produsul pe componente[282] a două matrice pozitiv semidefinite]

Fie K_1 şi $K_2 \in \mathbb{R}^{n\times n}$ două matrice pozitiv semidefinite. Demonstraţi că *produsul lor pe componente*, care este prin definiţie matricea K cu proprietatea $K(i,j) = K_1(i,j)K_2(i,j)$ pentru orice $i, j \in \{1, \ldots, n\}$, este de asemenea pozitiv semidefinită.

***Sugestie*: Pornind de la doi vectori n-dimensionali $u = (u_1, \ldots, u_n)^\top \sim \mathcal{N}(0, K_1)$ şi $v = (v_1, \ldots, v_n)^\top \sim \mathcal{N}(0, K_2)$,[283] puteţi considera matricea de covarianţă a vectorului $w = (u_1v_1, \ldots, u_nv_n)^\top$. Vă readucem aminte că matricea de covarianţă a oricărui vector de variabile aleatoare este pozitiv semidefinită.[284]**

b. [Produsul a două matrice pozitiv semidefinite]

Fie A şi $B \in \mathbb{R}^{n\times n}$ două matrice pozitiv semidefinite. Arătaţi că

i. **matricea AB nu este în mod neapărat pozitiv semidefinită.**

ii. **matricea A^m este pozitiv semidefinită, pentru orice $m \in \mathbb{Z}_+$.**

***Sugestie*: Puteţi folosi un rezultat teoretic, cunoscut sub numele de *teorema de factorizare spectrală finit-dimensională*, care afirmă că orice matrice simetrică M [de numere reale] poate fi „diagonalizată“ cu ajutorul unei matrice ortogonale. Mai exact, aceasta înseamnă că pentru orice matrice simetrică $M \in \mathbb{R}^{n\times n}$ există o matrice ortogonală $U \in \mathbb{R}^{n\times n}$ cu proprietatea că $D = U^\top MU \in \mathbb{R}^{n\times n}$ este matrice diagonală. (Faptul că matricea U este ortogonală înseamnă că $U^\top U = I$, unde I este matricea identitate.)**

[281] Vedeţi problema 19 de la capitolul *Reţele neuronale artificiale*.
[282] Engl., elementwise product.
[283] Aşadar, vectorii u şi v urmează fiecare câte o distribuţie gaussiană multidimensională de medie 0 şi matrice de covarianţă K_1, respectiv K_2.
[284] Vedeţi problema 20.

164. **(Funcţii-nucleu: Adevărat sau Fals?)**

□ • ○ ***Stanford, 2009 fall, Andrew Ng, practice midterm, pr. 6.c***

Fie x_1, x_2 şi x_3 trei puncte oarecare din $\mathbb{R}^p$, cu $x_1 \neq x_2$, $x_1 \neq x_3$ şi $x_2 \neq x_3$. Considerăm de asemenea punctele z_1, z_2 şi z_3 din $\mathbb{R}^q$, arbitrar alese, dar fixate. În aceste condiţii putem defini o funcţie-nucleu $K : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}$ [LC: căreia îi va corespunde o anumită funcţie de „mapare“ ϕ, definită, desigur, pe $\mathbb{R}^p$] astfel încât pentru orice $i, j \in \{1, 2, 3\}$ să aibă loc egalitatea $K(x_i, x_j) = z_i \cdot z_j$. Adevărat sau fals?

0.2.7 Metode de optimizare în învăţarea automată

165. **(Calcularea derivatei (respectiv a gradientului) unor funcţii relativ simple. Funcţii convexe: câteva chestiuni simple, pornind de la definiţie)**

prelucrare de Liviu Ciortuz, după
□ • ○ ***CMU, 2019s, Nina Blacan, HW0, pr. Calculus***
CMU, 2009 fall, Carlos Guestrin, HW2, pr. 2.a
CMU, 2013 fall, A. Smola, G. Gordon, midterm practice questions, pr. 1.c

a. Dacă $y = x^3 + x - 5$, cât este derivata lui y în raport cu x?
Dacă $f(x_1, x_2) = x_1 (\sin x_2)\, e^{-x_1}$, cât este gradientul lui f (notaţie: $\nabla_f(x)$)?

b. Fie funcţia f cu valori reale, definită pe un interval $[a, b]$ (sau, mai general, pe o submulţime convexă a unui spaţiu vectorial). Conform definiţiei,[285] funcţia f este convexă dacă pentru orice două puncte x_1 şi x_2 din $[a, b]$ urmează că $f(tx_1 + (1-t)x_2) \leq tf(x_1) + (1-t)f(x_2)$, pentru orice $t \in [0, 1]$. Demonstraţi că funcţia $f(x) = |x|$, cu $x \in \mathbb{R}$, este convexă.

c. Dacă o funcţie f nu este dublu derivabilă, adică matricea sa hessiană nu este definită, atunci funcţia f nu poate fi convexă. Adevărat sau fals?

166. **(Funcţii de cost / pierdere în învăţarea automată [profundă]: studiul convexităţii lor)**

prelucrare de Liviu Ciortuz, după
□ • ○ ***CMU, 2015 fall, A. Smola, B. Poczos, HW1, pr. 3.2.1***

În acest exerciţiu vom lucra cu câteva dintre cele mai des folosite funcţii de cost / perdere (engl., loss functions) din învăţarea automată [profundă]. Vom introduce mai întâi anumite *notaţii* pe care le vom folosi în continuare:

– $w \in \mathbb{R}^d$ reprezintă o pondere;

– $x_i \in \mathbb{R}^d$ este o instanţă (vector de trăsături); vom considera că primul element din vectorul x_i are întotdeauna valoarea 1, deci în cele ce urmează nu vom folosi [niciun] termen liber (engl., bias term);

[285] Vedeţi problema 78.

– $r_i = w \cdot x_i \in \mathbb{R}$.

Stabiliţi, pe bază de demonstraţie, pentru fiecare dintre funcţiile de la punctele $a - d$ de mai jos dacă ele sunt (sau nu) convexe.[286]

***Sugestie*: Puteţi face apel la *proprietăţile* prezentate la problema 78.**

a. Funcţia de cost *logistică* (engl., logistic loss function): $L(r_i) = \ln(1 + e^{-r_i})$.

b. Funcţia *ReLU* (engl., Rectified Linear [Unit] function): $R(r_i) = \max(0, r_i)$.

c. Funcţia de cost *hinge* (engl., hinge loss): $H([r_1, r_2, \ldots, r_N]) = \sum_{i=1}^{N} \max(0, 1 - r_i)$.

d. Considerăm matricea $W \in \mathbb{R}^{d \times k}$, pentru care vom nota coloana i cu w_i, iar după aceea $W^\top x_i = [w_1 \cdot x_i, \ldots, w_k \cdot x_i]^\top \overset{not.}{=} [r_i(1), r_i(2), \ldots, r_i(k)]^\top$. Funcţia de cost *softmax* (engl., softmax loss) pentru straturi cu conexiuni totale (engl., fully connected layers) în reţele neuronale, ia ca argument matricea W şi se defineşte prin relaţia

$$S_s(W) = \ln\Big(\sum_{j=1}^{k} e^{r_i(j)}\Big) - r_i(s), \text{ pentru } s = 1, \ldots, k.$$

167. **(Metoda gradientului: exemple de aplicare, şi un rezultat teoretic relativ la convergenţă)**

□ • ***CMU, 2019 spring, Nina Blacan, HW3, pr. 1***

[**Metoda gradientului descendent**]

Fie $f : \mathbb{R}^d \to \mathbb{R}$ o funcţie derivabilă. Algoritmul *gradientului descendent* — la care ne vom referi aici în forma abreviată, GD — porneşte de la un anumit punct $x^{(0)} \overset{not.}{=} (x_1^{(0)}, \ldots, x_d^{(0)}) \in \mathbb{R}^d$, iar după aceea algoritmul „actualizează“ în mod iterativ poziţia $x^{(k)}$, folosind o *regulă de actualizare*, care se exprimă astfel pentru coordonata i:

$$x_i^{(k+1)} \leftarrow x_i^{(k)} - \eta \, \frac{\partial}{\partial x_i} f(x^{(k)}) \text{ unde } i \in \{1, \ldots, d\}. \tag{150}$$

Aici $\dfrac{\partial f}{\partial x_i} : \mathbb{R}^d \to \mathbb{R}$ este derivata parţială a funcţiei f în raport cu coordonata i, apoi $\dfrac{\partial}{\partial x_i} f(x^{(k)})$ reprezintă valoarea acestei derivate parţiale în punctul $x^{(k)}$, iar $\eta > 0$ se numeşte *rata de învăţare* (engl., learning rate).

Scriind unitar, în manieră vectorială regulile (150) pentru toate valorile $i \in \{1, \ldots, d\}$, obţinem *regula gradientului descendent*, ca *regulă de actualizare* a valorilor succesive $x^{(1)}$, $x^{(2)}$, …, $x^{(k)}$, $x^{(k+1)}$, …:

$$x^{(k+1)} \leftarrow x^{(k)} - \eta \, \nabla_x f(x^{(k)}),$$

unde $\nabla_x f(x^{(k)})$ este *vectorul gradient*, format din toate derivatele parţiale ale lui $f(x)$ calculate în punctul $x^{(k)}$:

$$\nabla_x f(x^{(k)}) = \left(\frac{\partial}{\partial x_1} f(x^{(k)}), \ldots, \frac{\partial}{\partial x_d} f(x^{(k)})\right)^\top.$$

[286] Pentru punctele *a* şi *c*, puteţi vedea graficele din figura de la problema 88.

A. Exemple de aplicare a metodei GD...

a. ...în $\mathbb{R}$

Fie funcţia $f(x) = 4x^2 - 2x + 1$ şi rata de învăţare $\eta = 0.1$. Mai întâi, scrieţi expresia derivatei $\frac{\partial}{\partial x} f(x)$. Apoi, pornind de la $x^{(0)} = 1$, pentru fiecare din paşii $k = 0, 1$ şi 2 ai algoritmului GD, scrieţi vectorul gradient $\frac{\partial}{\partial x} f(x^{(k)})$, noua poziţie $x^{(k+1)}$, precum şi noua valoare a funcţiei, $f(x^{(k+1)})$.

b. ...în $\mathbb{R}^2$

Fie $f(x_1, x_2) = x_1^2 + \sin(x_1 + x_2) + x_2^2$. Mai întâi, scrieţi regula de actualizare pentru gradientul descendent, folosind o rată de învăţare oarecare, $\eta > 0$. După aceea, pentru fiecare dintre următoarele *două cazuri* faceţi grafice pentru funcţia f, care va fi considerată ca fiind definită pe domeniul $[-4, +4] \times [-4, +4]$, folosind [soft pentru] *diagrame de izocontur*,[287] precum şi săgeţi de la o poziţie a algoritmului GD către următoarea poziţie.[288]

– *Cazul i*: Determinaţi punctele (x_1, x_2) pentru primii 10 paşi făcuţi de GD, începând cu $(x_1^{(0)}, x_2^{(0)}) = (3, -3)$ şi folosind $\eta = 0.4$,

– *Cazul ii*: Determinaţi punctele (x_1, x_2) pentru primii 10 paşi făcuţi de GD, începând cu $(x_1^{(0)}, x_2^{(0)}) = (3, -3)$ şi folosind $\eta = 0.8$,

Ce observaţi?

B. Analiza algoritmului GD

Metoda gradientului descendent este în sine un *algoritm de căutare locală*: la fiecare pas, el foloseşte informaţia din punctul curent (şi anume, vectorul gradient) pentru a determina mişcarea pe care trebuie să o facă, în direcţia minimizării funcţiei (care este tocmai direcţia descreşterii vectorului gradient). Algoritmul GD nu are cunoştinţă despre modul în care se comportă funcţia pe întreg domeniul de definiţie. Totuşi, deşi GD este un algoritm de căutare locală, dacă f are proprietăţi convenabile — mai precis, dacă f este convexă şi β-netedă (engl., β-smooth), noţiune care este definită mai jos — şi are un punct de minim [de abscisă] x^*, atunci există un $\eta > 0$ pentru care $x^{(k)} \to x^*$ atunci când $k \to +\infty$. În cele ce urmează vă vom ghida cum să faceţi demonstraţia acestei proprietăţi în cazul unidimensional.

c. [Lema descreşterii]

Presupunem că f este o funcţie derivabilă pe tot domeniul de definiţie [LC: cu derivata continuă peste tot[289]] şi β-netedă, unde $\beta > 0$. Prin *definiţie*, faptul

[287] O *diagramă de izocontur* este o modalitate de reprezentare a unor suprafaţe, care în mod normal ar trebui să fie reprezentate în spaţiul 3D, în spaţiul 2D. Ea constă în a desena în plan mai multe curbe, fiecare curbă fiind formată din acele puncte din domeniul de definiţie al funcţiei în care aceasta (adică, funcţia) ia o anumită valoare, fixată. Procedura aceasta este similară cu modul în care se face reprezentarea pe hărţi a regiunilor muntoase, folosind curbele de nivel / altitudine.

[288] Exemple de *diagrame de izocontur* găsiţi la capitolul de *Clusterizare*, la rezolvarea problemei 17.b, pagina 894.

[289] Proprietatea aceasta este implicată dacă este satisfăcută o condiţie mai „tare": funcţia f să fie dublu derivabilă, adică să existe $f''(x)$, pe tot domeniul de definiţie. În acest caz, condiţia de convexitate va implica $f''(x) \geq 0$ pentru orice x.

că f este β-netedă înseamnă că

$$f(y) \leq f(x) + f'(x)(y-x) + \frac{\beta}{2}(y-x)^2 \text{ pentru orice } x \text{ şi } y. \tag{151}$$

Ştim că $x^{(k+1)} = x^{(k)} - \eta f'(x^{(k)})$ la pasul curent al algoritmului GD. Substituind $x = x^{(k)}$ şi $y = x^{(k+1)}$ în relaţia de mai sus, obţinem

$$f(x^{(k+1)}) \leq f(x^{(k)}) + f'(x^{(k)}) \left(x^{(k+1)} - x^{(k)}\right) + \frac{\beta}{2}\left(x^{(k+1)} - x^{(k)}\right)^2.$$

***Observaţie*: Conform problemei 78.a, proprietatea 2 (vedeţi *Observaţia* de acolo), faptul că f este convexă (în ipoteza în care f este derivabilă, cu derivata continuă peste tot) este echivalent cu:**

$$f(y) \geq f(x) + f'(x)(y-x) \text{ pentru orice } x \text{ şi } y. \tag{152}$$

Remarcaţi faptul că relaţia (151) din definiţia pentru proprietatea de β-netezire a lui f introduce o margine superioară pentru $f(y)$, pe lângă marginea inferioară din relaţia (152) dată de proprietatea de convexitate.

Vă cerem să rezolvaţi următoarele două *cerinţe*:

i. **Folosind regula de actualizare din algoritmul GD, arătaţi că din relaţia precedentă rezultă că**

$$f(x^{(k+1)}) \leq f(x^{(k)}) - \eta\left(1 - \frac{\eta\beta}{2}\right)(f'(x^{(k)}))^2.$$

ii. **Identificaţi intervalul în care trebuie să se situeze valorile ratei de învăţare η astfel încât să rezulte $f(x^{(k+1)}) < f(x^{(k)})$.**[290]

d. Presupunem că există un singur punct de minim pentru f, adică un punct x^* astfel încât $f(x^*) < f(x),\ \forall x \neq x^*$. Arătaţi că pentru $\eta = \frac{1}{\beta}$ vom avea $f'(x^{(k)}) \to 0$ atunci când $k \to +\infty$. Explicaţi de ce această convergenţă implică faptul că $x^{(k)} \to x^*$ atunci când $k \to +\infty$.

***Sugestie*: Folosiţi *lema descreşterii* (de la punctul c) pentru a arăta că pentru orice $K \geq 1$ are loc inegalitatea următoare: $\sum_{k=1}^{K}(f'(x^{(k)}))^2 \leq 2\beta(f(x^{(1)} - f(x^*))$.**

***Observaţie*: O demonstraţie pentru convergenţa metodei gradientului descendent în cazul unor funcţii cu proprietăţi mai generale decât cele de mai sus poate fi găsită în cartea *Understanding Machine Learning: From Theory to Algorithms* de Shai Shalev-Schwartz şi Shai Ben-David, Cambridge University Press, 2014.**[291]

168. **(Metoda gradientului descendent: aplicare / implementare)**

• ***Caltech, 2012, Abu Mostafa, HW5, pr. 4***

Considerăm că o anumită funcţie de eroare are expresia $E(u,v) = (ue^v - 2ve^{-u})^2$.

[290] Aşadar, vrem să ne asigurăm că valoarea lui η este aleasă astfel încât, la executarea unei iteraţii a algoritmului GD, valoarea funcţiei obiectiv să descrească.

[291] http://www.cs.huji.ac.il/∼shais/UnderstandingMachineLearning.

a. Calculaţi derivatele parţiale ale acestei funcţii în raport cu u şi respectiv v, adică $\frac{\partial E}{\partial u}(u, v)$ şi $\frac{\partial E}{\partial v}(u, v)$.

b. Pentru a identifica minimul funcţiei de eroare E, veţi aplica metoda gradientului descendent, începând cu punctul $(u_0, v_0) = (1, 1)$. Veţi folosi *rata de învăţare* $\eta = 0.1$.

Câte iteraţii se efectuează până când valoarea diferenţei $E(u_{t+1}, v_{t+1}) - E(u_t, v_t)$ scade pentru prima dată sub pragul de 10^{-14}? Aveţi grijă ca în programul pe care îl veţi implementa să folosiţi [variabile în] dublă precizie, pentru a putea obţine acurateţea cerută.

169. **(Alegerea unei funcţii de eroare convenabilă pentru învăţarea unor concepte geometrice şi studierea aplicabilităţii metodei gradientului descendent în acest context)**

• ⋆ ***T. Mitchell, "Machine Learning", 1997, pr. 4.12***
CMU, 2011 spring, Roni Rosenfeld, HW4, pr. 3

Considerăm că avem de învăţat o clasă de concepte definită de mulţimea dreptunghiurilor din planul bidimensional care au laturile paralele cu axele sistemului de coordonate.

Fiecare ipoteză este descrisă cu ajutorul a 4 coordinate: *llx, lly, urx, ury*. Semnificaţia acestor coordonate este următoarea: perechea (*llx*, *lly*) desemnează colţul din stânga-jos, iar (*urx*, *ury*) desemnează colţul din dreapta-sus al dreptunghiului de învăţat. O instanţă (x, y) este etichetată pozitiv de către ipoteza *llx, lly, urx, ury* dacă şi numai dacă punctul (x, y) este situat în interiorul dreptunghiului respectiv.

Ne propunem să stabilim cum anume ar trebui să procedăm pentru a minimiza eroarea la clasificare automată, pentru a „învăţa" astfel de dreptunghiuri.

a. Definiţi o funcţie de eroare (E) adecvată pentru această problemă de învăţare.

b. Se poate aplica un algoritm de tip gradient descendent pentru a identifica minimul acestei funcţii de eroare? De ce da, sau de ce nu?

c. Dacă răspunsul la punctul b este *da*, descrieţi pe scurt procedura de tip gradient descendent. Dacă răspunsul la punctul b este *nu*, puteţi sugera o aproximare a funcţiei de eroare de la punctul a astfel încât să puteţi aplica metoda gradientului descendent pentru funcţia de eroare nou-definită?

170. **(Metoda subgradientului stochastic descendent (engl., stochastic subgradient descent, SSGD): deducerea regulii de actualizare a parametrilor în cazul unei regresii liniare particulare)**

□ • ○ ***MIT, 2018 spring, Tommi Jaakkola, midterm review, pr. 6***

Funcţia de cost / pierdere (engl., loss) pătratică care a fost definită la problema 88, poate fi generalizată astfel:

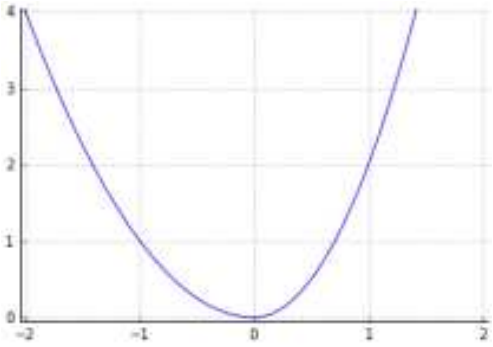
$$L(g,y) = \begin{cases} c_1(g-y)^2 & \textbf{dacă } g > y \\ c_2(g-y)^2 & \textbf{în cazul contar}, \end{cases}$$

unde c_1 şi c_1 sunt constante reale pozitive şi, în contextul folosirii acestei funcţii pentru regresie (adică, învăţare de funcţii cu valori reale), g este valoarea *prezisă* pentru funcţia care este învăţată [şi anume, într-un anumit punct x], iar y este valoarea *adevărată* a funcţiei [în punctul respectiv].

O astfel de definiţie este utilă, de exemplu, în situaţia următoare:
Nişte organizatori de concerte vor să creeze un *model* care să prezică / estimeze cât de mulţi participanţi vor avea la un eveniment oarecare, folosind ca trăsături precum sunt data concertului, genul [muzical al] formaţiei invitate, etc. Ei vor utiliza acest model pentru a decide ce formaţie să angajeze pentru următorul eveniment, dar ar vrea ca predicţiile pe care le fac să fie „protectoare" (engl., conservative), în sensul următor: dacă modelul lor sub-estimează numărul de participanţi, este mai bine aşa decât să-l supra-estimeze.

a. Cât trebuie să fie valorile lui c_1 şi c_2 ca această funcţie de cost să fie chiar loss-ul pătratic (engl., squared error)?

b. În figura alăturată am alcătuit un grafic al valorilor funcţiei de cost L în raport cu diferenţa $g-y$, dar am uitat care sunt valorile c_1 şi c_2 care au fost folosite pentru a realiza acest grafic.



Conform acestui grafic, ce este mai rău:
☐ să sub-estimezi, ori ☐ să supra-estimezi,
dată fiind o aceeaşi diferenţă în valoare absolută, $|g-y|$? Justificaţi riguros.

c. Considerăm un model liniar în care $g(x) = w \cdot x + w_0$. Folosiţi *metoda subgradientului descedent* pentru a deduce *regulile de actualizare* pentru w şi w_0, cu rata de învăţare / pasul (engl., step size) η.[292] Scrieţi apoi pseudo-codul *algoritmului subgradientului descedent stohastic* care foloseşte aceste reguli de actualizare pentru a „învăţa" valorile optime pentru w şi w_0.

171. **(O legătură între Perceptronul Rosenblatt şi metoda subgradientului)**

• ○ ***CMU, 2016 fall, N. Balcan, M. Gormley, HW4, pr. 3.b***

Algoritmul Perceptron este unul dintre primii şi cei mai simpli algoritmi de clasificare automată.[293] El ia ca input un set de date de antrenament format din instanţele etichetate $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, cu $x_i \in \mathbb{R}^d$ şi $y_i \in \{-1, +1\}$ pentru

[292] *Sugestie*: Dacă vă este greu să gândiţi despre w şi x ca fiind vectori, vă sugerăm să procedaţi la început considerând că [w şi x] sunt valori reale / scalare.
[293] Frank Rosenblatt. *The perceptron: A probabilistic model for information storage and organization in the brain.* Psychological Review, 65(6), 386–408, 1958.

$i = 1, \ldots, n$. **Outputul algoritmului este un set de ponderi** $w \in \mathbb{R}^d$. **Pseudo-codul acestui algoritm este următorul:**[294]

initialize $w \leftarrow \bar{0} \stackrel{not.}{=} (0, \ldots, 0) \in \mathbb{R}^d$
for $i = 1, \ldots, n$
  if $y_i \, (w \cdot x_i) \leq 0$ then
    $w \leftarrow w + y_i x_i$
  end if
end for

***Comentariu*:** Dacă setul de date de antrenament este liniar separabil, algoritmul Perceptron converge, adică outputul său, notat cu $\bar{w} \in \mathbb{R}^d$, satisface inegalităţile

$$y_i \, (\bar{w} \cdot x_i) \leq 0 \text{ pentru } i = 1, \ldots, n.$$

Mai mult, se poate arăta că există o margine superioară pentru numărul de greşeli comise de Perceptron (deci, de ajustări ale ponderilor sale) până se ajunge la convergenţă. (Vedeţi pr. 18 de la capitolul *Reţele neuronale artificiale*.)

Acest exerciţiu vă va ajuta să identificaţi o legătură între algoritmul Perceptron şi algoritmul subgradientului descendent.[295]

Considerând funcţia $F(w) = \frac{1}{n} \sum_{i=1}^{n} F_i(w)$**, *algoritmul subgradientului descendent (ciclic)* este următorul:**

initialize the variable $w \in R^d$
do until convergence:
  for $i = 1 \ldots n$
    find a subgradient $v_i(w)$ of $F_i(w)$
    $w \leftarrow w - \eta \, v_i(w)$, where $\eta \in \mathbb{R}^+$ is the *learning rate*
  end for
end do

Arătaţi că Perceptronul este un caz particular de algoritm de tip subgradient descendent.

172. **(Metoda lui Newton: o proprietate interesantă în cazul funcţiilor pătratice)**

prelucrare de Liviu Ciortuz, după
□ • ○ ***Stanford, 2009 fall, Andrew Ng, practice midterm, pr. 6.d***

Fie o funcţie $f : \mathbb{R}^d \to \mathbb{R}$ **definită prin expresia** $f(x) = \frac{1}{2} x^\top A x + bx + c$**, unde** A **este o matrice simetrică şi pozitiv definită.**

[294] În acest pseudo-cod nu se foloseşte termen liber (w_0), deci separarea se face printr-un hiperplan care trece prin originea sistemului de coordonate. (Alternativ, se poate considera că toate instanţele x_i au pe poziţia 1 constanta 1.) Pentru un exemplu de aplicare a algoritmului Perceptron, vedeţi problema 16 de la capitolul *Reţele neuronale artificiale*.

[295] Vă readucem aminte că, dată fiind o funcţie $F : \mathbb{R}^d \to \mathbb{R}$ care poate fi nederivabilă, vectorul $v(x) \in \mathbb{R}^d$ este un *subgradient* al lui $F(x)$ dacă şi numai dacă pentru orice x' are loc inegalitatea:

$$F(x') \geq F(x) + v(x) \cdot (x' - x).$$

a. Arătaţi că f este funcţie convexă.
(*Sugestie*: Puteţi folosi fie definiţia fie proprietăţile formulate la problema 78.)

b. Demonstraţi că atunci când se foloseşte metoda lui Newton pentru a afla minimul funcţiei f, este suficient să se execute o singură iteraţie. (Puteţi considera că metoda lui Newton face iniţializarea cu vectorul 0 din $\mathbb{R}^d$.)

173. **(Reparametrizarea liniară nu afectează metoda [de optimizare a] lui Newton, însă afectează metoda gradientului)**

□ • · ***Stanford, 2014 fall, Andrew Ng, HW1, pr. 5***

Presupunem că folosim un algoritm de optimizare iterativă (de exemplu, metoda lui Newton sau metoda gradientului descendent) pentru a calcula [punctul în care se atinge] minimul unei funcţii derivabile în mod continuu $f(x)$, cu $x \in \mathbb{R}^n$. Presupunem că iniţializăm algoritmul cu vectorul $x^{(0)} = \vec{0}$ din $\mathbb{R}^n$. La execuţie, algoritmul va produce pentru un input oarecare $x \in \mathbb{R}^n$ (fixat) câte o valoare din $\mathbb{R}^n$ la fiecare iteraţie: $x^{(1)}$, $x^{(2)}, \ldots$.

Considerăm acum că ni se dă o matrice oarecare pătratică nesingulară (adică, inversabilă) $A \in \mathbb{R}^{n \times n}$ şi, de asemenea, că definim o nouă funcţie $g(z) = f(Az)$. Presupunem că folosim acelaşi algoritm de optimizare iterativă pentru a calcula optimul funcţiei g, cu iniţializarea $z^{(0)} = \vec{0}$ din $\mathbb{R}^n$.

Dacă valorile $z^{(1)}$, $z^{(2)}, \ldots$ produse folosind această metodă satisfac [în mod necesar] condiţiile $z^{(i)} = A^{-1}x^{(i)}$ pentru orice i, vom spune că acest algoritm de optimizare este *invariant la reparametrizare liniară*.

a. Arătaţi că *metoda lui Newton* (aplicată pentru găsirea minimului unei funcţii) este invariantă la reparametrizare liniară.

***Sugestie*: Remarcaţi faptul că deoarece $z^{(0)} = \vec{0} = A^{-1}x^{(0)}$, este suficient să *demonstraţi* următoarea *implicaţie*:**

> **dacă la aplicarea metodei lui Newton asupra lui $f(x)$, din $x^{(i)}$ se obţine $x^{(i+1)}$,**
> **atunci când metoda lui Newton se va aplica asupra lui $g(z) = f(Az)$, din $z^{(i)} = A^{-1}x^{(i)}$ se va obţine $z^{(i+1)} = A^{-1}x^{(i+1)}$.**

b. Este oare metoda *gradientului descendent* invariantă la reparametrizare liniară? Justificaţi în mod riguros răspunsul dumneavoastră.

174. **(Metoda multiplicatorilor lui Lagrange: aplicare într-un caz simplu)**

prelucrare de Liviu Ciortuz, după
• · ***CMU, 2021 fall, Aarti Singh, Recitation 7, pr. 4***

Folosiţi condiţiile Karush-Kuhn-Tucker pentru a rezolva următoarea problemă de optimizare cu restricţii:

$$\begin{array}{ll} & \min_{x,y} x^2 + y^2 \\ \textbf{a. î.} & x + y - 1 = 0. \end{array}$$

***Cerinţe preliminare rezolvării*:**
Veţi face reprezentarea grafică corespunzătoare [rezolvării] acestei probleme. Folosind această reprezentare veţi indica soluţia problemei de optimizare date înainte de rezolvarea ei folosind metoda multiplicatorilor lui Lagrange. (Faceţi justificarea în mod riguros.)

175. **(Metoda dualităţii Lagrange: rezolvarea formei duale pentru o problemă de optimizare convexă simplă)**

prelucrare de Liviu Ciortuz, după
□ • ○ ***CMU, 2013 fall, A. Smola, G. Gordon, midterm practice questions, pr. 2.c***

Fie următoarea problemă de optimizare cu restricţii, în formă primală:

$$\begin{aligned} &\min_x (x^2+1) \\ &\text{a. î. } (x-2)(x-4) \leq 0. \end{aligned}$$

Obţineţi forma duală a acestei probleme de optimizare, apoi rezolvaţi această problemă duală prin 3 metode: *i*. direct; *ii*. rezolvând sistemul de condiţii Karush-Kuhn-Tucker şi folosind partea a doua din teorema Karush-Kuhn-Tucker; *iii*. rezolvând mai întâi problema primală şi apoi folosind relaţia de legătură dintre soluţiile celor două probleme (primală şi respectiv duală) obţinută din condiţia de staţionaritate / optimalitate din sistemul de condiţii Karush-Kuhn-Tucker.[296]

176. **(Metoda dualităţii Lagrange: aplicare în $\mathbb{R}$, rezolvând mai întâi problema duală şi folosind apoi relaţia de corespondenţă cu soluţia problemei primale)**

□ · ∗ ***Liviu Ciortuz, 2019, după CMU, Aarti Singh, 2010 fall, SVM – Lecture notes***

Se dă următoarea problemă de optimizare convexă, scrisă în forma primală:

$$\begin{aligned} &\min_x x^2 \\ &\textbf{a. î. } x \geq b, \end{aligned}$$

unde b este o constantă reală.

a. Rezolvaţi această problemă direct, în funcţie de valorile lui b. (Faceţi în prealabil reprezentarea grafică.)

b. Este oare îndeplinită *condiţia lui Slater* pentru această problemă de optimizare convexă? În cazul afirmativ, precizaţi care sunt implicaţiile satisfacerii acestei condiţii în contextul problemei date.

c. Scrieţi lagrangeanul generalizat $L_P(x, \alpha)$, unde α este multiplicatorul Lagrange corespunzător restricţiei din problema de optimizare convexă dată.

[296] Pentru varianta *iii* este necesar ca în prealabil să verificaţi faptul că este îndeplinită condiţia lui Slater.

Calculaţi x^*, rădăcina ecuaţiei $\frac{\partial}{\partial x} L_P(x, \alpha) = 0$, în funcţie de α.[297]
Aflaţi expresia lagrangeanului dual $L_D(\alpha)$, înlocuind în expresia lagrangeanului generalizat, $L_P(x, \alpha)$, argumentul x cu expresia pe care tocmai aţi obţinut-o pentru x^*, soluţia unică a ecuaţiei $\frac{\partial}{\partial x} L_P(x, \alpha) = 0$.

d. Scrieţi forma duală a problemei de optimizare convexă din enunţ.[298] **Rezolvaţi această problemă şi apoi calculaţi soluţia x^* a problemei date în enunţ, folosind relaţia pe care aţi obţinut-o la punctul c.**

177. **(Metoda dualităţii Lagrange: un exemplu de formalizare a unei probleme de optimizare convexă pornind de la descrierea ei în limbaj natural, urmată de rezolvarea ei)**

prelucrare de Liviu Ciortuz, după
□ • · ***CMU, 2021 fall, Aarti Singh, recitation 7, pr. 1***[299]

Să zicem că o companie fabrică două produse notate cu X şi Y, folosind pentru realizarea fiecăruia dintre acestea două maşini, care sunt notate cu A şi respectiv B. Realizarea fiecărui produs X necesită o procesare de 50 de minute pe maşina A şi o [altă] procesare de 30 de minute pe maşina B. Realizarea fiecărui produs Y necesită o procesare de 24 de minute pe maşina A şi o [altă] procesare de 33 de minute pe maşina B.

***Timpul total* [de procesare] *disponibil* este estimat la 40 de ore în cazul maşinii A, şi la 35 ore în cazul maşinii B.**

Pentru săptămâna curentă, cererea [prognozată] este de cel puţin 45 de unităţi din produsul X şi de cel puţin 5 unităţi din produsul Y.

Politica / *obiectivul* companiei este să maximizeze numărul total de unităţi produse (X şi Y) la sfârşitul săptămânii.

a. Formulaţi ca o *problemă de optimizare convexă* problema care constă în a decide cât de multe unităţi din fiecare produs trebuie să producă această companie în săptămâna curentă pentru a atinge *obiectivul* formulat mai sus.

***Atenţie*! Problema de optimizare convexă având — prin definiţie — funcţia obiectiv supusă operatorului $\min$, va trebui să aveţi grijă cum definiţi funcţia obiectiv în cazul specific prezentat mai sus.**

b. Rezolvaţi problema de optimizare convexă de la punctul a prin *netoda grafică*. În prealabil veţi identifica *domeniul de fezabilitate*, adică mulţimea punctelor din planul euclidian pentru care sunt satisfăcute *restricţiile* care apar în formularea acestei probleme de optimizare convexă.

c. Stabiliţi dacă problema de optimizare convexă pe care aţi formulat-o la punctul a satisface condiţia lui Slater. În cazul afirmativ, care sunt consecinţele?

[297] În general, rădăcinile ecuaţiei $\frac{\partial}{\partial x} L_P(x, \alpha) = 0$ se mai numesc *punctele staţionare* ale lui L_P.
[298] Vedeţi problema 82.b.
[299] From http://people.brunel.ac.uk/mastjjb/jeb/or/morelp.html.

d. Scrieţi sistemul de condiţii Karush-Kuhn-Tucker pentru problema de optimizare convexă pe care aţi formulat-o la punctul a.

e. Rezolvaţi sistemul de condiţii Karush-Kuhn-Tucker de la punctul d, iar la final indicaţi soluţia problemei de optimizare convexă de la punctul a.

***Atenţie*! Veţi ignora faptul că numărul de unităţi care trebuie realizate din fiecare dintre produsele X şi Y trebuie să fie numere naturale; le veţi considera numere reale pozitive. (La final veţi putea aplica operatorul *parte întreagă inferioară* la soluţiile obţinute.)**

178. **(Metoda dualităţii Lagrange: un exemplu de aplicare)**

Liviu Ciortuz, 2021, pornind de la secţiunea 2 din "Convex optimization overview (cont'd)", Chuong B. Do, Stanford, 2009

Fie problema de optimizare convexă

$$\begin{aligned} \min_{x_1,x_2\in\mathbb{R}} \quad & (x_1^2 + x_2) \\ \text{a. î.} \quad & 2x_1 + x_2 \geq 4 \\ & x_2 \geq 1. \end{aligned}$$

a. Verificaţi că aceasta este într-adevăr o problemă de optimizare convexă.

Pe un sistem de axe de coordonate în plan, indicaţi *regiunea fezabilă* corespunzătoare acestei probleme de optimizare.

b. Este *condiţia lui Slater* satisfăcută în cazul acestei probleme? (Justificaţi riguros.)

În cazul afirmativ, precizaţi două *implicaţii* ce decurg din satisfacerea condiţiei lui Slater.

c. Deduceţi *forma duală* a acestei probleme de optimizare convexă.

d. Rezolvaţi problema de optimizare în forma duală pe care aţi dedus-o la punctul precedent. Veţi nota soluţia problemei duale cu $\bar\alpha_1, \bar\alpha_2$.

***Indicaţie*: Veţi observa că expresia lagrangeanului dual, $L_D(\alpha_1, \alpha_2)$, se poate scrie de fapt doar în funcţie de α_1. Faceţi graficul funcţiei L_D în raport cu α_1, într-un reper de axe ortogonale în care axa orizontală este $O\alpha_1$. Marcând pe grafic *zona de fezabilitate* pentru α_1, veţi putea identifica uşor soluţia problemei duale.**

e. Deduceţi soluţia problemei de optimizare în *forma primală* pornind de la soluţia problemei duale pe care aţi obţinut-o la punctul precedent. Veţi nota soluţia problemei primale cu $\bar x_1, \bar x_2$.

f. Scrieţi *condiţiile Karush-Kuhn-Tucker* pentru problema de optimizare dată. Are oare *soluţie* sistemul format din aceste condiţii?

În cazul afirmativ, găsiţi soluţiile problemelor de optimizare primală şi respectiv duală, pornind de la soluţia sistemului format din condiţiile Karush-Kuhn-Tucker.

179. **(Metoda dualităţii Lagrange: două exemple de aplicare)**

prelucrare de Liviu Ciortuz, după
• · ***CMU, 2021 fall, Aarti Singh, recitation 7, pr. 2***

Fie următoarele două probleme de optimizare:

A. $\begin{array}{rl} \min_{x,y} & (x+y) \\ \text{a. î.} & x+y \geq 2 \\ & x, y \geq 0 \end{array}$ **B.** $\begin{array}{rl} \min_{x,y} & (x+3y) \\ \text{a. î.} & x+y \geq 2 \\ & x, y \geq 0 \end{array}$

Pentru fiecare dintre aceste două probleme de optimizare răspundeţi la următoarele întrebări:

a. Este problema respectivă o problemă de optimizare convexă? Justificaţi riguros.

b. Cum se defineşte în mod *analitic* mulţimea punctelor fezabile corespunzătoare acestei probleme de optimizare?

c. Reprezentaţi grafic, pe un sistem de axe de coordonate ortogonale mulţimea punctelor fezabile corespunzătoare respectivei probleme de optimizare. (Haşuraţi regiunea fezabilă.)

d. Puteţi indica în mod direct soluţia problemei de optimizare respective? (Folosiţi cunoştinţe simple de geometrie şi / sau geometrie analitică.)

e. Fără a rezolva sistemul de condiţii Karush-Kuhn-Tucker corespunzător respectivei probleme de optimizare — şi chiar fără a scrie acel sistem —, puteţi spune dacă el este satisfiabil?

f. Rezolvaţi sistemul de condiţii Karush-Kuhn-Tucker corespunzător respectivei probleme de optimizare, iar după aceea indicaţi soluţia problemei de optimizare, precum şi valoarea optimă a funcţiei obiectiv.

180. **(Metoda dualităţii Lagrange: un exemplu de problemă de optimizare convexă pentru care condiţiile Karush-Kuhn-Tucker nu sunt satisfăcute)**

· ***Sebastian Ciobanu, 2018***[300]

Ştim că atunci când o problemă de optimizare convexă cu restricţii satisface condiţia lui Slater,[301] **urmează cu *necesitate* că forma primală şi forma duală a problemei de optimizare date au acelaşi optim (adică, $p^* = d^*$) şi, conform teoremei Karush-Kuhn-Tucker,**[302] **soluţiile acestor două forme ale problemei satisfac condiţiile Karush-Kuhn-Tucker.**

***Obiectivul* acestui exerciţiu este să vă arate că în cazul nesatisfacerii condiţiei lui Slater, este posibil ca sistemul de condiţii Karush-Kuhn-Tucker să nu fie satisfiabil (adică să nu aibă soluţie), chiar dacă problema de optimizare convexă dată este satisfiabilă.**

[300] Surse: https://en.wikipedia.org/wiki/Karush%E2%80%93Kuhn%E2%80%93Tucker_conditions şi https://math.stackexchange.com/questions/1346767/do-we-really-need-the-constraint-qualification?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

[301] Vedeţi *Observaţia importantă* de la pagina 188.

[302] Vedeţi *Comentariul* de la problema 85, pag. 187.

Fie următoarea problemă de optimizare cu restricţii:

$$\begin{aligned} \min_x \quad & x \\ \text{a. î.} \quad & x^2 \leq 0 \end{aligned}$$

a. Indicaţi soluţia problemei de optimizare din enunţ. (În prealabil, veţi determina mulţimea de fezabilitate.)

b. Stabiliţi dacă problema de optimizare este convexă.

c. Stabiliţi dacă este satisfăcută condiţia lui Slater.

d. Rezolvaţi sistemul reprezentat de condiţiile Karush-Kuhn-Tucker.

e. Orice soluţie a problemei de optimizare care a fost dată în enunţ satisface condiţiile Karush-Kuhn-Tucker?

181. **(O [altă] variantă a algoritmului Perceptron, pentru care relaţia de actualizare a ponderilor se obţine rezolvând o problemă de optimizare convexă cu restricţii)**

■ □ ● ○ ∗ ***University of Helsinki, 2014 spring, Jyrki Kivinen, HW5, pr. 3***

Asemănător cu problema 87, aici vom lucra asupra unui algoritm în care vectorul w_{t+1} este obţinut prin rezolvarea problemei de optimizare următoare:

$$\begin{aligned} \min_w \quad & d_{\text{re}}(w, w_t) \\ \text{a. î.} \quad & y_t w \cdot x_t \geq 0 \\ & w_i \geq 0 \text{ pentru } i = 1, \ldots, d \\ & \textstyle\sum_{i=1}^d w_i = 1, \end{aligned}$$

unde $w \overset{not.}{=} (w_1, \ldots, w_d)$, iar $d_{\text{re}}(u, v)$ desemnează entropia relativă[303]

$$d_{\text{re}}(u, v) = \sum_{i=1}^{d} u_i \ln \frac{u_i}{v_i}.$$

Demonstraţi că atunci când există o soluţie pentru această problemă, rezultă cu necesitate că există un număr real $\beta_t \geq 1$ astfel încât noul vector de ponderi satisface relaţia

$$w_{t+1,i} = \frac{1}{Z} w_{t,i} \beta_t^{y_t x_{t,i}}$$

unde

$$Z = \sum_{j=1}^{d} w_{t,j} \beta_t^{y_t x_{t,j}}.$$

Nu trebuie să găsiţi ca atare valoarea lui β_t (fiindcă nu există o formulă analitică (engl., closed-form solution)); trebuie doar să arătaţi că soluţia w_{t+1} are forma care a fost indicată.

***Observaţie*: Deşi acest algoritm nu este [probabil] foarte practic, astfel de probleme de optimizare au condus la crearea algoritmului Winnow, precum şi a altor algoritmi *multiplicativi* de învăţare automată *online*.**

[303] Pentru proprietăţi elementare ale entropiei relative (care este cunoscută şi sub numele de *divergenţa Kullback-Leibler*), vedeţi problema 65.

182. **(Teorema de reprezentare generalizată)**

prelucrare de Liviu Ciortuz după
□ • ***Stanford, John Duchi, Supplemental lecture notes***

O generalizare a *teoremei de reprezentare* care a fost demonstrată la problema 88.a a fost publicată de Bernhard Schölkopf, Ralf Herbrich şi Alex Smola în anul 2001.[304] Generalizarea se referă — pe lângă faptul că nu vom mai cere ca funcţia de cost $L(z, y)$ să fie derivabilă în raport cu argumentul z, aşa cum s-a considerat la rezolvarea problemei 88 — la faptul că termenul de regularizare $\frac{\lambda}{2}\|w\|^2$ din expresia (131) se înlocuieşte cu termenul mai general $\rho(\|w\|)$, unde ρ este o funcţie nedescrescătoare.[305]

Similar cu modul în care am procedat la problema 88, vom considera setul de date de antrenament $(x_i, y_i)_{i=1,\dots,n}$, cu $x_i \in \mathbb{R}^d$ şi $y_i \in \mathbb{R}$. Acum vom desemna *costul* (sau, *riscul*) *empiric regularizat* prin funcţia $J_\rho(w) \overset{not.}{=} \frac{1}{n}\sum_{i=1}^n L(w \cdot x_i, y_i) + \rho(\|w\|)$.

***Concluzia* teoremei de reprezentare generalizată este formulată astfel:**

$$\exists \alpha \in \mathbb{R}^n \text{ a. î. pentru } w^{(\alpha)} \overset{not.}{=} \sum_{i=1}^n \alpha_i x_i \in \mathbb{R}^d \text{ şi } \forall w \in \mathbb{R}^d \text{ are loc inegalitatea}$$
$$J_\rho(w^{(\alpha)}) \leq J_\rho(w).$$

***Observaţie*: Importanţa (foarte mare!) a acestor *teoreme de reprezentare* constă în faptul că datorită lor rezolvarea problemelor de învăţare automată constând în minimizarea unei funcţii de tip $J_\rho(w)$ se poate reduce la căutarea unor tupluri $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathbb{R}^n$,[306] iar aceasta se poate realiza aplicând algoritmi clasici pentru minimizarea funcţiilor.[307]**

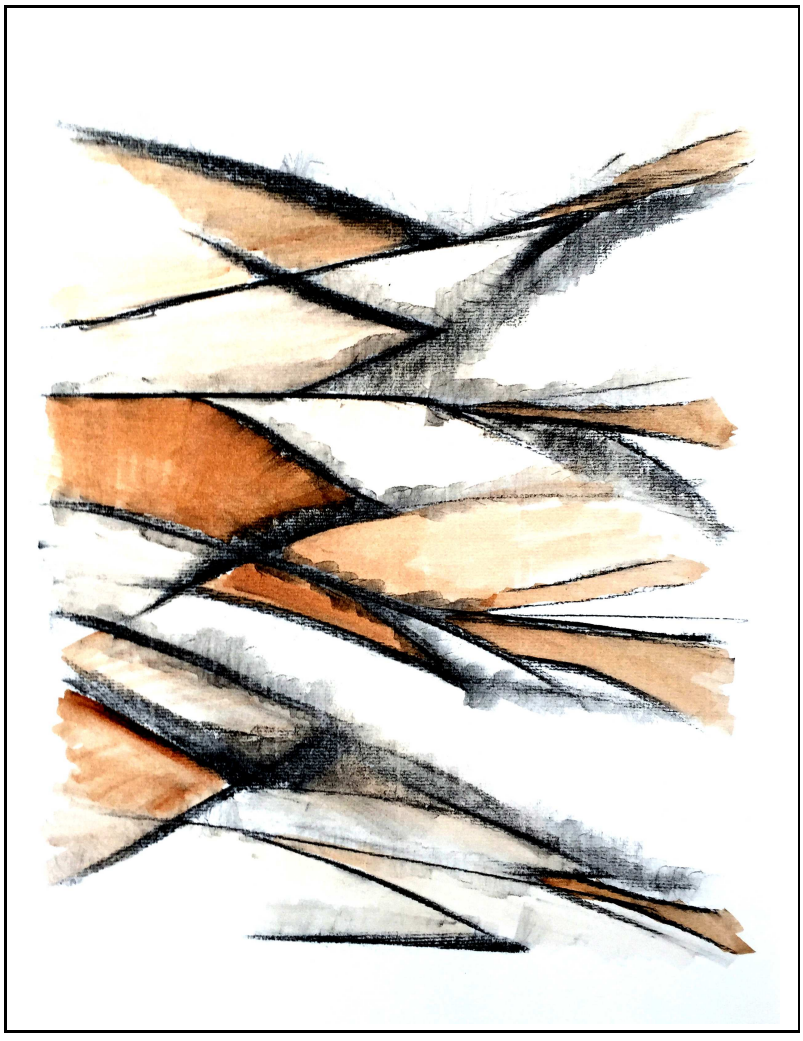
Demonstraţi teorema de reprezentare generalizată.

[304] Vedeţi articolul *A Generalized Representer Theorem* din volumul Computational Learning Theory, Lecture Notes in Computer Science, 2111: 416–426, 2001.

[305] Evident, pentru cazul particular $\rho(\|w\|) = \frac{\lambda}{2}\|w\|^2$ cu $\lambda > 0$, se obţine regularizarea de normă L_2. Într-un alt caz particular remarcabil, ρ poate fi considerată chiar funcţia constantă 0, deci *teorema de reprezentare generalizată* acoperă şi cazul când nu se face deloc regularizare.

[306] Altfel spus, căutarea soluţiilor se poate limita la *spaţiul liniar generat* de instanţele de antrenament x_i.

[307] Vedeţi https://en.wikipedia.org/wiki/Representer_theorem, accesat la data de 30.05.2018.



© *M. Romanică*

1 Metode de regresie

Sumar

Noţiuni preliminare

– **estimarea parametrilor unor distribuţii probabiliste uzuale (în special distribuţia Bernoulli, distribuţia gaussiană, distribuţia Laplace); vedeţi secţiunea corespunzătoare de la cap. *Fundamente*;**

– **funcţia sigmoidală: ex. 33; funcţia de cost logistică: ex. 13.d;**

– **elemente de calcul vectorial (în particular, produsul scalar) şi de calcul matriceal: ex. 36 de la cap. *Fundamente*;**
norma L_2 (euclidiană) şi norma L_1: ex. 3, ex. 26, ex. 11;
calculul derivatelor parţiale [de ordinul întâi şi al doilea]: ex. 7;
reguli de derivare cu argumente vectoriale: ex. 1;
calculul subderivatelor / subgradienţilor şi subdiferenţialelor: ex. 81 de la cap. *Fundamente*;

– **metode de optimizare (în speţă pentru aflarea maximului / minimului unei funcţii reale, derivabile): metoda analitică; metoda gradientului şi metoda lui Newton — exemplificare: ex. 80 de la cap. *Fundamente*; metoda subgradientului: ex. 171 de la capitolul de *Fundamente*; metoda descreşterii (resp., a creşterii) pe coordonate (engl., coordinate descent / ascent).**

Regresia liniară

- **prezentarea *generală* a metodei regresiei liniare:**[308]

  – MLE şi corespondenţa cu estimarea în sens LSE (least squared errors): ex. 3.A.abc; rezolvare folosind matricea de design şi *optimizare analitică*: ex. 3.A.d;
  particularizare pentru cazul unidimensional: ex. 1.ab, ex. 20;
  exemplificare pentru cazul unidimensional (ex. 2, ex. 21) şi pentru cazul bidimensional (ex. 23.a, ex. 32);

  – (P1) *scalarea atributelor* nu schimbă predicţiile obţinute (pentru instanţele de test) cu ajutorul *formulelor analitice*: ex. 4, 41.a;

  – (P2) adăugarea de noi trăsături / atribute nu măreşte suma pătratelor erorilor: ex. 25;

  – o *proprietate surprinzătoare* a regresiei liniare: adăugarea câtorva „observaţii“ suplimentare poate conduce la modificarea radicală a valorilor optime ale parametrilor de regresie: CMU, 2014 fall, Z. Bar-Joseph, W. Cohen, HW2, pr. 4;

  – [rezolvarea problemei de] regresie liniară folosind *metoda lui Newton*: ex. 7;

  – MAP şi corespondenţa cu *regularizarea* de normă L_2 (regresia *ridge*): ex. 3.C;
  particularizare pentru cazul unidimensional: ex. 1.c;

[308] În mod implicit, în această secţiune se va considera că termenul-zgomot este modelat cu distribuţia gaussiană (dacă nu se specifică altfel, în mod explicit).

– **(P3) efectul de diminuare a ponderilor (engl., weight decay) în cazul regularizării de normă L_2 (respectiv L_1) a regresiei liniare, în comparaţie cu cazul neregularizat: ex 26.b (respectiv ex. 26.a);**

- **bias-ul şi [co]varianţa estimatorului regresiei liniare; bias-ul regresiei *ridge*: ex. 5;**
- **regresia polinomială [LC: mai general: folosirea aşa-numitelor *funcţii de bază*]: ex. 3.B;**
  exemplificare pentru cazul bidimensional: CMU, 2015 spring, T. Mitchell, N. Balcan, HW4, pr. 1;

- **cazul regresiei liniare cu termen de regularizare L_2 (regresia *ridge*):**
  deducerea regulilor de actualizare pentru *medoda gradientului descendent*:
  varianta "batch" / "steepest descent": ex. 6.a;
  şi varianta stohastică / secvenţială / "online": ex. 6.b; exemplu de aplicare: ex. 24;
- **cazul regresiei liniare cu termen de regularizare de normă L_1:**
  rezolvare combinând *metoda descreşterii pe coordonate* cu *metoda optimizării analitice*: ex. 11;
  rezolvare combinând *metoda descreşterii pe coordonate* cu *metoda subgradientului* (aplicare la selecţia de trăsături): ex. 28;
- **regresia liniară în cazul zgomotului modelat cu distribuţia *Laplace* (în locul zgomotului gaussian): ex. 8.B;**
  exemplificare pentru cazul bidimensional: ex. 23.c;
  rezolvare în cazul unidimensional [chiar particularizat] cu ajutorul derivatei, acolo unde aceasta există: ex. 29;

- **regresia liniară şi *overfitting-ul*: ex. 12;**
- **regresie liniară folosită pentru *clasificare*: exemplificare: ex. 32;**

- **cazul *multivaluat* al regresiei liniare, reducerea la cazul uninomial: ex. 31;**
- **regresia liniară cu regularizare L_2 (regresia *ridge*), *kernel-izarea* ecuaţiilor „normale": ex. 9; aplicare: ex. 27;**
  (P4) folosind nucleu RBF, eroarea la antrenare devine 0 atunci când parametrul de regularizare λ tinde la 0: ex. 10;
- ***regresia liniară ponderată*: ex. 8.A;**
  particularizare pentru cazul unidimensional [neparametric]: CMU, 2010 fall, Aarti Singh, midterm, pr. 4
  particularizare / exemplificare pentru cazul bidimensional: ex. 23.b;
  cazul multivaluat, cu regularizare L_2: Stanford, 2015 fall, Andrew Ng, midterm, pr. 2;
  o *proprietate* a regresiei liniare local-ponderate [demonstrată în cazul unidimensional]: „netezirea" liniară: ex. 30.

Regresia logistică

– **prezentare generală,**
(•) calculul funcţiei de log-verosimilitate condiţională, estimarea parametrilor în sens M(C)LE, folosind *metoda gradientului* (i.e., deducerea regulilor de actualizare a parametrilor): ex. 13.bc, ex. 15.abc, ex. 35.a; ex. 36.a, 41.b;

particularizare **pentru cazul datelor din $\mathbb{R}^2$: ex. 35 (inclusiv *regularizare* L_1 / estimarea parametrilor în sens MAP, folosind o distribuţie a priori Laplace);**

- **(P0)** ***graniţa de decizie*** **pentru regresia logistică este liniară: ex. 35.d;**
- **(P1) funcţia de log-verosimilitate condiţională în cazul regresiei logistice este concavă (deci are un maxim global), fiindcă matricea hessiană este negativ semi-definită: ex. 14;**

  Observaţie**: Demonstraţia furnizează tot ce este necesar pentru obţinerea [ulterioară a] relaţiei de actualizare a parametrilor la aplicarea *metodei lui Newton* în cazul regresiei logistice; exemplificare: ex. 36.f, ex. 15.e;**
- **(P2) analiza efectului duplicării atributelor: ex. 37;**
- **(P3) echivalenţa cu minimizarea funcţiei de cost logistice: ex. 13.d, ex. 36.g;**
- **(P4) efectul de diminuare a ponderilor (engl., weight decay) în cazul regularizării de normă L_2 a regresiei logistice — adică la estimarea parametrilor în sens MAP, folosind ca distribuţie a priori distribuţia gaussiană multidimensională sferică —, în comparaţie cu cazul estimării parametrilor în sensul MLE: ex. 16;**
- ***Variante*** **/** ***extensii*** **ale regresiei logistice:**

  • regresia logistică local-ponderată, cu regularizare L_2:
  calcularea vectorului gradient şi a matricei hessiene (necesare pentru aplicarea metodei lui Newton în acest caz): ex. 17;

  • regresia logistică kernel-izată:
  adaptarea metodei gradientului: ex. 18;

  • regresia logistică n-ară (aşa-numita regresie *softmax*):
  calculul funcţiei de log-verosimilitate, cu regularizare L_2, deducerea regulilor de actualizare a ponderilor, folosind metoda gradientului: ex. 19;

  (P5) echivalenţa cu un anumit tip de mixtură de distribuţii gaussiene multidimensionale: ex. 39;
- **(P6) o [interesantă] proprietate comună pentru regresia liniară şi regresia logistică: ex. 38;**
- **întrebări (cu răspuns A/F) cu privire la aplicarea *metodei lui Newton* comparativ cu *metoda gradientului* (în contextul rezolvării problemelor de regresie liniară şi / sau regresie logistică): ex. 41.c;**
- **comparaţii între regresia logistică şi alţi clasificatori (Bayes Naiv, ID3): ex. 35.c, ex. 40.ab.**

Modele liniare generalizate (GLM)

- **condiţii suficiente pentru *concavitatea* funcţiei de log-verosimilitate: ex. 44;**
- **particularizare pentru cazul distribuţiei geometrice: ex. 42;**
- **particularizare pentru cazul distribuţiei gaussiene unidimensionale: ex. 43.**

1.1 Metode de regresie — Probleme rezolvate

1.1.1 Regresia liniară

1. **(Estimare [în sens MLE şi MAP] pentru parametrul unui model de regresie liniară unidimensională fără termen liber, în prezenţa unei componente-zgomot modelate de o distribuţie gaussiană)**

■ • ***CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, midterm, pr. 3***
CMU, 2011 spring, Tom Mitchell, midterm, pr. 4

***Introducere*:** *Obiectivul* acestei probleme este acela de a face cunoştinţă cu metoda / problema de regresie liniară într-una dintre cele mai simple variante ale sale: cazul unidimensional. Aceasta înseamnă că variabila de ieşire (Y) depinde de o singură variabilă de intrare (X),[309] dar şi de o componentă „zgomot" (ε). Modelarea acestei componente va fi făcută în manieră probabilistă, şi anume (aici) cu ajutorul unei distribuţii gaussiene.

Fie variabilele aleatoare X şi Y. Presupunem că valorile variabilei Y se generează în funcţie de valorile variabilei X conform relaţiei:

$$Y = aX + \varepsilon,$$

unde fiecare ε reprezintă valoarea unei variabile aleatoare (care este independentă de variabilele precedente), numită „zgomot", şi care urmează o distribuţie gaussiană de medie 0 şi deviaţie standard $\sigma > 0$. Acest fapt se notează îndeobşte astfel: $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.

Acesta este un model probabilist (de tip „regresie liniară", cu un singur atribut, X), în care a este singurul parametru (o pondere; engl., weight).

Probabilitatea condiţionată a lui Y în raport cu X urmează distribuţia $\mathcal{N}(aX, \sigma^2)$, deci poate fi descrisă ca

$$p(Y|X, a) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(-\frac{1}{2\sigma^2}(Y - aX)^2\Big).$$

Următoarele întrebări se referă la acest model de regresie liniară.

***Estimare în sens MLE*:**

a. Presupunem că avem un set de date de antrenament constând din n perechi (X_i, Y_i), cu $i = 1, \ldots, n$, şi că varianţa σ este cunoscută. Care dintre următoarele expresii reprezintă în mod corect *problema de optimizare* pentru estimarea lui a în sensul verosimilităţii maxime? Răspundeţi cu *da* sau *nu* la fiecare dintre

[309] Evident, dependenţa dintre Y şi X va fi considerată de tip liniar.

ele. Nu neapărat una singură dintre aceste expresii trebuie să fie însoţită de răspunsul *da*.

i. $\arg\max_a \sum_i \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2\Big)$

ii. $\arg\max_a \prod_i \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2\Big)$

iii. $\arg\max_a \sum_i \exp\Big(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2\Big)$

iv. $\arg\max_a \prod_i \exp\Big(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2\Big)$

v. $\arg\max_a \sum_i (Y_i - aX_i)^2$

vi. $\text{argmin}_a \sum_i (Y_i - aX_i)^2$

b. Obţineţi estimarea de verosimilitate maximă pentru parametrul a în funcţie de exemplele de antrenament X_i şi Y_i ($i = 1, \ldots, n$). Vă recomandăm / cerem să porniţi de la forma cea mai simplă a problemei de optimizare pe care aţi identificat-o la punctul precedent.

***Estimare în sens MAP*:**

Să presupunem că $a \sim \mathcal{N}(0, \lambda^2)$, deci

$$p(a|\lambda) = \frac{1}{\sqrt{2\pi}\lambda} \exp\left(-\frac{1}{2\lambda^2}a^2\right) \text{ cu } \lambda > 0.$$

Probabilitatea (de fapt, p.d.f.) a posteriori a parametrului a este

$$\begin{aligned} p(a|Y_1, \ldots, Y_n, X_1, \ldots, X_n, \lambda) &\overset{T.B.}{=} \frac{p(Y_1, \ldots, Y_n|X_1, \ldots, X_n, a)\, p(X_1, \ldots, X_n, a|\lambda)}{p(Y_1, \ldots, Y_n|X_1, \ldots, X_n, \lambda)} \\ &\overset{F.P.T.}{=} \frac{p(Y_1, \ldots, Y_n|X_1, \ldots, X_n, a)\, p(a|\lambda)}{\int_{a'} p(Y_1, \ldots, Y_n|X_1, \ldots, X_n, a')\, p(a'|\lambda)\, da'}. \end{aligned}$$

Putem să ignorăm numitorul acestei fracţii atunci când facem estimare în sens MAP, fiindcă el nu depinde de parametrul a.

c. Presupunem că $\sigma = 1$ şi că parametrul a priori λ a fost fixat. Găsiţi estimarea de probabilitate maximă a posteriori (engl., maximum a posteriori probability, MAP) pentru parametrul a:

$$\arg\max_a [\ln p(Y_1, \ldots, Y_n|X_1, \ldots, X_n, a) + \ln p(a|\lambda)].$$

Soluţia pe care o veţi da trebuie să fie în funcţie de X_i, Y_i ($i = 1, \ldots, n$) şi λ.

Răspuns:

a. Funcţia de *verosimilitate condiţională* a datelor de ieşire $Y = (Y_1, \ldots, Y_n)$ în raport cu datele de intrare $X = (X_1, \ldots, X_n)$ şi cu parametrul a este:

$$\begin{aligned} L(a) &\overset{def.}{=} p(Y_1, \ldots, Y_n|X_1, \ldots, X_n, a) \\ &\overset{i.i.d.}{=} \prod_{i=1}^n p(Y_i|X_i, a) = \prod_{i=1}^n \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2\right). \end{aligned}$$

Prin urmare,

$$a_{MLE} \overset{def.}{=} \arg\max_a L(a) = \arg\max_a \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2\right).$$

Aşadar, soluţia ii **propusă în enunţ este corectă.**

Mai departe, se poate scrie imediat o formă mai simplă pentru expresia funcţiei de verosimilitate:

$$\begin{aligned} L(a) &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \prod_{i=1}^{n} \exp\left(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2\right) \\ &= \frac{1}{(\sqrt{2\pi}\sigma)^n} \exp\left(-\sum_{i=1}^{n} \frac{1}{2\sigma^2}(Y_i - aX_i)^2\right). \end{aligned}$$

Factorul $\frac{1}{(\sqrt{2\pi}\sigma)^n}$ **este constant în raport cu parametrul** a **şi este de asemenea pozitiv, pentru că deviaţia standard** σ **este prin definiţie un număr pozitiv. În consecinţă,**

$$a_{MLE} \overset{def.}{=} \arg\max_a L(a) = \arg\max_a \prod_{i=1}^{n} \exp\left(-\frac{1}{2\sigma^2}(Y_i - aX_i)^2\right),$$

ceea ce înseamnă că şi soluţia iv **propusă în enunţ este corectă.**

Funcţia de log-verosimilitate a datelor are expresia:

$$\begin{aligned} \ell(a) &\overset{def.}{=} \ln L(a) = \ln\left(\frac{1}{(\sqrt{2\pi}\sigma)^n} \exp\left(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - aX_i)^2\right)\right) \\ &= -n\ln(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - aX_i)^2. \end{aligned}$$

Datorită faptului că funcţia $\ln$ **(ca şi orice funcţie** $\log_b$ **cu** $b > 1$**) este strict crescătoare, rezultă că la compunere de funcţii conservă / păstrează monotonia. În cazul nostru, această proprietate implică**

$$\begin{aligned} a_{MLE} &\overset{def.}{=} \arg\max_a L(a) = \arg\max_a \ell(a) \\ &= \arg\max_a \Big(-n\ln(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - aX_i)^2\Big). \end{aligned}$$

Termenul $-n\ln(\sqrt{2\pi}\sigma)$ **fiind constant în raport cu** a**, rezultă că**

$$\begin{aligned} a_{MLE} &= \arg\max_a \Big(-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(Y_i - aX_i)^2\Big) = \arg\max_a \Big(-\sum_{i=1}^{n}(Y_i - aX_i)^2\Big) \\ &= \arg\min_a \sum_{i=1}^{n}(Y_i - aX_i)^2. \end{aligned}$$

Penultima egalitate de mai sus are loc fiindcă factorul $\frac{1}{2\sigma^2}$ **este pozitiv şi constant în raport cu** a**. Ultima egalitate se explică prin faptul că la schimbarea**

de semn (adică, prin renunţarea la semnul – din faţa simbolului $\sum$), operatorul $\arg\max$ se transformă în $\arg\min$. Aşadar, soluţia *vi* din enunţ este, de asemenea, corectă. Celelalte soluţii propuse în enunţ (*i*, *iii* şi *v*) sunt incorecte.

b. Vom face calculele pornind de la expresia *vi* de la punctul precedent.

$$a_{MLE} = \arg\min_a \sum_{i=1}^n (Y_i - aX_i)^2 = \arg\min_a \left(a^2 \sum_{i=1}^n X_i^2 - 2a \sum_{i=1}^n X_i Y_i + \sum_{i=1}^n Y_i^2 \right).$$

Putem observa că expresia $a^2 \sum_{i=1}^n X_i^2 - 2a \sum_{i=1}^n X_i Y_i + \sum_{i=1}^n Y_i^2$ reprezintă o funcţie de gradul 2 în raport cu parametrul / variabila a, care are — în ipoteza că există măcar un $X_i \neq 0$ — coeficientul dominant pozitiv, deci va avea un (singur) minim. Abscisa acestui punct de minim este chiar a_{MLE}. Prin urmare,[310]

$$a_{MLE} = -\frac{-2 \sum_{i=1}^n X_i Y_i}{2 \sum_{i=1}^n X_i^2} = \frac{\sum_{i=1}^n X_i Y_i}{\sum_{i=1}^n X_i^2}.$$

c. Ţinând cont şi de faptul că

$$\arg\max_a \prod_i \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2} (Y_i - aX_i)^2 \right) = \arg\max_a \left(-\frac{1}{2} \sum_i (Y_i - aX_i)^2 \right),$$

problema de optimizare în sens MAP devine:

$$\begin{aligned} &\arg\max_a \left(-\frac{1}{2} \sum_{i=1}^n (Y_i - aX_i)^2 + \ln \frac{1}{\sqrt{2\pi}\lambda} - \frac{1}{2\lambda^2} a^2 \right) \\ &= \arg\max_a \left(-\frac{1}{2} \sum_{i=1}^n (Y_i - aX_i)^2 - \frac{1}{2\lambda^2} a^2 \right) = \arg\min_a \left(\sum_{i=1}^n (Y_i - aX_i)^2 + \frac{a^2}{\lambda^2} \right) \quad (153) \\ &= \arg\min_a \left(a^2 \left(\sum_{i=1}^n X_i^2 + \frac{1}{\lambda^2} \right) - 2a \sum_{i=1}^n X_i Y_i + \sum_{i=1}^n Y_i^2 \right). \end{aligned}$$

Ca şi la punctul *b*, observăm că funcţia de gradul al doilea căreia i se aplică operatorul $\arg\max$ are coeficientul dominant pozitiv, deci va avea un (singur) minim. Prin urmare, obţinem pentru a_{MAP} următorul rezultat:

$$a_{MAP} = \frac{\sum_{i=1}^n X_i Y_i}{\sum_{i=1}^n X_i^2 + \frac{1}{\lambda^2}}.$$

2. (Regresie liniară [cu anumite restricţii impuse]: exemplu de aplicare)

□ • ○ ***CMU, 2005 fall, T. Mitchell, A. Moore, midterm exam, pr. 3***

Presupunem că vrem să „antrenăm“ (engl., fit) un model de regresie liniară pe următoarele date:

x	-1	0	2
y	1	-1	1

[310] Se ştie că pentru funcţia de gradul al doilea $ax^2 + bx + c$, abscisa punctului de optim este $-\frac{b}{2a}$.

a. Antrenaţi modelul $Y_i = \beta_1 X_i + \varepsilon_i$ (reprezentând o regresie liniară unidimensională, dar fără termenul constant (β_0)); găsiţi valoarea optimă pentru parametrul β_1.[311]

b. Antrenaţi modelul $Y_i = \beta_0 + \varepsilon_i$ (reprezentând o regresie liniară „degenerată", fiindcă nu avem nicio variabilă de intrare); găsiţi valoarea optimă pentru parametrul β_0.

Răspuns:

a. Adaptând definiţia regresiei liniare pentru acest caz particular, vom scrie: $Y_i|X_i \sim \mathcal{N}(\beta_1 X_i, \sigma^2)$ pentru $i = 1, 2, 3$. Apoi, pornind de la expresia funcţiei de verosimilitate a datelor — în scrierea căreia vom folosi notaţia $Y = (Y_1, Y_2, Y_3)$ şi $X = (X_1, X_2, X_3)$ —, vom putea calcula $\hat{\beta}_1$, valoarea optimă a lui β_1, astfel:

$$\begin{aligned}
\hat{\beta}_1 &= \operatorname*{argmax}_{\beta_1} P(Y|X;\beta_1) \overset{i.i.d.}{=} \operatorname*{argmax}_{\beta_1} \prod_{i=1}^{3} P(Y_i|X_i;\beta_1) \\
&= \operatorname*{argmax}_{\beta_1} \prod_{i=1}^{3} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i-\beta_1 X_i)^2}{2\sigma^2}} = \operatorname*{argmax}_{\beta_1} \prod_{i=1}^{3} e^{-\frac{(Y_i-\beta_1 X_i)^2}{2\sigma^2}} \\
&= \operatorname*{argmax}_{\beta_1} \ln \prod_{i=1}^{3} e^{-\frac{(Y_i-\beta_1 X_i)^2}{2\sigma^2}} = \operatorname*{argmax}_{\beta_1} \sum_{i=1}^{3} -\frac{(Y_i-\beta_1 X_i)^2}{2\sigma^2} \\
&= \arg\min_{\beta_1} \sum_{i=1}^{3} (Y_i - \beta_1 X_i)^2 = \arg\min_{\beta_1} \left(\beta_1^2 \sum_{i=1}^{3} X_i^2 - 2\beta_1 \sum_{i=1}^{3} X_i Y_i + \sum_{i=1}^{3} Y_i^2 \right) \\
&= \frac{\sum_{i=1}^{3} X_i Y_i}{\sum_{i=1}^{3} X_i^2} = \frac{(-1)\cdot 1 + 0 \cdot (-1) + 2 \cdot 1}{(-1)^2 + 0^2 + 2^2} = \frac{1}{5}.
\end{aligned}$$

b. În acest caz, $Y_i|X_i \sim \mathcal{N}(\beta_0, \sigma^2)$ pentru $i = 1, 2, 3$. Similar raţionamentului de la punctul precedent, vom avea:

$$\begin{aligned}
\hat{\beta}_0 &= \operatorname*{argmax}_{\beta_0} P(Y|X;\beta_0) \overset{i.i.d.}{=} \operatorname*{argmax}_{\beta_0} \prod_{i=1}^{3} P(Y_i|X_i;\beta_0) \\
&= \operatorname*{argmax}_{\beta_0} \prod_{i=1}^{3} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i-\beta_0)^2}{2\sigma^2}} = \operatorname*{argmax}_{\beta_0} \prod_{i=1}^{3} e^{-\frac{(Y_i-\beta_0)^2}{2\sigma^2}} \\
&= \operatorname*{argmax}_{\beta_0} \ln \prod_{i=1}^{3} e^{-\frac{(Y_i-\beta_0)^2}{2\sigma^2}} = \operatorname*{argmax}_{\beta_0} \sum_{i=1}^{3} -\frac{(Y_i-\beta_0)^2}{2\sigma^2} \\
&= \arg\min_{\beta_0} \sum_{i=1}^{3} (Y_i - \beta_0)^2 = \arg\min_{\beta_0} \left(3\beta_0^2 - 2\beta_0 \sum_{i=1}^{3} Y_i + \sum_{i=1}^{3} Y_i^2 \right) \\
&= \frac{\sum_{i=1}^{3} Y_i}{3} = \frac{1 + (-1) + 1}{3} = \frac{1}{3}.
\end{aligned}$$

[311] Acest punct poate fi văzut ca o aplicaţie / exemplificare a modelului de regresie liniară de la problema 1.ab.

3. (Regresia liniară: prezentare generală, rezolvare folosind MLE; regresie neliniară [LC: polinomială]; regularizare de normă L_2 (*regresia ridge*), corespondenţa cu estimarea MAP)

prelucrare de Liviu Ciortuz, după
■ □ • ○ ***CMU, 2015 spring, Alex Smola, HW1, pr. 2***

***Introducere*:**
Obiectivul acestei probleme este acela de a ne furniza o privire mai generală[312] asupra regresiei liniare, mai întâi din perspectiva estimării de verosimilitate maximă (engl., Maximum Likelihood Estimation, MLE) a parametrilor, iar apoi din perspectiva estimării de probabilitate maximă a posteriori (engl., Maximum-a-Posteriori Estimation probability, MAP). Vom vedea că în cel de-al doilea caz în expresia funcţiei obiectiv apar şi nişte termeni de *regularizare*.[313]
În fiecare dintre cele două cazuri vom pune în evidenţă *i.* cum anume problema de *estimare* a parametrilor pentru respectiva problemă de regresie liniară conduce la [şi, chiar mai mult, este echivalentă cu] a rezolva o anumită problemă de *optimizare* scrisă sub formă matriceală şi *ii.* cum se calculează *soluţia analitică* a acelei probleme de optimizare.
Vom prezenta succint şi o variantă de regresie neliniară, şi anume regresia polinomială.

A. Regresia liniară: prezentare generală, bazată pe estimarea în sens MLE şi, echivalent, folosind *criteriul* LSE

Considerăm un *model liniar* care conţine o componentă de tip „zgomot“ (engl., noise) care urmează o distribuţie gaussiană:

$$Y_i = X_i \cdot w + b + \varepsilon_i \text{ cu } \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \text{ pentru } i = 1, \ldots, n, \tag{154}$$

unde $Y_i \in \mathbb{R}$ este „răspunsul“ pentru intrarea $X_i \in \mathbb{R}^d$ (acesta din urmă fiind văzut ca vector-linie), $b \in \mathbb{R}$ este un număr real (care în terminologia de limbă engleză se numeşte *bias*, iar în limba română *termen liber*), $w \in \mathbb{R}^d$ este tot un vector-linie, format din *ponderi* (engl., weights) care „acţionează“ asupra instanţelor X_i, iar ε_i sunt „zgomote“ i.i.d.[314] de tip gaussian, cu varianţa σ^2.

Având instanţele $\{X_i | i = 1, \ldots, n\}$, *scopul* nostru este să estimăm valorile parametrilor w şi b, care specifică / determină modelul de regresie.

Vom arăta că a rezolva modelul liniar (154) prin metoda MLE revine în mod echivalent la a rezolva următoarea problemă de tip *Least* [*Sum of*] *Squared Errors* (rom., *suma celor mai mici pătrate*):

$$\arg\min_{\beta} \underbrace{(Y - X'\beta)^\top (Y - X'\beta)}_{\|Y - X'\beta\|^2}, \tag{155}$$

unde notaţia $\| \ \|$ indică norma euclidiană (L_2),[315] $Y = (Y_1, \ldots, Y_n)^\top$, $X'_i = (1, X_i)^\top$, $X' = (X'_1, \ldots, X'_n)^\top$, iar $\beta = (b, w)^\top$.[316]

[312] La precedentele două probleme (1 şi 2) am lucrat cu o singură variabilă de intrare; aici vom lucra cu un număr arbitrar (d) de variabile de intrare (X_i, cu $i = 1, \ldots, d$).

[313] La problema 1, termenul de regularizare apare sub forma $\frac{a^2}{\lambda^2}$ în relaţia (153).

[314] Adică independente şi identic distribuite.

[315] Vă readucem aminte că dat fiind vectorul $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, norma sa de ordin k, cu $k \in \mathbb{N}$, se defineşte astfel: $\|x\|_k = \left(x_1^k + \ldots + x_d^k\right)^{1/k} = \sqrt[k]{x_1^k + \ldots + x_d^k}$. În general, atunci când nu se precizează indicele din scrierea lui $\| \ \|_k$, se consideră că este vorba despre valoarea $k = 2$, care corespunde normei euclidiene.

[316] Matricea X' se numeşte *matrice de design*. Linia i a acestei matrice este instanţa X_i extinsă cu elementul 1 pe prima poziţie, adică $(X'_i)^\top$.

a. Pornind de la modelul (154), calculaţi funcţia densitate de probabilitate (p.d.f.) condiţională a „răspunsului“ Y_i văzut ca variabilă aleatoare, în raport cu intrarea X_i şi cu parametrii w şi b. (Notaţie: $\Pr(Y_i|X_i, w, b)$.) Remarcaţi faptul că putem să-l privim pe X_i pur şi simplu ca pe o instanţă (punct fixat) din $\mathbb{R}^d$, nu ca pe o variabilă aleatoare.

b. Calculaţi în mod explicit expresia funcţiei de log-verosimilitate a datelor $\ell(\beta) \stackrel{def.}{=} \Pr(Y|X, \beta)$, unde $X \stackrel{not.}{=} (X_1, \ldots, X_n)$.

c. Demonstraţi că a găsi estimarea de tip MLE a parametrului β — adică, valoarea lui β pentru care funcţia de log-verosimilitate condiţională $\ell(\beta)$ îşi atinge maximul — este echivalent cu a rezolva problema de tip *Least Squared Errors* (155).

d. Deduceţi valoarea lui β care maximizează funcţia de log-verosimilitate condiţională. Puteţi presupune că *matricea de design* X', care a fost definită mai sus, are rangul maxim posibil (engl., full rank) în spaţiul determinat de vectorii-coloană din care este formată această matrice.

***Sugestie* (1): Puteţi, eventual, să folosiţi următoarele formule de calcul cu derivate vectoriale:**[317]

(5a) $\frac{\partial}{\partial X} a^\top X = \frac{\partial}{\partial X} X^\top a = a$

(5b) $\frac{\partial}{\partial X} X^\top A X = (A + A^\top) X$

B. Regresie neliniară [LC: polinomială]

e. Considerăm vectorul $\phi(X_i) = (1, X_i, X_i^2, \ldots, X_i^k)^\top$, cu $X_i \in \mathbb{R}$. În noul model de regresie pe care-l construim acum (şi pe care-l putem numi / considera un *model de ordin* k), vom exprima vectorul Y astfel:

$$Y_i = \phi(X_i) \cdot \beta' + \varepsilon_i \text{ cu } \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \text{ pentru } i = 1, \ldots, n, \tag{156}$$

unde $\beta' \in \mathbb{R}^{k+1}$.

Arătaţi că atunci când facem estimare în sens MLE şi presupunem că matricea $\phi(X) \stackrel{not.}{=} (\phi(X_1), \phi(X_2), \ldots, \phi(X_n))^\top$ are rangul maxim în spaţiul determinat de coloanele acestei matrice, valoarea optimă a parametrului β' din relaţia (156) este

$$(\phi(X)^\top \phi(X))^{-1} \phi(X)^\top Y.$$

***Sugestie* (2): Nu este nevoie să elaboraţi toţi paşii demonstraţiei, cum aţi procedat în secţiunea A. Concentraţi-vă pe schimbările care trebuie operate asupra expresiei funcţiei de log-verosimilitate, şi deduceţi noua formă a problemei de optimizare.**

C. Regresia liniară cu „zgomot“ gaussian şi regularizare de normă L_2 (*regresia ridge*); estimarea ponderilor în sensul MAP

Comentariu:[318] Multe probleme de regresie pot avea sute sau mii de atribute / variabile de intrare (numite şi *variabile de predicţie*). Dacă aceste atribute sunt corelate,

[317] Cf. *Matrix Identities*, Sam Roweis, 1999, http://www.cs.nyu.edu/∼roweis/notes/matrixid.pdf.
[318] Cf. CMU, 2008 fall, Eric Xing, HW1, pr. 4.2.

tehnicile standard de regresie (precum cele de la punctele A şi B) vor conduce în faza de antrenament la modele foarte complexe, precum şi la erori de generalizare mari în faza de testare / generalizare. O altă chestiune care poate să apară este faptul că unele probleme pot avea mai multe atribute decât numărul de instanţe din setul de date de antrenament. Atunci când se întâmplă aşa ceva, metodele regresie standard vor *eşua*. (Vom vedea mai jos de ce.) O metodă care ne permite să rezolvăm ambele chestiuni de mai sus este *regresia ridge*. *Ideea* pe care se bazează această metodă de regresie este una simplă: este bine să modificăm *funcţia de cost / pierdere* (engl., loss function), adăugând un *termen de penalizare* aplicat ponderilor (engl., weights), pentru a le determina să ia valori mici. Acest termen de penalizare este cunoscut ca *regularizator*; el controlează *complexitatea modelului* prin faptul că permite ponderi mari (în valoare absolută) doar pentru *cele mai importante atribute* din model. Regresia *ridge* penalizează pătratul lungimii / mărimii vectorului de ponderi (β). Aceasta este numită uneori *penalizare* L_2, fiindcă este pătratul normei L_2 (adică, norma euclidiană) aplicate vectorului de ponderi (β).

f. În situaţia în care matricea $X'^\top X'$ din secţiunea A (respectiv matricea $\phi(X)^\top \phi(X)$ din secţiunea B) nu este inversabilă,[319] se poate arăta că există $\lambda > 0$, astfel încât matricea $X'^\top X' + \lambda I$ — unde I este matricea identitate de dimensiune $d+1$ — să fie inversabilă.[320]

Arătaţi că

$$\hat{\beta}'' \overset{not.}{=} (X'^\top X' + \lambda I)^{-1} X'^\top Y$$

este soluţia problemei de optimizare

$$\arg\min_{\beta''} \left(\|Y - X'\beta''\|^2 + \lambda \|\beta''\|^2 \right). \tag{157}$$

g. Considerăm cazul în care β'' urmează o distribuţie a priori $\beta'' \sim \mathcal{N}(0, \eta^2 I)$.

Scrieţi distribuţia a posteriori a parametrului β'' în raport cu Y_i şi instanţa X_i, precum şi distribuţia a posteriori a lui β'' în raport cu Y şi întreg setul de date de antrenament, X. Veţi considera că β'' şi „zgomotele" $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, pentru $i = 1, \ldots, n$ sunt independente.

***Sugestie* (3): Folosiţi regula lui Bayes:** $\Pr(\beta''|Y_i, X_i) = \dfrac{\Pr(Y_i|X_i, \beta'')\Pr(\beta''|X_i)}{\Pr(Y_i|X_i)} = \dfrac{\Pr(Y_i|X_i, \beta'')\Pr(\beta'')}{\Pr(Y_i|X_i)}$**. Apoi urmaţi aceiaşi paşi ca în secţiunea A.**

***Sugestie* (4): Folosiţi faptul că prin corelarea a două sau mai multe variabile gaussiene unidimensionale independente se obţine o distribuţie (multidimensională) tot de tip gaussian. (Vedeţi problema 34 de la capitolul de *Fundamente*.)**

h. $\hat{\beta}''$, estimarea în sens MAP a parametrului β'', este definită ca fiind acea valoare a lui β'' pentru care se obţine *modul* [adică, valoarea maximă a] probabilităţii a posteriori $\Pr(\beta''|Y, X)$. Arătaţi că identificarea acestei estimări MAP

[319] Aceasta este situaţia când, spre exemplu, numărul trăsăturilor, $d+1$ din secţiunea A — respectiv $k+1$ din secţiunea B —, este mai mare decât numărul de instanţe, n (adică, $d+1 > n$). În acest caz, vom avea $\text{rang}(X'^\top X') = \text{rang}(X')$ (vedeţi documentul *Matrix identities*, de Sam Roweis, formula (2f)). Aşadar, $\text{rang}(X'^\top X')$ este mai mic sau egal cu $\min(n, d+1) = n$, care este mai mic decât $d+1$. Prin urmare, matricea $X'^\top X'$, care are dimensiunea $(d+1) \times (d+1)$, nu are rangul maxim şi în consecinţă nu poate fi inversată.

[320] Pentru a înţelege în mod intuitiv această chestiune, observaţi că dacă două coloane din matricea $X^\top X$ ar fi liniar dependente, atunci λI ar adăuga exact aceeaşi valoare (λ) la doar două dintre componente (diferite) ale acestor coloane. Aşadar, aceste coloane ar deveni liniar independente în matricea $X^\top X + \lambda I$.

conduce la problema de optimizare (157) în cazul în care λ poate fi exprimat în funcţie de σ şi η, adică $\lambda = h(\sigma, \eta)$. Găsiţi expresia funcţiei $h(\sigma, \eta)$.

i. Indicaţi o problemă care poate eventual să apară dacă am elimina termenul de regularizare din relaţia (157). Precizaţi cum anume poate termenul de regularizare să rezolve această potenţială problemă.

Răspuns:

a. Observaţi că $Y_i|X_i; w, b \sim \mathcal{N}(X_i \cdot w + b, \sigma^2)$, aşadar putem scrie funcţia densitate de probabilitate (p.d.f.) a lui $Y_i|X_i, w, b$ în forma următoare:

$$p(Y_i = y_i|X_i; w, b) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{(y_i - X_i \cdot w - b)^2}{2\sigma^2}\right).$$

b. Notând $y = (y_1, \ldots, y_n)^\top$, întrucât instanţele X_i sunt date, iar variabilele reprezentând „zgomotele" ε_i sunt i.i.d., funcţia de verosimilitate a lui Y în raport cu X şi β se scrie astfel:

$$\begin{aligned} L(\beta) &\overset{def.}{=} p(Y = y|X, \beta) = \prod_{i=1}^{n} p(y_i|X_i, w, b) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_i - X_i \cdot w - b)^2}{2\sigma^2}\right) \\ &= \left(\frac{1}{\sqrt{2\pi}\sigma}\right)^n \exp\left(-\frac{\sum_{i=1}^{n}(y_i - X_i \cdot w - b)^2}{2\sigma^2}\right). \end{aligned}$$

Apoi, aplicând logaritmul natural (ln), rezultă că funcţia de log-verosimilitate a lui $Y|\beta$ este următoarea:

$$\ell(\beta) \overset{def.}{=} \ln L(\beta) = -n \ln(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - X_i \cdot w - b)^2. \tag{158}$$

c. Pentru a calcula maximul funcţiei de log-verosimilitate $\ell(\beta)$, ne vom concentra asupra celui de-al doilea termen din expresia (158), fiindcă primul termen este constant în raport cu β. Observăm că pentru a maximiza cel de-al doilea termen din expresia (158), trebuie să minimizăm $\sum_{i=1}^{n}(y_i - X_i \cdot w - b)^2$. Scriind această sumă în notaţie matriceală / vectorială, obţinem:

$$\max_{\beta} \ell(\beta) = \min_{\beta} \sum_{i=1}^{n} (y_i - X_i \cdot w - b)^2 = \min_{\beta} (y - X'\beta)^\top (y - X'\beta), \tag{159}$$

unde, din nou, $Y = (Y_1, \ldots, Y_n)^\top$, $X_i' = (1, X_i)^\top, X' = (X_1', \ldots, X_n')^\top$, iar $\beta = (b, w)^\top$.

d. Definind funcţia obiectiv

$$J(\beta) \overset{def.}{=} (y - X'\beta)^\top (y - X'\beta), \tag{160}$$

şi aplicând regulile (5a) şi (5b) menţionate în *Sugestia* (1) din enunţ, rezultă:[321]

[321] Într-adevăr,

$$\begin{aligned} J(\beta) &= (y - X'\beta)^\top (y - X'\beta) = (y^\top - (X'\beta)^\top)(y - X'\beta) = (y^\top - \beta^\top X'^\top)(y - X'\beta) \\ &= y^\top y - y^\top X'\beta - \beta^\top X'^\top y + \beta^\top X'^\top X'\beta \\ \overset{(5a)}{\Rightarrow}\overset{(5b)}{} \nabla_\beta J(\beta) &= 0 - (y^\top X')^\top - X'^\top y + 2X'^\top X'\beta = -2X'^\top y + 2X'^\top X'\beta = 2X'^\top (X'\beta - y). \end{aligned}$$

$$\nabla_\beta J(\beta) = 2X'^\top(X'\beta - y).$$

Se poate demonstra relativ uşor că matricea hessiană corespunzătoare funcţiei $J(\beta)$ este pozitiv definită (vedeţi problema 7.a), deci această funcţie este convexă.

Valoarea $\hat\beta$ pentru care se obţine maximul funcţiei de log-verosimilitate poate fi găsită rezolvând următoarea ecuaţie:

$$\nabla_\beta J(\hat\beta) = 0 \Leftrightarrow X'^\top(X'\hat\beta - y) = 0 \Leftrightarrow X'^\top X'\hat\beta = X'^\top y \Leftrightarrow$$
$$\hat\beta = (X'^\top X')^{-1}X'^\top y. \tag{161}$$

Remarcaţi faptul că matricea $(X'^\top X')^{-1}$ există fiindcă s-a presupus că matricea X' are rangul maxim în spaţiul determinat de coloanele sale.[322] (Acest fapt, coroborat cu observaţia de mai sus referitoare la convexitatea funcţiei $J(\beta)$, implică faptul că punctul de maxim al lui J este unic.)

e. Întrucât s-a modificat relaţia dintre Y_i şi X_i, se vor modifica şi distribuţia condiţională (adică, p.d.f.-ul variabilei $Y_i|X_i, \beta'$) şi funcţia de verosimilitate. Noile lor expresii pot fi obţinute pur şi simplu înlocuind X_i cu $\phi(X_i)$, după cum urmează:

$$\begin{aligned} p(Y_i = y_i|X_i; w, b) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(\frac{(y_i - \phi(X_i)\cdot\beta')^2}{2\sigma^2}\right) \\ \ell(\beta) &= -n\ln(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \phi(X_i)\cdot\beta')^2. \end{aligned}$$

La fel cum am procedat mai înainte, putem aplica şi acum metoda estimării de verosimilitate maximă:

$$\max_{\beta'}\ell(\beta') = \min_{\beta'}\sum_{i=1}^{n}(y_i - \phi(X_i)\cdot\beta')^2 = \min_{\beta'}(y_i - \phi(X)\beta')^\top(y_i - \phi(X)\beta'),$$

iar estimatorul de verosimilitate maximă este

$$\hat\beta' = (\phi(X)^\top\phi(X))^{-1}\phi(X)^\top y. \tag{162}$$

f. Demonstraţia urmează aproape acelaşi curs ca la punctul d. Mai întâi, vom defini funcţia obiectiv J'' astfel:

$$\begin{aligned} J''(\hat\beta'') &= (y - X'\beta'')^\top(y - X'\beta'') + \lambda\underbrace{\|\beta''\|^2}_{\beta^\top\beta} \\ \overset{(5a),(5b)}{\Longrightarrow} \nabla_{\beta''}J''(\beta') &= 2X'^\top(X'\beta'' - y) + 2\lambda\beta''. \end{aligned}$$

[322] LC: Afirmaţia „matricea X' are rangul maxim în spaţiul determinat de coloanele sale“ trebuie interpretată astfel: numărul coloanelor din matricea X — matrice ale cărei linii sunt chiar instanţele X'_i — care sunt liniar independente este maxim, deci $d+1$ (dar şi $n \geq d+1$, unde n este numărul de instanţe de antrenament, deci şi numărul de linii din matricea X'). Folosind următoarea proprietate din documentul *Matrix Identities* de Sam Roweis

(2f) $\quad \text{rang}(A^\top A) = \text{rang}(AA^\top) = \text{rang}(A)$ pentru orice matrice A,

rezultă că rangul matricei $X'^\top X'$ este $d+1$. Prin urmare, matricea $X'^\top X'$ este inversabilă.

Similar cu observaţia făcută la punctul d, se poate arăta că şi în cazul funcţiei J'' matricea hessiană este pozitiv definită, deci J'' este funcţie convexă. Valoarea $\hat{\beta}''$ pentru care se obţine maximul funcţiei de log-verosimilitate poate fi găsită rezolvând următoarea ecuaţie:

$$\begin{aligned} & \nabla_{\beta''} J''(\hat{\beta}'') = 0 \\ \Leftrightarrow \quad & X'^{\top}(X'\hat{\beta}'' - y) + \lambda\hat{\beta}'' = 0 \\ \Leftrightarrow \quad & (X'^{\top}X' + \lambda I)\hat{\beta}'' = X'^{\top}y \\ \Leftrightarrow \quad & \hat{\beta}'' = (X'^{\top}X' + \lambda I)^{-1}X'^{\top}y. \end{aligned} \tag{163}$$

g. Ştim că $Y_i|X_i, \beta'' \sim \mathcal{N}(X_i \cdot \beta'', \sigma^2)$, iar $\beta'' \sim \mathcal{N}(0, \eta^2 I)$. Folosind regula lui Bayes, rezultă că

$$f(\beta''|Y_i, X_i) \propto p(Y_i|X_i, \beta'')\, g(\beta'') \propto \exp\left(-\frac{1}{2\sigma^2}(Y - X_i \cdot \beta'')^2\right) \cdot \exp\left(-\frac{1}{2\eta^2}\beta''^{\top}\beta''\right),$$

unde semnul $\propto$ înseamnă 'proporţional cu', $g(\cdot)$ este funcţia de densitate de probabilitate (p.d.f.) pentru β'', iar $f(\cdot)$ este funcţia de densitate de probabilitate pentru $\beta''|Y_i, X_i$. Factorul de normalizare pentru p.d.f.-ul f este definit ca $Z_i = \int_{-\infty}^{\infty} f(Y_i|X_i, \beta'')g(\beta'')d\beta''$, aşadar putem rescrie p.d.f.-ul $f(\beta''|Y_i, X_i)$ astfel:

$$f(\beta''|Y_i, X_i) = \frac{1}{Z_i}\exp\left(-\frac{1}{2\sigma^2}(Y - X_i \cdot \beta'')^2 - \frac{1}{2\eta^2}\beta''^{\top}\beta''\right).$$

În mod similar, ţinând cont de *Sugestia* (4) din enunţ, rezultă că p.d.f.-ul lui $\beta''|Y, X$ este

$$\begin{aligned} f(\beta''|Y) &= \frac{1}{Z}\exp\left(-\frac{1}{2\sigma^2}(Y - X'\beta'')^{\top}(Y - X'\beta'') - \frac{1}{2\eta^2}\beta''^{\top}\beta''\right) \\ &= \frac{1}{Z}\exp\left(-\frac{1}{2\sigma^2}\|Y - X'\beta''\|^2 - \frac{1}{2\eta^2}\|\beta''\|^2\right), \end{aligned}$$

unde Z este factorul de normalizare, definit ca $Z = \int_{-\infty}^{\infty} f(Y|X, \beta'')g(\beta'')d\beta''$.

h. De la punctul g este clar că

$$\max_{\beta''} f(\beta''|Y_i) = \min_{\beta''}\Big(\frac{1}{2\sigma^2}\|Y - X\beta''\|^2 + \frac{1}{2\eta^2}\|\beta''\|^2\Big). \tag{164}$$

Dacă η^2 devine $\dfrac{\sigma^2}{\lambda}$, atunci problema de minimizare devine echivalentă cu problema (157),

$$\arg\min_{\beta''}\big(\|Y - X\beta''\|^2 + \lambda\|\beta''\|^2\big).$$

Altfel spus, prin stabilirea corespondenţei termenilor, suntem conduşi la a lua $\lambda = h(\sigma, \eta) = \dfrac{\sigma^2}{\eta^2}$, ceea ce forţează relaţia (164) să devină (157).

i. Termenul de regularizare penalizează componentele din β care au valori mari; în consecinţă, β va avea valori mici (în normă). Prin urmare, termenul de regularizare „încurajează" modelul să evite *overfitting*-ul, împiedicându-l astfel să se adecveze la instanţele care sunt *outlier*-e (altfel, acestea ar influenţa în mod drastic valoarea lui β).

4. **(O proprietate a regresiei liniare, varianta LSE: la rezolvarea cu ajutorul *formulelor analitice*, scalarea atributelor nu schimbă predicţiile obţinute pentru instanţele de test)**

■ □ • ○ ***MIT, 2001 fall, Tommi Jaakkola, HW1, pr. 1.6***

Considerăm un set de exemple de antrenament $D = \{(x_i, y_i)|i = 1, \ldots, n\}$ cu $x_i \in \mathbb{R}^d$ şi $y_i \in \mathbb{R}$, precum şi o instanţă de test $x_{test} \in \mathbb{R}^d$ şi, în plus, factorii de scalare $\alpha_1, \ldots, \alpha_d \in \mathbb{R}^*$.[323] Demonstraţi următoarea egalitate:

$$predicted\ y_{test} = predicted\ \tilde{y}_{test},$$

unde

predicted y_{test} **este predicţia pentru outputul sau *valoarea de răspuns*[324] pentru instanţa x_{test} după ce am „antrenat“ / produs un model de regresie liniară de tip LSE pe setul D folosind *soluţia analitică*;[325]**

predicted $\tilde{y}_{test}$ **este predicţia obţinută în mod similar, antrenând un model de regresie LSE pe setul de date $D' = \{(\tilde{x}_i, y_i)|i = 1, \ldots, n\}$, cu $\tilde{x}_{i,j} = \alpha_j\, x_{i,j}$ şi facând apoi predicţie pentru $\tilde{x}_{test}$, unde $\tilde{x}_{\text{test},j} = \alpha_j\, x_{test,j}$, pentru $j \in \{1, \ldots, d\}$.**

***Observaţie*: O proprietate similară este valabilă şi în cazul metodei lui Newton; vedeţi problema 173 de la capitolul de *Fundamente*. Rezultatul respectiv este însă mai general; el nu este limitat la folosirea metodei lui Newton pentru rezolvarea problemelor de regresie [liniară].**

Răspuns:

Putem să formalizăm relaţia dintre noua *matrice de design* ($\tilde{X}$) şi cea veche (X) folosind o *matrice de transformare*, $A \in \mathbb{R}^{d\times d}$, pe a cărei diagonală sunt plasaţi coeficienţii de scalare α_j (şi care are celelalte elemente 0):

$$\tilde{X} = XA.$$

Utilizând această relaţie în „ecuaţia normală“ care ne dă valoarea optimă a vectorului de ponderi β, vom obţine:[326]

$$\begin{aligned}
\tilde{\beta} &= (\tilde{X}^\top \tilde{X})^{-1} \tilde{X}^\top y \\
&= ((XA)^\top XA)^{-1} (XA)^\top y \\
&= (A^\top X^\top XA)^{-1} A^\top X^\top y \\
&= A^{-1} (X^\top X)^{-1} \underbrace{(A^\top)^{-1} A^\top}_{I} X^\top y \\
&= A^{-1} (X^\top X)^{-1} X^\top y \\
&= A^{-1}\, \beta.
\end{aligned}$$

[323] Pentru atributele pe care nu le scalăm vom considera în mod implicit $\alpha_j = 1$.
[324] Vedeţi problema 5, unde pentru această noţiune am folosit notaţia $\hat{y}$.
[325] Adică, „ecuaţia normală“ reprezentată de relaţia (161) de la problema 3.d.
[326] În continuare, X şi y se referă la datele de antrenament.

Aşadar, outputul care va fi prezis este următorul:

$$\begin{aligned} \textit{predicted}\,\tilde{y}_{test} &= \tilde{X}_{test}\,\tilde{\beta} \\ &= (X_{test}\,A)(A^{-1}\beta) = X_{test}\,(AA^{-1})\,\beta \\ &= X_{test}\,\beta \\ &= \textit{predicted}\,y_{test}. \end{aligned}$$

***Observaţie*:**
La acelaşi rezultat se poate ajunge făcând *o altă demonstraţie*, pe care doar o schiţăm:
Întrucât instanţele $\tilde{x}_i$ sunt obţinute prin scalarea atributelor instanţelor x_i, mulţimea de transformări liniare asupra [atributelor] unei instanţe oarecare $x \in \mathbb{R}^d$ este exact aceeaşi cu mulţimea de transformări liniare asupra [atributelor] instanţei scalate $\tilde{x}$. *Predictor*-ul, în ambele cazuri, este [vectorul de ponderi β care determină] funcţia liniară din această mulţime care minimizează eroarea la antrenare.
[LC:] Se poate demonstra uşor că valorile optime ale funcţiilor obiectiv în cele două cazuri sunt identice. Aceasta implică faptul că pătratele erorilor (de forma $\|y_i - x_i\beta\|^2$ şi respectiv $\|y_i - \tilde{x}_i\tilde{\beta}'\|^2$) sunt identice pentru orice $i = 1, \ldots, n$. Ca să arătăm că avem şi $x_i\beta = \tilde{x}_i\tilde{\beta}'$ pentru $i = 1, \ldots, n$ (ceea ce este adevărat dacă $\tilde{\beta}' = A^{-1}\,\beta$), observăm mai întâi că cele două funcţii obiectiv sunt convexe, fiindcă matricele lor hessiene sunt pozitiv definite; vedeţi problema 7.a), iar apoi că cele două matrice hessiene sunt chiar inversabile, deoarece conform enunţului putem scrie ecuaţiile „normale“. Aşadar, punctele de minim pentru cele două funcţii obiectiv sunt unice. Rezultă cu necesitate că relaţia între cei doi predictori este cea de mai sus, $\tilde{\beta}' = A^{-1}\,\beta$.
În concluzie, cei doi predictori vor produce rezultate identice şi atunci când sunt aplicaţi pe instanţele de test X_{test} şi respectiv $\tilde{X}_{test}$.

5. **(Regresia liniară, varianta LSE: bias-ul şi matricea de covarianţă a estimatorului; Regresia *ridge*: bias-ul)**

■ □ • ***CMU, (?) spring, (10-701 course), HW1, pr. 3ab+4a***

După cum ştiţi, modelul regresiei liniare are forma

$$y = x\beta + \varepsilon,$$

unde $x = (x_1, \ldots, x_d)^\top$, $\beta = (\beta_1, \ldots, \beta_d)^\top$, $y \in \mathbb{R}$, iar ε este un „zgomot“, pe care aici îl vom considera de tip gaussian, având $E(\varepsilon) = 0$ şi $Var(\varepsilon) = \sigma^2$. Dat fiind setul de date de antrenament $(x_1, y_1), \ldots, (x_n, y_n)$, în care fiecare instanţă x_i este un vector de forma $(x_{i1}, \ldots, x_{id})^\top \in \mathbb{R}^d$, iar $y_i \in \mathbb{R}$, vrem să estimăm parametrii β. Notăm cu X matricea de dimensiune $n \times d$ în care linia i este vectorul $x_i^\top$; în mod similar, vectorul-coloană Y, cu n-componente, este format din ieşirile / „răspunsurile“ corespunzătoare instanţelor din setul de antrenament.

***Remember*:** Am arătat la problema 3.A că a minimiza suma pătratelor erorilor — vă reamintim că în notaţia matriceală această sumă se scrie $\|Y - X\beta\|^2$ — revine la a calcula următoarea *estimare* a parametrului β:

$$\hat{\beta} = (X^\top X)^{-1} X^\top Y. \tag{165}$$

Rezultă imediat că vectorul de *predicţii* corespunzător lui Y este calculat astfel:

$$\hat{y} = X\hat{\beta} = X(X^\top X)^{-1}X^\top Y.$$

Remarcaţi faptul că vectorul Y este o variabilă aleatoare (n-ară), fiindcă pentru orice instanţă x_i, valoarea outputului y_i este afectată de „zgomotul" aleatoriu ε_i. Într-adevăr, vectorul $\hat{\beta}$ se calculează în funcţie de vectorul Y deci şi el ($\hat{\beta}$) este o cantitate aleatoare.

a. Demonstraţi că $\hat{\beta}$, estimatorul regresiei liniare în varianta LSE, este *nedeplasat* (engl., unbiased), adică $E(\hat{\beta}) = \beta$.

b. Demostraţi că matricea de covarianţă a lui $\hat{\beta}$ este egală cu $\sigma^2(X^\top X)^{-1}$.

c. *Remember*: La problema 3.C am arătat că putem obţine estimatorul *ridge*, o variantă a estimatorului de tip LSE pentru regresia liniară, impunând un factor de penalizare asupra mărimii vectorului format din coeficienţii de regresie ($\hat{\beta}$):

$$\|X\beta - Y\|^2 + \lambda\beta^\top\beta,$$

unde parametrul λ controlează contribuţia / efectul termenului de regularizare. Minimizarea acestei funcţii de cost ne conduce la următorul estimator al vectorului de coeficienţi ai regresiei liniare regularizate:

$$\hat{\beta} = (X^\top X + \lambda I)^{-1}X^\top Y, \tag{166}$$

care în statistică este cunoscut sub numele de *estimator ridge*.[327] Predicţia de tip *ridge* a lui Y este [dată de formula]

$$\hat{y} = X\hat{\beta} = X(X^\top X + \lambda I)^{-1}X^\top Y.$$

Demonstraţi că estimatorul *ridge* este deplasat (engl., biased).

Răspuns:

a. Pornind de la relaţia (165)**, putem scrie:**

$$\hat{\beta} \;=\; (X^\top X)^{-1}X^\top Y = (X^\top X)^{-1}X^\top(X\beta + \varepsilon) = \beta + (X^\top X)^{-1}X^\top\varepsilon \tag{167}$$

Apoi, aplicând proprietatea de liniaritate a mediei, vom obţine:

$$E[\hat{\beta}] = E[\beta + (X^\top X)^{-1}X^\top\varepsilon] = \beta + (X^\top X)^{-1}X^\top \underbrace{E[\varepsilon]}_{0} = \beta.$$

Aşadar, estimatorul $\hat{\beta}$ este nedeplasat.

b. Vom demonstra mai întâi că pentru orice vector de variabile aleatoare $V = (V_1, \ldots, V_d)$, matricea sa de covarianţă, care este notată cu $Cov[V]$ şi are ca element generic $Cov(V_i, V_j) \overset{def.}{=} E[(V_i - E[V_i])(V_j - E[V_j])]$, satisface egalitatea $Cov[V] = E[VV^\top] - E[V](E[V]^\top)$:

$$\begin{aligned}
Cov[V] &\overset{def.}{=} [Cov(V_i, V_j)]_{i,j\in\{1,\ldots,d\}} \\
&\overset{def.}{=} [E[(V_i - E[\beta_i])(V_j - E[V_j])]]_{i,j\in\{1,\ldots,d\}} \\
&= [E[V_iV_j] - E[\beta_i]E[V_j]]_{i,j\in\{1,\ldots,d\}} \\
&= E[VV^\top] - E[V](E[V]^\top).
\end{aligned}$$

[327] Remarcaţi faptul că atât estimatorul LSE cât şi estimatorul *ridge* sunt *estimatori liniari*. Prin definiţie, un estimator $\tilde{\beta}$ este liniar dacă el poate fi scris ca o combinaţie liniară de vectorul de ieşirile Y, adică $\tilde{\beta} = KY$, unde K este o matrice oarecare de dimensiune $d \times n$.

***Observaţie*: Această egalitate este o generalizare a egalităţii** $Cov\,(X,Y) = E[XY] - E[X]E[Y]$**, unde** X **şi** Y **sunt variabile aleatoare oarecare; vedeţi problema 9.c de la capitolul de *Fundamente*.**

Înlocuind V **cu** $\hat{\beta}$ **în relaţia pe care am demonstrat-o mai sus, obţinem**

$$Cov[\hat{\beta}] = E[\hat{\beta}\hat{\beta}^\top] - E[\hat{\beta}](E[\hat{\beta}]^\top). \tag{168}$$

În continuare, făcând uz de relaţia (167)**, în membrul drept al egalităţii** (168) **îl vom înlocui pe** $\hat{\beta}$ **cu** $\beta + (X^\top X)^{-1}X^\top\varepsilon$ **şi apoi vom folosi rezultatul** $E[\hat{\beta}] = \beta$ **demonstrat la punctul** a**. Aşadar,**

$$\begin{aligned}
Cov[\hat{\beta}] &= E[(\beta + (X^\top X)^{-1}X^\top\varepsilon)(\beta + (X^\top X)^{-1}X^\top\varepsilon)^\top] - \beta\beta^\top \\
&= E[(\beta + (X^\top X)^{-1}X^\top\varepsilon)(\beta^\top + \varepsilon^\top X((X^\top X)^{-1})^\top)] - \beta\beta^\top \\
&= E[(\beta + (X^\top X)^{-1}X^\top\varepsilon)(\beta^\top + \varepsilon^\top X(\underbrace{(X^\top X)^\top}_{X^\top X})^{-1})] - \beta\beta^\top \\
&\overset{lin.\ med.}{=} \underbrace{E[\beta\beta^\top]}_{\beta\beta^\top} + E[\beta\varepsilon^\top X(X^\top X)^{-1} + (X^\top X)^{-1}X^\top\varepsilon\beta^\top] + \\
&\quad E[(X^\top X)^{-1}X^\top\varepsilon\,\varepsilon^\top X(X^\top X)^{-1}] - \cancel{\beta\beta^\top} \\
&\overset{lin.\ med.}{=} \beta\underbrace{E[\varepsilon]}_{0}^\top X(X^\top X)^{-1} + (X^\top X)^{-1}X^\top\underbrace{E[\varepsilon]}_{0}\beta^\top + \\
&\quad E[(X^\top X)^{-1}X^\top\varepsilon\,\varepsilon^\top X(X^\top X)^{-1}] \\
&= E[(X^\top X)^{-1}X^\top\varepsilon\,\varepsilon^\top X(X^\top X)^{-1}] \\
&\overset{lin.\ med.}{=} (X^\top X)^{-1}X^\top\underbrace{E[\varepsilon\,\varepsilon^\top]}_{\sigma^2 I_{n\times n}}X(X^\top X)^{-1} \\
&= \sigma^2(X^\top X)^{-1}(X^\top X)(X^\top X)^{-1} \\
&= \sigma^2(X^\top X)^{-1}.
\end{aligned}$$

c. Pentru a arăta că estimatorul $\hat{\beta}$ **al regresiei *ridge* este deplasat, va trebui să demonstrăm că** $E[\hat{\beta}] \neq \beta$ **sau, alternativ, că** $E[\hat{\beta}] - \beta$ **este diferit de vectorul nul. Într-adevăr, folosind relaţia** (166)**, putem scrie:**

$$\begin{aligned}
E[\hat{\beta}] - \beta &= E[(X^\top X + \lambda I)^{-1}X^\top y] - \beta \\
&= E[(X^\top X + \lambda I)^{-1}X^\top(X\beta + \varepsilon)] - \beta \\
&\overset{lin.\ med.}{=} (X^\top X + \lambda I)^{-1}X^\top X\beta + (X^\top X + \lambda I)^{-1}X^\top\underbrace{E[\varepsilon]}_{0} - \beta \\
&= (X^\top X + \lambda I)^{-1}X^\top X\beta - \beta \\
&= (X^\top X + \lambda I)^{-1}(X^\top X + \lambda I - \lambda I)\beta - \beta \\
&\overset{distrib.}{=} (X^\top X + \lambda I)^{-1}(X^\top X + \lambda I)\beta - (X^\top X + \lambda I)^{-1}\lambda I\beta - \beta \\
&= \beta - \lambda(X^\top X + \lambda I)^{-1}\beta - \beta \\
&= -\lambda(X^\top X + \lambda I)^{-1}\beta \\
&\neq 0.
\end{aligned}$$

6. **(Regresia liniară cu regularizare L_2 (Regresia *ridge*): rezolvare cu metoda gradientului, varianta "batch" şi varianta stochastică)**

■ □ • ○ ***CMU, 2008 fall, Eric Xing, HW1, pr. 4.2.2***

Fie $D = \{(x_i, y_i)|i = 1, \ldots, n\}$ un set de n exemple de antrenament pentru o problemă de regresie liniară. Fie $y_i \in \mathbb{R}$ eticheta (sau, variabila de răspuns) pentru exemplul i, iar $y \in \mathbb{R}^n$ vectorul-coloană alcătuit din toate variabilele de răspuns. Fiecare exemplu de antrenament are d atribute (sau, variable de intrare / predicţie). Fie $x_i \in \mathbb{R}^{d\times 1}$ vectorul-coloană format din toate variabilele de predicţie pentru exemplul i, iar $X \in \mathbb{R}^{n\times d}$ matricea de design, formată din toate variabilele de predicţie, linia i a matricei X fiind $x_i^\top$. Fie $\beta \in \mathbb{R}^{d\times 1}$ un vector-coloană care conţine *ponderile* / *parameterii* pe care vrem să le / îi „învăţăm".

***Comentariu*:** La problema 3.C am arătat (folosind calculul matriceal şi derivatele vectoriale) că *soluţia analitică* a problemei

$$\arg\min_{\beta} \left[\|y - X\beta\|^2 + \lambda\|\beta\|^2\right]$$

este

$$\hat{\beta}_{ridge} = (X^\top X + \lambda I)^{-1} X^\top y.$$

Deşi această *soluţie analitică* este uşor de implementat, ea poate deveni total nepractică atunci când numărul atributelor este foarte mare, fiindcă trebuie să calculăm atât produsul $X^\top X$ cât şi inversa acestei matrice. O tehnică / soluţie pe care o putem folosi pentru a depăşi acest neajuns este metoda *gradientului descendent* (engl., gradient descent), care este o metodă de optimizare, iterativă.

a. La acest punct vom lucra cu varianta "batch" (sau "steepest descent") a metodei gradientului descendent. La fiecare iteraţie se îmbunătăţesc ponderile β, aplicând următoarea *regulă de actualizare*:

$$\beta^{(t)} = \beta^{(t-1)} - \eta \nabla_\beta \ell(\beta^{(t-1)}),$$

unde constanta η este aşa-numita *rată de învăţare* (engl., learning rate), iar $\nabla_\beta \ell(\beta)$ este transpusul vectorului gradient pentru funcţia de cost / pierdere $\ell(\beta)$.

Deduceţi regula de actualizare pentru metoda gradientului, varianta "batch", ştiind că expresia funcţiei de log-verosimilitate $\ell(\beta)$ este:

$$\ell(\beta) = \frac{1}{2}\sum_{i=1}^{n}(y_i - x_i^\top \beta)^2 + \frac{\lambda}{2}\beta^\top \beta.$$

***Sugestie*: Puteţi folosi următoarele formule de calcul pentru derivare vectorială (din documentul *Matrix Identities*, de Sam Roweis, 1999):**[328]

$$(5a)\ \frac{\partial}{\partial X} a^\top X = \frac{\partial}{\partial X} X^\top a = a \qquad (5b)\ \frac{\partial}{\partial X} X^\top A X = (A + A^\top)X.$$

b. Varianta secvenţială / stochastică a metodei gradientului descendent actualizează parametrii β la procesarea fiecărui exemplu. Acest fapt este extrem de folositor în cazul seturilor de date de antrenament foarte mari, precum şi atunci când exemplele sunt furnizate în manieră secvenţială, printr-un

[328] http://www.cs.nyu.edu/~roweis/notes/matrixid.pdf.

"stream" de date. În acest caz, *regula de actualizare* a poderilor β este următoarea:

$$\beta^{(t)} = \beta^{(t-1)} - \eta \nabla_\beta \ell_i(\beta^{(t-1)}; x_i, y_i),$$

unde η este rata de învăţare (o constantă), iar $\ell_i(\beta; x_i, y_i)$ este transpusul vectorului gradient pentru funcţia de cost / pierdere calculată pentru exemplul particular i.

Calculaţi regula de actualizare pentru varianta stochastică a metodei gradientului descendent, ştiind că $\ell_i(\beta; x_i, y_i)$ este

$$\ell_i(\beta; x_i, y_i) = (y_i - x_i^\top \beta)^2 + \frac{\lambda}{2}\beta^\top \beta.$$

Răspuns:

a. Mai întâi vom aplica unele transformări elementare asupra funcţiei de log-verosimilitate ℓ (în aşa fel încât să putem aplica ulterior regulile de derivare vectorială):

$$\begin{aligned}\ell(\beta) &= \frac{1}{2}\sum_{i=1}^n (y_i - x_i^\top \beta)^2 + \frac{\lambda}{2}\beta^\top \beta = \frac{1}{2}\sum_{i=1}^n (y_i^2 - 2y_i x_i^\top \beta + \underbrace{(x_i^\top \beta)^2}_{(x_i^\top \beta)^\top (x_i^\top \beta)}) + \frac{\lambda}{2}\beta^\top \beta \\ &= \frac{1}{2}\sum_{i=1}^n (y_i^2 - 2y_i x_i^\top \beta + \beta^\top x_i x_i^\top \beta) + \frac{\lambda}{2}\beta^\top \beta.\end{aligned}$$

Acum vom calcula gradientul funcţiei ℓ, aplicând regulile de derivare vectorială:

$$\nabla_\beta \frac{\partial \ell(\beta)}{\partial \beta} \overset{(5a)\,(5b)}{=} \frac{1}{2}\sum_{i=1}^n (-2y_i x_i + 2x_i x_i^\top \beta) + \lambda\beta = \sum_{i=1}^n [-y_i x_i + x_i (x_i^\top \beta)] + \lambda\beta. \quad (169)$$

Aşadar, pentru regresia *ridge*, metoda gradientului foloseşte în varianta "batch" următoarea regulă de actualizare:

$$\beta^{(t)} = \beta^{(t-1)} - \eta\Big[\sum_{i=1}^n [-y_i x_i + x_i (x_i^\top \beta^{(t-1)})] + \lambda\beta^{(t-1)}\Big].$$

b. Procedând similar cu rezolvarea de la punctul precedent, vom ajunge la următorul rezultat:

$$\beta^{(t)} = \beta^{(t-1)} - \eta\big[-y_i x_i + x_i (x_i^\top \beta^{(t-1)}) + \lambda\beta^{(t-1)}\big].$$

7. (Regresia liniară, varianta LSE: rezolvare cu metoda lui Newton)

■ □ • ○ ***Stanford, 2007 fall, Andrew Ng, HW1, pr. 1***

***Introducere*:** La problema 3 am rezolvat problemele de regresie prezentate acolo folosind *metoda analitică* (şi obţinând aşa-numitele *ecuaţii normale*, (161), (162), (163)). În schimb, aici (şi la problema 6) vom folosi *metode de optimizare iterativă*: metoda lui Newton şi respectiv metoda gradientului.

În această problemă vom demonstra că la folosirea *metodei lui Newton*[329] pentru rezolvarea problemei de regresie liniară în forma *sumei celor mai mici pătrate* (engl., least squared error, LSE), avem nevoie de o singură iteraţie pentru a ajunge la convergenţă (şi, deci, pentru a obţine soluţia $\hat{\beta}$).

a. Calculaţi *matricea hessiană*[330] pentru funcţia de cost / pierdere

$$J(\beta) = \frac{1}{2} \sum_{i=1}^{n} (\beta^\top x^{(i)} - y^{(i)})^2.$$

***Observaţie*: Funcţia de cost $J(\beta)$ coincide (până la factorul $1/2$) cu funcţia obiectiv din problema de optimizare (155) de la problema 3.A.**[331]

b. Arătaţi că indiferent de valoarea $\beta^{(0)}$ pe care o atribuim iniţial vectorului β, la prima iteraţie a metodei lui Newton se obţine vectorul $\beta^{(1)} = (X^\top X)^{-1} X^\top y = \hat{\beta}$, adică exact soluţia problemei de regresie liniară de tipul (sau, în varianta) *suma celor mai mici pătrate*.

Răspuns:

a. Calculăm mai întâi derivatele parţiale ale funcţiei $J(\beta)$ în raport cu β_j, pentru $j = 1, \ldots, d$, unde d este numărul de atribute din instanţele $x^{(i)}$:

$$\frac{\partial J(\beta)}{\partial \beta_j} = \sum_{i=1}^{n} (\beta^\top x^{(i)} - y^{(i)})\, x_j^{(i)}.$$

Rezultă că derivatele parţiale de ordin secund ale lui $J(\beta)$ sunt de forma următoare:

$$\frac{\partial^2 J(\beta)}{\partial \beta_j \, \partial \beta_k} = \sum_{i=1}^{n} \frac{\partial}{\partial \beta_k} (\beta^\top x^{(i)} - y^{(i)})\, x_j^{(i)} = \sum_{i=1}^{n} x_j^{(i)} x_k^{(i)} = (X^\top X)_{jk}.$$

Aşadar, matricea hessiană pentru funcţia $J(\beta)$ este $X^\top X$.[332]

***Observaţie*: Se poate demonstra imediat că matricea $X^\top X$ este pozitiv definită: pentru orice $z \in \mathbb{R}^d$ avem $z^\top (X^\top X) z = (Xz)^\top (Xz) = (Xz)^2 \geq 0$. În consecinţă, J este funcţie convexă, deci admite [măcar un punct de] minim. În anumite condiţii — de exemplu, atunci când $X^\top X$ este matrice inversabilă, aşa cum s-a presupus la problema 3.A —, punctul respectiv de minim este unic.**

b. Pornind de la o valoarea arbitrară $\beta^{(0)}$ pentru vectorul de ponderi β, metoda lui Newton calculează $\beta^{(1)}$ conform formulei

$$\beta^{(1)} = \beta^{(0)} - H^{-1} \nabla_\beta J(\beta^{(0)}).$$

[329] Pentru o introducere la metoda lui Newton, vedeţi *Comentariul* din enunţul problemei 80 de la capitolul de *Fundamente*.

[330] Denumirea de matrice hessiană este derivată de la numele matematicianului german Ludwig Otto Hesse (1811 - 1874), care l-a avut ca profesor pe un alt matematician german vestit, Carl Gustav Jacob Jacobi (1804 - 1851). Prin definiţie, matricea hessiană a unei funcţii — pentru care există derivatele parţiale de ordinul întâi şi al doilea — este o matrice pătratică formată din derivatele parţiale de ordin secund ale funcţiei respective.

[331] În ce priveşte notaţiile din aceste două probleme, avem corespondenţele următoare: $x^{(i)} \to X_i'$, $y^{(i)} \to Y_i$ şi $X \to X'$.

[332] Se poate ajunge la acest rezultat şi aplicând pur şi simplu regulile de derivare vectorială [care au fost învăţate la cursul de algebră liniară].

Se poate arăta relativ uşor că $\nabla_\beta J(\beta) = X^\top X\beta - X^\top y$.[333] Prin urmare,

$$\begin{aligned}\beta^{(1)} &= \beta^{(0)} - (X^\top X)^{-1}(X^\top X\beta^{(0)} - X^\top y)\\ &= \beta^{(0)} - \beta^{(0)} + (X^\top X)^{-1}X^\top y\\ &= (X^\top X)^{-1}X^\top y.\end{aligned}$$

În consecinţă, indiferent de valoarea $\beta^{(0)}$ pe care o atribuim iniţial vectorului β, metoda lui Newton obţine soluţia $\hat{\beta}$ (adică, optimul funcţiei obiectiv $J(\beta)$) după o singură iteraţie.

8. ([Alte] două variante ale regresiei liniare: regresia liniară local-ponderată, cu „zgomot“ gaussian; regresia liniară cu „zgomot“ modelat cu distribuţia Laplace)

■ □ • ○ ***CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, HW2, pr. 3.1-2***
CMU, 2007 fall, Carlos Guestrin, HW1, pr. 2.2

***Remember*:** Pentru regresia liniară, se dă un set de date de antrenament de forma $D = (X, y) = \{(x_i, y_i)|i = 1, 2, \ldots, n\}$, unde $x_i \in \mathbb{R}^{d\times 1}$, adică $x_i = (x_{i,1}, \cdots, x_{i,d})^\top$, $y_i \in \mathbb{R}$, $X \in \mathbb{R}^{n\times d}$, linia i a matricei X fiind $x_i^\top$ şi, în sfârşit, $y = (y_1, \cdots, y_n)^\top$. Folosind un model [parametrizat] de forma $y_i = x_i\beta + \varepsilon_i$, unde ε_i sunt „zgomote“ generate de o anumită distribuţie probabilistă, metoda regresiei liniare caută să afle acea valoare a vectorului de parametri β care asigură cea mai bună „potrivire“ / adecvare (engl., *fit*) a modelului de mai sus pe datele de antrenament D. O modalitate (sau, un *criteriu*) de a măsura / evalua această „potrivire“ / adecvare este să găsim acel β care minimizează o funcţie de cost / pierdere (engl., *loss function*) dată, $J(\beta)$. La problema 3.A am introdus modelul regresiei lineare în varianta *sumei pătratelor erorilor* (engl., least squared errors, LSE), iar la secţiunile B şi C din aceeaşi problemă am prezentat două extensii / variante ale acestui model: regresia polinomială şi regresia *ridge*.

În această problemă vom explora alte două extensii / variante ale modelului de regresie LSE: *regresia local-ponderată*, în cazul căreia asociem fiecărei instanţe x_i un coeficient w_i care reprezintă ponderea / importanţa respectivei instanţe, precum şi un model de regresie în care „zgomotele“ ε_i sunt modelate cu distribuţia Laplace.

A. Regresie liniară *local-ponderată*, cu „zgomot“ gaussian[334]

Presupunem că „zgomotele“ $\varepsilon_1, \ldots, \varepsilon_n$ sunt [tot de tip gaussian, tot] independente, dar acum varianţele nu mai sunt identice, deci $\varepsilon_i \sim \mathcal{N}(0, \sigma_i^2)$, pentru $i = 1, \ldots, n$.

a. Stabiliţi formula de calcul pentru estimarea de verosimilitate maximă (MLE) a vectorului de parametri β în această variantă de regresie.

[333] Cunoscând $\frac{\partial J(\beta)}{\partial \beta_j}$ de la punctul a, se poate scrie vectorul gradient $\nabla_\beta J(\beta) \stackrel{def.}{=} \left(\frac{\partial J(\beta)}{\partial \beta_1}, \ldots, \frac{\partial J(\beta)}{\partial \beta_d}\right)$ astfel:

$$\nabla_\beta J(\beta) = \sum_{i=1}^{n}(\beta^\top x^{(i)} - y^{(i)})x^{(i)} = \sum_{i=1}^{n}(\beta^\top x^{(i)})x^{(i)} - \sum_{i=1}^{n} y^{(i)}x^{(i)} = X^\top X\beta - X^\top y.$$

Pentru a justifica ultima egalitate, este suficient să scrieţi componentele generice ale celor două matrice, $X^\top X\beta$ şi $X^\top y$.

[334] Acest model de regresie corespunde sumei ponderate a celor mai mici pătrate (engl., locally-weighted least squares).

***Sugestie*: Următoarele formule (din documentul *Matrix Identities*, de Sam Roweis, 1999)[335] vă pot fi de fi de folos:**

$$(5a)\ \frac{\partial}{\partial X} a^\top X = \frac{\partial}{\partial X} X^\top a = a \qquad (5b)\ \frac{\partial}{\partial X} X^\top A X = (A + A^\top) X.$$

b. Arătaţi că estimarea în sens MLE pe care aţi calculat-o la punctul a coincide cu valoarea parametrului β pentru care se obţine minimul unei funcţii de cost / pierdere de forma următoare:[336]

$$J_W(\beta) = \sum_i w_i (y_i - x_i^\top \beta)^2.$$

Exprimaţi ponderea w_i în funcţie de σ_i^2, varianţa „zgomotului“ ε_i din exemplul de antrenament i.

c. Explicaţi de ce acest nou estimator — suma ponderată a celor mai mici pătrate — este [uneori] preferată în raport cu versiunea neponderată.

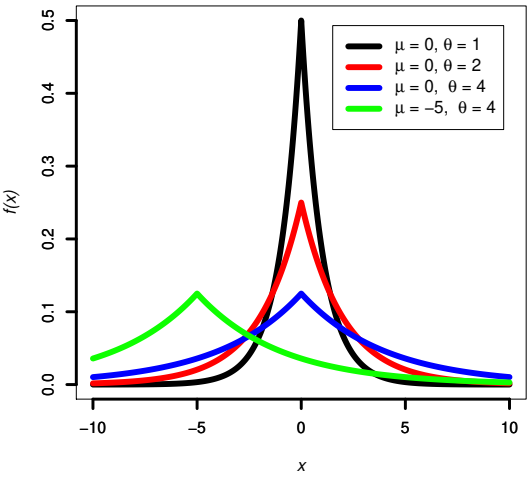
***Sugestie*: Analizaţi cazul / cazurile în care σ_i^2 are o valoare mare, comparativ cu cazul / cazurile în care valoarea sa este mică.**

B. Regresie liniară [neponderată], cu „zgomot“ de tip *Laplace*

Presupunem că „zgomotele“ $\varepsilon_1, \ldots, \varepsilon_n$ sunt distribuite în mod identic şi independent, conform unei distribuţii de tip Laplace de parametri $\mu = 0$ şi $\theta > 0$. Vă readucem aminte că funcţia de densitate de probabilitate (p.d.f.) pentru această distribuţie este

$$\text{Laplace}(x; \mu, \theta) \overset{def.}{=} \frac{1}{2\theta} \exp\left(-\frac{|x-\mu|}{\theta}\right).$$

Distribuţia Laplace: p.d.f.



d. Găsiţi funcţia de cost / pierdere $J_{Laplace}(\beta)$ a cărei minimizare este echivalentă cu găsirea estimării în sens MLE pentru parametrul β în cazul acestui tip de modelare a „zgomotului“.

e. Care credeţi că este avantajul acestui model, comparativ cu varianta standard (cea în care modelararea „zgomotului“ se face folosind o distribuţie gaussiană)?

***Sugestie*: Gândiţi-vă la excepţii / anomalii (engl., outliers).**

[335] http://www.cs.nyu.edu/∼roweis/notes/matrixid.pdf.

[336] Această funcţie se numeşte *suma ponderată a celor mai mici pătrate*; engl., weighted least squares loss function.

Răspuns:

a. Ştim că $y_i = x_i^\top \beta + \varepsilon_i$, iar $p(y_i|x_i;\beta) = \mathcal{N}(x_i^\top\beta, \sigma_i^2)$. Aşadar, formula care ne dă estimarea în sens MLE pentru parametrul β este:

$$\begin{aligned}
\beta_{MLE} &= \underset{\beta}{\operatorname{argmax}} \ln \prod_i p(y_i|x_i;\beta) = \underset{\beta}{\operatorname{argmax}} \sum_i \ln p(y_i|x_i;\beta) \\
&= \underset{\beta}{\operatorname{argmax}} \sum_i \ln\left(\frac{1}{\sqrt{2\pi}\sigma_i}\exp\left(-\frac{(y_i - x_i^\top\beta)^2}{2\sigma_i^2}\right)\right) \\
&= \underset{\beta}{\operatorname{argmax}} \sum_i \left(\ln\frac{1}{\sqrt{2\pi}\sigma_i} - \frac{(y_i - x_i^\top\beta)^2}{2\sigma_i^2}\right) = \underset{\beta}{\operatorname{argmax}} \sum_i -\frac{(y_i - x_i^\top\beta)^2}{2\sigma_i^2} \\
&= \arg\min_\beta \sum_i \frac{(y_i - x_i^\top\beta)^2}{2\sigma_i^2} \\
&= \arg\min_\beta \sum_i \frac{1}{\sigma_i^2}(y_i - x_i^\top\beta)^2. \qquad (170)
\end{aligned}$$

Vom scrie acum expresia (170) în notaţie matriceală. Considerând W matricea diagonală de dimensiune $n \times n$, în care $w_{ii} = \frac{1}{\sigma_i^2}$ pentru $i = 1, \ldots, n$, vom obţine:

$$\beta_{MLE} = \arg\min_\beta (y - X\beta)^\top W (y - X\beta). \qquad (171)$$

Ca şi în cazul regresiei liniare neponderate în varianta LSE, se poate demonstrea relativ uşor că matricea hessiană corespunzătoare funcţiei de cost $J_W(\beta) = (y - X\beta)^\top W(y - X\beta)$ este *pozitiv definită*, deci această funcţie este *convexă* şi prin urmare admite punct de minim.

Valoarea efectivă a estimatorului β_{MLE} este soluţia ecuaţiei care se obţine egalând gradientul expresiei $(y - X\beta)^\top W(y - X\beta)$ cu vectorul 0:

$$\begin{aligned}
0 &= \frac{\partial}{\partial\beta}\left((y - X\beta)^\top W(y - X\beta)\right) \\
&\overset{distrib.}{=} \frac{\partial}{\partial\beta}\Big(y^\top Wy - \underbrace{y^\top WX\beta}_{\in\mathbb{R}} - \underbrace{\beta^\top X^\top Wy}_{(y^\top W^\top X\beta)^\top} + \beta^\top X^\top WX\beta\Big). \qquad (172)
\end{aligned}$$

Pentru orice număr real z are loc egalitatea $z^\top = z$, aşadar $((\beta^\top X^\top)Wy)^\top = y^\top W^\top X\beta = y^\top WX\beta$, întrucât $W^\top = W$. (Vă reamintim că matricea W este diagonală.) Ţinând cont de aceasta, ecuaţia (172) va fi rescrisă astfel:

$$\begin{aligned}
0 = \frac{\partial}{\partial\beta}\left(y^\top Wy - 2\beta^\top X^\top Wy + \beta^\top X^\top WX\beta\right) &\overset{(5a)(5b)}{\Leftrightarrow} 0 = -2X^\top Wy + 2X^\top WX\beta \\
&\Leftrightarrow X^\top Wy = X^\top WX\beta.
\end{aligned}$$

În cazul în care matricea $X^\top WX$ este inversabilă, vom avea:

$$\beta_{MLE} = (X^\top WX)^{-1}X^\top Wy. \qquad (173)$$

Aceasta este ecuaţia „normală“ corespunzătoare regresiei LSE local-ponderate.

b. La relaţia (170) am demonstrat că estimatorul MLE al parametrului β este valoarea pentru care se atinge minimul pentru suma ponderată a celor

mai mici pătrate, $\sum_i \frac{1}{\sigma_i^2}(y_i - x_i^\top \beta)^2$. Coroborând această expresie cu relaţia $J_W(\beta) \stackrel{def.}{=} \sum_i w_i (y_i - x_i^\top \beta)^2$ din enunţ, obţinem $w_i = \frac{1}{\sigma_i^2}$.

c. Atunci când varianţa σ_i^2 este mare, „zgomotul" ε_i poate fi oricât de mare, deci punctul (x_i, y_i) poate fi un outlier. Nu este de dorit ca în astfel de cazuri estimatorul β_{MLE} să fie deplasat (engl., biased) în aşa fel încât el să „explice" aceste outlier-e (mai ales atunci când se foloseşte ca funcţie de cost / pierdere suma pătratelor erorilor). Modelul regresiei liniare local-ponderate formulat în problema de faţă rezolvă această chestiune ponderând „importanţa" (sau, „contribuţia") fiecărui exemplu de antrenament la funcţia obiectiv (J_W) cu ajutorul unui factor care este tocmai inversul varianţei corespunzătoare exemplului respectiv. În consecinţă, instanţele care au o varianţă mare nu vor afecta mult funcţia de cost / pierdere; ele pot fi ignorate, sau [cel puţin] li se poate acorda o [mai] mică importanţă atunci când se caută valoarea optimă a lui β.

d. Ştim că $y_i = x_i^\top \beta + \varepsilon_i$, iar $p(y_i|x_i;\beta) = \text{Laplace}(x_i^\top \beta; 0, \theta)$. Aşadar, formula care ne dă estimarea în sens MLE pentru parametrul β în acest caz este:

$$\begin{aligned}
\beta_{MLE} &= \underset{\beta}{\text{argmax}} \ln \prod_i p(y_i|x_i;\beta) = \underset{\beta}{\text{argmax}} \sum_i \ln p(y_i|x_i;\beta) \\
&= \underset{\beta}{\text{argmax}} \sum_i \ln \left(\frac{1}{2\theta} \exp\left(-\frac{|y_i - x_i^\top \beta|}{\theta} \right) \right) \\
&= \underset{\beta}{\text{argmax}} \sum_i \left(\ln \frac{1}{2\theta} - \frac{|y_i - x_i^\top \beta|}{\theta} \right) = \underset{\beta}{\text{argmax}} \sum_i -\frac{|y_i - x_i^\top \beta|}{\theta} \\
&\stackrel{\theta \geq 0}{=} \arg\min_\beta \sum_i |y_i - x_i^\top \beta|.
\end{aligned}$$

În concluzie,

$$J_{Laplace}(\beta) = \sum_i |y_i - x_i^\top \beta|. \tag{174}$$

e. Dacă o instanţă de antrenament este "outlier", eroarea produsă atunci când se face predicţia corespunzătoare acestei instanţe — folosind valoarea corectă pentru parametrul β — este mult mai mare în cazul în care se lucrează cu „zgomot" de tip gaussian (pentru că atunci se foloseşte pătratul diferenţei dintre valoarea "target" şi valoarea prezisă de model) decât în cazul zgomotului de tip laplacian (fiindcă în acest caz se lucrează cu modulul diferenţei respective). Prin urmare, outlier-ele afectează estimarea lui β mult mai mult în cazul „zgomotului" gaussian decât în cazul celui laplacian.

Din punctul de vedere al modelării, ţinând cont că $y_i = x_i^\top \beta + \varepsilon_i$, atunci când y_i este outlier, modelul poate „explica" acest fapt dând lui ε_i o valoare mare (în modul), pentru a se putea „accepta" diferenţa dintre valoarea "target" şi valoarea prezisă pentru outlier-ul respectiv. Acest fapt este posibil în cazul modelului Laplace, fiindcă funcţia densitate de probabilitate (p.d.f.) a distribuţiei Laplace are braţele (engl., tails) mai ridicate decât p.d.f.-ul distribuţiei gaussiene.

***Observaţie*: Făcând legătura cu secţiunea A a problemei de faţă, putem spune că pentru a obţine un efect similar [cu cazul „zgomotului" Laplace], acolo am presupus că fiecărei instanţe îi asociem o varianţă proprie, σ_i^2. Remarcaţi totuşi că aceste varianţe trebuie estimate (folosind, de exemplu, un algoritm de tip EM — vedeţi capitolul *Schema algoritmică EM*) fiindcă ele influenţează problema de optimizare, în vreme ce în modelul Laplace nu trebuie să facem astfel de estimări. Pe de altă parte, este mai dificil să facem optimizare cu o funcţie de cost / pierdere care foloseşte norma L_1 (cazul distribuţiei Laplace) decât atunci când se foloseşte norma L_2 (cazul distribuţiei gaussiene), fiindcă funcţia modul nu este derivabilă pe tot domeniul ei de definiţie.**

9. **(Regresia liniară cu regularizare L_2 (Regresia *ridge*): kernel-izare)**

□ • ○ ***CMU, 2014 spring, B. Poczos, A. Singh, HW2, pr. 4.B***

În această problemă vom arăta cum anume poate fi obţinută varianta kernel-izată a regresiei *ridge*.[337] După cum se procedează în general la regresia liniară, vom lucra cu un set de instanţe / date $\{x_i\}_{i=1}^n$ din $\mathbb{R}^d$ şi cu valorile de „răspuns" asociate lor, $\{y_i\}_{i=1}^n$ din $\mathbb{R}$.

Pentru a obţine rezultate mai bune pentru problema noastră de regresie, putem folosi o funcţie de „mapare" a atributelor (engl., attribute mapping function) ϕ, care asociază fiecărui vector d-dimensional x câte un nou vector, $\bar{x} = \phi(x)$ din $\mathbb{R}^m$, numit vector de *trăsături* (engl., feature vector), unde m este mai mare decât d.[338] Considerăm matricea de design $\Phi \in \mathbb{R}^{n \times m}$, în care linia i este vectorul de trăsături $\bar{x}_i^\top$ corespunzător instanţei i. Notăm cu y vectorul-coloană format din valorile de răspuns; în acest vector, componenta de pe poziţia i este valoarea de răspuns y_i, care a fost asociată instanţei i.

Aşa cum am arătat la problema 3.C, pentru regresia *ridge* funcţia obiectiv este

$$J(\beta) = \|\mathbf{y} - \Phi\beta\|^2 + \lambda\|\beta\|^2, \tag{175}$$

unde $\lambda > 0$ este parametrul de *regularizare*. Vom nota cu $\hat{\beta}$ soluţia / estimatorul acestei variante de regresie *ridge*. În cele ce urmează vom arăta cum anume se poate calcula această soluţie.

a. Mai întâi, arătaţi că $\hat{\beta}$, care este un vector din $\mathbb{R}^m$, este în *spaţiul* generat de liniile din matricea Φ. Aceasta înseamnă că $\hat{\beta}$ poate fi scris ca o combinaţie liniară de vectorii-linie care alcătuiesc matricea Φ. Altfel spus, există vectorul $\hat{\alpha} \in \mathbb{R}^{n \times 1}$ astfel încât $\hat{\beta}$ se poate scrie sub forma

$$\hat{\beta} = \Phi^\top \hat{\alpha}. \tag{176}$$

Arătaţi apoi că, în ipoteza că matricele $\Phi\Phi^\top$ şi $\Phi\Phi^\top + \lambda I$ sunt inversabile, urmează că vectorul de coeficienţi $\hat{\alpha}$ se poate exprima cu ajutorul matricei de design Φ astfel:[339]

$$\hat{\alpha} = (\Phi\Phi^\top + \lambda I)^{-1} y. \tag{177}$$

[337] Pentru o introducere în subiectul funcţiilor-nucleu, vedeţi secţiunea corespunzătoare din capitolul de *Fundamente*. Similar, pentru regresia *ridge*, vedeţi problema 3.C.

[338] În general, m este mult mai mare decât d; vom nota acest fapt prin $m \gg d$.

[339] Pentru regresia liniară nekernel-izată, am abţinut deja o astfel de relaţie la problema 3.f.

***Sugestie*: Vectorul de ponderi β din expresia (175) — ca şi orice alt vector din $\mathbb{R}^m$ — poate fi descompus în două componente, şi anume $\beta = \beta_{\|} + \beta_{\perp}$, unde componenta $\beta_{\|}$ aparţine spaţiului generat de liniile din Φ, iar componenta $\beta_{\perp}$ este ortogonală pe acest spaţiu.[340] Această ultimă proprietate înseamnă că produsul scalar dintre orice vector din spaţiul generat de liniile din Φ şi $\beta_{\perp}$ este 0.**

b. În practică, este foarte dificil să construim funcţii de „mapare" ϕ. Dar chiar şi atunci când putem să construim astfel de funcţii, calcularea produselor $\phi(x)^\top \phi(x')$ — care sunt necesare pentru aflarea vectorului $\hat{\alpha}$, conform relaţiei (177) — poate fi ineficientă dacă se foloseşte maniera clasică de calcul.[341] În schimb, putem calcula în mod *implicit* aceste produse, folosind o funcţie-nucleu $k(x, x')$ care are proprietatea $k(x, x') = \phi(x)^\top \phi(x')$. În principiu, a *kernel*-iza înseamnă să folosim „apeluri" la o funcţie-nucleu pentru a înlocui produsele $\phi(x)^\top \phi(x')$.

La acest punct, vă cerem să kernel-izaţi regresia *ridge*, presupunând că vi se dă o funcţie-nucleu $k(x, x')$, unde x şi x' sunt vectorii originali, de dimensiune d, care corespund instanţelor de antrenament. Va trebui să lucraţi atât asupra antrenării cât şi asupra testării. Mai exact, va trebui ca pentru faza de antrenare să înlocuiţi cu „apeluri" la funcţia-nucleu k toate produsele scalare care apar în calculul necesar obţinerii vectorului $\hat{\alpha}$ (vedeţi relaţia (177)). De asemenea, pentru faza de testare, dată fiind o instanţă de test x, va trebui să înlocuiţi toate produsele scalare care apar în calculul lui $\phi(x)^\top \hat{\beta} = \phi(x)^\top (\Phi^\top \hat{\alpha}) = (\phi(x)^\top \Phi^\top)\hat{\alpha}$ cu „apeluri" la funcţia-nucleu k.[342]

Răspuns:

a. Ţinând cont de *Sugestia* din enunţ, vom putea scrie astfel funcţia obiectiv a regresiei *ridge* kernel-izate:

$$\begin{aligned} J(\beta) &= \|y - \Phi\hat{\beta}\|^2 + \lambda\|\hat{\beta}\|^2 = \\ J(\beta_{\|} + \beta_{\perp}) &= \|y - \underbrace{\Phi\beta_{\perp}}_{0} - \Phi\beta_{\|}\|^2 + \lambda\|\beta_{\perp}\|^2 + \lambda\|\beta_{\|}\|^2 \\ &= \|y - \Phi\beta_{\|}\|^2 + \lambda\|\beta_{\perp}\|^2 + \lambda\|\beta_{\|}\|^2. \end{aligned}$$

Minimizarea acestei funcţii obiectiv va trebui să fie făcută în raport cu cele două componente, $\beta_{\perp}$ şi $\beta_{\|}$, care sunt ortogonale una în raport cu cealaltă. Din expresia lui J rezultă că, pentru minimizare, componenta $\beta_{\perp}$ trebuie să fie vectorul $0 \in \mathbb{R}^d$. În consecinţă, $\hat{\beta} = \hat{\beta}_{\|}$. (Şi $J(\beta) = J(\hat{\beta}_{\|})$.) Rezultă, deci, că vectorul β aparţine spaţiului generat de liniile din matricea Φ, ceea ce înseamnă că există într-adevăr vectorul $\hat{\alpha} \in \mathbb{R}^{n\times 1}$ astfel încât $\hat{\beta}$ să se poată scrie sub forma $\hat{\beta} = \Phi^\top \hat{\alpha}$.

***Observaţie*: Dacă analizaţi problema 88 de la capitolul de *Fundamente*, unde este prezentată *Teorema de reprezentare*, veţi putea identifica o altă cale de**

[340] Pentru explicaţii mai detaliate, puteţi vedea documentul *Supplemental Lecture Notes* (pentru cursul de *Machine Learning*), de John Duchi, de la Universitatea Stanford, mai precis secţiunea *A more general representer theorem*, pag. 11-12.

[341] Observaţi că putem rescrie în mod echivalent produsul $\phi(x)^\top \phi(x')$ cu ajutorul produsului scalar al vectorilor: $\phi(x) \cdot \phi(x')$.

[342] Veţi observa că, din cauza acestei ultime expresii, la faza de antrenare este suficient să calculăm vectorul $\hat{\alpha}$, nu şi vectorul $\hat{\beta} = \Phi^\top \hat{\alpha}$.

demonstrare a faptului că $\hat{\beta}$ se poată scrie sub forma $\hat{\beta} = \Phi^\top \hat{\alpha}$, şi anume, calculând şi apoi egalând cu 0 derivata funcţiei $J(\beta)$, din relaţia (175).[343]

Acum, pentru a calcula efectiv vectorul de coeficienţi $\hat{\alpha}$, vom folosi relaţia $\hat{\beta} = \Phi^\top \hat{\alpha}$ şi-l vom înlocui pe $\hat{\beta}$ cu $\Phi^\top \hat{\alpha}$ în expresia lui $J(\hat{\beta})$:

$$J(\hat{\beta}) = J(\hat{\beta}_{||}) = \|y - \Phi\hat{\beta}_{||}\|^2 + \lambda\|\hat{\beta}_{||}\|^2 = \|y - \Phi\Phi^\top\hat{\alpha}\|^2 + \lambda\hat{\alpha}^\top\Phi\Phi^\top\hat{\alpha}. \tag{178}$$

Întrucât $\hat{\beta}$ minimizează funcţia $J(\beta)$, putem calcula $\hat{\alpha}$ egalând cu vectorul $0 \in \mathbb{R}^n$ derivata în raport cu $\hat{\alpha}$ a expresiei-rezultat din relaţia (178). Aşadar, vom avea:[344]

$$-2\Phi\Phi^\top(y - \Phi\Phi^\top\hat{\alpha}) + 2\lambda\Phi\Phi^\top\hat{\alpha} = 0 \Leftrightarrow \Phi\Phi^\top(\Phi\Phi^\top + \lambda I)\hat{\alpha} = \Phi\Phi^\top y,$$

ceea ce implică, în ipoteza că matricele $\Phi\Phi^\top$ şi $\Phi\Phi^\top + \lambda I$ sunt inversabile, faptul că

$$\hat{\alpha} = (\Phi\Phi^\top + \lambda I)^{-1} y.$$

b. La faza de antrenare, trebuie să calculăm vectorul $\hat{\alpha}$. Dacă notăm cu K matricea ale cărei elemente sunt $K_{ij} = k(x_i, x_j)$ rezultă că aceasta este chiar matricea $\Phi\Phi^\top$. Aşadar, conform relaţiei (177), rezultă că $\hat{\alpha} = (K + \lambda I)^{-1} y$. (Matricea $K = \Phi\Phi^\top$ se numeşte *matrice-nucleu* sau *matrice Gram*.)

La faza de testare va trebui să calculăm

$$\phi(x)^\top\hat{\beta} = \phi(x)^\top(\Phi^\top\hat{\alpha}) = (\phi(x)^\top\Phi^\top)\hat{\alpha} = \sum_{i=1}^{n}\hat{\alpha}_i k(x_i, x).$$

10. **(Regresia *ridge* kernel-izată, cu nucleu RBF; un rezultat interesant: atunci când parametrul de regularizare λ tinde la 0, eroarea la antrenare devine 0, dacă instanţele x_i sunt mutual distincte)**

prelucrare de Liviu Ciortuz, după
□ • ○ ***MIT, 2006 fall, Tommi Jaakkola, HW2, pr. 2.a-c***

La problema 9 am prezentat versiunea kernel-izată a regresiei *ridge*. Vă reamintim că regresie *ridge* înseamnă regresia liniară cu „zgomot" gaussian şi regularizare (L_2). Regula de predicţie pentru acest tip de regresie kernel-izată este

$$y(x) = \sum_{i=1}^{n}\hat{\alpha}_i K(x_i, x), \tag{179}$$

unde $\hat{\alpha}_i$ este valoarea optimală a coeficientului α_i, pentru $i = 1, \ldots, n$. Valorile $\hat{\alpha}_i$ au fost obţinute astfel:

$$\hat{\alpha} = (\mathbf{K} + \lambda\mathbf{I})^{-1} y. \tag{180}$$

[343] Exact aşa se procedează în problema de la Stanford, Andrew Ng, 2007 fall, HW2, pr. 1.b.

[344] Pentru derivarea efectivă, putem folosi mai întâi descompunerea

$$\begin{aligned}\|y - \Phi\Phi^\top\hat{\alpha}\|^2 &\overset{def.}{=} (y - \Phi\Phi^\top\hat{\alpha})^\top(y - \Phi\Phi^\top\hat{\alpha}) = (y^\top - \hat{\alpha}^\top\Phi\Phi^\top)(y - \Phi\Phi^\top\hat{\alpha}) \\ &= y^\top y - \hat{\alpha}^\top\Phi\Phi^\top y - y^\top\Phi\Phi^\top\hat{\alpha} + \hat{\alpha}^\top(\Phi\Phi^\top)(\Phi\Phi^\top)\hat{\alpha} \\ &= y^\top y - 2\hat{\alpha}^\top\Phi\Phi^\top y + \hat{\alpha}^\top(\Phi\Phi^\top)(\Phi\Phi^\top)\hat{\alpha}\end{aligned}$$

şi apoi putem aplica regulile de derivare vectorială (5a) şi (5b) care au fost folosite şi la problema 6.

În această formulă, $\hat{\alpha} \overset{not.}{=} (\hat{\alpha}_1, \ldots, \hat{\alpha}_n)^\top$, $y \overset{not.}{=} (y_1, \ldots, y_n)^\top$, iar K este aşa-numita *matrice-nucleu* (sau, *matricea Gram*) corespunzătoare instanţelor $x_1, \ldots, x_n$.[345]

În acest exerciţiu vrem să studiem evoluţia erorii la antrenare pentru acest model de regresie liniară kernel-izată în condiţiile următoare:
1. folosim ca nucleu aşa-numita funcţie cu baza radială (engl., radial basis function, RBF), care este definită astfel:

$$K(x, x') = \exp\left(-\frac{\gamma}{2}\|x - x'\|^2\right),\ \gamma > 0 \text{ pentru orice } x, x' \in \mathbb{R}^d;$$

2. valorile parametrului de regularizare λ tind la 0.[346]

a. Arătaţi că, în ipoteza $x_i \neq x_j$ pentru orice $i \neq j$, matricea Gram corespunzătoare funcţiei-nucleu RBF este inversabilă (indiferent de valoarea parametrului γ).[347]

***Sugestie*: Puteţi folosi *teorema lui Michelli* (1986): dacă o funcţie ρ de variabilă t este monotonă pentru $t \in [0, \infty)$, atunci pentru orice set de instanţe distincte $x_1, \ldots, x_n$, matricea pătratică definită prin relaţia $\rho_{ij} = \rho(\|x_i - x_j\|)$ pentru $i, j \in \{1, \ldots, n\}$ este inversabilă.**

b. Care este valoarea funcţiei $y(x)$ atunci când $\lambda \to 0$? Cu alte cuvinte, care este formula de calcul pentru predicţiile făcute de către modelul de regresie *ridge* kernel-izată atunci când se face această trecere la limită?

c. Presupunem, ca mai înainte, că $x_i \neq x_j$ pentru orice $i \neq j$. Demonstraţi că eroarea la antrenare (calculată ca sumă a pătratelor „erorilor" $\lim_{\lambda \to 0} y(x_t) - y_t$) produsă la limită de către regresia *ridge* kernel-izată cu nucleu RBF este zero.

Răspuns:

a. Funcţia $f(t) = -\gamma t^2/2$ este monoton descrescătoare în raport cu t (pentru orice $\gamma > 0$ fixat). De asemenea, funcţia $g(t) = e^t$ este monoton crescătoare în raport cu t. Prin compunere, funcţia $(g \circ f)(t) \overset{not.}{=} h(t) = e^{-\gamma t^2/2}$ va fi monoton descrescătoare în raport cu t.
Funcţia-nucleu RBF, care este definită prin relaţia

$$K(x, x') = e^{-\frac{\gamma}{2}\|x - x'\|^2} = (g \circ f)(\|x - x'\|), \text{ pentru orice } x, x' \in \mathbb{R}^d,$$

satisface condiţiile teoremei lui Michelli. Prin urmare, ea va da naştere, pentru orice set de instanţe $x_1, \ldots, x_n \in \mathbb{R}^d$ care sunt diferite între ele, la o matrice Gram care este inversabilă.

b. Calculăm mai întâi valoarea, la limită, a vectorului de parametri $\hat{\alpha}$:

$$\begin{aligned} \lim_{\lambda \to 0} \hat{\alpha} &= \lim_{\lambda \to 0} \Big((\mathbf{K} + \lambda \mathbf{I})^{-1} y\Big) = \Big(\lim_{\lambda \to 0} \big((\mathbf{K} + \lambda \mathbf{I})^{-1}\big)\Big) y \\ &= \Big[\lim_{\lambda \to 0} (\mathbf{K} + \lambda \mathbf{I})\Big]^{-1} y = K^{-1} y. \end{aligned}$$

[345] *Observaţie*: Spre deosebire de problema 9, aici am notat matricea Gram (care acolo era desemnată cu K) cu simbolul $\mathbf{K}$, iar funcţia-nucleu (care acolo era $k(\,.\,,\,.\,)$) cu $K(\,.\,,\,.\,)$.

[346] Facem aceasta, deşi a seta $\lambda = 0$ — ceea ce înseamnă a lucra fără regularizare — nu este deloc convenabil, din cauza overfitting-ului.

[347] *Justificare*: Trecerea la limită pentru $\lambda \to 0$ în relaţia (179) necesită calcularea valorii pe care o are vectorul de coeficienţi $\hat{\alpha}$ la limită. Din cauza relaţiei (180), întrucât $\lim_{\lambda \to 0}(\mathbf{K} + \lambda \mathbf{I}) = \mathbf{K}$, trebuie ca [în prealabil] să ne asigurăm că matricea Gram $\mathbf{K}$ este inversabilă.

Această limită este [întotdeauna] bine-definită, fiindcă

– matricea $\mathbf{K}$ este inversabilă (vedeţi punctul a),

– matricea $\mathbf{K}+\lambda\mathbf{I}$, pentru valori mici $\lambda > 0$, este şi ea inversabilă (LC: datorită continuităţii), iar

– pentru astfel de matrice, limita matricelor inversate este inversa matricei-limită.

Folosind acum relaţia (179)**, vom putea scrie:**

$$\begin{aligned}\lim_{\lambda\to 0} y(x) &= \lim_{\lambda\to 0}\Big(\sum_{i=1}^{n}\hat{\alpha}_i\,K(x_i,x)\Big) = \sum_{i=1}^{n}\lim_{\lambda\to 0}\hat{\alpha}_i\,K(x_i,x)\\ &= \sum_{i=1}^{n} B_i\,K(x_i,x),\ \textbf{unde } B \overset{not.}{=} [B_1,\ldots,B_n]^\top = \mathbf{K}^{-1}y.\end{aligned}$$

În consecinţă, prin trecerea la limita specificată în enunţ, vom avea $y(x) = \sum_{i=1}^{n} B_i\,K(x_i,x)$**.**

c. Pentru a demonstra că eroarea la antrenare devine zero atunci când $\lambda\to 0$**, trebuie să arătăm că** $\lim_{\lambda\to 0} y(x_t) = y_t$ **pentru orice** $t\in\{1,\ldots,n\}$**. Plecând de la rezultatul obţinut la punctul** b**, putem scrie:**

$$\begin{aligned}\lim_{\lambda\to 0} y(x_t) &= \sum_{i=1}^{n} B_i\,K(x_i,x_t) = \sum_{i=1}^{n}\Big(\sum_{j=1}^{n} y_j\,\mathbf{K}^{-1}(i,j)\Big)K(x_i,x_t)\\ &\overset{asoc.}{=} \sum_{j=1}^{n} y_j\sum_{i=1}^{n}\mathbf{K}^{-1}(i,j)\,K(x_i,x_t) \overset{sim.}{=} \sum_{j=1}^{n} y_j\sum_{i=1}^{n}\mathbf{K}^{-1}(j,i)\,K(x_i,x_t)\\ &= \sum_{j=1}^{n} y_j\sum_{i=1}^{n}\mathbf{K}^{-1}(j,i)\,\mathbf{K}(i,t) = \sum_{j=1}^{n} y_j\,\delta(j,t)\\ &= y_t,\end{aligned}$$

unde $\mathbf{K}^{-1}(i,j)$ **elementul de pe poziţia** (i,j) **a matricei** $\mathbf{K}^{-1}$**, iar**

$$\delta(i,j) = \begin{cases} 0 & \textbf{pentru } i\neq j\\ 1 & \textbf{pentru } i=j.\end{cases}$$

În cele de mai sus, am ţinut cont în primul rând de faptul că funcţia-nucleu K **(şi, în consecinţă şi matricea** $\mathbf{K}$**) este simetrică, ceea ce implică faptul că matricea** $\mathbf{K}^{-1}$ **este simetrică, deci** $\mathbf{K}^{-1}(i,j) = \mathbf{K}^{-1}(j,i)$ **pentru orice indici** i **şi** j**. În al doilea rând, din relaţia** $\mathbf{K}^{-1}\mathbf{K} = \mathbf{I}$ **rezultă că** $\sum_i \mathbf{K}^{-1}(j,i)\,\mathbf{K}(i,t) = \delta(j,t)$ **pentru orice indici** j **şi** t**.**

***Observaţie importantă*:**
O *consecinţă* remarcabilă a rezultatului obţinut la acest exerciţiu este următoarea: dacă valorile pentru componentele y_i **sunt doar două (de exemplu,** 0 **sau** 1**), atunci setul de date** $(\phi(x_1,y_1),\ldots,\phi(x_n,y_n))$ **este liniar separabil când** $\lambda\to 0$**. (Am notat cu** ϕ **maparea corespunzătoare nucleului RBF.) Altfel spus, pentru orice etichetare consistentă a setului de instanţe** $x_1,\ldots,x_n$ **se obţine separabilitate liniară în spaţiul de „trăsături“ generat de nucleul RBF, când** $\lambda\to 0$**.**

11. **(Regresia liniară cu zgomot gaussian şi cu regularizare L_1: rezolvare cu metoda *descreşterii pe coordonate*)**

□ • ○ ***Stanford, 2007 fall, Andrew Ng, HW3, pr. 3.a***[348]

La problema 3.C am prezentat cazul regresiei *ridge*, adică regresia liniară cu „zgomot" gaussian la care funcţia obiectiv este augmentată cu termenul de *regularizare* $\lambda\|\beta\|^2$. De această dată, vom analiza cazul în care regularizarea se face folosind norma L_1. Aşadar, acum dorim să minimizăm funcţia obiectiv

$$J(\beta) = \frac{1}{2}\sum_{i=1}^{n}(\beta^\top x^{(i)} - y^{(i)})^2 + \lambda\|\beta\|_1,$$

unde $\|\beta\|_1$ este definit ca fiind $\sum_{i=1}^{d}|\beta_i|$.

Regularizarea de normă L_1 este foarte utilă pentru că, aşa cum vom vedea, ea oferă avantajul creării de soluţii „rare" (engl., sparse), ceea ce înseamnă că multe dintre componentele vectorului rezultat β vor fi egale cu 0.

***Comentariu*:** Acest tip de regresie este mai dificil de rezolvat decât regresia neregularizată ori cea cu regularizare de normă L_2, pentru că termenul L_1 nu este derivabil. Totuşi, în acest scop au fost dezvoltaţi mai mulţi algoritmi eficienţi, care lucrează foarte bine în practică. Între aceştia, o abordare foarte directă este reprezentată de metoda *descreşterii pe coordonate* (engl., *coordinate descent*).[349]

În această problemă veţi concepe un algoritm de tipul descreşterii pe coordonate pentru regresia liniară de tip LSE cu regularizare L_1. Mai specific, întrucât metoda descreşterii pe coordonate este iterativă — ca de altfel şi metoda gradientului, care se aplică în cazul regresiei *ridge*, aşa cum aţi văzut la problema 6 — veţi deduce *regula de actualizare* pentru o pondere oarecare β_i, cu $i \in \{1, \ldots, d\}$.

Considerând *matricea de design* X şi vectorul y aşa cum au fost definite la problemele 3 şi 6, precum şi vectorul de parametri β, [ne întrebăm] cum am putea [oare] să „ajustăm" valoarea lui β_i astfel încât să minimizăm funcţia obiectiv?

Pentru a răspunde la această întrebare, vom rescrie funcţia obiectiv de mai sus astfel:

$$\begin{aligned} J(\beta) &= \frac{1}{2}\|X\beta - y\|^2 + \lambda\|\beta\|_1 \\ &= \frac{1}{2}\|X\bar{\beta} + X_i\beta_i - y\|^2 + \lambda\|\bar{\beta}\|_1 + \lambda|\beta_i|, \end{aligned}$$

unde $X_i \in \mathbb{R}^n$ reprezintă coloana i a matricei X,[350] **iar vectorul $\bar{\beta}$ este identic cu vectorul β, doar că $\bar{\beta}_i = 0$. Tot ce am făcut aici a fost să rescriem expresia precedentă, în aşa fel încât termenul β_i să apară în mod explicit în funcţia obiectiv.**

[348] Punctele *b* şi *c* ale acestei probleme sunt de tip implementare. Acolo sunteţi solicitaţi să aplicaţi algoritmul dezvoltat în acest exerciţiu pe un anumit set de date de antrenament.

[349] Alţi algoritmi de învăţare automată care folosesc metoda descreşterii (sau, aternativ, a creşterii) pe coordonate sunt menţionaţi în nota de subsol 691 de la capitolul *Maşini cu vectori-suport*.

[350] Vă reamintim că în matricea de design $X \in \mathbb{R}^{n\times d}$, linia i este instanţa $x^{(i)}$ (văzută ca vector-coloană) transpusă.

Totuşi, ultima expresie încă îl are în componenţa sa pe $|\beta_i|$, care este nederivabil, deci este dificil de optimizat. Pentru a depăşi acest impediment, *remarcăm* faptul că semnul parametrului β_i este fie pozitiv fie negativ (mai precis, fie negativ fie nenegativ). Dacă am şti semnul lui β_i, atunci $|\beta_i|$ ar deveni pur şi simplu un termen liniar. Aşadar, putem rescrie funcţia obiectiv astfel:

$$J(\beta) = \frac{1}{2}\|X\bar{\beta} + \beta_i X_i - y\|^2 + \lambda\|\bar{\beta}\|_1 + \lambda s_i \beta_i, \tag{181}$$

unde prin s_i am notat semnul lui β_i, deci $s_i \in \{-1, +1\}$.

Pentru a actualiza valoarea parametrului β_i, putem să calculăm valoarea sa optimă pentru fiecare dintre cele două valori posibile ale lui s_i în parte — asigurându-ne apoi că restricţionăm aceste valori optime ale lui β_i astfel încât ele să satisfacă restricţia de semn pe care am fixat-o pentru cazul respectiv —, iar apoi să vedem care dintre cele două soluţii corespunde valorii celei mai bune pentru funcţia obiectiv la iteraţia respectivă.

Vă cerem ca pentru fiecare dintre valorile posibile pentru semnul s_i, să calculaţi valorile optime corespunzătoare pentru parametrul β_i.

***Sugestie*: În acest scop, puteţi fixa valoarea lui s_i în expresia (181), pe care apoi urmează să o derivaţi în raport cu β_i pentru a-i determina valoarea optimă. În final, impuneţi restricţii asupra valorii optime obţinute pentru β_i, astfel încât ea să aparţină domeniului corespunzător. (De exemplu, pentru $s_i = +1$, trebuie să impuneţi asupra lui β_i restricţia $\beta_i \geq 0$.)**

Răspuns:

În cazul $s_i = +1$,

$$\begin{aligned} J(\beta) &= \frac{1}{2}(X\bar{\beta} + \beta_i X_i - y)^\top (X\bar{\beta} + \beta_i X_i - y) + \lambda\|\bar{\beta}\|_1 + \lambda\beta_i \\ &= \frac{1}{2}(\beta_i^2 X_i^\top X_i + 2\beta_i X_i^\top (X\bar{\beta} - y) + \|X\beta - y\|^2) + \lambda\|\bar{\beta}\|_1 + \lambda\beta_i. \end{aligned}$$

Prin urmare,

$$\frac{\partial J(\beta)}{\partial \beta_i} = X_i^\top X_i \beta + X_i^\top (X\bar{\beta} - y) + \lambda,$$

ceea ce înseamnă că valoarea optimă pentru β_i în acest caz este dată de expresia

$$\beta_i = \max\left\{\frac{-X_i^\top (X\bar{\beta} - y) - \lambda}{X_i^\top X_i}, 0\right\}.$$

În mod similar, în cazul $s_i = -1$ valoarea optimă care se obţine pentru β_i este:

$$\beta_i = \min\left\{\frac{-X_i^\top (X\bar{\beta} - y) + \lambda}{X_i^\top X_i}, 0\right\}.$$

12. (Regresia liniară şi fenomenul de overfitting: Adevărat sau Fals?)

□ • ○ ***Stanford, 2015 fall, Andrew Ng, midterm, pr. 1.a***

Presupunem că, folosind un set [fix] format din m exemple, antrenaţi un model de *regresie liniară*, $h_\beta(x) = \beta^\top x$, unde β şi $x \in \mathbb{R}^{d+1}$. După antrenare, vă

daţi seama că *varianţa* modelului dumneavoastră este relativ mare, deci se produce *supra-specializare* (engl., *overfitting*). Pentru fiecare dintre metodele listate mai jos, asociaţi fie răspunsul *Adevărat* în caz că metoda respectivă poate contracara / reduce fenomenul de overfitting, fie răspunsul *Fals* în cazul contrar. Explicaţi de ce.

a. Extindeţi vectorul de trăsături prin adăugarea de noi atribute / trăsături.

b. Impuneţi condiţia ca parametrul (de fapt, vectorul de parametri) β să urmeze o distribuţie a priori, şi anume o distribuţie de forma $\mathcal{N}(0, \tau^2 I)$, iar apoi deduceţi valoarea lui β prin metoda estimării de probabilitate maximă a posteriori (engl., maximum a posteriori probability, MAP).

Răspuns:

a. Fals. Adăugarea de noi trăsături va face ca modelul dumneavoastră să fie şi mai *complex*. El va reuşi să se adapteze la şi mai multe excepţii (engl., *outliers*) din setul de antrenament şi, în consecinţă să producă şi mai mult overfitting.

b. Adevărat. Impunând / cerând ca parametrul β să urmeze o distribuţie probabilistă a priori [precum cea indicată în enunţ], se limitează în mod efectiv norma vectorului β, fiindcă vectorii mai mari [în normă] au o probabilitate mai mică.[351] Astfel, acest procedeu face ca modelul nostru să fie mai puţin predispus la overfitting.

1.1.2 Regresia logistică

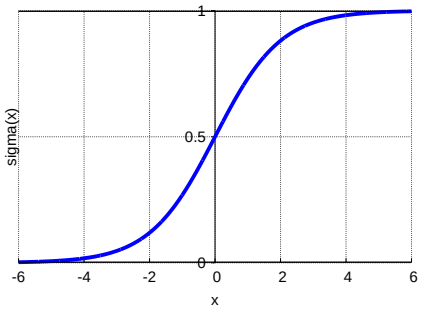
13. **(Regresia logistică, chestiuni introductive: estimare MLE; deducerea regulilor de actualizare a parametrilor, folosind metoda gradientului; echivalenţa cu minimizarea costului logistic)**

formulare de Liviu Ciortuz, după
■ □ • ○ ***Stanford, 2016 spring, Chris Piech, Introduction to Probability for Computer Scientists course (CS109), Logistic Regression***
CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW2, pr. 4

În învăţarea automată, *algoritmii de clasificare* primesc ca date de intrare n instanţe de antrenament care sunt identic şi independent distribuite, $(x^{(1)}, y^{(1)})$, $(x^{(2)}, y^{(2)}), \ldots, (x^{(n)}, y^{(n)})$, fiecare vector $x^{(i)}$ având d atribute. În cazul de faţă vom presupune că $y^{(i)} \in \{0, 1\}$ pentru $i = 1, \ldots, n$.

[351] Vedeţi şi problema 26.

Regresia logistică **este un algoritm de clasificare, care lucrează urmărind să înveţe o funcţie care să aproximeze în mod convenabil distribuţia $P(Y|X)$.[352] *Presupunerea ei de bază* este că $P(Y|X)$ poate fi aproximată cu o *funcţie sigmoidală* (numită şi *funcţie logistică*) aplicată unei combinaţii liniare de atributele de intrare.**



Din punct de vedere matematic, dată fiind o (singură) instanţă de antrenament (x, y), regresia logistică va considera

$$P(Y = 1|X = x; w) = \sigma(z) \text{ şi, echivalent, } P(Y = 0|X = x; w) = 1 - \sigma(z), \text{ unde} \quad (182)$$

$$\sigma(z) \overset{def.}{=} \frac{1}{1 + e^{-z}} = \frac{e^z}{1 + e^z}, \text{ cu}$$

$$z \overset{not.}{=} w_0 + \sum_{i=1}^{d} w_i x_i = w \cdot x \text{ şi } w \overset{not.}{=} (w_0, w_1, \ldots, w_d) \in \mathbb{R}^{d+1}, \text{ presupunând că}$$

vectorul x a fost extins cu componenta $x_0 = 1$.

Operatorul $\cdot$ desemnează produsul scalar al vectorilor.[353] Pornind de la aceste formule pentru probabilitatea condiţionată $P_{Y|X}$, putem crea un algoritm care selectează valori pentru vectorul w care maximizează această probabilitate pentru întreg setul de date de antrenament.

În acest exerciţiu mai întâi se va arăta că a maximiza log-verosimiliatea completă a datelor de antrenament revine la a maximiza log-verosimiliatea condiţională, iar după aceea veţi calcula expresia funcţiei de log-verosimilitate [condiţională] pentru datele $(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})$. Apoi vă vom arăta modul în care un algoritm iterativ bazat pe *metoda gradientului* poate să selecteze valorile optime pentru parametrii w. În continuare, veţi descoperi cum anume se obţin *regulile de actualizare* a parametrilor din acest algoritm, folosind derivatele parţiale ale funcţiei de log-verosimilitate în raport cu componentele vectorului w. În final, veţi vedea că a maximiza funcţia de log-verosimilitate condiţională $\ell(w)$ este echivalent cu a minimiza o anumită funcţie de cost.

a. Arătaţi că funcţia de *log-verosimilitate completă* a setului de date de antrenament $\ln \prod_{i=1}^{n} P(x^{(i)}, y^{(i)}|w)$ se poate scrie ca o sumă dintre *log-verosimilitatea condiţională* $\ell(w) \overset{def.}{=} \ln \prod_{i=1}^{n} P_{Y|X}(y^{(i)}|x^{(i)}; w)$ şi un termen care nu depinde de vectorul w.

***Consecinţă imediată*: Întrucât facem estimarea în sensul verosimilităţii maxime (MLE) a parametrului w, rezultă că putem să ne limităm la a maximiza $\ell(w)$.**

[352] La capitolul *Clasificare bayesiană* veţi vedea că în acest tip de clasificare ideea de bază este că, dată fiind o instanţă x, urmărim să alegem ca output acea etichetă y care maximizează probabilitatea condiţionată $P(Y = y|X = x)$. Algoritmul Bayes Naiv este [tot] un algoritm de clasificare care modelează / aproximează probabilitatea $P(Y = y|X = x)$ folosind *presupoziţia naivă* că toate trăsăturile / atributele sunt mutual independente în raport cu eticheta clasei. Regresia logistică *nu* foloseşte o astfel de presupoziţie.

[353] În notaţie matriceală, dacă ne referim la w şi x ca vectori-coloană din acelaşi spaţiu vectorial, am putea scrie $w^\top x$ în loc de $w \cdot x$. Operatorul $\top$ desemnează operaţia de transpunere a matricelor / vectorilor.

b. Demonstraţi că log-verosimilitatea condiţională $\ell(w)$ se calculează — ţinând cont de *presupoziţia de bază* a regresiei logistice — în felul următor:[354]

$$\ell(w) = \sum_{i=1}^{n} \left(y^{(i)} \ln \sigma(w \cdot x^{(i)}) + (1 - y^{(i)}) \ln(1 - \sigma(w \cdot x^{(i)})) \right). \tag{183}$$

***Comentariu*:** Dispunând de expresia (183) pentru funcţia de log-verosimilitate condiţională, putem să trecem la identificarea acelor valori ale lui w care maximizează această funcţie. Spre deosebire de alte situaţii în care se pune(a) problema găsirii optimului funcţiei de log-verosimilitate, în acest caz nu există o *formulă analitică* (engl., closed form formula) care să ne permită să obţinem în mod direct valoarea optimă a lui w. De aceea, vom identifica această valoare utilizând o *metodă de optimizare*. Mai precis, vom folosi un algoritm iterativ numit *gradientul ascendent*.[355] Ideea acestui algoritm este că atunci când facem în mod continuu paşi mici în direcţia gradientului funcţiei de optimizat — gradientul fiind, conform definiţiei, vectorul de derivate parţiale al funcţiei respective —, vom ajunge la un moment dat într-un punct de maxim local al acelei funcţii. În cazul regresiei logistice, se poate demonstra că rezultatul aplicării metodei / algoritmului gradientului ascendent va fi întotdeauna punctul de *maxim global*.[356] Paşii mici pe care îi parcurgem în mod iterativ, dat fiind setul de date de antrenament, sunt calculaţi / determinaţi folosind următoarea *regulă de actualizare*:

$$w_j^{new} = w_j^{old} + \eta \frac{\partial}{\partial w_j} \ell(w^{old}), \tag{184}$$

unde η este un factor constant care determină mărimea pasului pe care îl facem la fiecare iteraţie, cunoscut sub numele de *rată de învăţare*.

c. Arătaţi că forma derivatelor parţiale ale funcţiei de log-verosimilitate condiţională în raport cu fiecare componentă w_j a vectorului w este următoarea:

$$\frac{\partial}{\partial w_j} \ell(w) = \sum_{i=1}^{n} [y^{(i)} - \underbrace{\sigma(w \cdot x^{(i)})}_{P(y^{(i)}=1 | X = x^{(i)}; w)}] x_j^{(i)} \text{ pentru } j = 0, 1, \ldots, d. \tag{185}$$

***Sugestie*: Puteţi folosi următoarea *proprietate* pentru derivata funcţiei logistice σ în raport cu argumentul ei (vedeţi problema 33.b):**

$$\frac{\partial}{\partial z} \sigma(z) = \sigma(z)[1 - \sigma(z)] \text{ pentru } \forall z \in \mathbb{R}.$$

***Observaţie*: Din relaţia (185) urmează că vectorul gradient $\nabla_w \ell(w)$ se poate scrie astfel:**

$$\nabla_w \ell(w) = \sum_{i=1}^{n} [y^{(i)} - \sigma(w \cdot x^{(i)})] x^{(i)}. \tag{186}$$

[354] *Observaţie importantă*: Expresia acestei log-verosimilităţi condiţionale, adică partea dreaptă a egalităţii (183) are forma unei cross-entropii, cu semnul schimbat. (De fapt, câte o cross-entropie (cu semnul schimbat) pentru fiecare instanţă.) Pentru *definiţia noţiunii de cross-entropie*, vedeţi problema 63 de la capitolul de *Fundamente*. Valorile $y^{(i)}$ şi $1 - y^{(i)}$ corespund unei distribuţii de probabilitate, iar valorile $\sigma(w \cdot x^{(i)})$ şi $1 - \sigma(w \cdot x^{(i)})$ corespund altei distribuţii de probabilitate (şi anume, cea calculată de regresia logistică), care aproximează prima distribuţie.

[355] Pentru o ilustrare simplă a modului cum funcţionează algoritmul gradientului *descendent*, vedeţi problema 80.c de la capitolul de *Fundamente*.

[356] Vedeţi pr. 14.

La problema 17 vom arăta că $\nabla_w \ell(w)$ se poate scrie sub o formă şi mai compactă, folosind aşa-numita *matrice de design* X.[357]

d. [Legătura cu *funcţia de cost logistică*][358]

Arătaţi că dacă în loc de $y^{(i)} \in \{0,\ 1\}$ vom considera $y'^{(i)} \in \{-1,\ 1\}$ pentru $i = 1, \ldots, n$,[359] **atunci se poate demonstra că următoarea *funcţie de cost mediu* (sau de *risc empiric*; engl., empirical risk)**

$$J(w) \stackrel{def.}{=} \frac{1}{n} \sum_{i=1}^{n} \ln(1 + \exp(-y'^{(i)} w \cdot x^{(i)})) \tag{187}$$

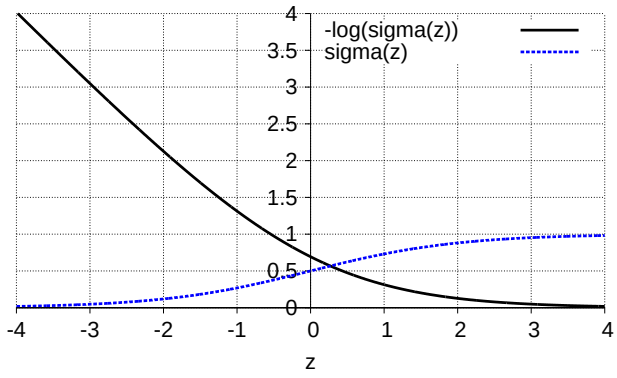
este chiar $-\frac{1}{n}\ell(w)$, unde $\ell(w)$ este funcţia de log-verosimilitate condiţională din relaţia (183). Ca o *consecinţă*, a maximiza funcţia de log-verosimilitate condiţională $\ell(w)$ este echivalent cu a minimiza funcţia de cost mediu $J(w)$.

***Observaţie*: Funcţia $\phi(y'w \cdot x) \stackrel{not.}{=} \ln(1 + \exp(-y'w \cdot x))$ sau, mai simplu, $\phi(z) \stackrel{not.}{=} \ln(1 + e^{-z})$ este numită *funcţia de cost* (sau, pierdere) *logistică*.**[360]

Legătura dintre $\phi(z) = \ln(1 + e^{-z})$ pe de o parte şi (atenţie!) *funcţia logistică* (sau *funcţia sigmoidală*) $\sigma(z) \stackrel{def.}{=} \frac{1}{1+e^{-z}}$ pe de altă parte este următoarea:

$$\phi(z) = \ln(1 + e^{-z}) = -\ln \sigma(z).$$

Din această cauză, funcţia de cost logistică este numită uneori şi *funcţia log-sigmoidală*.



Răspuns:

a. Putem scrie log-verosimilitatea completă a setului de date de antrenament astel:

$$\begin{aligned}
\textit{log-likelihood} &= \ln \prod_{i=1}^{n} P(x^{(i)}, y^{(i)}; w) = \ln \prod_{i=1}^{n} (P_{Y|X}(y^{(i)}|x^{(i)}; w)\, P_X(x^{(i)})) \\
&= \ln \left(\left(\prod_{i=1}^{n} P_{Y|X}(y^{(i)}|x^{(i)}; w) \right) \cdot \left(\prod_{i=1}^{n} P_X(x^{(i)}) \right) \right) \\
&= \ln \prod_{i=1}^{n} P_{Y|X}(y^{(i)}|x^{(i)}; w) + \ln \prod_{i=1}^{n} P_X(x^{(i)}) \stackrel{not.}{=} \ell(w) + \ell_x.
\end{aligned}$$

[357] Acolo de fapt se lucrează cu o formă mai generală de regresie logistică, şi anume *regresia logistică local-ponderată*.

[358] Cf. secţiunii 2 (Logistic Regression) din documentul *Supplemental lecture notes* (pentru cursul de *Machine Learning*), de John Duchi, de la Universitatea Stanford.

[359] Definim $y'^{(i)} = -1$ dacă $y^{(i)} = 0$ şi $y'^{(i)} = 1$ dacă $y^{(i)} = 1$.

[360] *Atenţie*! În acest context, $z \stackrel{not.}{=} y'w \cdot x$. În enunţ, la început (când $y \in \{0, 1\}$), am avut $z \stackrel{not.}{=} w \cdot x$.

Observăm că ℓ_x nu depinde de parametrul w.

b. Pentru început, iată o modalitate de a scrie în mod compact expresia (182) din enunţ, care reprezintă probabilitatea condiţionată a unei *singure* instanţe de antrenament:[361]

$$P(Y = y|X = x; w) = \sigma(w \cdot x)^y \, [1 - \sigma(w \cdot x)]^{1-y}, \text{ presupunând că } y \in \{0, 1\}.$$

Întrucât toate instanţele de antrenament sunt independente, verosimilitatea condiţională a întregului set de date este următoarea:

$$\prod_{i=1}^{n} P(Y = y^{(i)}|X = x^{(i)}; w) = \prod_{i=1}^{n} \left(\sigma(w \cdot x^{(i)})^{y^{(i)}} \, [1 - \sigma(w \cdot x^{(i)})]^{1-y^{(i)}} \right). \tag{188}$$

Dacă aplicăm logaritmul natural acestei funcţii, obţinem imediat expresia indicată în enunţ (183) pentru log-verosimilitatea condiţională a datelor în cazul regresiei logistice.

c. Calculul derivatelor parţiale ale expresiei $\sigma(w \cdot x)$ în raport cu componentele vectorului w va fi făcut folosind formula de derivare pentru funcţii compuse.

Concret, valoarea derivatei parţiale a lui $\sigma(w \cdot x)$ în raport cu componenta w_j a vectorului w, pentru o instanţă (x, y), se calculează astfel:

$$\begin{aligned}
&\frac{\partial}{\partial w_j} \ln[\sigma(w \cdot x)^y \, [1 - \sigma(w \cdot x)]^{1-y}] \\
&= \frac{\partial}{\partial w_j} y \ln \sigma(w \cdot x) + \frac{\partial}{\partial w_j} (1 - y) \ln[1 - \sigma(w \cdot x)] \\
&= \left[\frac{y}{\sigma(w \cdot x)} - \frac{1 - y}{1 - \sigma(w \cdot x)} \right] \frac{\partial}{\partial w_j} \sigma(w \cdot x) \\
&= \left[\frac{y}{\sigma(w \cdot x)} - \frac{1 - y}{1 - \sigma(w \cdot x)} \right] \sigma(w \cdot x)[1 - \sigma(w \cdot x)] x_j \\
&= \frac{y - \sigma(w \cdot x)}{\sigma(w \cdot x)[1 - \sigma(w \cdot x)]} \sigma(w \cdot x)[1 - \sigma(w \cdot x)] x_j \\
&= [y - \sigma(w \cdot x)] x_j.
\end{aligned} \tag{189}$$

Întrucât derivata sumei este suma derivatelor, rezultă că derivata parţială a funcţiei de log-verosimilitate condiţională $\ell(w)$ în raport cu w_j este suma acestui tip de termeni, câte unul pentru fiecare instanţă de antrenament. Mai exact, după ce aplicăm funcţia $\ln$ expresiei (188) şi apoi calculăm derivatele ei parţiale în raport cu w_j (pentru $j \in \{0, 1, \ldots, d\}$), vom obţine rezultatul (185), datorită relaţiei (189), pe care am demonstrat-o mai sus.

d. Demonstraţia se bazează în esenţă pe proprietatea $1 - \sigma(z) = \sigma(-z)$, care este imediată (vedeţi problema 33.b).

Cazul $y'^{(i)} = 1 \; : \; P(Y = 1|X = x^{(i)}; w) = \sigma(w \cdot x^{(i)}) = \sigma(y'^{(i)} w \cdot x^{(i)})$;

Cazul $y'^{(i)} = -1 \; : \; P(Y' = -1|X = x^{(i)}; w) = 1 - P(Y = 1|X = x^{(i)}; w) = 1 - \sigma(w \cdot x^{(i)})$
$= \sigma(-w \cdot x^{(i)}) = \sigma(y'^{(i)} w \cdot x^{(i)})$.

Prin urmare,

[361] Acest procedeu este numit *artificiul exponenţierii* (engl., the exponentiation trick).

$$\begin{aligned}\ell(w) &= \ln \prod_{i=1}^{n} P(Y = y^{(i)} | X = x^{(i)}; w) = \sum_{i=1}^{n} \ln P(Y = y^{(i)} | X = x^{(i)}; w) \\ &= \sum_{i=1}^{n} \ln P(Y' = y'^{(i)} | X = x^{(i)}; w) = \sum_{i=1}^{n} \ln \sigma(y'^{(i)} w \cdot x^{(i)}) \\ &= \sum_{i=1}^{n} \ln \frac{1}{1 + \exp(-y'^{(i)} w \cdot x^{(i)}))} = -\sum_{i=1}^{n} \ln(1 + \exp(-y'^{(i)} w \cdot x^{(i)})) \\ &= -\sum_{i=1}^{n} \phi(y'^{(i)} w \cdot x^{(i)}).\end{aligned}$$

Aşadar, $-\frac{1}{n}\ell(w) = J(w)$**.**

14. **(Regresia logistică, o proprietate importantă: funcţia de log-verosimilitate este concavă, deci are un maxim global; calculul matricei hessiene pentru funcţia de log-verosimilitate)**

■ ● ○ ***Stanford, 2008 fall, Andrew Ng, HW1, pr. 1.a***

La problema 13.b am demonstrat că în cazul regresiei logistice funcţia de log-verosimilitate condiţională se scrie sub forma următoare:

$$\ell(w) = \sum_{i=1}^{n} \Big(y^{(i)} \ln \sigma_w(x^{(i)}) + (1 - y^{(i)}) \ln(1 - \sigma_w(x^{(i)})) \Big),$$

unde instanţele $x^{(i)}$ **sunt vectori-coloană de forma** $x = (x_1, \ldots, x_d)^\top$**, etichetele** $y^{(i)} \in \{0, 1\}$**, vectorul de ponderi** $w \in \mathbb{R}^d$**, iar** $\sigma_w(x) \stackrel{def.}{=} \sigma(w \cdot x) = \dfrac{1}{1 + \exp(-w \cdot x)}$**.**

a. Calculaţi *matricea hessiană* (H**) a acestei funcţii, care este prin definiţie matricea formată din derivatele parţiale de ordin secund ale funcţiei** ℓ**.**

b. Arătaţi că pentru orice $z = (z_1, \ldots, z_d)^\top$**, are loc proprietatea**

$$z^\top H z \leq 0.$$

***Observaţie*: Aceasta este una dintre modalităţile clasice de a arăta că matricea** H **este *negativ semidefinită*, ceea ce se notează sub forma** $H \leq 0$**. Aceasta implică faptul că** ℓ **este funcţie concavă şi, în consecinţă, ea nu are decât o singură valoare maximă (care este şi maxim global).**

Răspuns:

a. Deşi expresia derivatei parţiale a funcţiei de log-verosimilitate condiţională a fost obţintă deja la problema 13.c, de dragul alternanţei aici vom obţine pe o altă cale expresia respectivă, şi anume folosind faptul că funcţia logistică admite ca funcţie inversă funcţia *logit*, care este definită prin $\sigma^{-1}(z) = \ln \dfrac{z}{1 - z}$ **(vedeţi problema 33.b). Datorită acestui fapt, expresia funcţiei de log-verosimiliate condiţională poate fi elaborată astfel:**

$$\begin{aligned}
\ell(w) &= \sum_{i=1}^{n} \Big(y^{(i)} \ln \sigma_w(x^{(i)}) + (1-y^{(i)}) \ln(1-\sigma_w(x^{(i)})) \Big) \\
&= \sum_{i=1}^{n} \Big(y^{(i)} \ln \frac{\sigma_w(x^{(i)})}{1-\sigma_w(x^{(i)})} + \ln(1-\sigma_w(x^{(i)})) \Big) \\
&= \sum_{i=1}^{n} \Big(y^{(i)} \sigma_w^{-1}(\sigma_w(x^{(i)})) + \ln(1-\sigma_w(x^{(i)})) \Big) \\
&= \sum_{i=1}^{n} \Big(y^{(i)} \sigma_w^{-1}(\sigma_w(x^{(i)})) + \ln(1-\sigma_w(-x^{(i)})) \Big) = \sum_{i=1}^{n} \Big(y^{(i)} (w \cdot x^{(i)}) - \ln(1+e^{w \cdot x^{(i)}}) \Big).
\end{aligned}$$

Prin urmare,

$$\frac{\partial}{\partial w_k} \ell(w) = \sum_{i=1}^{n} \left(y^{(i)} x_k^{(i)} - \frac{e^{w \cdot x^{(i)}}}{1+e^{w \cdot x^{(i)}}} x_k^{(i)} \right) = \sum_{i=1}^{n} [y^{(i)} - \sigma_w(x^{(i)})] x_k^{(i)}. \tag{190}$$

În continuare vom folosi o altă proprietate importantă a funcţiei logistice: $\sigma'(z) = \sigma(z)(1-\sigma(z))$**. (Vedeţi tot problema 33.b.) Prin urmare,** $\dfrac{\partial \sigma_w(x)}{\partial w_k} = \sigma_w(x)(1-\sigma_w(x))\, x_k$**, fiindcă** $\sigma_w(x) = \sigma(w \cdot x)$**.**

Elementul generic al matricei hessiene (H**) este**

$$H_{lk} \;=\; \frac{\partial^2 \ell(w)}{\partial w_l \, \partial w_k} = \sum_{i=1}^{n} -\frac{\partial \sigma_w(x^{(i)})}{\partial w_l} x_k^{(i)} = \sum_{i=1}^{n} -\sigma_w(x^{(i)})(1-\sigma_w(x^{(i)}))\, x_l^{(i)} x_k^{(i)}. \tag{191}$$

Ţinând cont de faptul că putem scrie $xx^\top$ **ca o matrice** X **[dacă şi numai dacă** $X_{ij} = x_i x_j$**], rezultă că matricea hessiană** H **este:**[362]

$$H = -\sum_{i=1}^{n} \underbrace{\sigma_w(x^{(i)})(1-\sigma_w(x^{(i)}))}_{\in \mathbb{R}_+} \, x^{(i)} (x^{(i)})^\top. \tag{192}$$

b. Pentru orice vector-coloană $z \in \mathbb{R}^d$ **avem**

$$\begin{aligned}
z^\top H z &= -z^\top \left(\sum_{i=1}^{n} \sigma_w(x^{(i)})(1-\sigma_w(x^{(i)})) x^{(i)} (x^{(i)})^\top \right) z \\
&= -\sum_{i=1}^{n} \sigma_w(x^{(i)})(1-\sigma_w(x^{(i)})) z^\top x^{(i)} \underbrace{(x^{(i)})^\top z}_{(z^\top x^{(i)})^\top} \\
&= -\sum_{i=1}^{n} \sigma_w(x^{(i)})(1-\sigma_w(x^{(i)}))(z^\top x^{(i)})^2 \leq 0.
\end{aligned}$$

Ultima inegalitate de mai sus se justifică astfel: $(z^\top x^{(i)})^2 \geq 0$ **pentru orice** z **şi orice** $x^{(i)}$ **din** $\mathbb{R}^d$**, iar din definiţia funcţiei logistice** σ **rezultă** $0 < \sigma_w(x^{(i)}) < 1$**, ceea ce implică** $\sigma_w(x^{(i)})(1-\sigma_w(x^{(i)})) > 0$**.**

[362] La problema 17 vom arăta că H se poate scrie şi mai compact, folosind aşa-numita *matrice de design* X. (Veţi observa că colo de fapt se lucrează cu o formă mai generală de regresie logistică, şi anume *regresia logistică local-ponderată*.)

***Observaţie importantă*: Demonstraţia de mai sus furnizează tot ce este necesar pentru obţinerea [ulterioară a] relaţiei de actualizare a parametrilor la *aplicarea metodei lui Newton* în cazul regresiei logistice.**[363]

15. (Regresia logistică: particularizare pentru $d = 1$, adică pentru date din $\mathbb{R}$; aplicarea regulilor de actualizare pentru metoda gradientului şi respectiv metoda lui Newton-Raphson)

Liviu Ciortuz, 2025

În această problemă veţi aplica algoritmul de regresie logistică bazat pe metoda gradientului şi respectiv pe metoda lui Newton-Raphson pe un set de date din $\mathbb{R}$.

Considerăm un set oarecare de date de antrenament $\{(x^{(i)}, y^{(i)}), i = 1, \ldots, n\}$ în care fiecare $x^{(i)} \in \mathbb{R}^2$ este un vector de trăsături de forma $(1, x_1^{(i)})^\top$, iar $y^{(i)} \in \{0, 1\}$ este o etichetă binară. Vom nota cu w vectorul de ponderi $(w_0, w_1)^\top \in \mathbb{R}^2$. Presupunând că folosim un *model* parametrizat de forma

$$p(y = 1|x; w) \stackrel{def.}{=} \sigma(w \cdot x) = \frac{1}{1 + \exp(-w_0 - w_1 x_1)} = \frac{\exp(w_0 + w_1 x_1)}{1 + \exp(w_0 + w_1 x_1)},$$

***obiectivul* nostru este să găsim valorile $\bar{w}_j$ ale parametrilor w_j care maximizează verosimilitatea condiţională (engl., Maximum (Conditional) Likelihood Estimation, M(C)LE) a setului de date de antrenament.**

Funcţia de log-verosimilitate condiţională a acestui set de date este următoarea:[364]

$$\begin{aligned}
\ell(w) &= \ln \prod_{i=1}^{n} p(y^{(i)}|x^{(i)}, w) \\
&= \sum_{i=1}^{n} [y^{(i)} \ln \sigma(w \cdot x^{(i)}) + (1 - y^{(i)}) \ln(1 - \sigma(w \cdot x^{(i)}))] \qquad (193) \\
&= \sum_{i=1}^{n} \left[y^{(i)} \left(w_0 + w_1 x_1^{(i)}\right) - \ln\left(1 + \exp(w_0 + w_1 x_1^{(i)})\right)\right]. \qquad (194)
\end{aligned}$$

Componentele vectorului gradient pentru funcţia $\ell(w)$ sunt:[365]

$$\frac{\partial \ell(w)}{\partial w_j} = \sum_{i=1}^{n} (y^{(i)} - \sigma(w \cdot x^{(i)}))\, x_j^{(i)}. \qquad (195)$$

În consecinţă, vectorul gradient se poate calcula astfel:[366]

$$\nabla_w \ell(w) = \sum_{i=1}^{n} [y^{(i)} - \sigma(w \cdot x^{(i)})]\, x^{(i)}. \qquad (196)$$

[363] Pentru o descriere a metodei lui Newton, vedeţi *Comentariul* din enunţul problemei 80 de la capitolul de *Fundamente*. În prezentul capitol, la problema 7 se arată cum anume se aplică metoda lui Newton în cazul regresiei liniare, iar la problema 17 se calculează vectorul gradient şi matricea hessiană pentru o variantă uşor generalizată (şi anume, *local-ponderată* şi *regularizată*) a regresiei logistice.

[364] Vedeţi problema 13.b.

[365] Vedeţi problema 13.c.

[366] Vedeţi problema 14.a.

Matricea hessiană (notată cu H_w) a funcţiei $\ell(w)$ este prin definiţie matricea formată din derivatele parţiale de ordin secund ale acestei funcţii. Matricea H_w se poate calcula astfel:

$$H_w = -\sum_{i=1}^{n} \sigma(w \cdot x^{(i)})(1 - \sigma(w \cdot x^{(i)}))\, x^{(i)} (x^{(i)})^\top. \tag{197}$$

Considerăm acum setul de date alăturat.
Veţi aplica pe acest set de date regresia logistică folosind cele două metode — mai întâi metoda gradientului descendent şi apoi metoda lui Newton-Raphson — ale căror elemente principale au fost date mai sus. Veţi proceda conform punctelor următoare (a–g).

i	$x_1^{(i)}$	$y^{(i)}$
1	-1	0
2	$+1$	$+1$

a. Scrieţi funcţia de log-verosimilitate condiţională $\ell(w)$ corespunzătoare acestui set de date [în contextul regresiei logistice]. Puteţi să aplicaţi direct formula (194), particularizând-o pentru datele de antrenament de mai sus.

b. Calculaţi vectorul gradient pentru funcţia de log-verosimilitate condiţională de la punctul a. Puteţi să aplicaţi direct formula (196), particularizând-o pentru datele de antrenament de mai sus.

c. Considerând că pentru a afla maximul funcţiei de log-verosimilitate de la punctul a se aplică *metoda gradientului ascendent* cu rata de învăţare $\eta = 0.1$, vă cerem să aflaţi w^1, adică valoarea vectorului de ponderi după prima iteraţie, dacă la iniţializare s-a fixat $w^0 = 0$ (vectorul nul din $\mathbb{R}^2$).
Cum va clasifica modelul determinat de vectorul w^1 exemplele de antrenament?

d. Arătaţi că $\ell(w^0) \leq \ell(w^1)$, adică se confirmă (la finalul acestei prime iteraţii) faptul că metoda gradientului ascendent are ca efect creşterea valorii funcţiei ℓ în punctele succesive (w^0, w^1, ...) identificate de această metodă.

***Indicaţie*: Vă sugerăm să folosiţi formula (193), precum şi proprietăţi cunoscute ale funcţiei logistice / sigmoidale σ.**

e. Calculaţi matricea hessiană H_{w^0} pentru funcţia de log-verosimilitate condiţională de la punctul a, conform formulei (197).
Apoi, considerând din nou $w^0 = 0$ (vectorul nul din $\mathbb{R}^2$) şi folosind vectorul gradient ∇_{w^0} pe care l-aţi obţinut [în cursul rezolvării] la punctul c, precum şi matricea hessiană H_{w^0}, determinaţi valorile obţinute pentru ponderile w_0, w_1 după aplicarea primei iteraţii *conform metodei Newton-Raphson*:

$$w^{j+1} = w^j - (H_{w^j})^{-1} \nabla_{w^j}.$$

Cum va clasifica modelul determinat de vectorul w^1 instanţele de antrenament?

f. Arătaţi că şi în cazul aplicării metodei lui Newton-Raphson se verifică inegalitatea $\ell(w^0) \leq \ell(w^1)$.

g. Se ştie că în general „progresul" [către punctul de maxim / minim al funcţiei de optimizat] realizat de către metoda Newton-Raphson într-o singură iteraţie este mult mai mare decât cel realizat de către metoda gradientului. Verificaţi dacă această afirmaţie se susţine în contextul datelor noastre, limitându-vă

doar la analiza primei iteraţii. (Concret, vedeţi dacă valoarea pe care o are $\ell(w^1)$ în cazul metodei Newton-Raphson este mult mai mare decât valoarea lui $\ell(w^1)$ în cazul metodei gradientului.)

***Atenţie*! Următoarele două chestiuni (prezentate la curs)**
i. **forma regulii / regulilor de actualizare pentru metoda gradientului *ascendent***
ii. **regula de decizie pentru algoritmul regresiei logistice**
nu au fost date în enunţul de mai sus. Va trebui să le formulaţi dumneavoastră în cadrul rezolvării exerciţiului.

Răspuns:

a. Funcţia de log-verosimilitate condiţională corespunzătoare setului de date din enunţ este:

$$\begin{aligned}\ell(w) &\overset{(194)}{=} \sum_{i=1}^{2} \left[y^{(i)}\left(w_0 + w_1 x_1^{(i)}\right) - \ln\left(1+\exp(w_0 + w_1 x_1^{(i)})\right)\right] \\ &= -\ln\left(1+\exp(w_0 - w_1)\right) + w_0 + w_1 - \ln\left(1+\exp(w_0 + w_1)\right). \qquad (198)\end{aligned}$$

b. Pentru setul de date din enunţ, vectorul gradient este următorul:

$$\begin{aligned}\nabla_w &\overset{(196)}{=} \sum_{i=1}^{2}[y^{(i)} - \sigma(w \cdot x^{(i)})]\, x^{(i)} \\ &= -\sigma(w \cdot x^{(1)}) \begin{pmatrix} 1 \\ -1 \end{pmatrix} + (1 - \sigma(w \cdot x^{(1)}) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= -\sigma(w_0 - w_1) \begin{pmatrix} 1 \\ -1 \end{pmatrix} + (1 - \sigma(w_0 + w_1)) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= -\sigma(w_0 - w_1) \begin{pmatrix} 1 \\ -1 \end{pmatrix} + \sigma(-w_0 - w_1) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} -\sigma(w_0 - w_1) + \sigma(-w_0 - w_1) \\ \sigma(w_0 - w_1) + \sigma(-w_0 - w_1) \end{pmatrix}. \qquad (199)\end{aligned}$$

c. Aplicând regula de actualizare a vectorului de ponderi conform metodei gradientului ascendent, vom obţine:

$$w^1 \overset{(199)}{=} \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \eta \begin{pmatrix} -1/2 + 1/2 \\ 1/2 + 1/2 \end{pmatrix} = 0.1 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.1 \end{pmatrix}. \qquad (200)$$

Regula de predicţie / decizie pentru regresia logistică este următoarea:

> **dacă** $p(y^{(i)} = 1 | x^{(i)}; w) > p(y^{(i)} = 0 | x^{(i)}; w)$**, atunci alege predicţia** 1**,**
> **altfel alege predicţia** 0**.**

Se poate dovedi uşor că ea este echivalentă cu următoarea regulă:

> **dacă** $w \cdot x^{(i)} > 0$**, atunci alege predicţia** 1**,**
> **altfel alege predicţia** 0**.**

Avem

$w^1 \cdot x^{(1)} = \begin{pmatrix} 0 \\ 0.1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ -1 \end{pmatrix} = -0.1 < 0$**, deci** $x^{(1)}$ **este clasificat corect;**

$w^1 \cdot x^{(2)} = \begin{pmatrix} 0 \\ 0.1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 0.1 > 0$**, deci** $x^{(2)}$ **este clasificat corect.**

***Observaţie*: Se poate arăta că la orice iteraţie care urmează, componenta w_0^t va avea valoarea 0! (Această proprietate decurge din iniţializarea $w^0 = 0 \in \mathbb{R}^2$; ea nu este valabiă, în general, pentru alte iniţializări.)**

d. Făcând calculele, vom obţine:

$$\begin{aligned}
\ell(w^0) = \ell((0,0)^\top) &\overset{(193)}{=} \ln(1 - \sigma(w^0 \cdot x^{(1)})) + \ln(\sigma(w^0 \cdot x^{(2)})) \\
&= \ln(\sigma(-w^0 \cdot x^{(1)})) + \ln(\sigma(w^0 \cdot x^{(2)})) \overset{c}{=} 2\,\ln(\sigma(0)) \\
\ell(w^1) = \ell((0,0.1)^\top) &\overset{(193)}{=} \ln(\sigma(-w^1 \cdot x^{(1)})) + \ln(\sigma(w^1 \cdot x^{(2)})) = 2\,\ln(\sigma(0.1)).
\end{aligned}$$

Întrucât σ este funcţie strict crescătoare, rezultă că $\ell(w^0) < \ell(w^1)$.

e. Particularizând formula (197) pentru datasetul nostru şi ţinând cont de proprietatea $1 - \sigma(z) = \sigma(-z)$ care are loc pentru orice $z \in \mathbb{R}$, vom obţine:

$$\begin{aligned}
H_w &\overset{(197)}{=} -\sum_{i=1}^{2} \sigma(w \cdot x^{(i)})(1 - \sigma(w \cdot x^{(i)}))\, x^{(i)} (x^{(i)})^\top \\
&= -\sum_{i=1}^{2} \sigma(w \cdot x^{(i)})(\sigma(-w \cdot x^{(i)}))\, x^{(i)} (x^{(i)})^\top.
\end{aligned}$$

Întrucât

$$w^0 = 0 \in \mathbb{R}^2 \Rightarrow w^0 \cdot x^{(i)} = 0 \Rightarrow \sigma(w^0 \cdot x^{(i)}) = \sigma(-w^0 \cdot x^{(i)}) = \frac{1}{2} \text{ pentru } i \in \{1,2\},$$

rezultă

$$H_{w^0} = -\frac{1}{4} \begin{pmatrix} 1 \\ -1 \end{pmatrix} (1,-1) - \frac{1}{4} \begin{pmatrix} 1 \\ 1 \end{pmatrix} (1,1) = -\frac{1}{4} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} - \frac{1}{4} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} = -\frac{1}{2}\, I_2,$$

iar

$$w^1 = w^0 - (H_{w^0})^{-1} \nabla_{w^0} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} + 2\, I_2 \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}.$$

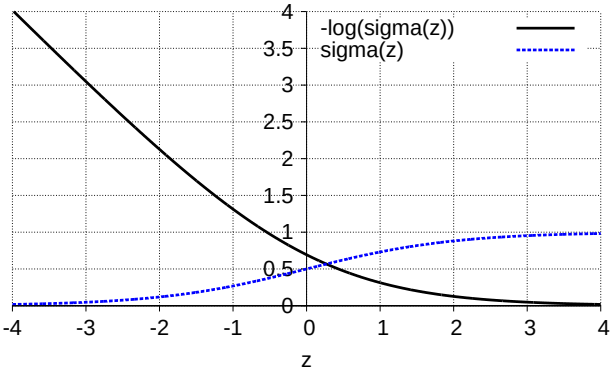
În consecinţă,

$w^1 \cdot x^{(1)} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ -1 \end{pmatrix} = -2 < 0$**, deci** $x^{(1)}$ **este clasificat corect;**

$w^1 \cdot x^{(2)} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 2 > 0$**, deci** $x^{(2)}$ **este clasificat corect.**

f. Faptul că $\ell(w^0) < \ell(w^1)$ se verifică exact ca la punctul i:
$\ell(w^0) = 2\,\ln(\sigma(0))$**, iar** $\ell(w^1) = 2\,\ln(\sigma(2))$**, deci** $\ell(w^0) < \ell(w^1)$**.**

g. Se constată imediat (fie numeric, ca mai jos, fie pe graficul alăturat, în care am reprezentat funcţia $\ln(\sigma(z))$ că la această primă iteraţie „progresul“ realizat de metoda Newton-Raphson — adică, „saltul“ înspre maximumul funcţiei ℓ — este mult mai mare decât cel obţinut cu metoda gradientului:



$$\ln \sigma(0) = \ln \frac{1}{2} = -\ln 2 \approx -0.693$$
$$\ln \sigma(0.1) = \ln \frac{1}{1+e^{-0.1}} = -\ln(1+e^{-0.1}) \approx -0.644$$
$$\ln \sigma(2) = \ln \frac{1}{1+e^{-2}} = -\ln(1+e^{-2}) \approx -0.127.$$

16. **(Regresia logistică: estimarea parametrilor în sens MAP — regularizare de normă L_2; efectul de *diminuare a ponderilor* (engl., weight decay))**

prelucrare de Liviu Ciortuz, după
□ • ○ ***Stanford, 2008 fall, Andrew Ng, HW3, pr. 4***

Presupunem că folosim un model de *regresie logistică* $h_\theta(x) = \sigma(\theta \cdot x)$, unde σ este *funcţia logistică* (sau *sigmoidală*), şi că dispunem de un set de date de antrenament $D = \{(x^{(i)}, y^{(i)}); i = 1, \ldots, n\}$ definit ca de obicei.

***Remember*:** Estimarea de verosimilitate maximă a parametrilor θ este definită astfel:

$$\theta_{ML} \overset{def.}{=} \underset{\theta}{\operatorname{argmax}}\, P(D|\theta) \overset{i.i.d.}{=} \underset{\theta}{\operatorname{argmax}} \prod_{i=1}^{n} P(y^{(i)}|x^{(i)};\theta). \tag{201}$$

Dacă dorim să *regularizăm* regresia logistică, putem impune ca [vectorul de] parametri θ să urmeze o distribuţie probabilistă a priori P. Estimarea de probabilitate maximă a posteriori pentru θ este definită astfel:

$$\theta_{MAP} \overset{def.}{=} \underset{\theta}{\operatorname{argmax}}\, P(\theta|D) \overset{F.B.}{=} \underset{\theta}{\operatorname{argmax}} \frac{P(D|\theta)P(\theta)}{P(D)} = \underset{\theta}{\operatorname{argmax}}\, P(D|\theta)P(\theta) \tag{202}$$
$$\overset{i.i.d.}{=} \underset{\theta}{\operatorname{argmax}}\, P(\theta) \prod_{i=1}^{m} P(y^{(i)}|x^{(i)};\theta). \tag{203}$$

Presupunem că alegem ca distribuţie a priori pentru θ distribuţia gaussiană multidimensională $\mathcal{N}(0, \tau^2 I)$, cu $\tau > 0$ şi I matricea identitate de dimensiune $(d+1) \times (d+1)$.[367]

[367] Am scris $d+1$ în loc de d fiindcă se consideră că se lucrează şi cu termenul liber θ_0, deci $\theta = (\theta_0, \theta_1, \ldots, \theta_d)$.

a. Arătaţi că atunci când parametrul θ urmează o distribuţie probabilistă de genul celei indicate mai sus, efectul practic este următorul: maximizarea funcţiei de probabilitate a posteriori $P(\theta|D)$ este echivalentă cu maximizarea sumei dintre funcţia de log-verosimilitate $\ln P(D|\theta)$ şi un termen de forma $-\lambda\|\theta\|^2$, unde $\lambda = \frac{1}{2\tau^2}$. (Astfel se justifică denumirea de *regularizare de normă* L_2 sau, mai simplu, regularizare L_2.)

b. Demonstraţi următoarea inegalitate:

$$\|\theta_{MAP}\|^2 \leq \|\theta_{ML}\|^2.$$

Răspuns:

a. Ţinând cont de faptul că funcţia $\ln$ este strict crecătoare pe întreg domeniul ei de definiţie, putem rescrie egalitatea reprezentată de primul şi ultimul termen din relaţia (201) sub forma următoare:

$$\theta_{ML} = \underset{\theta}{\operatorname{argmax}} \ln \prod_{i=1}^{n} P(y^{(i)}|x^{(i)};\theta) = \underset{\theta}{\operatorname{argmax}} \underbrace{\sum_{i=1}^{n} \ln P(y^{(i)}|x^{(i)};\theta)}_{not.:\ell(\theta)}.$$

Similar, rescriem relaţia (203) sub forma

$$\begin{aligned}
\theta_{MAP} &= \underset{\theta}{\operatorname{argmax}} \ln P(\theta) \prod_{i=1}^{m} P(y^{(i)}|x^{(i)};\theta) = \underset{\theta}{\operatorname{argmax}} (\ln P(\theta) + \sum_{i=1}^{n} \ln P(y^{(i)}|x^{(i)};\theta) \\
&= \underset{\theta}{\operatorname{argmax}} (\ln P(\theta) + \ell(\theta)) \\
&= \underset{\theta}{\operatorname{argmax}} \left(\ln \left(\frac{1}{(2\pi)^{(d+1)/2}\,\tau^{d+1}} \exp\Big(-\frac{1}{2}\theta^\top \frac{1}{\tau^2} I\,\theta \Big) \right) + \ell(\theta) \right) \\
&= \underset{\theta}{\operatorname{argmax}} \left(-\ln \left((2\pi)^{(d+1)/2}\,\tau^{d+1} \right) - \frac{1}{2\tau^2}\theta^2 + \ell(\theta) \right).
\end{aligned}$$

Întrucât expresia $(2\pi)^{(d+1)/2}\,\tau^{d+1}$ nu depinde de θ, rezultă că

$$\theta_{MAP} = \underset{\theta}{\operatorname{argmax}} \Big(-\underbrace{\frac{1}{2\tau^2}}_{not.:\lambda} \theta^2 + \ell(\theta) \Big) = \underset{\theta}{\operatorname{argmax}} \Big(\ell(\theta) - \lambda\|\theta\|^2 \Big). \tag{204}$$

b. Presupunem prin reducere la absurd că $\|\theta_{MAP}\|^2 > \|\theta_{ML}\|^2$. Rezultă că

$$\begin{aligned}
P(\theta_{MAP}) &= \frac{1}{(2\pi)^{\frac{d+1}{2}}\,|\tau^2 I|^{\frac{1}{2}}} \exp\Big(-\frac{1}{2\tau^2}\|\theta_{MAP}\|^2 \Big) \\
&< \frac{1}{(2\pi)^{\frac{d+1}{2}}\,|\tau^2 I|^{\frac{1}{2}}} \exp\Big(-\frac{1}{2\tau^2}\|\theta_{ML}\|^2 \Big) \\
&= P(\theta_{ML}).
\end{aligned}$$

În consecinţă,

$$\begin{aligned}
P(\theta_{MAP})P(D|\theta_{MAP}) &< P(\theta_{ML})P(D|\theta_{MAP}) \\
&\leq P(\theta_{ML})P(D|\theta_{ML}).
\end{aligned}$$

Ultima inegalitate de mai sus are loc întrucât prin definiţie θ_{ML} maximizează verosimilitatea datelor, $P(D|\theta)$. Însă rezultatul $P(\theta_{MAP})P(D|\theta_{MAP}) < P(\theta_{ML})P(D|\theta_{ML})$ constituie o contradicţie, fiindcă θ_{MAP} prin definiţie maximizează produsul $P(\theta)P(D|\theta)$. Prin urmare, presupunerea $\|\theta_{MAP}\|^2 > \|\theta_{ML}\|^2$ este falsă.

***Observaţii*:**

1. Remarcaţi faptul că în vreme ce la regresia liniară termenul de regularizare era precedat de semnul + (vedeţi relaţia (157) de la problema 3.C), în relaţia (204) termenul de regularizare este precedat de semnul −. Explicaţia rezidă în faptul că la regresia liniară — varianta LSE — se caută *minimul* sumei pătratelor erorilor, iar la regresia logistică se caută *maximul* funcţiei de verosimilitate condiţională.

2. Inegalitatea $\|\theta_{MAP}\|^2 \leq \|\theta_{ML}\|^2$ indusă de regularizarea de normă L_2 este numită uneori şi proprietatea de *diminuare a ponderilor* (engl., *weight decay*), fiindcă regularizarea aceasta încurajează ponderile / parametrii să ia valori care în general sunt mai mici (în valoare absolută) decât în cazul în care nu se foloseşte regularizare, ceea ce conduce la limitarea / prevenirea fenomenului de *overfitting*. Veţi observa din demonstraţie că această proprietate indusă de regularizarea L_2 este valabilă in general, nu doar pentru regresia logistică.[368]

17. (Regresia logistică local-ponderată regularizată (L_2): exprimarea vectorului gradient şi a matricei hessiene cu ajutorul matricei de design, X)

prelucrare de Liviu Ciortuz, după
□ • Stanford, 2007 fall, Andrew Ng, HW1, pr. 2

Conform relaţiilor (183) şi (204) de la problemele 13 şi respectiv 16, problema de regresie logistică cu regularizare L_2 constă în a maximiza o funcţie de log-verosimilitate condiţională extinsă cu un termen de regularizare:[369]

$$\sum_{i=1}^{n}[y^{(i)} \ln h_\theta(x^{(i)}) + (1 - y^{(i)}) \ln(1 - h_\theta(x^{(i)}))] - \frac{\lambda}{2}\|\theta\|^2, \tag{205}$$

unde $x^{(i)} \in \mathbb{R}^d$, $y^{(i)} \in \{0, 1\}$, $\theta \in \mathbb{R}^d$, iar h_θ desemnează o funcţie de tip sigmoidal / logistic definită prin $h_\theta(x) = \dfrac{1}{1 + e^{-\theta \cdot x}}$ pentru orice $x \in \mathbb{R}^d$.[370]

Făcând o uşoară *generalizare*, vom putea spune că problema de regresie logistică *local-ponderată* cu regularizare L_2 constă în a maximiza o funcţie de forma

$$\ell(\theta) = \sum_{i=1}^{n} w_i[y^{(i)} \ln h_\theta(x^{(i)}) + (1 - y^{(i)}) \ln(1 - h_\theta(x^{(i)}))] - \frac{\lambda}{2}\|\theta\|^2,$$

unde ponderea $w_i \in \mathbb{R}_+$ reprezintă „importanţa“ acordată instanţei de antrenament $(x^{(i)}, y^{(i)})$ în determinarea separatorului decizional produs de regresia

[368] Menţionăm că în contextul regresiei liniare, proprietatea de diminuarea a ponderilor apare la problema 26.
[369] Hiperparametrului λ din expresia (204) îi corespunde $\frac{\lambda}{2}$ în (205). Va fi mai convenabil aşa la derivare.
[370] Vectorului w de la problemele 13 şi 14 îi corespunde aici vectorul θ.

logistică pe aceste date. Separatorul decizional determinat de acest model, şi anume $\theta \cdot x = 0$, va fi tot liniar. Remarcaţi faptul că $-\frac{\lambda}{2}\|\theta\|^2 = -\frac{\lambda}{2}\theta^2$ din expresiile de mai sus este termenul de regularizare; el este necesar pentru a contracara fenomenul de *overfitting* (rom., supra-specializare).

a. Demonstraţi că vectorul gradient pentru $\ell(\theta)$ este

$$\nabla_\theta \ell(\theta) = X^\top z - \lambda\theta, \tag{206}$$

unde X este *matricea de design* (care are dimensiunea $n \times d$, fiecare linie a acestei matrice fiind reprezentată de câte o instanţă de antrenament), iar vectorul-coloană $z \in \mathbb{R}^n$ este de forma $(z_1, \ldots, z_n)^\top$, cu

$$z_i = w_i(y^{(i)} - h_\theta(x^{(i)})), \text{ pentru } i = 1, \ldots, n.$$

b. Demonstraţi că matricea hessiană pentru $\ell(\theta)$ este

$$H = X^\top D X - \lambda I,$$

unde $D \in \mathbb{R}^{n\times n}$ este o matrice diagonală, în care elementul generic (notat cu D_{ii}) este

$$D_{ii} = -w_i h_\theta(x^{(i)})(1 - h_\theta(x^{(i)})).$$

***Observaţie*: Dacă se consideră $w_i = 1$ pentru $i = 1, \ldots, m$ şi se elimină termenul corespunzător regularizării — adică, se revine la modelul de regresie logistică introdus la problema 13 —, atunci expresia (206) va corespunde expresiei (185) de la problema 13 (şi totodată expresiei (190) de la problema 14). De asemenea, elementul generic al matricei H de mai sus va corespunde expresiei (191) de la problema 14. În practică, folosirea matricei de design X este recomandată a fi folosită ori de câte ori este posibil, din motive de eficienţă computaţională.**

Răspuns:[371]

a. Mai întâi vom calcula expresia termenului generic pentru vectorul gradient $\nabla_\theta l(\theta)$, generalizând uşor raţionamentul de la problema 14, adică adăugând ponderile w_i, precum şi termenul de regularizare $-\frac{\lambda}{2}\theta^2$. Vă reamintim că $\frac{\partial}{\partial\theta_k} h_\theta(x) = h_\theta(x)(1 - h_\theta(x))x_k$.

$$\begin{aligned}
\frac{\partial}{\partial\theta_k}\ell(\theta) &= \sum_{i=1}^m w_i \left[y^{(i)} \frac{\partial}{\partial\theta_k} \ln(h_\theta(x^{(i)})) + (1 - y^{(i)}) \frac{\partial}{\partial\theta_k} \ln(1 - h_\theta(x^{(i)})) \right] - \frac{\lambda}{2}\frac{\partial}{\partial\theta_k}\theta^2 \\
&= \sum_{i=1}^m w_i \Bigg[y^{(i)} \cdot \frac{1}{\cancel{h_\theta(x^{(i)})}} \cdot \cancel{h_\theta(x^{(i)})}(1 - h_\theta(x^{(i)})) \cdot x_k^{(i)} + \\
&\qquad + (1 - y^{(i)}) \cdot \frac{1}{\cancel{1 - h_\theta(x^{(i)})}} \cdot (-1) \cdot h_\theta(x^{(i)}) \cancel{(1 - h_\theta(x^{(i)}))} \cdot x_k^{(i)} \Bigg] - \frac{\lambda}{\cancel{2}} \cdot \cancel{2}\theta_k \\
&= \sum_{i=1}^m w_i \left[y^{(i)} \cdot (1 - h_\theta(x^{(i)})) x_k^{(i)} - (1 - y^{(i)}) \cdot h_\theta(x^{(i)}) x_k^{(i)} \right] - \lambda\theta_k \\
&= \sum_{i=1}^m w_i \left(y^{(i)} x_k^{(i)} - \cancel{y^{(i)} h_\theta(x^{(i)}) x_k^{(i)}} - h_\theta(x^{(i)}) x_k^{(i)} + \cancel{y^{(i)} h_\theta(x^{(i)}) x_k^{(i)}} \right) - \lambda\theta_k
\end{aligned}$$

[371] Soluţia care urmează a fost redactată într-o formă iniţială de către Mădălina Racoviţă, MSc student, UAIC, Iaşi, 2020.

$$= \sum_{i=1}^{m} \underbrace{w_i\big(y^{(i)} - h_\theta(x^{(i)})\big)}_{z_i} x_k^{(i)} - \lambda\theta_k$$

$$= \sum_{i=1}^{m} z_i x_k^{(i)} - \lambda\theta_k, \text{ pentru } k = 1, \ldots, d.$$

În consecinţă, vectorul gradient, $\nabla_\theta \ell(\theta)$, se scrie astfel:

$$\nabla_\theta \ell(\theta) = \sum_{i=1}^{m} z_i x^{(i)} - \lambda\theta.$$

Acum vom arăta că $\nabla_\theta \ell(\theta) = X^\top z - \lambda\theta$, unde X este *matricea de design*. Conform rezultatului demonstrat mai sus, va fi suficient să arătăm că $X^\top z = \sum_{i=1}^{m} z_i x^{(i)}$. Într-adevăr,

$$X^\top z = ({x^{(1)}}^\top, \ldots, {x^{(m)}}^\top) \times (z_1, \ldots, z_m)^\top = \begin{pmatrix} x_1^{(1)} & x_1^{(2)} & \ldots & x_1^{(m)} \\ \vdots & \vdots & & \vdots \\ x_d^{(1)} & x_d^{(2)} & \ldots & x_d^{(m)} \end{pmatrix} \times \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_m \end{pmatrix}$$

$$= \left(\sum_{i=1}^{m} z_i x_1^{(i)}, \ldots, \sum_{i=1}^{m} z_i x_d^{(i)} \right)^\top = \sum_{i=1}^{m} z_i x^{(i)}.$$

b. Mai întâi vom calcula expresia termenului generic al matricei hessiene pentru funcţia de log-verosimilitate condiţională regularizată, $\ell(\theta)$.

$$H_{k,l} = \frac{\partial^2}{\partial\theta_l\, \partial\theta_k} \ell(\theta) = \frac{\partial}{\partial\theta_l} \left[\sum_{i=1}^{m} w_i \left(y^{(i)} - h_\theta(x^{(i)}) \right) x_k^{(i)} - \lambda\theta_k \right]$$

$$= \sum_{i=1}^{m} w_i \cdot (-1) \cdot \left(\frac{\partial}{\partial\theta_l} h_\theta(x^{(i)}) \right) \cdot x_k^{(i)} - \lambda \frac{\partial}{\partial\theta_l} \theta_k$$

$$= -\sum_{i=1}^{m} w_i\, h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))\, x_l^{(i)}\, x_k^{(i)} - \lambda \frac{\partial}{\partial\theta_l} \theta_k, \text{ pentru orice } k, l \in \{1, \ldots, d\}.$$

Aşadar,

$$H_{k,l} = \begin{cases} -\sum_{i=1}^{m} w_i\, h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))\, (x_k^{(i)})^2 - \lambda & \text{pentru } k = l, \\ -\sum_{i=1}^{m} w_i\, h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))\, x_l^{(i)} x_k^{(i)} & \text{pentru } k \neq l. \end{cases}$$

Folosind notaţia $D_{ii} = -w_i h_\theta(x^{(i)})(1 - h_\theta(x^{(i)}))$ dată în enunţ, putem scrie:

$$H_{k,l} = \begin{cases} \sum_{i=1}^{m} D_{ii} (x_k^{(i)})^2 - \lambda & \text{pentru } k = l, \\ \sum_{i=1}^{m} D_{ii} x_l^{(i)} x_k^{(i)} & \text{pentru } k \neq l. \end{cases}$$

Similar cu modul cum am procedat la punctul a, vom arăta că matricea H este egală cu $X^\top DX - \lambda I$, ceea ce revine la a demonstra că $X^\top DX = H + \lambda I$.

$$X^\top DX = \big(x^{(1)}, \ldots, x^{(m)}\big) \times \begin{bmatrix} D_{11} & \ldots & 0 \\ & \ddots & \\ 0 & \ldots & D_{mm} \end{bmatrix} \times \big(x^{(1)}, \ldots, x^{(m)}\big)^\top$$

$$= (D_{11}\, x^{(1)}, \ldots, D_{mm}\, x^{(m)}) \times \big(x^{(1)}, \ldots, x^{(m)}\big)^\top = \sum_{i=1}^{m} D_{ii}\, x^{(i)} {x^{(i)}}^\top$$

$$= \begin{bmatrix} \sum_{i=1}^m D_{ii} \cdot \left(x_1^{(i)}\right)^2 & \sum_{i=1}^m D_{ii} \cdot x_1^{(i)} x_2^{(i)} & \dots & \sum_{i=1}^m D_{ii} \cdot x_1^{(i)} x_d^{(i)} \\ \sum_{i=1}^m D_{ii} \cdot x_2^{(i)} x_1^{(i)} & \sum_{i=1}^m D_{ii} \cdot \left(x_2^{(i)}\right)^2 & \dots & \sum_{i=1}^m D_{ii} \cdot x_2^{(i)} x_d^{(i)} \\ \vdots & \vdots & \vdots & \vdots \\ \sum_{i=1}^m D_{ii} \cdot x_d^{(i)} x_1^{(i)} & \sum_{i=1}^m D_{ii} \cdot x_d^{(i)} x_2^{(i)} & \dots & \sum_{i=1}^m D_{ii} \cdot \left(x_d^{(i)}\right)^2 \end{bmatrix}$$

$$= H + \lambda I.$$

18. **(Regresia logistică: kernel-izare în cazul folosirii metodei gradientului [ascendent])**

prelucrare de Liviu Ciortuz, după CMU, 2005 fall, Tom Mitchell, HW3, pr. 2.ab

■ □ • ○

Introducere: Ideea centrală din spatele metodei kernel-izării, aşa cum este folosită în clasificare,[372] este transformarea / „maparea“ vectorilor care reprezintă instanţe de antrenament, provenind dintr-un spaţiu $\mathcal{X}$ care [în mod normal] are un număr mic de dimensiuni, într-un alt spaţiu $\mathcal{Z}$ (numit îndeobşte *spaţiu de trăsături*), care are [de obicei] un număr mare de dimensiuni. O astfel de transformare poate da naştere [în anumite condiţii] unui clasificator mai flexibil, care moşteneşte însă eficienţa computaţională din spaţiul iniţial $\mathcal{X}$. Mai concret, *kernel-izarea* implică găsirea unei transformări (sau, a unei mapări, de la engl. mapping) $\phi : \mathcal{X} \to \mathcal{Z}$, astfel încât

i. spaţiul $\mathcal{Z}$ să aibă un număr de dimensiuni mai mare decât cel al spaţiului $\mathcal{X}$;
ii. calculele pe care ar / va trebui să le facem cu vectori din spaţiul $\mathcal{Z}$ să se limiteze doar la folosirea produsului scalar;
iii. să existe o funcţie $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, astfel încât pentru orice $x^{(i)}$ şi $x^{(j)}$ din $\mathcal{X}$ produsul scalar dintre $\phi(x^{(i)})$ şi $\phi(x^{(j)})$ să fie egal cu $K(x^{(i)}, x^{(j)})$, însă valoarea $K(x^{(i)}, x^{(j)})$ să poată fi calculată folosind în mod direct $x^{(i)}$ şi $x^{(j)}$ (adică fără a face apel la $\phi(x^{(i)})$ şi $\phi(x^{(j)})$). Vom numi K *funcţie-nucleu* (engl., kernel function).[373]

Veţi vedea că un clasificator liniar din spaţiul $\mathcal{Z}$ va corespunde (în general) unui clasificator neliniar din spaţiul $\mathcal{X}$.

Aşa cum am arătat deja la problema 13, în varianta *standard* a regresiei logistice se consideră

$$P(Y = 1|X) = \sigma(w_0 + \sum_{i=1}^{d} w_i X_i)$$
$$P(Y = 0|X) = 1 - P(Y = 1|X),$$

unde σ este funcţia logistică, definită prin relaţia $\sigma(a) \stackrel{def.}{=} 1/(1 + e^{-a})$ pentru orice $a \in \mathbb{R}$.

Fie o funcţie ϕ care transformă o instanţă oarecare X din spaţiul $\mathcal{X}$ (de dimensiune d) într-un element din spaţiul $\mathcal{Z}$ (a cărui dimensine este m, cu $m > d$). Folosind „maparea“ ϕ, vom scrie:

[372] Metoda kernel-izării a fost introdusă iniţial pentru maşinile cu vectori-suport (engl., Support Vector Machines, SVM). Puteţi consulta capitolul respectiv din prezenta culegere şi (eventual, în prealabil) secţiunea *Funcţii-nucleu* de la capitolul de *Fundamente*.

[373] Spre exemplu, $K(x, x') \stackrel{def.}{=} \frac{1}{\sqrt{2\pi}\tau} \exp\left(\frac{\|x - x'\|^2}{2\tau^2}\right)$ este o astfel de funcţie-nucleu. Ea se numeşte nucleu gaussian sau funcţie cu baza radială (engl, Radial Basis Function, RBF).

$$P(Y=1|\phi(X)) = \sigma(w_0 + \sum_{i=1}^{m} w_i\, \phi(X)_i). \tag{207}$$

unde $\phi(X)$ este vectorul de „trăsături“ (engl., feature vector) m-dimensional, care îi corespunde vectorului X din spaţiul $\mathcal{X}$ (d-dimensional). În cele ce urmează, vom nota cu $\phi(X)_i$ componenta i a vectorului $\phi(X)$.

a. Presupunem că vectorul de ponderi w este o combinaţie liniară de toţi vectorii de trăsături $\phi(X^{(i)})$ cu $i = 1, \ldots, n$, unde n este numărul instanţelor de antrenament.[374] Din punct de vedere formal, putem exprima acest fapt astfel:

$$(w_1, \ldots, w_m)^\top = \sum_{j=1}^{n} \alpha_j \phi(X^{(j)}), \tag{208}$$

unde $X^{(j)}$ este instanţa cu numărul de ordine j, iar α_j sunt constante reale ($j = 1, \ldots, n$). Presupunem de asemenea că $w_0 = \alpha_0$.

Exprimaţi probabilitatea condiţionată $P(Y = 1|\phi(X))$ folosind procedeul kernelizării (engl., kernel trick). (Astfel veţi evita să faceţi calculele în mod explicit în spaţiul $\mathcal{Z}$.)

b. Deduceţi expresia funcţiei de log-verosimilitate condiţională $\ell(\alpha)$. Apoi determinaţi *regula de actualizare* specifică metodei gradientului ascendent pentru varianta kernel-izată a regresiei logistice, i.e. pentru maximizarea funţiei $\ell(\alpha)$.

c. Scrieţi *regula de decizie* pentru o instanţă oarecare de test X din spaţiul $\mathcal{X}$, presupunând că parametrii α_i (pentru $i = 1, \ldots, m$) au fost deja învăţaţi.

Răspuns:

a. Pornind de la relaţia (207), putem exprima probabilitatea condiţionată $P(Y = 1|\phi(X))$ astfel:

$$\begin{aligned} P(Y=1|\phi(X)) &= \sigma\big(w_0 + \sum_{i=1}^{m} w_i\, \phi(X)_i\big) = \sigma(w_0 + w \cdot \phi(X)) \\ &\overset{(208)}{=} \sigma\big(w_0 + \big(\sum_{j=1}^{n} \alpha_j \phi(X^{(j)})\big) \cdot \phi(X)\big) = \sigma\big(w_0 + \sum_{j=1}^{n} \alpha_j (\phi(X^{(j)})) \cdot \phi(X)\big) \\ &= \sigma\big(\alpha_0 + \sum_{j=1}^{n} \alpha_i K(X^{(j)}, X)\big). \end{aligned}$$

b. În cele ce urmează, vom folosi deja-cunoscutul „artificiu al exponenţierii“ necesar pentru a exprima în mod unitar probabilităţile $P(Y = 1|\phi(X))$ şi $P(Y = 0|\phi(X))$:[375]

[374] Dacă aplicăm metoda gradientului [ascendent] pornind cu valoarea $w = 0$, conform regulii de actualizare (184) şi expresiei (186) din cadrul problemei 13, rezultă că într-adevăr vom obţine vectorul de ponderi w sub forma unei combinaţii liniare de instanţe $X^{(j)}$ (respectiv $\phi(X^{(j)})$ în cazul kernel-izării).

Mai general, din *Teorema de reprezentare* care este demonstrată la problema 88 de la capitolul de *Fundamente*, rezultă că orice vector de ponderi w pentru care se atinge maximul funcţiei de log-verosimilitate condiţională a datelor pentru regresia logistică — vedeţi relaţia (183) — se poate scrie în modul indicat mai sus.

[375] Am folosit deja acest „artificiu“ la rezolvarea problemei 13.b.

$$P(Y = y|\phi(X)) \quad = \quad \sigma(z)^y(1-\sigma(z))^{1-y} \textbf{ pentru orice } y \in \{0,1\}, \tag{209}$$

unde

$$z \overset{not.}{=} w_0 + w \cdot \phi(X) = \alpha_0 + \sum_{j=1}^{n} \alpha_j K(X^{(j)}, X). \tag{210}$$

Prin aplicarea funcţiei ln**, relaţia** (209) **devine:**

$$\ln P(Y = y|\phi(X)) = y \ln \sigma(z) + (1-y)\ln(1-\sigma(z)) \tag{211}$$

$$\tag{212}$$

Întrucât va trebui să exprimăm funcţia de log-verosimilitate condiţională ℓ **în raport** α**, iar** z **a fost definit mai sus folosind coeficienţii** α_j**, vom explicita** $\sigma(z)$ **din relaţia** (212)**:**

$$\begin{aligned} \sigma(z) &= \frac{1}{1+e^{-z}} = \frac{e^z}{1+e^z} \\ \Rightarrow 1-\sigma(z) &= \frac{e^{-z}}{1+e^{-z}} = \frac{1}{1+e^z} \\ \Rightarrow \ln(1-\sigma(z)) &= -\ln(1+e^z), \end{aligned}$$

pentru orice $z \in \mathbb{R}$**.**

Prin urmare, vom putea exprima log-verosimilitatea condiţională a unei instanţe de antrenament oarecare $(\phi(X), Y = y)$ **astfel:**

$$\begin{aligned} &\ln P(Y = y|\phi(X)) = y \ln \sigma(z) + (1-y)\ln(1-\sigma(z)) \\ &= \quad y \ln \frac{e^z}{1+e^z} + (1-y)\ln \frac{1}{1+e^z} \\ &= \quad yz - \cancel{y \ln(1+e^z)} - \ln(1+e^z) + \cancel{y \ln(1+e^z)} \\ &= \quad yz - \ln(1+e^z). \end{aligned} \tag{213}$$

Datele de antrenament fiind îndeobşte considerate independente şi identic distribuite (abrev., i.i.d.), vom putea scrie verosimilitatea condiţionată a setului de date $(\phi(X^{(1)}), Y^{(1)})$**, ...,** $(\phi(X^{(n)}), Y^{(n)})$ **astfel:**

$$P(Y^{(1)}, \ldots, Y^{(n)}|\phi(X^{(1)}), \ldots, \phi(X^{(n)}); \alpha) \overset{i.i.d.}{=} \prod_{l=1}^{n} P(Y^{(l)}|\phi(X^{(l)}); \alpha),$$

unde $\alpha \overset{not.}{=} (\alpha_1, \ldots, \alpha_n)$ **este vectorul de coeficienţi care au fost introduşi la relaţia** (208)**.**

Aşadar, log-verosimilitatea condiţională a setului de date $\{(\phi(X^{(1)}), Y^{(1)})$**, ...,** $(\phi(X^{(n)}), Y^{(n)})\}$**, exprimată ca funcţie de** α**, este următoarea:**

$$\begin{aligned} \ell(\alpha) &= \ln \prod_{l=1}^{n} P(Y^{(l)}|\phi(X^{(l)}); \alpha) = \sum_{l=1}^{n} \ln P(Y^{(l)}|\phi(X^{(l)}); \alpha) \\ &\overset{(210)\,(213)}{=} \sum_{l=1}^{n} \Big[Y^{(l)} \big(\alpha_0 + \sum_{j=1}^{n} \alpha_j K(X^{(j)}, X^{(l)})\big) - \ln\big(1 + \exp(\alpha_0 + \sum_{j=1}^{n} \alpha_j K(X^{(j)}, X^{(l)}))\big)\Big]. \end{aligned}$$

De aici se obţine uşor derivata parţială a funcţiei $\ell(\alpha)$ în raport cu componenta α_i, pentru fiecare $i = 1, \ldots, n$:[376]

$$\begin{aligned}\frac{\partial\,\ell(\alpha)}{\partial\alpha_i} &= \sum_{l=1}^{n}\left(Y^{(l)} - \frac{\exp(\alpha_0 + \sum_{j=1}^{n}\alpha_j K(X^{(j)}, X^{(l)}))}{1+\exp(\alpha_0 + \sum_{j=1}^{n}\alpha_j K(X^{(j)}, X^{(l)}))}\right) K(X^{(i)}, X^{(l)}) \\ &= \sum_{l=1}^{n}\Big(Y^{(l)} - \sigma(\alpha_0 + \sum_{j=1}^{n}\alpha_j K(X^{(j)}, X^{(l)}))\Big) K(X^{(i)}, X^{(l)}). \qquad (214)\end{aligned}$$

şi respectiv derivata parţială a funcţiei $\ell(\alpha)$ în raport cu componenta α_0:

$$\begin{aligned}\frac{\partial\,\ell(\alpha)}{\partial\alpha_0} &= \sum_{l=1}^{n}\left(Y^{(l)} - \frac{\exp(\alpha_0 + \sum_{j=1}^{n}\alpha_j K(X^{(j)}, X^{(l)}))}{1+\exp(\alpha_0 + \sum_{j=1}^{n}\alpha_j K(X^{(j)}, X^{(l)}))}\right) \\ &= \sum_{l=1}^{n}\Big(Y^{(l)} - \sigma(\alpha_0 + \sum_{j=1}^{n}\alpha_j K(X^{(j)}, X^{(l)}))\Big).\end{aligned}$$

Regula de actualizare specifică metodei gradientului ascendent aplicată la rezolvarea acestei probleme de regresie logistică kernel-izată este:

$$\alpha_i^{(t+1)} = \alpha_i^{(t)} + \eta\frac{\partial\,\ell(\alpha)}{\partial\alpha_i} \text{ pentru } i = 1, \ldots, n,$$

sau, scriind vectorial: $\alpha^{(t+1)} = \alpha^{(t)} + \eta\nabla_\alpha\ell(\alpha)$.

c. Conform relaţiei (207), modelul învăţat de regresia logistică asociază instanţei X eticheta $Y = 1$ dacă şi numai dacă $\sigma(w_0 + \sum_{i=1}^{m} w_i\,\phi(X)_i) \geq 1/2$, ceea ce este echivalent cu $w_0 + \sum_{i=1}^{m} w_i\,\phi(X)_i \geq 0$, adică $w_0 + w\cdot\phi(X) \geq 0$. Ţinând cont de relaţia (208) şi de faptul că $w_0 = \alpha_0$, această ultimă inegalitate revine la $\alpha_0 + \sum_{j=1}^{n}\alpha_j\phi(X^{(j)})\cdot\phi(X) \geq 0$, adică

$$\alpha_0 + \sum_{j=1}^{n}\alpha_j K(X^{(j)}, X) \geq 0.$$

19. **(Regresia logistică: clasificare n-ară [regresie *softmax*] cu regularizare L_2, folosind metoda gradientului ascendent)**

■ □ • ○ ***CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, HW2, pr. 2***
MIT, 2016 fall, R. Barzilay, S. Sra, Weekly Exercises, week 4, pr. 5.ad

În acest exerciţiu vom arăta că putem extinde uşor modelul binar al regresiei logistice în aşa fel încât să realizăm clasificare n-ară. Să presupunem că avem K clase distincte şi că probabilitatea a posteriori pentru clasa k este definită prin

$$\begin{aligned}P(Y = k|X = x) &= \frac{\exp(w_k\cdot x)}{1+\sum_{l=1}^{K-1}\exp(w_t\cdot x)} \text{ pentru } k = 1, \ldots, K-1 \\ P(Y = K|X = x) &= \frac{1}{1+\sum_{l=1}^{K-1}\exp(w_t\cdot x)},\end{aligned}$$

[376] A se compara expresia (214) cu expresia (189) care a fost obţinută la problema 13, pentru regresia logistică nekernel-izată.

unde x şi w_k cu $k = 1, \ldots, K$ sunt vectori d-dimensionali. Remarcaţi că pentru a simplifica expresia acestor probabilităţi nu am folosit componentele w_{k0}.[377] Remarcaţi de asemenea că pentru $K = 2$ se obţine regresia logistică binară, aşa cum a fost prezentată la problema 13.

***Observaţie*: Putem simplifica expresia de mai sus pentru regresia logistică n-ară (engl., multiclass logistic regression) introducând un parametru fix, vectorul $w_K = 0$ (un vector d-dimensional format în întregime din zerouri).[378]**

***Scopul* nostru este să estimăm parametrii / ponderile w_t folosind ca metodă de optimizare gradientul ascendent. Vom stabili de asemenea distribuţiile *a priori* pe care le urmează parametrii w_t, pentru a evita ca ei să [ajungă să] aibă valori foarte mari [în normă euclidiană] şi să se producă *overfitting* (rom., supra-specializare).**

a. Presupunem că se dă setul de date de antrenament $\{(x^1, y^1), \ldots (x^n, y^n)\}$, cu $x^i \in \mathbb{R}^d$ şi $y^i \in \{1, \ldots, K\}$ pentru $i = 1, \ldots, n$. Scrieţi expresia funcţiei de log-verosimilitate condiţională, $\ell(w_1, \ldots, w_K)$, folosind *regularizare* de normă L_2 pentru ponderi.[379] Arătaţi paşii pe care i-aţi urmat ca să deduceţi această expresie.

b. Remarcaţi că, asemenea cazului clasificării binare (vedeţi problema 13), nu există o expresie pentru calculul direct (engl., closed form) al valorii parametrului w_k care corespunde maximului funcţiei de log-verosimilitate condiţională, $\ell(w_1, \ldots, w_K)$. Totuşi, putem găsi soluţia folosind *metoda gradientului ascendent*, cu ajutorul derivatelor parţiale. Calculaţi expresia componentei de pe poziţia k din vectorul gradient pentru $\ell(w_1, \ldots, w_K)$, şi anume derivata parţială a lui $\ell(w_1, \ldots, w_K)$ în raport cu w_k.

c. Dând ponderilor valoarea iniţială 0, scrieţi *regula de actualizare* (engl., update rule) pentru w_k, folosind η ca rată a învăţării (engl., step size). Va converge oare soluţia la maximul global?

Răspuns:

a. Presupunem că dispunem de n instanţe de antrenament. Folosind funcţia-indicator $1_{\{l=k\}}$, care este definită prin $1_{\{l=k\}} = 1$ dacă $Y^l = k$ şi 0 în caz contrar, putem scrie funcţia de verosimilitate condiţională astfel:

$$L(w_1, \ldots, w_K) = \prod_{l=1}^{n} \prod_{k=1}^{K} P(Y^l = k | X^l = x^l; w)^{1_{\{l=k\}}} = \prod_{l=1}^{n} \prod_{k=1}^{K} \left(\frac{\exp(w_k \cdot x^l)}{\sum_r \exp(w_r \cdot x^l)} \right)^{1_{\{l=k\}}}.$$

[377] Alternativ, se poate considera că extindem „artificial“ vectorul x — şi, la fel, fiecare vector x^l din setul de date de antrenament — cu componenta $x_0 = 1$ şi, respectiv, $x_0^l = 1$.

[378] Funcţia $f : \{w_1 \cdot x, \ldots, w_K \cdot x\} \to [0, 1]$ definită prin $f(w_k \cdot x) = \dfrac{\exp(w_k \cdot x)}{1 + \sum_{l=1}^{K-1} \exp(w_t \cdot x)}$ şi numită *funcţia softmax*, are proprietatea că menţine ordinea, adică $w_k \cdot x < w_j \cdot x \Rightarrow f(w_k \cdot x) < f(w_j \cdot x)$. Întrucât, de asemenea, f este derivabilă în raport cu w_k — ceea ce justifică particula *soft* în denumirea softmax —, această funcţie este folosită adeseori ca funcţie de activare pentru stratul de ieşire al reţelelor neuronale cu output multiplu. În mecanica statistică funcţia softmax este cunoscută sub numele *distribuţia Bolzmann* (introdusă de fizicianul austriac Ludwig Bolzmann în anul 1868) sau *distribuţia Gibbs*.

[379] Termenii de *regularizare* şi *overfitting* au fost introduşi pentru regresia liniară la problema 3.C. La secţiunea de *Regresie logistică*, am folosit pentru prima dată regularizarea (de normă) L_2 la problema 16.

Aplicând funcţia $\ln$**, vom obţine funcţia de log-verosimilitate condiţională:**

$$\ell(w_1, \ldots, w_K) = \sum_{l=1}^{n} \sum_{k=1}^{K} 1_{\{l=k\}} \left(\left(w_k \cdot x^l - \ln \sum_r \exp(w_r \cdot x^l) \right) \right).$$

Adăugând termenul de regularizare corespunzător normei L_2**, rezultă:**[380]

$$\ell(w_1, \ldots, w_K) = \sum_{l=1}^{n} \sum_{k=1}^{K} 1_{\{l=k\}} \left(w_k \cdot x^l - \ln \sum_r \left(\exp(w_r \cdot x^l) \right) \right) - \frac{\lambda}{2} \sum_{k=1}^{K} \|w_k\|^2.$$

b. Calculăm derivata parţială — de fapt, vectorul de derivate parţiale — pentru funcţia de log-verosimilitate condiţională, în raport cu parametrul w_i**:**[381]

$$\begin{aligned} \frac{\partial}{\partial w_i} \ell(w_1, \ldots, w_K) &= \sum_{l=1}^{n} \left(1_{\{l=i\}} \, x^l - \frac{\exp(w_i \cdot x^l) \, x^l}{\sum_r \exp(w_r \cdot x^l)} \right) - \lambda w_i \\ &= \sum_{l=1}^{n} \left(1_{\{l=i\}} - P(Y^l = i | X^l = x^l) \right) x^l - \lambda w_i. \end{aligned}$$

c. Regula de actualizare pentru parametrul w_i**, folosind gradientul ascendent, este următoarea:**

$$w_i \leftarrow w_i + \eta \sum_{l=1}^{n} \left(1_{\{l=i\}} - P(Y^l = i | X^l = x^l) \right) x^l - \eta \lambda w_i.$$

Convergenţa la punctul de maxim global are loc întrucât funcţia de log-verosimilitate condiţională este concavă.[382]

[380] Vedeţi problema 16. Hiperparametrului λ din expresia (204) îi corespunde $\frac{}{2}$ în (205). Va fi mai convenabil aşa la derivare.

[381] Vă reamintim că acest tip de derivată se numeşte *derivată vectorială*. Pentru expresia $w \cdot x = w^\top x$, derivata în raport cu vectorul w este vectorul x. (Vedeţi documentul *Matrix Identities* de Sam Roweis.)

[382] Demonstraţia acestei proprietăţi extinde în mod natural demonstraţia din cazul clasificării binare, care a fost prezentată la problema 14.

1.2 Metode de regresie — Probleme propuse

1.2.1 Regresia liniară

20. **(Regresia liniară unidimensională: deducerea directă a soluţiei analitice MLE / LSE)**

• · ***CMU, 2011 fall, Eric Xing, HW1, pr. 2.1***

Considerăm un caz simplu de regresie liniară, în care modelul [parametric] este de forma $y = ax + b + \varepsilon$, unde $x, y, a, b, \in \mathbb{R}$, iar ε este un termen „zgomot" care urmează o distribuţie gaussiană de medie zero, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$.[383] Avem n instanţe de antrenament, (x_i, y_i), $i = 1, 2, \ldots, n$, cu $x_i, y_i \in \mathbb{R}$.

a. Scrieţi expresia funcţiei de cost reprezentând suma pătratelor erorilor în acest model. *Nu* folosiţi forma matriceală. Funcţia aceasta va fi exprimată folosind modelul parametric unidimensional (1-D).

b. Calculaţi valorile $\hat{a}$ şi $\hat{b}$ care minimizează funcţia de cost / pierdere şi arătaţi că

$$\begin{aligned} \hat{a} &= \frac{n(\sum_i x_i y_i) - (\sum_i x_i)(\sum_i y_i)}{n(\sum_i x_i^2) - (\sum_i x_i)^2} \\ \hat{b} &= \frac{(\sum_i x_i^2)(\sum_i y_i) - (\sum_i x_i)(\sum_i x_i y_i)}{n(\sum_i x_i^2) - (\sum_i x_i)^2}, \end{aligned}$$

bineînţeles, în condiţiile în care expresia de la numitorul celor două fracţii este nenulă. Ca şi mai sus, *nu* folosiţi notaţia matriceală. Minimizaţi în manieră directă funcţia de cost / pierdere de la punctul a.

c. Deduceţi formula pentru estimarea de verosimilitate maximă (MLE) pentru modelul 1-D. Arătaţi că estimarea aceasta (de tip MLE) coincide cu estimarea corespunzătoare minimizării *criteriului sumei pătratelor erorilor*, obţinută la punctul a.

d. Arătaţi că soluţia optimă calculată la punctul b este aceeaşi cu forma matriceală obţinută în cazul modelului [mai] general de regresie liniară,

$$\hat{\beta} = (X^\top X)^{-1} X^\top y,$$

unde, în cazul nostru $\hat{\beta} \overset{not.}{=} (\hat{a}, \hat{b})$, $X \in \mathbb{R}^{n \times 2}$, cu linia i din X fiind $(1, x_i)$ pentru $i = 1, \ldots, n$, iar $y = (y_1, \ldots, y_n)^\top$.[384]

[383] Aşadar, acest model este uşor mai elaborat decât cel care a fost prezentat la problema 1, dar nu [atât de] general precum cel de la problema 3.A.

[384] Vedeţi relaţia (161) de la problema 3.d.

21. **(Compararea a două modele de regresie unidimensională de tip LSE)**

□ • · ***CMU, 2014 fall, Z. Bar-Joseph, W. Cohen, HW2, pr. 3***

Fie X şi Y două variabile aleatoare, iar $(x_1, y_1), \ldots, (x_n, y_n)$ un set de n exemple de antrenament generate în mod independent.

Considerăm că modelele de regresie de mai jos sunt de tipul sumei celor mai mici pătrate (engl., least squared errors, LSE). Aşadar, „zgomotul" ε urmează o distribuţie gaussiană $N(0, \sigma^2)$

i. $Y = \beta X + \varepsilon$

ii. $Y = \beta^2 X + \varepsilon$.

a. Calculaţi soluţiile acestor două modele de regresie.

b. Care dintre cele două modele de mai sus produce o eroare la antrenare mai mică?

***Sugestie*: Răspunsul poate să depindă de setul de date de antrenament. În consecinţă, explicaţi în ce situaţie un anumit model este mai bun decât celălalt.**

22. **(Regresia liniară: aplicaţie pentru „învăţarea" parametrului unei funcţii neliniare, componenta „zgomot" fiind modelată cu o distribuţie gaussiană)**

• ***CMU, 2009 spring, Tom Mitchell, midterm, pr. 6***

Se dă un set de date constituit din n instanţe, (x_1, y_1), (x_2, y_2), ..., (x_n, y_n). Atât atributul de intrare x cât şi atributul de ieşire y au ca valori numere reale. Valoarea atributului de ieşire este generată de o distribuţie gaussiană având media $\sin(wx_i)$ şi varianţa 1:

$$p(y_i \mid x_i, w) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(y_i - \sin(wx_i))^2}{2}}$$

În modelul de mai sus avem un parametru necunoscut w şi dorim să deducem / „învăţăm" din aceste date care este estimarea sa în sensul verosimilităţii maxime.

a. Scrieţi expresia verosimilităţii [condiţionale a] datelor ca funcţie de parametrul w, date fiind valorile observate x_i şi y_i, cu $i = 1, \ldots, n$.

b. Precizaţi care dintre egalităţile de mai jos este adevărată atunci când se doreşte / face estimarea parametrului w în sensul maximizării verosimilităţii datelor. În cazul în care niciuna dintre relaţiile i-v nu este adevărată, marcaţi opţiunea vi.

i. $\sum_i \cos^2(wx_i) = \sum_i y_i \sin(x_i)$

ii. $\sum_i \cos^2(wx_i) = \sum_i y_i \sin(2wx_i)$

iii. $\sum_i \sin(wx_i)\cos(wx_i) = \sum_i y_i \sin(wx_i/2)$

iv. $\sum_i x_i \sin(wx_i)\cos(wx_i) = \sum_i x_i y_i \cos(wx_i)$

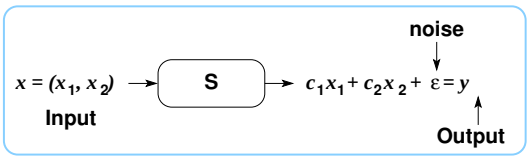
v. $w\cos(x_i) = \sum_i x_i y_i \cos(x_i)$

vi. **Niciuna dintre variantele de mai sus.**

23. **(Estimarea parametrilor unui model de regresie liniară bidimensională, în care componenta-zgomot este modelată fie cu o distribuţie gaussiană — inclusiv o variantă de regresie local-ponderată —, fie cu o distribuţie Laplace)**

■ □ • ○ ***CMU, 2009 fall, Carlos Guestrin, HW1, pr. 1.5.2***
CMU, 2012 fall, E. Xing, A. Singh, HW1, pr. 2

Figura alăturată reprezintă un sistem S care ia ca intrări [valorile] x_1 şi x_2 şi produce la ieşire o combinaţie liniară a celor două intrări, $c_1x_1 + c_2x_2$, unde c_1 şi c_2 sunt două numere reale necunoscute.



Dispozitivul folosit ca să măsoare outputul sistemului S introduce în plus o „eroare“ ε, care este [asimilată cu] o variabilă aleatoare ce urmează o anumită distribuţie. Astfel, outputul observat (y) este dat de ecuaţia

$$y = c_1x_1 + c_2x_2 + \varepsilon.$$

Presupunem că avem $n > 2$ instanţe (x_{j1}, x_{j2}, y_j), $j = 1, \ldots, n$ sau, echivalent, (x_j, y_j), $j = 1, \ldots, n$, unde $x_j = [x_{j1}, x_{j2}]$. Prin urmare, a avea la dispoziţie n măsurători revine la a avea n egalităţi de forma $y_j = c_1x_{j1} + c_2x_{j2} + \varepsilon_j$, cu $j = 1, \ldots, n$. Scopul nostru este să estimăm parametrii c_1 şi c_2 din aceste măsurători cu ajutorul criteriului verosimilităţii maxime, făcând [la fiecare dintre punctele de mai jos] diferite presupuneri în legătură cu „zgomotul“.

a. Presupunem că ε_i pentru $i = 1, \ldots, n$ sunt variabile aleatoare gaussiene i.i.d. (independente şi identic distribuite) cu media 0 şi varianţa σ^2. Calculaţi funcţia de log-verosimilitate a datelor. Apoi folosiţi această funcţie pentru a demonstra că estimarea de verosimilitate maximă $c^* = [c_1^*, c_2^*]$ este soluţia unei probleme de aproximare prin aşa-numita *metodă a celor mai mici pătrate* (engl., least squares approximation method). Nu este necesar să găsiţi efectiv soluţia problemei celor mai mici pătrate [pentru acest caz].

b. [Varianta regresiei local-ponderate] Presupunem că ε_i pentru $i = 1, \ldots, n$ sunt variabile aleatoare independente de tip gaussian având media 0 şi varianţa σ_i^2. Calculaţi funcţia de log-verosimilitate şi găsiţi perechea de valori $c^* = [c_1^*, c_2^*]$ care o maximizează, adică estimarea de verosimilitate maximă (MLE) a parametrilor c_1 şi c_2.

c. Presupunem că pentru orice $i = 1, \ldots, n$, variabila ε_i are funcţia de densitate

$$f_{\varepsilon_i}(x) = f(x) = \frac{1}{2\theta} e^{-\frac{|x|}{\theta}}, \text{ cu } \theta > 0.$$

Cu alte cuvinte, zgomotul este i.i.d. şi urmează o distribuţie Laplace având parametrul de *locaţie* $\mu = 0$ şi parametrul de *scalare* θ. Calculaţi funcţia de log-verosimilitate a datelor pentru acest model al zgomotului şi indicaţi motivul / motivele pentru care acest model conduce la *soluţii mai robuste* [la „zgomote“ şi outlier-e] decât în cazurile precedente.[385]

[385] Pentru reprezentarea grafică a mai multor distribuţii Laplace, vedeţi problema 8.B.

24. **(Regresia liniară bidimensională: exemplu de aplicare a metodei gradientului, varianta stochastică)**

• ***CMU, 2014 spring, Seyoung Kim, HW2, pr. 1.2.B***

În această problemă vom presupune că ori de câte ori se înmulţeşte o instanţă [reprezentată printr-un vector] x cu un vector de ponderi, x a fost deja augmentat [în aşa fel încât să înceapă] cu o componentă 1. De exemplu, x poate fi $(1, x_1)$ sau $(1, x_1, x_2)$, iar constanta 1 va fi înmulţită cu ponderea β_0.

Considerăm că folosim metoda gradientului pentru a antrena un model de regresie liniară pentru a „învăţa" funcţia $f : \mathbb{R}^2 \to \mathbb{R}$, ale cărei argumente (sau atribute) sunt notate cu x_1 şi respectiv x_2. Facem iniţializarea vectorului de ponderi astfel: $\beta = (0.5, 0.5, 0.5)$. Primul exemplu considerat este $x = (2, 6)$ şi, corespunzător, $y = 8.5$.

a. Folosind aceste valori iniţiale pentru ponderi, cât este eroarea $y - f(x) = y - x \cdot \beta$ produsă de modelul de regresie?

b. Cât va fi β_1^{new}, noua valoare a ponderii β_1 (care corespunde atributului / trăsăturii x_1) la finalul primei iteraţii executate de algoritmul gradientului descendent, varianta stochastică,[386] folosind exemplul indicat mai sus? Veţi utiliza rata de învăţare (engl., learning rate) $\eta = 0.05$.

În cele ce urmează vom presupune că am terminat de antrenat modelul de regresie liniară şi că am obţinut vectorul de ponderi $\hat{\beta} = (0.5, 1, 1)$. Acum vom considera instanţa de test $x_q = (3, 4)$ şi vom prezice $f(x_q)$. Presupunem că în prealabil am estimat în mod empiric varianţa componentei-zgomot (ε) din modelul de regresie şi am obţinut $\sigma = 1$.

c. Cât este valoarea „aşteptată" (engl., expected value) pentru $f(x_q)$?

d. Cât este probabilitatea ca valoarea adevărată pentru $f(x_q)$ să fie în intervalul $[9, 10]$?

***Sugestie*: Puteţi folosi tabela de valori ale funcţiei cumulative de distribuţie (Φ) pentru distribuţia gaussiană standard care apare la pr. 33 de la capitolul de *Fundamente*.**

25. **(Regresia liniară, varianta LSE: o proprietate interesantă: adăugarea de noi trăsături / atribute nu măreşte suma pătratelor erorilor)**

□ • ○ ***Stanford, 2014 fall, Andrew Ng, midterm, pr. 1***

După cum am arătat la problema 3.A, la regresia liniară în varianta LSE (în care modelarea „zgomotului" se face folosind distribuţia gaussiană) se lucrează cu o funcţie de cost de tipul sumei celor mai mici pătrate (engl., Least Squared Errors):[387]

$$J(\beta) = \sum_{i=1}^{n} (\beta^\top x^{(i)} - y^{(i)})^2 = (X\beta - y)^\top (X\beta - y).$$

[386] Pentru prezentarea metodei gradientului aplicată la rezolvarea regresiei liniare, vedeţi pr. 6.

[387] Corespondenţa dintre notaţiile de aici şi cele din problema 3.A este următoarea: $x^{(i)} \to X_i'$ şi $X \to X'$.

***Obiectivul* la aplicarea acestei metode de regresie este să găsim acea valoare a vectorului de parametri β pentru care se atinge minimul expresiei $J(\beta)$, pentru respectivul set de date de antrenament.**

Presupunem că în acest set de date avem iniţial d *trăsături* şi, în consecinţă, inputul la antrenare este dat sub forma lui $X \in \mathbb{R}^{n\times(d+1)}$, aşa-numita *matrice de design*.

Să zicem că acum avem acces la încă o trăsătură pentru fiecare exemplu de antrenament. Aşadar, avem un vector adiţional de trăsături $v \in \mathbb{R}^{n\times 1}$ pentru setul nostru de antrenament şi dorim să-l folosim în problema noastră de regresie. Putem face acest lucru creând o nouă matrice de design: $\tilde{X} \overset{not.}{=} [X\, v] \in \mathbb{R}^{n\times(d+2)}$. Prin urmare, noul nostru vector de parametri este $\beta_{new} \overset{not.}{=} [\beta, p]^\top$, unde $p \in \mathbb{R}$ este un parametru care corespunde noului vector de trăsături v.

***Observaţie importantă*: Pentru a asigura o anumită simplitate pentru demonstraţia matematică, în cele ce urmează vom presupune că $X^\top X = I \in \mathbb{R}^{(d+1)\times(d+1)}$ şi $\tilde{X}^\top \tilde{X} = I \in \mathbb{R}^{(d+2)\times(d+2)}$, $v^\top v = 1$. Aceasta este aşa-numita presupoziţie de *ortonormalitate*. Altfel spus, coloanele matricei X (respectiv $\tilde{X}$) sunt ortonormale. Concluziile demonstraţiei din această problemă se menţin chiar şi fără această presupoziţie de ortonormalitate, însă cu ea demonstraţia va fi mai uşoară.**

a. Fie $\hat{\beta} \overset{not.}{=} \arg\min_\beta J(\beta)$ valoarea vectorului de parametri β care minimizează funcţia obiectiv originală (cea care foloseşte matricea de design X). Folosind presupoziţia de ortornormalitate, demonstraţi că

$$J(\hat{\beta}) = (XX^\top y - y)^\top (XX^\top y - y),$$

adică, arătaţi că aceasta este valoarea lui $\min_\beta J(\beta)$ (valoarea funcţiei obiectiv în punctul de minim).

b. Vom nota cu $\tilde{\beta}_{new}$ valoarea vectorului de parametri β pentru care se atinge minimul funcţiei

$$\tilde{J}(\beta_{new}) \overset{not.}{=} (\tilde{X}\beta_{new} - y)^\top (\tilde{X}\beta_{new} - y).$$

Determinaţi $\tilde{J}(\tilde{\beta}_{new})$, valoarea minimă a noii funcţii obiectiv, şi scrieţi această expresie sub forma

$$\tilde{J}(\beta_{new}) = J(\hat{\beta}) + f(X, v, y),$$

unde termenul $J(\hat{\beta})$ a fost obţinut la punctul a, iar f este o anumită funcţie de argumentele X, v şi y.

c. Demonstraţi că valoarea optimă a funcţiei obiectiv nu se măreşte dacă adăugăm o nouă trăsătură la matricea de design. Altfel spus, arătaţi că

$$\tilde{J}(\hat{\beta}_{new}) \leq J(\hat{\beta}).$$

d. Arată oare rezultatul de mai sus că adăugând [din ce în ce] mai multe trăsături vom obţine întotdeauna (sau, cu necesitate) un model de regresie care generalizează mai bine decât un model cu mai puţine trăsături? Explicaţi de ce *da*, ori, dimpotrivă de ce *nu*.

26. **(Regresia liniară cu regularizare L_2 (*ridge*) şi respectiv regularizare L_1: efectul de *diminuare a ponderilor* (engl., weight decay))**

prelucrare de Liviu Ciortuz, după
□ • ○ ***CMU, 2011 fall, Eric Xing, HW1, pr. 2.2***

În regresia liniară, ni se dă un set de date de antrenament de forma $D = \{(x_i, y_i)|i = 1, \ldots, n\}$, cu $x_i = (x_{i,1}, \ldots, x_{i,d})^\top$ şi $y_i \in \mathbb{R}$. Fie matricea de design $X \in \mathbb{R}^{n\times d}$, unde linia i este $x_i^\top$ şi, de asemenea, vectorul $y = (y_1, \ldots, y_n)^\top$. Considerând că lucrăm cu un model parametrizat (engl., parametric model) de forma $y_i = x_i^\top \beta + \varepsilon_i$, unde $\beta = (\beta_1, \ldots, \beta_d)^\top$, iar ε_i este un termen „zgomot“ (engl., noise term) care urmează o [anumită] distribuţie dată, regresia liniară caută să găsească un vector de parametri β care furnizează cel mai bun "fit" pentru modelul de regresie de mai sus. Un exemplu (de *criteriu*) pentru a măsura fitness-ul, este găsirea acelui β care minimizează o anumită *funcţie de cost / pierdere* (engl., loss function) $J(\beta)$, dată.

a. Pentru regresia liniară cu termen de regularizare de normă L_1, minimizăm următoarea funcţie de cost / pierdere:[388]

$$J_L(\beta) = \sum_i (x_i^\top \beta - y_i)^2 + \lambda \sum_{j=1}^{d} |\beta_j| = \underbrace{(X\beta - y)^\top (X\beta - y)}_{\|X\beta - y\|^2} + \lambda \|\beta\|_1.$$

Presupunem că $X^\top X = I$ (este aşa-numita proprietate de *ortonormalitate* a matricei X). Calculaţi valoarea parametrului β care minimizează criteriul J_L. (Veţi nota această valoare cu $\hat{\beta}_L$).

***Indicaţie* (1): Vă readucem aminte următoarea regulă de derivare:**

$$\frac{\partial |\beta_a|}{\partial \beta_a} = \begin{cases} 1 & \text{dacă } \beta_a > 0 \\ -1 & \text{dacă } \beta_a < 0 \\ \text{nedefinit} & \text{dacă } \beta_a = 0. \end{cases}$$

***Indicaţie* (2): Pentru λ situat într-un anumit interval, nu există soluţie analitică (ecuaţii „normale“) pentru această problemă de optimizare. Ce înseamnă acest fapt?**

b. *Remember*: La problema 3.A am arătat că atunci când luăm ca funcţie de cost suma pătratelor erorilor (engl., the square-error), adică

$$J(\beta) = \sum_i (x_i^\top \beta - y_i)^2 = (X^\top \beta - y)^\top (X^\top \beta - y),$$

obţinem (atunci când matricea $X^\top X$ este inversabilă) următoarea soluţie pentru problema de optimizare $\arg\min_\beta J(\beta)$:

$$\hat{\beta} = (X^\top X)^{-1} X^\top y.$$

[388] *Atenţie*! A nu se confumda acest tip de regresie regularizată cu varianta de regresie liniară în care zgomotul ε este modelat cu distribuţia Laplace. Pentru regresia aceasta din urmă, criteriul de optimizat este — conform problemei 8.B — o sumă a erorilor în modul / norma L_1 (vezi relaţia (174)). În problema de faţă, doar termenul de regularizare este în norma L_1 (este vorba despre $\lambda\|\beta\|_1$), celălalt termen din criteriul J_L este o sumă de pătrate (deci este în norma L_2).

În cazul regresiei liniare cu termen de regularizare de normă L_2 (regresia *ridge*), minimizăm următoarea funcţie de cost / pierdere:

$$J_R(\beta) = \sum_i (x_i^\top \beta - y_i)^2 + \lambda \sum_{j=1}^{d} \beta_j^2 = (X\beta - y)^\top (X\beta - y) + \lambda \|\beta\|^2.$$

La problema 3.C am arătat că valoarea parametrului β care minimizează criteriul J_R este următoarea (atunci când matricea $X^\top X + \lambda I$ este inversabilă):

$$\hat{\beta}_R = (X^\top X + \lambda I)^{-1} X^\top y.$$

Presupunem că $X^\top X = I$. Scrieţi valorile parametrului β care minimizează criteriile J şi J_R. Comparaţi şi explicaţi cum anume cele două metode de regularizare, *ridge* şi L_1, afectează (eventual, diminuează) valorile parametrilor $\hat{\beta}_j$, pentru $j \in \{1, \ldots, d\}$. (Am folosit notaţia $\hat{\beta} = (\hat{\beta}_1, \ldots, \hat{\beta}_d)^\top$.)

27. **(Regresia *ridge* kernel-izată: exemplu de aplicare)**

□ • · ***CMU, 2020 spring, Pat Virtue, recitation 9, pr. 1.2***

Algoritmul *regresiei ridge kernel-izate* este următorul:[389]

Pasul 1 (Antrenare):

Calculează vectorul-coloană $\alpha = (K + \lambda I)^{-1} y$, unde $K_{ij} = k(x^{(i)}, x^{(j)})$, iar k este funcţia-nucleu.

Pasul 1 (Predicţie):

Dată fiind o instanţă nouă x, calculează *predicţia* $\hat{y} = \sum_{i=1}^{N} \alpha_i k(x, x^{(i)})$.

În cele ce urmează veţi folosi funcţia-nucleu

$$k(x, x^{(i)}) = \begin{cases} 1 & \text{dacă } \|x - x^{(i)}\|^2 \leq \frac{1}{2} \\ 0 & \text{altfel,} \end{cases}$$

unde simbolul $\|\cdot\|$ desemnează norma euclidiană.

Vi se dau următoarele instanţe etichetate (x, y), cu x şi y din $\mathbb{R}$: $(3, 4)$, $(3.7, 1)$, $(4.2, -2)$.

a. Calculaţi matricea K şi vectorul α. Pentru simplitate, veţi considera $\lambda = 0$.

b. Preziceţi valoarea $\hat{y}$ pentru punctul $x = 3.4$.

***Remember*:**
Dacă A este matrice pătratică de ordin $n \times n$ *nesingulară* (adică, $det(A) \neq 0$), atunci există A^{-1}, *inversa* matricei A, şi se calculează astfel:

$$A^{-1} = \frac{1}{det(A)} (A^*)^\top,$$

unde operatorul $\top$ desemnează transpusa unei matrice oarecare, iar A^* este matricea de ordin $n \times n$ (numită matricea adjunctă) pentru care elementul generic $[A^*]_{i,j}$ este produsul dintre $(-1)^{i+j}$ şi determinantul matricei obţinute din matricea A în urma eliminării liniei i şi a coloanei j.

[389] Vă readucem aminte că regresia eridge este regresia liniară cu termen de regularizare de normă L_2. Pentru detalii privind kernel-izarea regresiei *ridge*, vedeţi problema 9.

28. **(Regresia liniară cu zgomot gaussian şi cu regularizare L_1: rezolvare folosind metoda descreşterii pe coordonate şi noţiunile de *subderivată* / *subgradient* şi *subdiferenţială*)**

□ • · ***CMU, 2009 fall, Carlos Guestrin, HW2, pr. 2.cde***
MIT, 2003 fall, Tommi Jaakkola, HW4, pr. 1.ab[390]

Se consideră un set de puncte împreună cu output-urile asociate: $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, unde $x_i \in \mathbb{R}^d$, iar $y_i \in \mathbb{R}$ pentru $i = 1, \ldots, n$. Vom folosi aceste date ca să antrenăm un predictor liniar $y = w \cdot F(x)$, unde $w = (w_1, \ldots, w_d)$, iar $F(x) = (f_1(x), \ldots, f_d(x))$, cu d finit. După cum ştiţi, f_i este numită *trăsătura* (engl., feature) sau *funcţia de bază* (engl., basis function) de indice i pentru problema noastră de învăţare. Fie următoarea funcţie-obiectiv:

$$J(w, \lambda) = \frac{1}{n} \sum_{j=1}^{n} \frac{1}{2} (y_j - w \cdot F(x_j))^2 + \lambda ||w||_1, \tag{215}$$

unde $\|w\|_1 \overset{def.}{=} \sum_{i=1}^{d} |w_i|$ este norma L_1 pentru vectorul w, iar $\lambda > 0$. Vom folosi datele noastre ca să învăţăm vectorul $w^* \overset{not.}{=} w^*(\lambda)$ minimizând funcţia-obiectiv $J(w, \lambda)$, adică,

$$w^* = \arg\min_{w \in \mathbb{R}^d} J(w, \lambda). \tag{216}$$

Acest criteriu de optimizare conduce în mod tipic la realizarea unei bune selecţii de trăsături atunci când se alege o valoare mare pentru parametrul λ. Cu alte cuvinte, dacă atribuim lui λ o valoare mare, atunci multe ponderi w_i vor deveni 0. În consecinţă, trăsăturile / funcţiile de bază corespunzătoare acestor ponderi w_i nu vor fi folosite de către predictorul nostru.

***Comentariu*:** Pentru a rezolva această problemă de regresie liniară având termenul de regularizare de normă L_1, putem folosi o metodă de optimizare simplă cunoscută sub numele de *metoda descreşterii pe coordonate* (engl., *coordinate descent*). Aceasta implică faptul că vom ajusta pe rând fiecare parametru w_i, minimizând funcţia objiectiv în care toţi ceilalţi parametri sunt fixaţi:

$$w_i^* = \arg\min_{w_i} J(w, \lambda).$$

Făcând indicele i să ia valori succesive în mulţimea $\{1, \ldots, d\}$ şi ajustând pe rând parametrii w_i, cu repetarea de mai multe ori a acestui ciclu iterativ, vectorul de parametri va converge asimptotic la minimul global al funcţiei obiectiv convexe. În general, metoda descreşterii pe coordonate apelează o subrutină care realizează minimizarea pe coordonata aleasă la un moment dat — „minimizarea pe linie“ (engl., *line minimization*). În problema noastră, această minimizare se poate realiza într-o formă analitică (engl., *closed form*).

a. Calculaţi $\partial_{w_i} J$, subdiferenţiala funcţiei $J(w, \lambda)$ în raport cu parametrul w_i.[391]

***Sugestie* (1): Calculaţi derivata întâi a primului termen al funcţiei J în raport cu w_i şi apoi subdiferenţiala celui de-al doilea termen. Asiguraţi-vă că trataţi toate cele trei cazuri: $w_i < 0$, $w_i = 0$ şi $w_i > 0$. Scrieţi derivata primului termen sub forma $a_i w_i - c_i$, unde c_i este dat (mai jos) de expresia (217).**

[390] Punctele *c* şi *d* ale acestei probleme sunt de tip implementare. Acolo sunteţi solicitaţi să aplicaţi algoritmul dezvoltat în acest exerciţiu pe un anumit set de date de antrenament.
[391] Pentru definiţia noţiunii de *subderivată*, vedeţi pr. 81 de la capitolul de *Fundamente*.

***Sugestie* (2): O subderivată a lui $J(w, \lambda)$ va fi egală cu suma dintre derivata primului termen (fiindcă acest termen este derivabil) şi o subderivată celui de-al doilea termen.**

***Sugestie* (3): Subderivata astfel calculată pentru $J(w, \lambda)$ va putea fi folosită ulterior în cadrul algoritmului subgradientului descendent (ciclic) care a fost introdus la pr. 171 de la capitolul de *Fundamente*, obţinând astfel un algoritm de antrenare pentru regresia liniară cu regularizare de normă L_1.**

b. Calculaţi punctul de optim global w_i^* ţinând cont de următoarea *proprietate*: w^* este un punct de optim global pentru funcţia convexă f dacă şi numai dacă $0 \in \partial f(w^*)$. Cu alte cuvinte, găsiţi w_i^* astfel încât $0 \in \partial_{w_i^*} J$.

c. Să explorăm mai atent relaţia dintre parametrul de regularizare λ, ponderea w_i pentru trăsătura / funcţia de bază de indice i şi cantitatea c_i definită astfel:

$$c_i = \frac{1}{n} \sum_{j=1}^{n} f_i(x_j)(y_j - \sum_{m=1, m \neq i}^{d} w_m f_m(x_j)). \tag{217}$$

Ce semnificaţie are c_i? Trasaţi graficul lui w_i^* în raport cu c_i. Unde apare λ în acest grafic?

29. (Regresia liniară cu „zgomot“ modelat cu distribuţia *Laplace*: rezolvare în cazul unidimensional [chiar particularizat] cu ajutorul derivatei, acolo unde există)

□ • *CMU, 2009 fall, Carlos Guestrin, HW2, pr. 1*

Presupunem că dorim să prezicem o valoare necunoscută Y, după ce ne-au fost puse la dispoziţie [ca date de antrenament] o secvenţă de „observaţii“ $x_1, \ldots, x_n$ pentru Y, care sunt afectate de prezenţa unor „zgomote“ i.i.d., mai precis $y_i = x_i + \epsilon_i$, pentru $i = 1, \ldots, n$.

***Comentariu*:** Se poate arăta[392] că atunci când presupunem că „zgomotele“ sunt independente şi distribuite conform aceleiaşi distribuţii gaussiene ($\epsilon_i \sim \mathcal{N}(0, \sigma^2)$), a găsi estimarea în sens MLE pentru Y echivalează cu a găsi acea valoare $\hat{y}$ care minimizează suma pătratelor erorilor [adică, a diferenţelor] în raport cu aceste valori date, x_i. Altfel spus,

$$\hat{y} = \arg\min_y \sum_{i=1}^{n} (y - x_i)^2,$$

iar această valoare se calculează cu o formulă analitică (engl., closed form solution) simplă:

$$\hat{y} = \frac{1}{n} \sum_{i=1}^{n} x_i.$$

În acest exerciţiu veţi demonstra că atunci când presupunem că „zgomotele“ ε_i sunt independente şi distribuite conform distribuţiei $Laplace(0, b)$, al cărei p.d.f., vă readucem aminte, este

$$f_{\epsilon_i}(x) = \frac{1}{2b} \exp\left(-\frac{|x|}{b}\right),$$

[392] Observaţi că acesta este un caz particular în raport cu regresia liniară de tip LSE prezentat la problema 3.A.

obţinem o estimare mai robustă a soluţiei decât în cazul „zgomotului“ gaussian (vedeţi problema 3.A).

La problema 8.B am demonstrat că a găsi estimarea în sensul MLE pentru Y, atunci când presupunem că „zgomotul“ este de tip Laplace, este echivalent cu a găsi valoarea $\hat{y}$ care minimizează suma erorilor în valoare absolută (sau, în modul). Altfel spus,

$$L(y) = \sum_{i=1}^{n} |y - x_i| \text{ şi } \hat{y} = \arg\min_{y} L(y).$$

O modalitate standard de a calcula minimul funcţiei de cost / pierdere este să calculăm derivata ei şi să o egalăm cu 0. Problema este că funcţia $L(y)$ nu este derivabilă pe tot domeniul său de definiţie. Totuşi, este uşor de observat că $L(y)$ este nederivabilă doar atunci când y ia aceeaşi valoare ca una dintre instanţele x_i.

a. Presupunem că instanţele x_i sunt diferite între ele şi că sunt sortate în ordine crescătoare ($x_i < x_j$ pentru orice $i < j$).

Calculaţi $L'(y)$, derivata lui $L(y)$ în raport cu y, pentru cazul când y este situat între două valori consecutive ale lui x (adică, $x_i < y < x_{i+1}$).

***Sugestie*: Analizaţi situaţia x-ilor care sunt mai mici decât y separat de cea a x-ilor care sunt mai mari decât y.**

b. Presupunând că n, numărul instanţelor de antrenament este par, care sunt valorile lui y pentru care $L'(y) = 0$?

c. Dacă n este impar, [se poate arăta că] nu există nicio valoare a lui y astfel încât $L'(y) = 0$. Totuşi, există o valoare y_0 astfel încât $L'(y) < 0$ pentru orice $y < y_0$ şi $L'(y) > 0$ pentru orice $y > y_0$. Cât este acest y_0?

d. Valoarea lui $\hat{y}$ este determinată de răspunsurile pe care le-aţi obţinut la punctele b şi c. Daţi pe scurt o explicaţie pentru faptul că această soluţie poate fi mai robustă la prezenţa outlier-elor din date în comparaţie cu soluţia obţinută în cazul regresiei de tip LSE (engl., least squared errors).

30. **(Regresia liniară local-ponderată, cazul unidimensional: o proprietate: „netezirea“ liniară)**

□ • ○ ⋆ ***CMU, 2014 spring, B. Poczos, A. Singh, HW2, pr. 4.A***

Presupunem că lucrăm cu un model de regresie liniară unidimensională local-ponderată, în care „răspunsul“ prezis pentru o valoare oarecare a variabilei $x \in \mathbb{R}$ este $\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 x$, unde parametrii $\hat{\beta}_0$ şi $\hat{\beta}_1$ sunt definiţi printr-o relaţie de forma următoare:

$$\hat{\beta}_0, \hat{\beta}_1 = \arg\min_{\beta_0, \beta_1} \Big(\sum_{i=1}^{n} w_i(x) \, (y_i - \beta_0 - \beta_1 x_i)^2 \Big),$$

cu ponderile $w_i(x) \in \mathbb{R}$.[393]

a. Arătaţi că funcţia obiectiv din definiţia dată mai sus pentru ponderile $\hat{\beta}_0, \hat{\beta}_1$ poate fi rescrisă astfel:[394]

$$(y - X\beta)^\top W(x)\,(y - X\beta),$$

unde $y \overset{not.}{=} (y_1, \ldots, y_n)^\top$, $\beta \overset{not.}{=} (\beta_0,\, \beta_1)^\top$, $W(x)$ este o matrice diagonală, care are diagonala $(w_1(x), \ldots, w_n(x))$, iar

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}.$$

b. Arătaţi că funcţia $\hat{f}(x)$, care a fost definită mai sus prin relaţia $\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 x$, este o combinaţie liniară de valorile $\{y_i\}_{i=1}^n$.[395] **Mai precis, valoarea $\hat{f}(x)$ poate fi scrisă sub forma $\hat{f}(x) = \sum_{i=1}^n \ell_i(x)\, y_i = \ell(x)^\top y$, unde $\ell_i(x)$ desemnează o anumită cantitate / funcţie definită în raport cu x, iar $\ell(x) \overset{not.}{=} (\ell_1(x), \ell_2(x), \ldots, \ell_n(x))^\top$. Cu alte cuvinte, regresia liniară [unidimensională] local-ponderată manifestă proprietatea de *netezire liniară* (cf. CMU, 2007 fall, Carlos Guestrin, HW3, pr. 1).**

31. (O extensie / generalizare a regresiei liniare, varianta LSE: cazul „răspunsului“ multivaluat)

□ • · ***Stanford, 2007 fall, Andrew Ng, HW1, pr. 3***

La toate problemele de până acum, în ce priveşte regresia liniară, am considerat că variabila de „răspuns“ y are o valoare reală. Acum vom presupune că

[393] Deşi nu este neapărat necesar [pentru a demonstra rezultatele care fac obiectul acestui exerciţiu], am putea considera că pentru fiecare $i \in \{1, \ldots, n\}$, ponderea $w_i(x)$ are valoarea $\dfrac{K\left(\frac{x - x_i}{h}\right)}{\sum_{i=1}^n K\left(\frac{x - x_i}{h}\right)}$, unde funcţia K poate fi, de exemplu, aşa-numitul *nucleu RBF*. Această funcţie este definită pe $\mathbb{R}^d$ prin relaţia $K\left(\frac{x}{h}\right) \overset{def.}{=} \exp\left(-\frac{\|x\|^2}{h^2}\right)$, unde $h > 0$ este [pentru problema de regresie] un hiper-parametru, numit *lăţimea de bandă* (engl., *bandwidth*). De obicei, într-un astfel de caz, regresia liniară local-ponderată este numită regresie [locală] kernel-izată (engl., kernelized [local] regression).

Observaţie importantă: Vă rugăm să reţineţi că termenul *kernel* (sau, *kernel*-izat) în acest context nu are nimic de a face cu metoda de *kernel*-izare (engl., kernel trick) care este folosită pentru maşinile cu vectori-suport (engl., Support Vector Machines, SVMs), regresia *ridge* kernel-izată (vedeţi problema 9) şi regresia logistică kernel-izată (problema 18).

[394] Remarcaţi faptul că expresia care urmează este, ca formă, identică cu relaţia (171) din cazul regresiei liniare (în varianta LSE) local-ponderate cu n atribute / variabile de intrare. Aici vă cerem însă să lucraţi în cazul (particular) unidimensional.

[395] Pentru cazul regresiei liniare neponderate în varianta LSE (care a fost prezentată la problema 3.A), această proprietate decurge imediat din relaţia (161). Pentru cazul regresiei liniare (în varianta LSE) local-ponderate în general (deci nu doar pentru cazul unidimensional despre care discutăm aici), proprietatea decurge imediat din relaţia (173). Totuşi, ca şi mai sus, în problema de faţă vă cerem să elaboraţi demonstraţia pentru cazul (particular) al regresiei local-ponderate unidimensionale.

ni se cere să facem antrenarea pe un set de date având răspunsuri / outputuri multiple pentru toate exemplele de antrenament:

$$\{(x^{((i)}, y^{(i)})|i = 1, \ldots, n\}, x^{(i)} \in \mathbb{R}^d, y^{(i)} \in \mathbb{R}^p.$$

Aşadar, pentru oricare exemplu de antrenament, $y^{(i)}$ este un vector cu p componente.

Vrem să folosim un *model liniar* pentru a prezice outputurile, asemănător celui definit prin estimarea în sensul MLE, adică prin minimizarea sumei pătratelor erorilor (engl., least squared errors, LSE),[396] şi anume specificând acum (în locul vectorului d dimensional) o matrice de parametri $\beta \in \mathbb{R}^{n\times p}$ în relaţia de definiţie a modelului de regresie:

$$y = \beta^\top x.$$

a. Funcţia de cost / pierdere în acest caz este

$$J(\beta) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}\left((\beta^\top x^{(i)})_j - y_j^{(i)}\right)^2.$$

Similar cu modul în care am procedat în cazul univaluat,[397] scrieţi $J(\beta)$ în notaţie vectorial-matriceală.

***Sugestie*: Începeţi considerând aşa-numita *matrice de design* (X), de dimensiune $n \times d$**

$$X = \begin{bmatrix} — & (x^{(1)})^\top & — \\ — & (x^{(2)})^\top & — \\ & \vdots & \\ — & (x^{(n)})^\top & — \end{bmatrix}$$

şi matricea de „răspunsuri“ (Y), de dimensiune $n \times p$

$$Y = \begin{bmatrix} — & (y^{(1)})^\top & — \\ — & (y^{(2)})^\top & — \\ & \vdots & \\ — & (y^{(n)})^\top & — \end{bmatrix},$$

iar apoi căutaţi să vedeţi cum poate fi exprimat $J(\beta)$ cu ajutorul acestor matrice.

b. Găsiţi *formula analitică* (engl., the closed form solution) pentru acea valoare a lui β care minimizează expresia $J(\beta)$.[398] Altfel spus, aceasta înseamnă să obţineţi aşa-numitele ecuaţii „normale“ pentru cazul multivaluat al regresiei liniare în varianta LSE.

c. Să presupunem că în loc să considerăm vectorii $y^{(i)}$ ca atare, decidem să calculăm fiecare variabilă / componentă $y_j^{(i)}$ în mod separat, pentru $j = 1, \ldots, p$. În această abordare, vom avea p modele liniare individuale, de forma

$$y_j = \beta_j^\top x^{(i)}, \text{ cu } \beta_j \in \mathbb{R}^d \text{ pentru } j = 1, \ldots, p.$$

[396] Vedeţi cum am procedat la problema 3.A.

[397] Adică, atunci când $y \in \mathbb{R}$; vedeţi relaţiile (159) şi (160) de la problema 3.A.

[398] Această formulă va constitui o generalizare a relaţiei (161) de la regresia liniară LSE univaluată (problema 3.A).

Care este relaţia dintre soluţiile (adică valorile optime pentru parametrii β_j) acestor p probleme independente de tip "least squared errors" (LSE) şi soluţia (adică valoarea optimă pentru matricea de parametri β) pentru varianta regresiei liniare LSE multivaluate?

32. **(Regresie liniară folosită pentru clasificare; aplicare pe date din $\mathbb{R}^2$)**

prelucrare de Liviu Ciortuz, după
□ • ○ ***MIT, 2011 fall, Leslie Kälbling, HW1, pr. 2.2***

În acest exerciţiu veţi lucra cu un algoritm de regresie liniară de tipul sumei celor mai mici pătrate (engl., least squared errors, LSE), folosind ca etichete / ieşiri y doar valorile ± 1. În acest fel, reducem problema de *clasificare* la una de *regresie*.[399]

Concepeţi un set de exemple de antrenament în planul euclidian ($\mathbb{R}^2$), care să fie separabile printr-o dreaptă care trece prin originea sistemului de coordonate, astfel încât, atunci când aceste exemple sunt folosite pentru antrenarea unui model de regresie liniară de tip LSE cu etichete binare, să rezulte [în urma antrenării] că unele dintre exemplele de antrenament vor fi clasificate în mod eronat.

1.2.2 Regresia logistică

33. **(Funcţia sigmoidală / logistică: definiţie şi proprietăţi de bază)**

• · ***Liviu Ciortuz, 2024***

Considerăm *funcţia sigmoidală* (sau *logistică*)

$$\sigma : \mathbb{R} \to (0,1), \text{ definită prin } \sigma(z) = \frac{1}{1+e^{-z}} = \frac{e^z}{1+e^z}, \text{ pentru orice } z \in \mathbb{R}.$$

a. Elaboraţi graficul acestei funcţii. Daţi toate justificările necesare. (*Indicaţie*: O parte dintre ele sunt enunţate la punctele b.*i*, b.*ii* şi b.*iii* de mai jos.)

b. Demonstraţi următoarele patru proprietăţi:

i. $\sigma(z) \in (0,\, 1)$ **pentru orice** $z \in \mathbb{R}$.

ii. $\sigma(-z) = 1 - \sigma(z)$.

iii. $\sigma'(z) = \sigma(z)[1 - \sigma(z)]$.

iv. **Funcţia σ este inversabilă (veţi justifica de ce este aşa!), aşadar există funcţia inversă $\sigma^{-1} : (0,1) \to \mathbb{R}$, astfel încât**

$$\sigma^{-1}(\sigma(z)) = \sigma(\sigma^{-1}(z)) = z \text{ pentru orice } z \in \mathbb{R}. \tag{218}$$

[399] Această metodă este descrisă în secţiunea 4.1.3 din cartea *Pattern Recognition and Machine Learning* de Ch. Bishop (Springer, 2006) şi în secţiunea 4.2 din cartea *The Elements of Statistical Learning* de T. Hastie, R. Tibshirani and J. Friedman (Springer, 2009).

Arătaţi mai întâi că

$$\sigma^{-1}(z) = \ln \frac{z}{1-z} \text{ pentru orice } z \in (0,\, 1).$$

Aceasta este aşa-numita funcţie *logit*.
Verificaţi apoi că dubla egalitate (218) **este satisfăcută.**
La final, faceţi graficul funcţiei logit.

34. **(Regresia logistică: o proprietate a funcţiei de verosimilitate condiţională, atunci când setul de antrenament este liniar separabil)**

Liviu Ciortuz, 2025

Fie $D = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$ **un set de date de antrenament, cu** $x^{(i)} \in \mathbb{R}^{d+1}$**, cu** $x_0^{(i)} = 1$ **şi** $y^{(i)} \in \{0, 1\}$**, pentru** $i = 1, \ldots, n$**. Spunem că datasetul** D **este *liniar separabil* dacă există** $w^* \overset{not.}{=} (w_0^*, w_1^*, \ldots, w_d^*)^\top \in \mathbb{R}^{d+1}$ **astfel încât** $sign(w^* \cdot x^{(i)})$ **corespunde clasei** $y^{(i)}$**, pentru** $i = 1, \ldots, n$**, ceea ce înseamnă că pentru fiecare instanţă de antrenament** $(x^{(i)}, y^{(i)})$ **valoarea combinaţiei liniare** $w_0^* + w_1^* x_1^{(i)} + \ldots w_d^* x_d^{(i)}$ **este strict pozitivă dacă** $y^{(i)} = 1$ **şi respectiv este strict negativă dacă** $y^{(i)} = 0$**.**

a. Pentru acest set de date, scrieţi expresia lui $L_D(w)$**, funcţia de verosimilitate condiţională — *Atenţie*!, *nu* funcţia de log-verosimilitate condiţională — în raport cu vectorul de parametri** $w \overset{not.}{=} (w_0, w_1, \ldots, w_d)^\top \in \mathbb{R}^{d+1}$**, în contextul regresiei logistice.**

b. Arătaţi că dacă datasetul D **este liniar separabil, atunci** $L_D(w)$ **poate lua valori oricât de apropiate de** 1 **(probabilitatea maximă). Ce puteţi spune în acest caz despre valorile funcţiei de log-verosimilitate condiţională?**

35. **(Regresia logistică: particularizare în** $\mathbb{R}^2$**; deducerea regulilor de actualizare pentru metoda gradientului; regularizare** L_1**)**

prelucrare de Liviu Ciortuz, după
• ∗ ***CMU, 2018 spring, Nina Balcan, HW3, pr. 1, 4***

A. În prima parte a acestei probleme, veţi elabora algoritmul de regresie logistică bazat pe metoda gradientului, în cazul bidimensional. Considerăm setul de date de antrenament $\{(x^{(i)}, y^{(i)}), i = 1, \ldots, n\}$ **în care fiecare** $x^{(i)} \in \mathbb{R}^2$ **este un vector de trăsături, iar** $y^{(i)} \in \{0, 1\}$ **este o etichetă binară. Presupunând că folosim un model parametrizat de forma**

$$p(y = 1|x; w) = \frac{1}{1 + \exp(-w_0 - w_1 x_1 - w_2 x_2)} = \frac{\exp(w_0 + w_1 x_1 + w_2 x_2)}{1 + \exp(w_0 + w_1 x_1 + w_2 x_2)},$$

***obiectivul* nostru este să găsim valorile** $\bar{w}_j$ **ale parametrilor** w_j **care maximizează verosimilitatea condiţională (engl., Maximum (Conditional) Likelihood Estimation, M(C)LE) a setului de date de antrenament.**

a. Mai jos, vom face *noi* calculul pentru log-verosimilitatea condiţională a datelor de antrenament. Relativ la acest calcul, *dumneavoastră* veţi elabora câte o scurtă justificare pentru fiecare linie din demonstraţie.

$$\ell(w) = \ln \prod_{i=1}^{n} p(y^{(i)}|x^{(i)}, w) \tag{219}$$

$$= \sum_{i=1}^{n} \ln p(y^{(i)}|x^{(i)}, w) \tag{220}$$

$$= \sum_{i=1}^{n} \ln \left(p(y^{(i)} = 1|x^{(i)}, w)^{y^{(i)}} \, p(y^{(i)} = 0|x^{(i)}, w)^{1-y^{(i)}} \right) \tag{221}$$

$$= \sum_{i=1}^{n} [y^{(i)} \ln p(y^{(i)} = 1|x^{(i)}, w) + (1 - y^{(i)}) \ln p(y^{(i)} = 0|x^{(i)}, w)] \tag{222}$$

$$= \sum_{i=1}^{n} [y^{(i)} \ln \sigma(w \cdot x^{(i)}) + (1 - y^{(i)}) \ln(1 - \sigma(w \cdot x^{(i)}))] \tag{223}$$

$$= \sum_{i=1}^{n} \Big[y^{(i)} \ln \frac{\exp(w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)})}{1 + \exp(w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)})} + (1 - y^{(i)}) \ln \frac{1}{1 + \exp(w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)})} \Big] \tag{224}$$

$$= \sum_{i=1}^{n} \Big[y^{(i)} \ln \left(\exp(w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)}) \right) + \ln \frac{1}{1 + \exp(w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)})} \Big] \tag{225}$$

$$= \sum_{i=1}^{n} \left[y^{(i)} \left(w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)} \right) - \ln \left(1 + \exp(w_0 + w_1 x_1^{(i)} + w_2 x_2^{(i)}) \right) \right]. \tag{226}$$

b. Acum veţi calcula gradientul expresiei de mai sus în raport cu vectorul de ponderi (w_0, w_1, w_2). Componentele vectorului gradient sunt derivatele parţiale $\frac{\partial \ell(w)}{\partial w_j}$, unde $\ell(w)$ este funcţia de log-verosimilitate condiţională de la punctul *a*. Arătaţi că

$$\frac{\partial \ell(w)}{\partial w_j} = \sum_{i=1}^{n} (y^{(i)} - \sigma(w \cdot x^{(i)})) x_j^{(i)}. \tag{227}$$

***Observaţie*: Considerând că fiecare instanţă/vector $x^{(i)}$ (cu $i \in \{1, \ldots, n\}$) a fost extinsă/extins cu componenta $x_0^{(i)} = 1$, pentru ca expresia (227) să „acopere" şi cazul derivatei parţiale $\frac{\partial \ell(w)}{\partial w_0}$, vectorul gradient pentru funcţia ℓ va putea fi calculat astfel:**

$$\nabla_w \ell(w) = \sum_{i=1}^{n} [y^{(i)} - \sigma(w \cdot x^{(i)})] x^{(i)}.$$

c. Se poate demonstra că funcţia de log-verosimilitate condiţională ℓ este concavă,[400] deci este uşor să determinăm maximul ei folosind metoda gradientului

[400] Pentru demonstraţia acestei afirmaţii în cazul general (deci nu doar pentru cazul datelor din $\mathbb{R}^2$), vedeţi problema 14.

ascendent. *Algoritmul* procedează după cum urmează:

Se fixează mai întâi rata de învăţare $\eta > 0$ şi se aleg nişte valori iniţiale w_0^0, w_1^0 şi w_2^0 pentru ponderile w_0, w_1 şi respectiv w_2, după care se execută un număr de iteraţii [în care partea esenţială este aplicarea *regulii de actualizare* a ponderilor w_0, w_1 şi w_2], până când diferenţa dintre valorile funcţiei ℓ la două iteraţii consecutive scade sub un anumit prag (fixat dinainte) $\varepsilon > 0$.

Scrieţi regula (regulile) de actualizare a ponderilor w_0, w_1 şi w_2, care sunt folosite de metoda gradientului ascendent pentru rezolvarea acestei probleme de regresie logistică.

d. Vă readucem aminte că regula de predicţie / decizie pentru regresia logistică este următoarea:

> **dacă $p(y^{(i)} = 1|x^{(i)}) > p(y^{(i)} = 0|x^{(i)})$, atunci alege predicţia 1,**
> **altfel alege predicţia 0.**

De ce tip este *graniţa de decizie* (engl., decision boundary) corespunzătoare modelului de regresie logistică? Scrieţi ecuaţia corespunzătoare graniţei de decizie în funcţie de $\bar{w}_0$, $\bar{w}_1$, $\bar{w}_2$ şi $x_1^{(i)}, x_2^{(i)}$.

e. Explicaţi de ce regresia logistică este un clasificator de tip *discriminativ*, spre deosebire de clasificatorii de tip *generativ*, cum este de exemplu algoritmul Bayes Naiv (pentru acesta din urmă, vedeţi capitolul *Clasificare bayesiană*).

B. În a doua parte a acestei probleme ne vom concentra asupra estimării parametrilor regresiei logistice în sensul probabilităţii condiţionate maxime a posteriori (engl., Maximum (Conditional) A posteriori Probability, M(C)AP).

f. Considerând o distribuţie a priori de tip Laplace $p(w) = \prod_j \frac{1}{2b} e^{-\frac{|w_j|}{b}}$, calculaţi valoarea parametrului w care maximizează expresia funcţiei de probabilitate a posteriori (M(C)AP). Aşadar, veţi calcula

$$\hat{w} = \underset{w}{\operatorname{argmax}} \ln[p(w) \prod_i p(y^{(i)}|x^{(i)}, w)].$$

***Sugestie*: Rezultatul pe care îl veţi obţine ar trebui să fie foarte asemănător cu expresia (226), cu singura diferenţă că noul rezultat va conţine un termen suplimentar, care corespunde distribuţiei a priori.**

g. Care este expresia pe care ar trebui s-o aibe acum derivata parţială a funcţiei de verosimilitate condiţională a posteriori?

***Sugestie*: Ar trebui să puteţi identifica / separa în expresia funcţiei de *verosimilitate condiţională a posteriori* termenul corespunzător distribuţiei a priori de termenul corespunzător funcţiei de *verosimilitate condiţională*. La final ar trebui să obţineţi o expresie similară cu expresia (227), având însă un termen suplimentar.**

36. **(Regresia logistică, chestiuni introductive: exemplificare, folosind datele de la problema 2 de la capitolul *Arbori de decizie*)**

■ ∗ ***Liviu Ciortuz, 2024***

Considerăm setul de date de mai jos, pe care l-am preluat de la problema 2 de la capitolul *Arbori de decizie*, cu următoarele schimbări referitoare la *notaţii* (pentru a face uşor legătura cu rezultatele de la problema 13, la care am descris fundamentele regresiei logistice):

Atributele de intrare au fost redenumite $x_1, \ldots, x_4$, atributul de ieşire a fost renotat cu y, iar instanţele de antrenament au fost indexate cu $i \in \{1, \ldots, 8\}$ şi vor fi desemnate în continuare sub forma $x^{(1)}, \ldots, x^{(8)}$. Valoarea atributului x_j pentru instanţa de antrenament $x^{(i)}$ va fi desemnată cu $x_j^{(i)}$.

i	x_1	x_2	x_3	x_4	y
1	1	0	0	0	1
2	1	0	1	0	1
3	0	1	0	1	1
4	0	0	0	1	0
5	1	1	1	0	0
6	1	0	1	1	0
7	1	0	0	1	0
8	0	1	0	0	0

În acest exerciţiu veţi crea un model / clasificator discriminativ pentru acest set de date, aplicând metoda regresiei logistice care a fost descrisă la problema 13.

a. Scrieţi funcţia de log-verosimilitate condiţională $\ell(w)$ corespunzătoare acestui set de date [în contextul regresiei logistice]. *Indicaţie*: Puteţi să aplicaţi direct formula (183) de la problema 13. Nu uitaţi să extindeţi vectorii $x^{(i)}$ care reprezintă instanţele de antrenament cu componenta $x_0^{(i)} = 1$ corespunzătoare ponderii w_0!

b. La acest punct veţi calcula vectorul gradient pentru funcţia de log-verosimilitate condiţională de la punctul a.

i. **Aplicaţi direct formula (186) de la problema 13:**

$$\nabla_w \ell(w) = \sum_{i=1}^{n} [y^{(i)} - \sigma(w \cdot x^{(i)})] x^{(i)}.$$

ii. **Pentru o valoare oarecare a lui $j \in \{0, 1, \ldots, 4\}$, calculaţi în mod clasic derivata parţială a funcţiei de log-verosimilitate condiţională $\ell(w)$ în raport cu ponderea w_j (această derivată parţială se notează cu $\frac{\partial}{\partial w_j}\ell(w)$) şi apoi verificaţi că rezultatul pe care l-aţi obţinut coincide cu componenta de pe poziţia j din vectorul gradient care a fost calculat mai sus (vedeţi punctul i).**

c. Considerând că pentru a afla maximul funcţiei de log-verosimilitate de la punctul a, se aplică metoda gradientului ascendent cu rata de învăţare $\eta = 0.1$ (vedeţi problema 80 de la capitolul de *Fundamente*), care este valoarea vectorului gradient după prima iteraţie, dacă la iniţializare s-a fixat $w = 0$ (vectorul nul din $\mathbb{R}^5$)?

d. [Implementare] Folosind metoda gradientului ascendent, determinaţi valorile optime pentru ponderile $w_0, w_1, \ldots, w_4$ din funcţia de log-verosimilitate condiţională de la punctul a.

Cum va clasifica modelul astfel obţinut instanţele de test din tabelul alăturat?

	x_1	x_2	x_3	x_4	y
U	0	1	1	1	?
V	1	1	0	1	?
W	1	1	0	0	?

e. La acest punct veţi calcula matricea hessiană corespunzătoare funcţiei de log-verosimilitate condiţională $\ell(w)$ pe care aţi obţinut-o la punctul a.

i. **Aplicaţi direct formula** (192) **de la problema 14:**

$$H_w = -\sum_{i=1}^{n} \underbrace{\sigma(w \cdot x^{(i)})(1 - \sigma(w \cdot x^{(i)}))}_{\in \mathbb{R}_+} \, x^{(i)} (x^{(i)})^\top.$$

(Şi aici se consideră că vectorii $x^{(i)}$ au fost extinşi cu componenta $x_0^{(i)} = 1$ corespunzătoare ponderii w_0!) Nu este nevoie să aduceţi matricea astfel obţinută la o formă mai simplă.

ii. **Pentru acea valoare $j \in \{0, \ldots, 4\}$ care a fost aleasă la punctul $b.i$ şi pentru un $k \in \{0, \ldots, 4\}$, calculaţi în mod clasic derivata parţială de ordin secund $\frac{\partial^2}{\partial w_k \, \partial w_j} \ell(w) \stackrel{def.}{=} \frac{\partial}{\partial w_k} \Big(\frac{\partial}{\partial w_j} \ell(w) \Big)$ şi apoi verificaţi că rezultatul pe care l-aţi obţinut coincide cu componenta de pe linia k şi coloana j din matricea hessiană H_w care a fost calculată mai sus (vedeţi punctul i).**

iii. **Un rezultat matematic interesant afirmă că matricea hessiană este simetrică dacă derivatele parţiale de ordin secund sunt continue.[401] Verificaţi (tot prin derivare directă) că pentru o pereche oarecare de indici $j \neq k$ are loc egalitatea $\frac{\partial^2}{\partial w_k \, \partial w_j} \ell(w) = \frac{\partial^2}{\partial w_j \, \partial w_k} \ell(w)$.**

f. [Implementare] Folosind metoda Newton-Raphson (vedeţi problema 80 de la capitolul de *Fundamente*), vectorul gradient $\nabla_w \ell(w)$ pe care l-aţi obţinut la punctul b şi matricea hessiană H obţinută la punctul e, determinaţi valorile optime pentru ponderile $w_0, w_1, \ldots, w_4$ din funcţia de log-verosimilitate $\ell(w)$ de la punctul d.

Cum va clasifica modelul astfel obţinut instanţele de test din tabelul de la punctul c?

Comparaţi numărul de iteraţii şi respectiv timpii de execuţie până la convergenţă pentru cele două metode, adică metoda gradientului şi metoda Newton-Raphson, pe setul de date din enunţ.

g. Scrieţi *funcţia de cost logistică* corespunzătoare acestui set de date, notată cu $J(w)$.[402] Pentru aceasta, vă cerem să aplicaţi în mod direct formula (187) **de la problema 13. După ce veţi aplica această formulă şi veţi compara rezultatul pe care l-aţi obţinut cu expresia funcţiei $\ell(w)$ pe care aţi obţinut-o la punctul a, veţi constata că nu este imediat evidentă egalitatea $J(w) = -\frac{1}{n} \ell(w)$ care a fost demonstrată la problema 13. (Observaţi că în expresia lui $J(w)$ nu apare funcţia σ.)**

[401] https://en.wikipedia.org/wiki/Hessian_matrix.

[402] Nu există nicio legătură între această funcţie şi criteriul J de la algoritmul K-means (vedeţi capitolul de *Clusterizare*). Este pur şi simplu o coincidenţă de notaţie.

Ce transformări (adică, operaţii matematice) trebuie făcute asupra expresiei funcţiei $\ell(w)$ de la punctul a, pentru ca egalitatea $J(w) = -\frac{1}{n}\ell(w)$ să devină evidentă în mod direct?

***Observaţie*: Ţinând cont de rezultatul care a fost demonstrat la problema 13, minimul funcţiei de cost $J(w)$ poate fi obţinut cu ajutorul metodei gradientului descendent[403] sau cu metoda Newton-Raphson.[404]**

37. **(Regresia logistică: analiza efectului duplicării atributelor de intrare)**

■ □ • ○ ***CMU, 2011 spring, Tom Mitchell, midterm, pr 5.3***

Vom considera o problemă de clasificare binară pe date caracterizate de variabila / atributul $X_1 \in \{0, 1\}$ şi eticheta $Y \in \{0, 1\}$.

Fie un set de date de antrenament format din n exemple: $D_1 = \{(x_1^1, y^1), \ldots, (x_1^n, y^n)\}$. Presupunem că generăm un alt set de date de antrenament, D_2, format tot din n exemple, $D_2 = \{(x_1^1, x_2^1, y^1), \ldots, (x_1^n, x_2^n, y^n)\}$, unde în fiecare exemplu x_1 şi y sunt aceleaşi ca în D_1, iar x_2 este copia lui x_1.

Învăţăm un model de regresie logistică folosind setul de date D_1; acest model va avea doi parametri: w_0 şi w_1. De asemenea, învăţăm un alt model de regresie logistică folosind setul de date D_2; acesta va avea trei parametri: w_0, w_1 şi w_2.

Mai întâi, scrieţi *funcţia obiectiv* pe care o folosim pentru a estima (în sensul maximizării verosimilităţii condiţionale) parametrii (w_0, w_1) şi respectiv (w_0, w_1, w_2), pe baza datelor de antrenament.

Apoi, analizând cele două funcţii obiectiv, precizaţi care este *relaţia* dintre seturile de parametri (w_0, w_1) şi (w_0, w_1, w_2), care se estimează din datele D_1 şi respectiv D_2. Pornind de la această relaţie, argumentaţi de ce (sau cum anume) va fi afectată regresia logistică (sau *nu* va fi afectată) de duplicarea variabilei X_1.

38. **(Regresia liniară şi regresia logistică: definiţii [„revizitate“], o interesantă proprietate comună)**

prelucrare de Liviu Ciortuz, după

■ □ • ○ ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW2, pr. 4***

Dată fiind o instanţă (sau, un input) $X \in \mathbb{R}^d$ împreună cu „răspunsul“ / outputul corespunzător $Y \in \mathbb{R}$, *regresia liniară* [cu „zgomot“ gaussian] construieşte un model de forma

$$Y|X \sim Normal(\mu(X), \sigma^2),$$

[403] Concret, puteţi folosi acelaşi vector gradient care a fost calculat la punctul $b.i$, dar regula de actualizare a ponderilor în acets caz va fi $w \leftarrow w - \frac{\eta}{n}\nabla_w \ell(w)$, unde $\eta > 0$ este rata de învăţare, iar n este numărul instanţelor de antrenament.

[404] Forma regulii de actualizare pentru metoda lui Newton-Raphson este aceeaşi, indiferent dacă se calculează un punct de maxim pentru o funcţie concavă ori un punct de minim pentru o funcţie convexă.

unde media $\mu(X)$ este o funcţie liniară de componentele / atributele inputului: $\mu(X) = \theta^\top X = \theta_0 + \theta_1 X_1 + \ldots + \theta_d X_d$.

Dacă $Y \in \{0, 1\}$, *regresia logistică* — care, spre deosebire de regresia liniară serveşte pentru *clasificare* — modelează outputul Y astfel:

$$Y|X \sim Bernoulli(h_\theta(X)),$$

unde $h_\theta(X)$, parametrul acestei distribuţii Bernoulli, este obţinut din $\theta^\top X$ aplicând funcţia *logistică / sigmoidală*:

$$h_\theta(X) = g(\theta^\top X),$$

unde prin g am notat *funcţia logistică*

$$g(z) \stackrel{def.}{=} \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$$

sau, echivalent, folosind funcţia *logit* (care este inversa funcţiei *logistice / sigmoidale*):[405]

$$logit\,(h_\theta(X)) \stackrel{def.}{=} \ln \frac{h_\theta(X)}{1-h_\theta(X)} = \theta^\top X$$

***Comentariu*:** Definiţiile date mai sus pun în evidenţă un anumit „paralelism“ pentru cele două modele de regresie. În această problemă,

– mai întâi, veţi putea vedea încă o *similaritate*, şi anume între *vectorii gradient* corespunzători funcţiilor de log-verosimilitate condiţională pentru cele două metode de regresie. Ne referim aici la expresia $\nabla_\theta \ell(\theta) = \sum_{i=1}^n (y_i - h_\theta(x_i)) x_i$ (care corespunde relaţiei (186) de la problema 13) pentru gradientul regresiei logistice şi la expresia $\nabla_\theta \ell(\theta) = \sum_{i=1}^n (y_i - \theta^\top x_i) x_i$ pentru gradientul regresiei liniare, conform termenului principal din relaţia (169), care a fost obţinută la problema 6;

– iar acum veţi demonstra o *proprietate de tip probabilist*, care se scrie în mod identic(!) pentru cele două modele de regresie şi care foloseşte media condiţională a outputului Y în raport cu inputul X şi cu $\hat{\theta}$, estimarea în sens MLE pentru parametrul θ.

Veţi considera setul de date de antrenament $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$.

Demonstraţi că pentru fiecare dintre cele două modele de regresie de mai sus, estimarea de verosimilitate maximă a parametrului θ (notaţie: $\hat{\theta}$) satisface următoarea proprietate:

$$\sum_{i=1}^n y_i x_i = \sum_{i=1}^n E[Y|X = x_i, \theta = \hat{\theta}]\; x_i.$$

***Observaţii*:**

1. Remarcaţi faptul că lui y_i din partea stângă a acestei egalităţi îi corespunde în partea dreaptă o medie, şi anume media condiţională a outputului Y, dat fiind inputul $X = x_i$ şi $\hat{\theta}$, estimarea în sens MLE pentru parametrul θ.

2. Datorită faptului că $y_i \in \{0, 1\}$, rezultă că membrul stâng al acestei egalităţi este de fapt suma acelor instanţe x_i pentru care $y_i = 1$. Membrul drept al egalităţii este o sumă ponderată a tuturor instanţelor x_i. Ponderile respective sunt mediile variabilelor aleatoare condiţionate (de tip gaussian, respectiv de tip Bernoulli) scrise în mod unitar sub forma $Y|X = x_i, \theta = \hat{\theta}$.

[405] Într-adevăr, $\sigma(z) = \frac{1}{1+e^{-z}} = \frac{e^z}{1+e^z} = y \in (0,1) \Rightarrow e^z(1-y) = y \Rightarrow z = \ln \frac{y}{1-y}$.

39. **(Regresia logistică multi-class (i.e., regresia softmax): echivalenţa cu un caz particular de mixtură de distribuţii gaussiene multidimensionale)**

prelucrare de Liviu Ciortuz, după
■ □ • ○ ***MIT, 2001 fall, Tommi Jaakkola, HW2, pr. 3.6***
MIT, 2016 fall, R. Barzilay, S. Sra, Weekly Exercises, week 4, pr. 5.bc

***Introducere*:** La problema 19 am introdus modelul de regresie logistică multi-class (i.e., regresia softmax). Distribuţia gaussiană multidimensională a fost introdusă la problema 34 de la capitolul *Fundamente*. Definiţia modelului de tip mixtură de distribuţii gaussiene multidimensionale a fost prezentată la problema 119 tot de la capitolul *Fundamente*.

În acest exerciţiu veţi demonstra că pentru orice $K \in \mathbb{N}$, $K \geq 2$, modelul probabilist învăţat de regresia softmax este echivalent cu un anumit tip de mixtură de distribuţii gaussiene multi-variate.

Considerăm un model de regresie softmax (adică, regresie logistică multi-class) cu K clase, cu parametrii desemnaţi astfel: $w_i \in \mathbb{R}^d$ şi $w_{i0} \in \mathbb{R}$ corespund clasei i, pentru $i = 1, \ldots, K$, iar $(w_i)_j \overset{not.}{=} w_{ij}$, pentru $j = 1, \ldots, d$. Folosind aceste notaţii, distribuţia de probabilitate a posteriori *softmax* este definită astfel:[406]

$$Pr(y|x) = \frac{e^{w_y^\top x + w_{y0}}}{\sum_i e^{w_i^\top x + w_{i0}}}.$$

a. Arătaţi că acestui model softmax îi putem asocia o mixtură de distribuţii gaussiene multi-variate de parametri $(\pi_i, \mu_i, \Sigma_i)_{i=1,\ldots,K}$ astfel încât distribuţia de probabilitate a posteriori softmax de mai sus să coincidă cu distribuţia de probabilitate a posteriori a modelului de mixtură de distribuţii gaussiene:

$$Pr\big(y|x; (\pi_i, \mu_i = w_i, \Sigma_i)_{i=1,\ldots,K}\big) = \frac{\pi_y |\Sigma_y|^{-d/2} \exp\Big(-\frac{1}{2}(x-\mu_y)^\top \Sigma_y^{-1}(x-\mu_y)\Big)}{\sum_i \pi_i |\Sigma_i|^{-d/2} \exp\Big(-\frac{1}{2}(x-\mu_i)^\top \Sigma_i^{-1}(x-\mu_i)\Big)}.$$

b. Rezolvaţi cerinţa de la punctul a pentru cazul particular $K = 2$. Simplificând şi mai mult, puteţi considera $d = 2$ sau $d = 1$.

40. **(Comparaţii între regresia logistică şi alţi clasificatori... şi încă o chestiune, legată de regresia liniară)**

□ • ○ ***CMU, 2009 spring, Tom Mitchell, midterm, pr. 1.1***

Explicaţi pe scurt de ce

a. pentru a rezolva o anumită problemă de clasificare automată, s-ar putea să fie mai bine să folosim [un algoritm de învăţare de] arbori de decizie decât regresia logistică.

[406] Nu este dificil de făcut legătura între definiţia de aici şi cea de la problema 19, chiar dacă ele diferă uşor. De asemenea, folosind definiţia de aici se poate demonstra uşor că separatorii decizionali corespunzători regresiei softmax sunt de tip liniar.

b. pentru a rezolva o anumită problemă de clasificare automată, s-ar putea să fie mai bine să folosim regresia logistică decât algoritmul Bayes Naiv.

c. atunci când aplicăm regresia liniară, selectăm acele valori ale parametrilor care minimizează suma pătratelor erorilor la antrenare.

41. **(Regresia liniară şi regresia logistică; comparaţie între metoda gradientului şi metoda lui Newton: Adevărat sau Fals?)**

□ • ○ ***Stanford, 2014 fall, Andrew Ng, midterm, pr. 6.gba***

a. Presupunem că tu şi prietenul tău vreţi să folosiţi un model de *regresie liniară* pentru a prezice preţul locuinţelor. Tu foloseşti în modelul tău trăsăturile $x_0 = 1$, $x_1 =$ mărimea locuinţei în metri pătraţi şi $x_2 =$ înălţimea acoperişului măsurată în metri. Să zicem că prietenul tău face aceeaşi analiză, folosind exact acelaşi set de date de antrenament, doar că el reprezintă datele folosind trăsăturile $x'_0 = 1$, $x'_1 = x_1$ şi $x'_2 =$ înălţimea acoperişului măsurată în cm (aşadar, $x'_2 = 100\, x_2$).

***i.* Să zicem că atât tu cât şi prietenul tău rezolvaţi problema de regresie liniară folosind metoda analitică (adică, aşa-numitele *ecuaţii normale*, conform problemei 3). Presupunem că nu suntem într-un *caz de degenerare*, aşadar aceste ecuaţii ne dau o soluţie unică pentru parametrii regresiei liniare. Tu obţii parametrii / soluţia $\theta_0, \theta_1, \theta_2$, în vreme ce prietenul tău obţine $\theta'_0, \theta'_1, \theta'_2$. Urmează că $\theta'_0 = \theta_0$, $\theta'_1 = \theta_1$, $\theta'_2 = \frac{1}{100}\theta_2$. Adevărat sau Fals?**

***ii.* Presupunem că atât tu cât şi prietenul tău rezolvaţi problema de regresie liniară iniţializând parametrii cu 0 şi comparând rezultatele pe care le-aţi obţinut după rularea unei singure iteraţii a algoritmului / metodei *gradientului* [*descendent*], varianta *batch*. [LC: Amândoi folosiţi aceeaşi rată a învăţării η.] Tu obţii parametrii / soluţia $\theta_0, \theta_1, \theta_2$, iar prietenul tău obţine $\theta'_0, \theta'_1, \theta'_2$. Urmează că $\theta'_0 = \theta_0$, $\theta'_1 = \theta_1$, $\theta'_2 = \frac{1}{100}\theta_2$. Adevărat sau Fals?**

***Observaţie*: Punctele b şi c care urmează în continuare nu au nicio legătură cu punctul a de mai sus.**

b. Dacă utilizăm metoda *gradientului* [*ascendent*] în varianta *stochastică* pentru antrenarea *regresiei logistice* şi folosim o rată fixă a învăţării, atunci algoritmul va converge neapărat [în mod exact] la soluţia optimă pentru parametrii θ, şi anume $\hat{\theta} = \text{argmax}_\theta \prod_{i=1}^{n} p(y^{(i)}|x^{(i)}; \theta)$, presupunând că s-a ales o rată a învăţării rezonabilă. Adevărat sau Fals?

c. Presupunem că dispuneţi de o implementare pentru *metoda lui Newton* şi metoda *gradientului* [*descendent*] pentru aflarea punctului de optim al unei funcţii convexe. Să zicem că o iteraţie a metodei lui Newton durează dublu faţă de o iteraţie a metodei gradientului [descendent]. Prin urmare, rezultă că metoda gradientului [descendent] va converge mai repede la soluţia optimă. Adevărat sau Fals?

1.2.3 Modele liniare generalizate

***Introducere*:**[407]

Considerăm că avem o *problemă* de clasificare sau de regresie, în care am dori să prezicem valoarea unei variabile aleatoare y în funcţie de [o altă variabilă aleatoare] x. Pentru a deriva un model liniar generalizat (engl., Generalized Linear Model, GLM) pentru această problemă, vom considera următoarele *trei presupoziţii* în legătură cu distribuţia condiţională a lui y în raport cu x, precum şi în legătură cu modelul nostru:

1. **$y|x;\theta \sim ExponentialFamily(\eta)$. Aşadar, daţi fiind x şi θ, variabila y urmează o anumită distribuţie din familia exponenţială, caracterizată de parametrul η:**[408]
$$p(y|\eta) = b(y)\exp(\eta^\top T(y) - a(\eta)).$$
2. ***Scopul* nostru este ca, dat fiind x, să prezicem media valorii lui $T(y)$ în raport cu x. În cele mai multe dintre exemplele didactice, se consideă $T(y) = y$. Aşadar, dorim ca predicţia $h(x)$ generată (ca output) de către *ipoteza învăţată* h să satisfacă relaţia $h(x) = E[y|x]$.**[409]
3. **Legătura dintre *parametrul natural* η şi variabila de intrare x este de tip liniar: $\eta = \theta^\top x$. (Sau, dacă η este vector, atunci $\eta_i = \theta_i^\top x$.)**

Cea de-a treia presupoziţie poate părea mai puţin justificată decât celelalte două, şi ar fi probabil mai indicat să o gândim ca fiind un element de *concepţie* (engl., design choice) pentru modelele liniare generalizate (engl., Generalized Linear Models, GLM).

Aceste trei presupoziţii / elemente de concepţie ne permit să deducem o foarte elegantă clasă de algoritmi de învăţare, numită GLM, care au multe proprietăţi dezirabile, cum ar fi de pildă facilitatea învăţării. Mai mult, modelele rezultate sunt adeseori foarte potrivite pentru a „caracteriza“ diverse tipuri de distribuţii peste y. De exemplu, se poate arăta că atât regresia logistică cât şi regresia liniară (în varianta celor mai mici pătrate) pot fi văzute ca fiind instanţe ale clasei de algoritmi de învăţare automată GLM.

Vom mai adăga câteva *chestiuni de terminologie*, precizăm că funcţia g care returnează media distribuţiei în raport cu parametrul natural η (adică, $g(\eta) = E[T(y);\eta]$) este numită *funcţia de răspuns canonică* (engl., canonical response function). Inversa sa, g^{-1}, este numită *funcţia de legătură canonică* (engl., canonical link function). Se poate demonstra uşor că funcţia de răspuns canonică pentru distribuţia gaussiană [cu varianţa 1] este funcţia identitate, iar pentru distribuţia Bernoulli este funcţia logistică.

[407] Cf. Andrew Ng, Stanford University, Machine learning course, Lecture notes, Part III — Generalized Linear Models.

[408] Vedeţi problema 123 de la capitolul de *Fundamente*.

[409] Veţi putea observa că această presupunere este satisfăcută atât în cazul regresiei logistice cât şi în cazul regresiei liniare, de către definiţia respectivă aleasă pentru $h_\theta(x)$. De exemplu, la regresia logistică avem $h_\theta(x) \overset{def.}{=} \sigma(\theta^\top x) = p(y = 1|x;\theta) = 0 \cdot p(y = 0|x;\theta) + 1 \cdot p(y = 1|x;\theta) = E[y|x;\theta]$.

42. **(GLM: particularizare pentru cazul distribuţiei geometrice şi al distribuţiei Poisson)**

□ • · ***Stanford, 2014 fall, Andrew Ng, midterm, pr. 3***
Stanford, 2008 fall, Andrew Ng, HW1, pr. 3.abc
University of Chicago, 2004 spring, Michael Eichler, HW3, pr. 4.bd

Vă readucem aminte că forma standard a familiei de distribuţii exponenţiale este

$$p(y|\eta) = b(y)\exp(\eta^\top T(y) - a(\eta)).$$

De asemenea, vă readucem aminte că distribuţia geometrică de parametru ϕ are funcţia masă de probabilitate (p.m.f.) definită în felul următor:

$$p(y|\phi) = (1-\phi)^{y-1}\phi, \text{ pentru } y = 1, 2, \ldots .$$

a. Arătaţi că distribuţia geometrică face parte din familia de distribuţii exponenţiale şi precizaţi valorile corespunzătoare pentru $b(y)$, η, $T(y)$ şi $a(\eta)$.

b. Presupunem că vrem să facem regresie folosind un model liniar generalizat (engl., Generalized Liniar Model, GLM) cu o *variabilă de răspuns* geometrică. Precizaţi care este – în acest caz – *funcţia canonică de răspuns*. (Puteţi utiliza faptul că media distribuţiei geometrice este by $1/\phi$.)

c. Vă readucem aminte că în modelul GLM, se consideră că $\eta = \theta^\top x$, cu θ şi x din $\mathbb{R}^d$. De asemenea, dat fiind un set de date de antrenament $\{(x_i, y_i)\}_{i=1}^m$ cu $x_i \in \mathbb{R}^d$ şi $y_i \in \mathbb{R}$, se ştie că log-verosimilitatea unui exemplu (x_i, y_i) este $\ln p(x_i, y_i)$. Calculaţi derivata acestei log-verosimilităţi în raport cu θ_j, pentru a obţine regula gradientului ascendent în varianta stochastică, în vederea antrenării unui model GLM cu răspunsuri geometrice y şi funcţie canonică de răspuns.

d. Calculaţi matricea hessiană H pentru $\ell(\theta) = \sum_{i=1}^m \ln p(y_i|x_i;\theta)$, funcţia de log-verosimilitate a datelor de antrenament, şi precizaţi cum va arăta aplicarea unui pas al metodei lui Newton pentru maximizarea acestei funcţii de log-verosimilitate.

e. Vă cerem să satisfaceţi din nou cerinţele de la punctele $a-d$ de mai sus, însă de data aceasta lucrând cu distribuţia Poisson de parametru λ. Vă readucem aminte că pentru această distribuţie funcţia masă de probabilitate are forma

$$p(y|\lambda) = \frac{e^{-\lambda}\lambda^y}{y!},$$

iar media este egală cu λ.

43. **(GLM: particularizare pentru cazul distribuţiei gaussiene unidimensionale)**

□ • · ***Stanford, 2015 fall, Andrew Ng, midterm, pr. 3***

Vă readucem aminte că forma standard a familiei de distribuţii exponenţiale este

$$p(y|\eta) = b(y)\exp(\eta^\top T(y) - a(\eta)).$$

De asemenea, vă readucem aminte că funcţia densitate de probabilitate pentru distribuţia gaussiană [unidimensională] este definită în felul următor:

$$p(y|\mu,\sigma^2) = \frac{1}{\sigma^2}\exp\Big(-\frac{1}{2\sigma^2}(y-\mu)^2\Big).$$

a. Arătaţi că distribuţia gaussiană face parte din familia de distribuţii exponenţiale. Specificaţi valorile corespunzătoare pentru $b(y)$, η, $T(y)$, şi $a(\eta)$.

***Indicaţie*: η şi $T(y)$ vor fi vectori bidimensionali; veţi folosi notaţia $\eta = (\eta_1, \eta_2)^\top$. Veţi exprima $a(\eta)$ în funcţie de η_1 şi η_2.**

b. Presupunem că dispunem de setul de date de antrenament $\{(x_i, y_i)\}_{i=1}^m$, cu $x_i \in \mathbb{R}^d$ şi $y \in \mathbb{R}$. Pornind de la $p(y|\eta)$, forma standard a familiei exponenţiale dată mai sus, calculaţi expresia [generală a] matricei hessiene pentru funcţia de log-verosimilitate $\ell(\theta) = \sum_{i=1}^m \ln p(y_i|x_i;\theta)$, cu $\theta \in \mathbb{R}^d$. (Vă readucem aminte că în modelul GLM se lucrează cu $\eta = \theta^\top x$.) Veţi scrie expresia acestei matrice hessiene în raport cu x, η_1 şi η_2.

c. Folosind rezultatul de la punctul b, arătaţi că matricea hessiană [pentru funcţia de log-verosimilitate $\ell(\theta)$] este negativ semidefinită, adică $z^\top H z \le 0$ pentru orice $z \in \mathbb{R}^d$.

44. (GLM: calculul funcţiei de log-verosimilitate în cazul general, condiţii suficiente pentru concavitate; calculul gradientului; calculul matricei hessiene)

prelucrare de Liviu Ciortuz, după
□ • · ***Stanford, 2009 fall, Andrew Ng, practice midterm, pr. 1***
Stanford, 2008 fall, Andrew Ng, HW1, pr. 3.d

Vă readucem aminte că în modelul liniar generalizat (engl., generalized linear models, GLM) se presupune că variabila de răspuns y (condiţionată în raport cu x) urmează o distribuţie care face parte din *familia de distribuţii exponenţiale*:

$$p(y|\eta) = b(y)\exp(\eta^\top T(y) - a(\eta)), \tag{228}$$

unde $\eta = \theta^\top x$. În acest exerciţiu vom presupune că $\eta \in \mathbb{R}$.

a. Ştiţi că fiind dat setul de date de antrenament $\{(x_i, y_i)\}_{i=1}^m$, funcţia de log-verosimilitate [condiţională] se defineşte astfel:

$$\ell(\theta) = \sum_{i=1}^m \ln p(y_i|x_i;\theta).$$

Formulaţi un set de condiţii pe care trebuie să le satisfacă $b(y)$, $T(y)$ şi $a(\eta)$ din relaţia (228) în aşa fel încât funcţia $\ell(\theta)$ să fie concavă (asigurându-ne astfel că ea are o valoare maximă unică). Condiţiile pe care le veţi formula trebuie să fie rezonabile şi, de asemenea, trebuie să fie cât se poate de laxe (engl., weak).[410]

[410] De pildă, răspunsul „orice $b(y)$, $T(y)$ şi $a(\eta)$ astfel încât funcţia $\ell(\theta)$ să fie concavă“ nu este rezonabil. Tot aşa, condiţii foarte restrictive, care se aplică – de exemplu – doar pentru unele modele particulare de GLMs, nu sunt rezonabile.

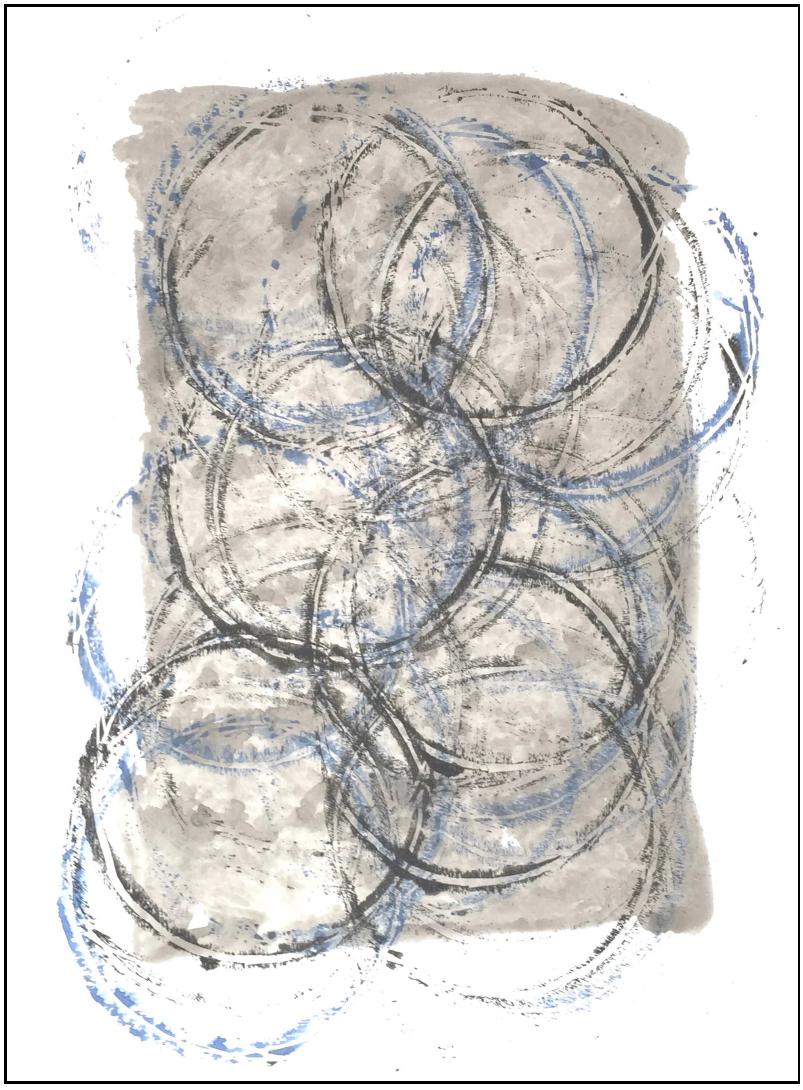
b. Verificaţi că în cazul în care variabila de răspuns urmează distribuţia gaussiană standard, avem $b(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}}$, $T(y) = y$ **şi** $a(\eta) = \frac{\eta^2}{2}$. **Verificaţi de asemenea că setul / sistemul de condiţii pe care le-aţi formulat la punctul** a **este satisfăcut în acest caz particular.**

c. Presupunem că folosim un model GLM având ca *variabilă de răspuns* orice membru al familiei de distribuţii exponenţiale, cu $T(y) = y$ **şi *funcţia de răspuns canonic* pentru această familie.**

Arătaţi că atunci când trebuie să aplicăm metoda gradientului ascendent în varianta stochastică pentru a determina maximul funcţiei de log-verosimilitate $\ln p(y|X, \theta)$**, obţinem următoarea *regulă de actualizare*:**

$$\theta_i \leftarrow \theta_i - \alpha(h(x) - y)x_i,$$

unde α **este *rata de învăţare*.**



© *M. Romanică*

2 Clasificare bayesiană

Sumar

Noţiuni preliminare

– **probabilităţi şi probabilităţi condiţionate;**

– **formula lui Bayes: ex. 5.b;**
cap. *Fundamente*, ex. 6, ex. 7, ex. 94, ex. 95;

– **independenţa [condiţională a] evenimentelor aleatoare:**
cap. *Fundamente*, ex. 4, ex. 91, ex. 92;

– **independenţa [condiţională a] variabilelor aleatoare: ex. 9, ex. 10, ex. 12, ex. 33-44; vedeţi şi cap. *Fundamente*, ex. 15, ex. 30, ex. 99.b, ex. 108, ex. 106;**

– **distribuţii probabiliste comune, marginale şi condiţionale: ex. 6, ex. 10, ex. 12, ex. 33; vedeţi şi cap. *Fundamente*, ex. 13, ex. 14;**

– **distribuţia gaussiană: de la cap. *Fundamente*, ex. 32, ex. 33 (pentru cazul unidimensional), ex. 35 (pentru cazul bidimensional), ex. 20, ex. 34, ex. 36, ex. 37 (pentru cazul multidimensional);**

– **estimarea parametrilor pentru distribuţii de tip Bernoulli (ex. 5.a), categorial şi gaussian (ex. 15.a);**[411]

– **ipoteze MAP vs. ipoteze ML:**
formulare [ca soluţii la] probleme de optimizare:[412] **ex. 26;**
exemplificare: ex. 1, ex. 2, ex. 3, ex. 24, ex. 25, ex. 43;
exemplificare în cazul arborilor de decizie: ex. 4;

– **regresia logistică, chestiuni introductive:**[413] **de la cap. *Metode de regresie*, ex. 13.**

Algoritmi de clasificare bayesiană

– **Algoritmul Bayes Naiv şi algoritmul Bayes Optimal:**[414]
formulare ca probleme de optimizare / estimare în sens MAP: cartea ML, pag. 157;
pseudo-codul algoritmului Bayes Naiv (cf. cartea ML, pag. 177):

***Training*:**

for each value v_j **of the output attribute**
$\hat{P}(v_j) \leftarrow$ **estimate** $P(v_j)$

[411] De la cap. *Fundamente*, pentru estimarea parametrului unei distribuţii Bernoulli vedeţi ex. 43 şi ex. 125.a, pentru estimarea parametrilor unei distribuţii categoriale vedeţi ex. 44, iar pentru estimarea parametrilor unei distribuţii gaussiene vedeţi ex. 50, ex. 51, ex. 135 (pentru cazul unidimensional) şi ex. 53 (pentru cazul multidimensional).

[412] Vedeţi cartea ML, pag. 156-157.

[413] Vedeţi draftul capitolului suplimentar pentru cartea ML a lui T. Mitchell, *Generative and discriminative classifiers: Naive Bayes and logistic regression* (în special secţiunea 3).

[414] La secţiunea aceasta, precum şi la următoarea secţiune, considerăm (implicit) că toate variabilele de intrare sunt de tip Bernoulli sau, mai general, de tip categorial. După aceea vom considera şi variabile de intrare de tip continuu, în genere de tip gaussian. Variabila de ieşire se consideră întotdeauna de tip Bernoulli / categorial.

for each value a_i **of each input attribute** a
$\hat{P}(a_i|v_j) \leftarrow$ **estimate** $P(a_i|v_j)$

Classification **of the test instance** $(a_1, a_2, \ldots, a_n)$**:**
$v_{NB} = \text{argmax}_{v_j \in V} \hat{P}(v_j) \prod_{a_i \in x} \hat{P}(a_i|v_j)$**.**

Justificarea regulii de decizie **pentru algoritmul Bayes Naiv:** [415]

$$v_{MAP} = \underset{v_j \in V}{\text{argmax}}\, P(v_j|a_1, a_2 \ldots a_n) = \underset{v_j \in V}{\text{argmax}}\, \frac{P(a_1, a_2 \ldots a_n|v_j)P(v_j)}{P(a_1, a_2 \ldots a_n)}$$

$$= \underset{v_j \in V}{\text{argmax}}\, P(a_1, a_2 \ldots a_n|v_j)P(v_j) \overset{indep.\ cdt.}{=} \underset{v_j \in V}{\text{argmax}} \prod_i P(a_i|v_j)P(v_j) \overset{not.}{=} v_{NB}.$$

– **exemple de aplicare: ex. 5, ex. 6, ex. 7, ex. 27, ex. 28, ex. 29;**

– **aplicarea / adaptarea algoritmului Bayes Naiv pentru clasificare de texte:**[416] **ex. 8, ex. 30;**
folosirea regulii "add-one" [a lui Laplace] pentru „netezirea" parametrilor: ex. 8, ex. 31;

– **calculul ratei medii a erorilor pentru algoritmii Bayes Naiv şi Bayes Optimal: ex. 10, ex. 11, ex. 32, ex. 33, ex. 34, ex. 35.a-d, ex. 36, ex. 44;**

– **evidenţierea grafică a neconcordanţei predicţiilor făcute de clasificatorii Bayes Naiv şi Bayes Optimal: ex. 12;**

– **exemple de clasificatori bayesieni care combină avantajele algoritmilor Bayes Naiv şi Bayes Optimal: ex. 9, ex. 35.e.**

Proprietăţi ale algoritmilor Bayes Naiv şi Bayes Optimal

- **(P0) dacă proprietatea de independenţă condiţională a atributelor de intrare în raport cu variabila de ieşire se verifică, atunci rezultatele produse de către cei doi algoritmi (Bayes Naiv şi Bayes Optimal) în faza de testare coincid; în caz contrar, rezultatele celor doi algoritmi pot sa nu coincidă;**
- **(P1) numărul de parametri necesari de estimat din date: liniar pentru Bayes Naiv** $(2d+1)$ **şi exponenţial pentru Bayes Optimal** $(2^{d+1}-1)$**:**[417] **ex. 7.e, ex. 27.b, ex. 29.ab, ex. 35.ac, ex. 37;**
- **(P2) complexitatea algoritmului Bayes Naiv:**[418]

  complexitatea de spaţiu: $\mathcal{O}(dK)$
  complexitatea de timp:
  la antrenare: $\mathcal{O}(dn)$
  la testare: $\mathcal{O}(dK)$**,**

[415] *Justificarea regulii de decizie* pentru algoritmul Bayes Comun / Optimal:

$$v_{MAP} = \underset{v_j \in V}{\text{argmax}}\, P(v_j|a_1, a_2 \ldots a_n) = \ldots$$

$$= \underset{v_j \in V}{\text{argmax}}\, P(a_1, a_2 \ldots a_n|v_j)P(v_j) = \underset{v_j \in V}{\text{argmax}}\, P(a_1, a_2 \ldots a_n, v_j) \overset{not.}{=} v_{JB}.$$

[416] Atenţie: Noi am folosit aici versiunea de bază a algoritmului Bayes Naiv; varianta "bag of words" (vedeţi cartea Machine Learning a lui Tom Mitchell, pag. 183) diferă uşor de aceasta.

[417] Numărul de parametri indicaţi în paranteze se referă la cazul când atât atributele de intrare cât şi atributul de ieşire sunt de tip Bernoulli.

[418] Şi aici se consideră că atât atributele de intrare cât şi atributul de ieşire sunt de tip Bernoulli.

unde n este numărul de exemple, d este numărul de atribute de intrare, iar K este numărul de clase;

- **(P3) algoritmul Bayes Optimal poate produce eroare [la clasificare] din cauza faptului că ia decizia în sensul unui vot majoritar. Algoritmul Bayes Naiv are şi el această „sursă“ de eroare; în plus el poate produce eroare şi din cauza faptului că lucrează cu presupoziţia de independenţă condiţională (care nu este satisfăcută în mod neapărat);**
- **(P4) Algoritmul Bayes Naiv nu garantează obţinerea ratei medii a erorii 0 la antrenare pe date consistente: ex. 11.a;**[419]
  - **(P5) este posibil ca rata medie a erorii la antrenare produsă de Bayes Naiv să fie 0 chiar dacă independenţa condiţională este încălcată: ex. 36;**
- **(P6) acurateţea [la clasificare a] algoritmului Bayes Naiv scade atunci când unul sau mai multe atribute de intrare sunt duplicate: ex. 10.d, ex. 33.def; în schimb, algoritmul Bayes Optimal nu este sensibil la duplicarea atributelor de intrare; atât Bayes Naiv cât şi Bayes Optimal sunt sensibili la duplicarea exemplelor de antrenament;**
- **(P7) în cazul „învăţării“ unei funcţii booleene (oarecare), rata medie a erorii produse la antrenare de către algoritmul Bayes Optimal (spre deosebire de Bayes Naiv!) este 0: ex. 35.d;**[420]
- **(P8) complexitatea de eşantionare: de ordin logaritmic pentru Bayes Naiv şi de ordin exponenţial pentru Bayes Optimal: ex. 13;**
- **(P9) corespondenţa dintre regula de decizie a algoritmului Bayes Naiv (când toate variabilele de intrare sunt de tip Bernoulli) şi regula de decizie a *regresiei logistice* şi, în consecinţă, liniaritatea graniţelor de decizie: ex. 14, ex. 38.c, ex. 39.**

– **o perspectivă info-teoretică asupra algoritmului Bayes Naiv: ex. 149 (şi ex. 148) de la capitolul de *Fundamente*.**[421]

– ***comparaţii* între algoritmul Bayes Naiv şi alţi algoritmi de clasificare automată: ex. 42, ex. 44.**

Algoritmii Bayes Naiv şi Bayes Optimal cu variabile de intrare *de tip gaussian*

• Aplicare: G[N]B: ex. 15, ex. 46 şi ex. 56; GJB: ex. 52, ex. 53 şi ex. 54; GNB vs. GJB: ex. 21.
• Numărul de parametri necesari de estimat din date: ex. 49.

• Proprietăţi:

– **(P0′) presupunem că variabila de ieşire este booleană, i.e. ia valorile 0 sau 1; dacă pentru orice atribut de intrare, variabilele condiţionale $X_i|Y = 0$ şi $X_i|Y = 1$ au distribuţii gaussiene de varianţe egale ($\sigma_{i0} = \sigma_{i1}$), atunci regula de decizie GNB (Gaussian Naive Bayes) este echivalentă (ca formă) cu cea a**

[419] În contextul învăţării conceptului XOR (care reprezintă un set de date consistent), algoritmul Bayes Naiv obţine pentru rata medie a erorii la antrenare valoarea maximă posibilă în cazul clasificării binare: 50%!
[420] Rezultatul este valabil pentru cazul (mai general) al funcţiilor definite în sens matematic (corespondenţe de tip “many-to-one”) cu argumente discrete şi valori discrete.
[421] Vedeţi legătura cu relaţia de echivalenţă max likelihood $\Leftrightarrow$ min cross-entropy.

regresiei logistice, deci separarea realizată de către algoritmul GNB este de formă liniară: demonstraţie: ex. 17; exemplificare în $\mathbb{R}$: ex. 46.a; exemplificare în $\mathbb{R}^2$: ex. 48.c;

- **(P1′) similar, presupunem că variabila de ieşire este booleană;**
  dacă variabilele de intrare (notaţie: $X = (X_1, \ldots, X_d)$) au distribuţiile [comune] condiţionale $X|Y = 0$ şi $X|Y = 1$ de tip gaussian [multidimensional], cu matricele de covarianţă egale ($\Sigma_0 = \Sigma_1$), atunci regula de decizie a algoritmului "full" / Joint Gaussian Bayes este şi ea echivalentă (ca formă) cu cea a regresiei logistice, deci separarea realizată este tot de formă liniară: ex. 18, ex. 20.a.$i - ii$;
- **(P2′) când variabilele de intrare satisfac condiţii mixte de tip (P0′) sau (P7), atunci concluzia – separare liniară – se menţine: ex. 51.b;**
- **(P3′) dacă în condiţiile de la propoziţiile (P0′)-(P2′) presupoziţia de independenţă condiţională este satisfăcută, iar numărul de instanţe de antrenament tinde la infinit, atunci rezultatul de clasificare obţinut de către algoritmul Bayes Naiv gaussian este identic cu cel al regresiei logistice: ex. 22.a.**

  Atunci când presupoziţia de independenţă condiţională nu este satisfăcută, iar numărul de instanţe de antrenament tinde la infinit, regresia logistică se comportă mai bine decât algoritmul Bayes Naiv [gaussian]: ex. 22.b;
- **(P4′) nu există o corespondenţă 1-la-1 între parametrii calculaţi de regresia logistică şi între parametrii calculaţi de algoritmul Bayes Naiv [gaussian]: ex. 23.a;**
- **(P5′) atunci când varianţele distribuţiilor gaussiene care corespund probabilităţilor condiţionale $P(X_i|Y = k)$ depind şi de eticheta k, separatorul decizional determinat de algoritmul Bayes Naiv gaussian nu mai are (în mod necesar) forma regresiei logistice: ex. 46.bc, ex. 50;**
  similar, pentru algoritmul Bayes Optimal gaussian, atunci când $\Sigma_0 \neq \Sigma_1$, ecuaţia separatorului decizional este de ordin pătratic: ex. 19, ex. 20.a.$iii - vi$, ex. 49.e (separatorul decizional este un cerc), ex. 49.f (o hiperbolă), ex 54 (o reuniune de două drepte);
- **(P6′) parametrii algoritmilor Bayes Naiv gaussian şi Bayes Optimal gaussian se pot estima în timp liniar în raport cu numărul de instanţe din setul de date de antrenament: ex. 23.b, ex. 48.bd.**

2.1 Clasificare bayesiană — Probleme rezolvate

2.1.1 Ipoteze de probabilitate maximă a posteriori (MAP)

1. **(Formula lui Bayes; medii ale unor variabile aleatoare discrete; [ipoteze MAP;] măsuri statistice folosite în clasificare)**

■ • ***CMU, 2009 fall, Geoff Gordon, HW1, pr. 2***

O anumită boală afectează una din 500 de persoane în medie. Identificarea persoanelor care au această boală se poate face cu ajutorul unei analize a sângelui, care costă 100 de dolari de persoană. Această analiză indică în cazul unui rezultat *pozitiv* faptul că se *poate* ca persoana respectivă să sufere de acea boală.

Testul / analiza are o *sensibilitate* (engl., *sensitivity* sau *recall*) perfectă — adică raportul dintre numărul instanţelor pozitive identificate ca atare de acel test şi numărul total de instanţe pozitive este 1 —, ceea ce înseamnă că pentru orice persoană care are boala respectivă, rezultatul testului este pozitiv cu probabilitate de 100%. Pe de altă parte, testul are o *specificitate* — raportul dintre numărul instanţelor negative identificate ca atare de acel test şi numărul total de instanţe negative — de 99%, adică o persoană care nu suferă de acea boală va avea cu probabilitate de 1% rezultatul testului pozitiv.

a. Se testează o persoană selectată în mod aleatoriu, iar rezultatul este pozitiv. Care este probabilitatea ca persoana respectivă să sufere de acea boală?

b. Există şi un al doilea test, care costă 10.000 de dolari şi are atât sensibilitatea cât şi specificitatea de 100%. Dacă am cere ca toate persoanele detectate pozitiv la testul precedent să fie supuse acestui test mult mai scump, care ar fi costul mediu pentru testarea / analiza uni individ?

c. O companie farmaceutică încearcă să reducă preţul celui de-al doilea test (care este perfect), adică are atât *sensibilitatea* cât şi *specificitatea* de 100%. Cât ar trebui să fie preţul acesta pentru ca primul test să nu mai fie necesar? (Adică, la ce preţ va rezulta că este mai ieftin să se utilizeze doar testul al doilea, decât să se facă ambele teste, ca la punctul b?)

Răspuns:

Definim următoarele variabile aleatoare:

B: ia valoarea 1 / adevărat pentru persoanele care suferă de această boală şi 0 / fals în caz contrar

T_1: rezultatul primului test, care poate fi + (în caz de boală) sau −

T_2: rezultatul celui de-al doilea test, care poate fi tot + sau −.

Folosind aceste variabile aleatoare, datele problemei se pot rescrie astfel:

$$P(B) = \frac{1}{500}$$
$$P(T_1 = + \mid B) = 1$$
$$P(T_1 = + \mid \bar{B}) = \frac{1}{100}$$
$$P(T_2 = + \mid B) = 1$$
$$P(T_2 = + \mid \bar{B}) = 0$$

a. Probabilitatea ca o persoană oarecare să sufere de boala respectivă, ştiind că rezultatul primului test este pozitiv, este $P(B \mid T_1 = +)$ şi se calculează cu ajutorul formulei lui Bayes:

$$\begin{aligned} P(B \mid T_1 = +) &= \frac{P(T_1 = + \mid B) \cdot P(B)}{P(T_1 = + \mid B) \cdot P(B) + P(T_1 = + \mid \bar{B}) \cdot P(\bar{B})} \\ &= \frac{1 \cdot \frac{1}{500}}{1 \cdot \frac{1}{500} + \frac{1}{100} \cdot \frac{499}{500}} = \frac{100}{599} \approx 0.1669 \end{aligned}$$

***Observaţie*: Remarcaţi faptul că $P(\bar{B} \mid T_1 = +) = 0.8331$. Este imediat care dintre cele două probabilităţi, $P(B \mid T_1 = +)$ şi $P(\bar{B})$, este mai mare. În mod echivalent, pentru a stabili acest fapt era suficient să comparăm $P(B \mid T_1 = +)$ cu $1/2$, sau să stabilim care dintre produsele $P(T_1 = + \mid B) \cdot P(B)$ şi $P(T_1 = + \mid \bar{B}) \cdot P(\bar{B})$ este mai mare. Aceste observaţii sunt utile pentru că ele fac legătura cu noţiunea de *ipoteză de probabilitate maximă a posteriori* (vedeţi pr. 3).**

b. Vom calcula costul mediu al testării unui individ folosind o nouă variabilă aleatoare, notată cu C, care reprezintă costul total de testare al unei persoane. Notând cu c_1 şi c_2 costurile celor două teste, putem scrie:

$$C = \begin{cases} c_1 & \text{dacă persoana este testată doar cu primul test} \\ c_1 + c_2 & \text{dacă persoana este testată cu ambele teste} \end{cases}$$

O persoană este testată cu al doilea test doar dacă are rezultatul pozitiv la primul test, deci probabilităţile pentru variabila aleatoare C sunt:

$$P(C = c_1) = P(T_1 = -) \text{ şi } P(C = c_1 + c_2) = P(T_1 = +)$$

Costul mediu cerut de problemă este media variabilei aleatoare C, deci se poate calcula astfel:

$$\begin{aligned} E[C] &= c_1 \cdot P(C = c_1) + (c_1 + c_2) \cdot P(C = c_1 + c_2) \\ &= c_1 \cdot P(T_1 = -) + (c_1 + c_2) \cdot P(T_1 = +) \end{aligned}$$

Ştim că $P(T_1 = -) = 1 - P(T_1 = +)$, iar din formula probabilităţii totale avem:

$$\begin{aligned} P(T_1 = +) &= P(T_1 = + \mid B) \cdot P(B) + P(T_1 = + \mid \bar{B}) \cdot P(\bar{B}) \\ &= 1 \cdot \frac{1}{500} + \frac{1}{100} \cdot \frac{499}{500} = \frac{599}{50000} = 0.01198 \end{aligned}$$

Aşadar, vom obţine:

$$\begin{aligned} E[C] &= c_1 \cdot (1 - P(T_1 = +)) + (c_1 + c_2) \cdot P(T_1 = +) \\ &= c_1 - c_1 \cdot P(T_1 = +) + c_1 \cdot P(T_1 = +) + c_2 \cdot P(T_1 = +) \\ &= c_1 + c_2 \cdot P(T_1 = +) \\ &= 100 + 10000 \cdot \frac{599}{50000} \\ &= 219.8 \approx 220\$ \end{aligned}$$

c. Notăm cu c_n noul preţ pentru al doilea test (T_2'). Acest preţ trebuie să fie mai mic sau egal cu costul mediu de aplicare al ambelor teste (T_1 şi T_2'), deci:

$$\begin{aligned} c_n \leq E[C'] &= c_1 \cdot P(C = c_1) + (c_1 + c_n) \cdot P(C = c_1 + c_n) \\ &= c_1 + c_n \cdot P(T_1 = +) = 100 + c_n \cdot \frac{599}{50000} \end{aligned}$$

Rezolvând ecuaţia $c_n = 100 + c_n \cdot 0.01198$, obţinem $c_n \approx 101.2125$.

Aşadar, dacă al doilea test ar costa cel mult 101.21 dolari, atunci primul test n-ar mai fi necesar.

2. **(Formula lui Bayes; [ipoteze MAP;] inferenţe statistice)**

prelucrare de Liviu Ciortuz, după
CMU, 2009 fall, Geoff Gordon, HW1, pr. 1
("Monty's haunted house" problem)

Fără să ştii cum s-a întâmplat, ai nimerit într-o casă plină de fantome. Acum eşti blocat în faţa unui perete care are 3 uşi (pentru convenienţă, le vom nota cu numerele 1, 2, 3). Apare o fantomă care îţi spune: „Scăparea ta este să ieşi din casă printr-una din aceste uşi. Însă doar una dintre ele dă în afară; celelalte două sunt păzite de câte un monstru care te va ucide imediat dacă încerci să ieşi pe acolo. Trebuie să alegi o uşă!"

Decizi să alegi la întâmplare una din cele trei uşi, să zicem uşa 1.

***Observaţie*: Probabilitatea *a priori* ca uşa aceasta să dea în afară este 1/3. (La fel este şi în cazul celorlalte două uşi.) Prin adăgarea altor informaţii — vedeţi continuarea problemei — probabilitatea *a posteriori* a aceluiaşi eveniment se poate modifica, deci într-un caz fericit ea poate creşte.**

Într-adevăr, tocmai când ai pus mâna pe clanţă ca să o deschizi, fantoma îţi spune: „Aşteaptă puţin! Îţi voi mai da o informaţie." Zicând aceasta, fantoma întredeschide o altă uşă (să zicem uşa 2) şi îţi arată că în spatele ei se află un monstru groaznic. Apoi fantoma te întreabă: „Vrei acum să alegi ultima uşă (adică uşa 3) sau consideri că este mai bine să rămâi la alegerea pe care ai făcut-o iniţial?"

Fantoma te mai ajută spunându-ţi că în alegerea ei, ea a urmat o *strategie* bazată pe două *principii*:

P1. După ce tu ai făcut alegerea iniţială, fantoma a ales una din celelalte două uşi, mai precis o uşă în spatele căreia se află un monstru. (Este evident că întotdeauna există o astfel de uşă, indiferent de alegerea ta.)

P2. În cazul în care ambele uşi între care are de ales fantoma au în spate câte un monstru, ea procedează după *una* din următoarele trei *variante* (pe care o alege a priori şi ţi-o aduce la cunoştinţă):

a. **Fantoma alege una din cele două uşi cu probabilitate egală** $(1/2)$**.**
b. **Dacă, aşa cum a fost cazul mai sus, tu ai ales uşa 1, fantoma alege uşa 2 (cu probabilitate 1).**
c. **Dacă, tot aşa, tu ai ales uşa 1, fantoma alege uşa 3 (cu probabilitate 1).**
(*Notă*: Alte variante nu interesează pentru rezolvarea care se cere mai jos.)

Se cere ca, pentru fiecare din aceste 3 variante în parte, să determini probabilităţile ca ieşirea să se afle în spatele uşii 1, respectiv în spatele uşii 3, dacă fantoma a deschis uşa 2.

***Indicaţie*: Pentru rezolvare, vom folosi două *variabile aleatoare*:**

- O **(de la engl. outside), cu valori în mulţimea** $\{1, 2, 3\}$**, indicând unde este ieşirea cea bună;**
- G **(de la engl. ghost), pentru a desemna uşa aleasă de fantomă.**

Pentru fiecare din variantele de strategie ale fantomei (a, b, c)**, folosind formula lui Bayes se calculează** $P(O = 1 \mid G = 2)$ **şi** $P(O = 3 \mid G = 2)$**.**

Răspuns:

Pentru variabila aleatoare O **avem probabilităţi *a priori* egale pentru toate ieşirile:**

$$P(O = 1) = P(O = 2) = P(O = 3) = \frac{1}{3}. \tag{229}$$

Folosind formula lui Bayes combinată cu formula probabilităţii totale, probabilităţile (*a posteriori*) cerute vor putea fi calculate astfel:

$$P(O = 1 \mid G = 2) = \frac{P(G = 2 \mid O = 1) \cdot P(O = 1)}{P(G = 2 \mid O = 1) \cdot P(O = 1) + P(G = 2 \mid O = 3) \cdot P(O = 3)}$$
$$P(O = 3 \mid G = 2) = \frac{P(G = 2 \mid O = 3) \cdot P(O = 3)}{P(G = 2 \mid O = 3) \cdot P(O = 3) + P(G = 2 \mid O = 1) \cdot P(O = 1)}.$$

Remarcăm că la fiecare din numitorii celor două fracţii de mai sus ar fi trebuit să mai scriem $P(G = 2 \mid O = 2) \cdot P(O = 2)$**, însă acesta este** 0 **fiindcă** $P(G = 2 \mid O = 2) = 0$**, conform principiului P1. Deci** $P(O = 3 \mid G = 2) = 1 - P(O = 1 \mid G = 2)$**.**

Pentru a exprima succint probabilităţile condiţionate necesare pentru calculul probabilităţilor $P(O = 1 \mid G = 2)$ **şi** $P(O = 3 \mid G = 2)$ **folosind formulele de mai sus, completăm următorul tabel:**[422]

G	O	$P(G \mid O)$		
		varianta a	**varianta** b	**varianta** c
2	1	1/2	1	0
3	1	1/2	0	1
2	2	0	0	0
3	2	1	1	1
2	3	1	1	1
3	3	0	0	0

[422] Se observă că pe fiecare dintre cele trei coloane din tabel, ultimele patru linii sunt identice, ceea ce este natural, conform principiilor P1 şi P2.

În aceste condiţii putem calcula uşor probabilităţile cerute în fiecare din cele trei variante:

***Varianta a*:**

$$P(O=1 \mid G=2)=\frac{\frac{1}{2}\cdot\frac{1}{3}}{\frac{1}{2}\cdot\frac{1}{3}+1\cdot\frac{1}{3}}=\frac{1}{3} \text{ şi } P(O=3 \mid G=2)=1-\frac{1}{3}=\frac{2}{3},$$

deci vei alege uşa a treia, fiindcă ea are probabilitatea mai mare de a te duce afară.

***Varianta b*:**

$$P(O=1 \mid G=2)=\frac{1\cdot\frac{1}{3}}{1\cdot\frac{1}{3}+1\cdot\frac{1}{3}}=\frac{1}{2} \text{ şi } P(O=3 \mid G=2)=1-\frac{1}{2}=\frac{1}{2},$$

deci vei alege la întâmplare oricare din cele două uşi rămase (uşa 1 şi uşa 3), ele având aceeaşi probabilitate de salvare.

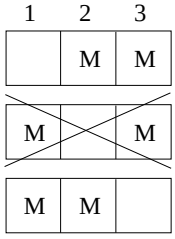
***Varianta c*:**

$$P(O=1 \mid G=2)=0 \text{ şi } P(O=3 \mid G=2)=1-0=1,$$

deci vei alege uşa a treia, care este cu siguranţă ieşirea cea bună.

***Observaţii*:**

1. Echivalent, pentru a determina maximul dintre $P(O=1 \mid G=2)$ **şi** $P(O=3 \mid G=2)$ **ar fi fost suficient, conform formulei lui Bayes, să comparăm** $P(G=2 \mid O=1)\cdot P(O=1)$ **şi** $P(G=2 \mid O=3)\cdot P(O=3)$**. Mai mult, ţinând cont de relaţia** (229)**, aceasta revine la a compara** $P(G=2 \mid O=1)$ **şi** $P(G=2 \mid O=3)$**. Răspunsul poate fi citit imediat din tabelul de mai sus (vedeţi prima linie şi penultima linie): în cazul variantei** a**,** $O=3$ **este varianta pentru care se obţine probabilitatea [a posteriori] maximă. Altfel spus, pentru varianta** a**,** $O=3$ **este *ipoteza de probabilitate maximă a posteriori* (engl., maximum a posteriori probability (MAP) hypothesis). Absolut similar se poate proceda şi pentru variantele** b **şi** c**.**[423]



2. Alternativ, pentru variantele b **şi** c **putem răspunde la întrebare şi făcând un raţionament care nu foloseşte formula lui Bayes. Ieşirea cea bună se poate găsi în spatele uneia dintre cele trei uşi (vedeţi figura alăturată). Cum fantoma a deschis deja uşa cu numărul 2, una dintre aceste situaţii (şi anume, a doua din figură) este eliminată, fiindcă în spatele ei este un monstru. În continuare, putem raţiona în felul următor:**

***Varianta b*: Întrucât fantoma alege uşa 2 cu probabilitate** 1**, vom putea afirma că ambele variante – 1 şi 3 – au probabilităţi egale, şi anume** $\frac{1}{2}$**. Într-adevăr,**

[423] *Observaţie*: În cazuri precum cel de mai sus ($P(O=1)=P(O=2)=P(O=3)=1/3$), ipoteza MAP coincide cu *ipoteza de verosimilitate maximă* (engl., maximum likelihood (ML) hypothesis).

- **fie uşa 1, cea aleasă de mine, dă înspre afară, iar atunci fantoma trebuie, conform principiului P2, să aleagă uşa 2;**
- **fie uşa 3 dă înspre afară, iar atunci, din nou conform principiului P2, fantoma trebuie să aleagă uşa 2;**
- **conform principiului P1, nu există o a treia posibilitate;**
- **nu dispun de alte informaţii pentru a decide între cele două situaţii de mai sus.**

***Varianta c*: Ştiind că fantoma nu a deschis uşa 3 (care ar fi opţiunea corespunzătoare principiului P2), ci a ales uşa 2, înseamnă că nu a putut face altfel, deci uşa 3 reprezintă ieşirea.**

3. **(Formula lui Bayes; inferenţe statistice; exemplificarea noţiunii de ipoteză / ipoteze MAP ("Maximum A posteriori Probability"))**

■ • ○ ***CMU, 2012 spring, Ziv Bar-Joseph, HW1, pr. 1.5***

Mickey dă cu zarul de mai multe ori, sperând să obţină un 6. Secvenţa celor 10 rezultate obţinute de el în urma acestor aruncări este următoarea: $1, 3, 4, 2, 3, 3, 2, 5, 1, 6$**. Mickey se întreabă dacă nu cumva zarul este măsluit (având tendinţa să producă de mai multe ori faţa 3 decât ar fi normal dacă zarul ar fi perfect).**

Concepeţi o analiză simplă bazată pe teorema lui Bayes care să-i furnizeze lui Mickey informaţia care-l interesează: [în ce măsură putem spune că] zarul este măsluit [sau nu]?

Explicaţi raţionamentul dumneavoastră.

Veţi presupune că în general fiecare set de 100 de zaruri conţine 5 zaruri măsluite (engl., unfair) în aşa fel încât este favorizată apariţia feţei 3, rezultând următoarea distribuţie de probabilitate a celor şase feţe, $(1, 2, 3, 4, 5, 6)$**:** $P = [0.1,\ 0.1,\ 0.5,\ 0.1,\ 0.1,\ 0.1]$**.**

Răspuns:

Acesta este un exerciţiu [tipic] de punere în evidenţă a noţiunii de ipoteză de *probabilitate maximă a posteriori* (engl., Maximum A posteriori Probability, MAP).

Vă reamintim definiţia:

$$h_{MAP} = \underset{h \in H}{\operatorname{argmax}}\, P(h|D),$$

unde D **este setul de date cu care se lucrează, iar** H **este mulţimea de ipoteze considerate.**

În cazul nostru, $D = \{1, 3, 4, 2, 3, 3, 2, 5, 1, 6\}$**, iar** $H = \{FD, LD\}$**, unde am notat cu** FD **zarul corect / cinstit (engl., fair die) şi cu** LD **zarul măsluit (engl., loaded die).**

Folosind formula lui Bayes, definiţia de mai sus, poate fi „rafinată" astfel:

$$h_{MAP} \overset{def.}{=} \underset{h \in H}{\operatorname{argmax}}\, P(h|D) \overset{F.B.}{=} \underset{h \in H}{\operatorname{argmax}}\, \frac{P(D|h) \cdot P(h)}{P(D)} = \underset{h \in H}{\operatorname{argmax}}\, P(D|h) \cdot P(h),$$

ultima egalitate având loc datorită faptului că $P(D)$ este o cantitate pozitivă care nu depinde de h.[424]

Aşadar, în cazul de faţă, a determina ipoteza de probabilitate maximă a posteriori (h_{MAP}) revine la a determina maximul dintre două produse: $P(D|FD) \cdot P(FD)$ şi $P(D|LD) \cdot P(LD)$.

Facem observaţia că pentru a calcula $P(D|FD)$ şi $P(D|LD)$, vom ţine cont de faptul că aruncările zarului au fost independente unele de altele. Prin urmare, notând $D = \{x_1, x_2, \ldots, x_{10}\}$, vom putea scrie:

$$\begin{aligned} P(D|FD) \cdot P(FD) &= P(x_1, x_2, \ldots, x_{10}|FD) \cdot P(FD) \overset{i.i.d.}{=} \Big(\prod_{i=1}^{10} P(x_i|FD)\Big) \cdot P(FD) \\ &= \left(\frac{1}{6}\right)^{10} \cdot \frac{95}{100} = \frac{1}{2^{10} \cdot 3^{10}} \cdot \frac{19}{20}. \end{aligned}$$

Similar,

$$\begin{aligned} P(D|LD) \cdot P(LD) &= P(x_1, x_2, \ldots, x_{10}|LD) \cdot P(LD) \overset{i.i.d.}{=} \Big(\prod_{i=1}^{10} P(x_i|LD)\Big) \cdot P(LD) \\ &= \left(\frac{1}{10} \cdot \frac{1}{2} \cdot \frac{1}{10} \cdot \frac{1}{10} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{10} \cdot \frac{1}{10} \cdot \frac{1}{10} \cdot \frac{1}{10} \cdot\right) \cdot \frac{5}{100} \\ &= \frac{1}{10^7 \cdot 2^3} \cdot \frac{1}{20} = \frac{1}{2^{10} \cdot 5^7} \cdot \frac{1}{20}. \end{aligned}$$

Aşadar, a vedea care dintre produsele $P(D|FD) \cdot P(FD)$ şi $P(D|LD) \cdot P(LD)$ este mai mare revine la a compara fracţiile $\frac{19}{3^{10}}$ şi $\frac{1}{5^7}$. În loc să facem ridicările la putere (3^{10} şi 5^7), este mai convenabil să logaritmăm, folosind ca bază un număr supra-unitar:

$$\ln \frac{19}{3^{10}} = \ln 19 - \ln 3^{10} = \ln 19 - 10 \ln 3 = 2.9444 - 10.9861 = -8.0417$$

şi

$$\ln \frac{1}{5^7} = -\ln 5^7 = -7 \ln 5 = -11.2661.$$

Conchidem că ipoteza de probabilitate maximă a posteriori este FD, deci că zarul lui Mickey *nu* este măsluit.

***Observaţie*: Se obişnuieşte ca, în loc să se lucreze cu cele două produse ($P(D|FD) \cdot P(FD)$ şi $P(D|LD) \cdot P(LD)$) în mod separat, aşa cum am procedat noi mai sus, să se facă raportul lor,**

$$\frac{P(D|LD) \cdot P(LD)}{P(D|FD) \cdot P(FD)} = \frac{P(LD|D)}{P(FD|D)}.$$

Acest raport se numeşte *raportul de şanse* (engl., odds ratio). (Dacă acest raport este supra-unitar, înseamnă că ipoteza LD este mai plauzibilă decât ipoteza FD.) Mai departe, aplicând logaritmul — fiindcă la calcule probabilitatea $P(D|\ldots)$ se exprimă ca produs de n factori —, se obţine ceea ce în limba

[424] Folosind formula probabilităţii totale, atunci când H este o mulţime discretă, putem exprima $P(D)$ ca $\sum_{h' \in H} P(D|h') \cdot P(h')$. Ar fi util să calculaţi această probabilitate *a priori* în cazul datelor din acest exerciţiu.

engleză se numeşte *log-odds ratio*. (Evident, dacă *log-odds ratio* are valoare pozitivă, ipoteza LD este mai plauzibilă.) Pe datele noastre,

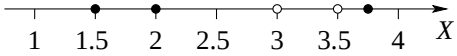
$$\ln \frac{P(LD|D)}{P(FD|D)} = -11.2661 - (-8.0417) = -3.2244 < 0.$$

În concluzie, folosind un astfel de raport (de „potrivire"), vom putea spune nu doar care ipoteză este mai plauzibilă, ci şi *în ce măsură* acea ipoteză (în cazul nostru, FD) este mai plauzibilă decât cealaltă ipoteză (LD).

4. (Algoritmul ID3: calculul erorii la antrenare şi la CVLOO. Arbori cu decizii probabiliste, ca ipoteze ML şi respectiv MAP)

■ ***CMU, 2009 spring, T. Mitchell, midterm, pr. 2***

Se consideră următorul set de date de antrenament din spaţiul real unidimensional:



Este vorba de 5 date caracterizate de atributul real X, împărţite în două clase: clasa 0 constituită din mulţimea $\{3, 3.5\}$, şi clasa 1 – mulţimea $\{1.5, 2, 3.75\}$.

A. Pe acest set de date se va aplica algoritmul ID3 pentru construirea unor arbori de decizie. Deoarece atributul are valori reale, testele vor fi de forma $X > t$, unde t reprezintă o valoare-prag.

Se notează cu DT^* algoritmul care construieşte arborele de decizie cu număr minim de noduri necesare pentru clasificarea perfectă a datelor de antrenament, şi cu $DT2$ algoritmul care construieşte un arbore de decizie cu un singur nod de test.

***Indicaţie*: Veţi presupune că atunci când algoritmul de învăţare de arbori de decizie găseşte două praguri astfel încât testele $x < t_1$ şi $x < t_2$ produc acelaşi câştig de informaţie, el (adică, algoritmul) va alege pragul cel mai din stânga. (Aşadar, dacă $t_1 < t_2$, atunci el va alege testul $(x < t_1)$).**

a. Care este eroarea la antrenare a lui $DT2$ pe datele specificate? Dar eroarea la cross-validare folosind metoda "Leave-One-Out" (CVLOO)?

b. Care este eroarea la antrenare a lui DT^* pe datele specificate? Dar eroarea la cross-validare folosind metoda "Leave-One-Out"?

B. În continuare se consideră o nouă clasă de arbori de decizie, care au *etichete probabiliste*. Fiecare nod frunză specifică probabilitatea fiecărei etichete posibile, probabilitate scrisă sub forma raportului dintre datele cu acea etichetă din nodul respectiv şi toate datele din acel nod.

De *exemplu*, un arbore de decizie neavând niciun nod de test, construit pe datele specificate mai sus, clasifică astfel: $P(Y = 1) = 3/5$ şi $P(Y = 0) = 2/5$. Un arbore de decizie cu un singur nod de test (în raport cu valoarea / pragul 2.5) conţine probabilităţile: $P(Y = 1) = 1$ dacă $X \leq 2.5$, şi $P(Y = 1) = 1/3$ dacă $X > 2.5$.

c. Pentru setul de date de mai sus, determinaţi arborele de decizie de tip ML (engl., maximum likelihood), adică acel arbore cu decizii probabiliste care maximizează *verosimilitatea* datelor de antrenament:

$$T_{ML} = \arg\max_T P_T(D), \text{ unde}$$

$$P_T(D) \stackrel{def.}{=} P(D|T) \stackrel{indep.}{=} \prod_{i=1}^{5} P(Y = y_i | X = x_i, T),$$

cu y_i eticheta / clasa instanţei $x_i \in \{1.5, 2, 3, 3.5, 3.75\}$.

d. Se consideră o distribuţie a priori $P(T)$ care penalizează numărul de teste / split-uri din arborele de decizie T, şi anume:

$$P(T) \propto \left(\frac{1}{4}\right)^{splits(T)^2}$$

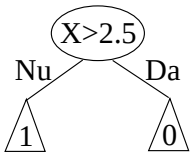
unde $splits(T)$ reprezintă numărul nodurilor de test din arborele T, iar simbolul $\propto$ înseamnă „este proporţional cu".

Pentru acelaşi set de date, folosind această distribuţie a priori $P(T)$, găsiţi arborele de decizie de tip MAP (engl., Maximum A posteriori Probability):

$$T_{MAP} = \arg\max_T P_T(T|D)$$

Răspuns:

a. Dacă se aplică algoritmul $DT2$, întrucât câştigurile de informaţie calculate pentru testele $X > 2.5$ şi $X > 3.625$ sunt 0.419 şi respectiv 0.171, rezultatul este:





Eroarea la antrenare este 1/5, deoarece punctul 3.75 este clasificat greşit.

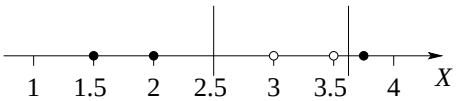
Eroarea la cross-validare cu metoda "Leave-One-Out" se determină astfel:

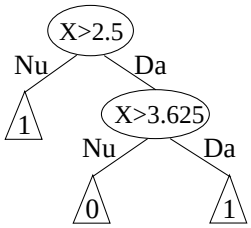
- **$x_1 = 1.5$. Testul se face relativ la pragul 2.5 — ţinând cont de *Indicaţia* din enunţ —, deci punctul 1.5 este clasificat corect.**
- **$x_2 = 2$. Testul se face relativ la pragul 2.25 — ţinând din nou cont de *Indicaţia* din enunţ —, deci punctul $x_2 = 2$ este clasificat corect.**
- **$x_3 = 3$. Testul se face relativ la pragul 2.75, fiindcă se verifică imediat că IG-ul acestui prag este 0.311, deci mai mare decât IG-ul pragului 3.625, şi anume 0.122. Prin urmare, punctul $x_3 = 3$ va fi clasificat fie corect (în cazul în care se consideră că decizia arborelui $DT2$ este 0 pentru $X > 2.75$), fie eronat (în cazul în care se consideră că decizia arborelui $DT2$ este 1 pentru $X > 2.75$). Însă în al doilea caz arborele $DT2$ s-ar reduce la un singur nod (care este nod frunză), ceea ce este contrar definiţiei sale, aşa că vom reţine doar primul caz.**

- $x_4 = 3.5$. **Testul se face din nou relativ la pragul** 2.5 **— justificarea este similară cu cea de la punctul precedent —, iar punctul** $x_4 = 3.5$ **este clasificat fie corect (în cazul în care se consideră că decizia arborelui** $DT2$ **este** 0 **pentru** $X > 2.5$**), fie eronat (în cazul în care se consideră că decizia arborelui** $DT2$ **este** 1 **pentru** $X > 2.5$**). Însă (din nou!) în al doilea caz arborele** $DT2$ **s-ar reduce la un singur nod (frunză), aşa că vom reţine doar primul caz.**
- $x_5 = 3.75$. **Testul se face tot relativ la pragul** 2.5**, iar punctul** $x_5 = 3.75$ **este clasificat greşit.**

Deci eroarea la cross-validare cu metoda "Leave-One-Out" este de $1/5$**, punctul** 3.75 **fiind clasificat eronat.**

b. Dacă se aplică algoritmul DT^***, rezultatul este:**





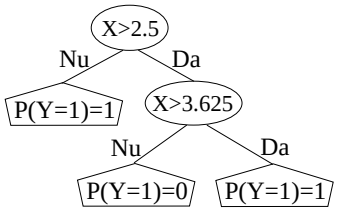
Eroarea la antrenare este bineînţeles 0, deoarece datele de antrenament nu conţin inconsistenţe.

La cross-validare cu metoda "Leave-One-Out", punctele care ar putea genera probleme sunt:

- $x_2 = 2$. **Cele două teste se fac la pragurile** 2.25 **şi la** 3.625 **(ordinea nu contează), deci punctul** $x_2 = 2$ **este corect clasificat.**
- $x_3 = 3$. **Primul test se face la pragul** 2.75**, deci punctul** $x_3 = 3$ **este corect clasificat.**
- $x_4 = 3.5$. **Al doilea test se face la pragul** 3.375**, deci punctul** $x_4 = 3.5$ **este clasificat greşit.**
- $x_5 = 3.75$. **Se face un singur test (la pragul** 2.5**), iar punctul** $x_5 = 3.75$ **este clasificat greşit.**

Deci eroarea la cross-validare pentru arborele DT^* **folosind metoda "Leave-One-Out" este** $2/5$**.**

c. Un arbore de decizie care maximizează probabilităţile datelor de antrenament va fi unul care clasifică în mod perfect aceste date. Deci un arbore de decizie ML poate fi obţinut din arborele construit de DT^***, extins cu etichete probabiliste în frunze, conform figurii alăturate.**



d. Pentru a determina arborele de decizie MAP folosind distribuţia a priori $P(T)$**, va trebui să comparăm probabilităţile a posteriori ale celor 3 arbori de decizie posibili pe setul de date de antrenament. Vom scrie aceste probabilităţi a posteriori folosind formula lui Bayes:**

$$P(T_j \mid D) = \frac{P(D \mid T_j) \cdot P(T_j)}{P(D)} = \frac{\prod_{i=1}^{5} P(Y = y_i \mid T_j, X = x_i) \cdot P(T_j)}{P(D)}$$

Evident, la compararea propriu-zisă a probabilităţilor $P(T_j \mid D)$ **cu** $j = 0, 1, 2$**, nu vom avea nevoie de numitorul** $P(D)$**. Aşadar,**

- **pentru 0 noduri de test:**
  După cum s-a precizat şi în enunţ, acest arbore va avea $P(Y = 1) = 3/5$ şi $P(Y = 0) = 2/5$. Deci

$$P(T_0 \mid D) \propto \left(\frac{3}{5}\right)^3 \cdot \left(\frac{2}{5}\right)^2 \cdot \left(\frac{1}{4}\right)^0 = \frac{3^3 \cdot 2^2}{5^5} = \frac{108}{3125} = 0.0336.$$

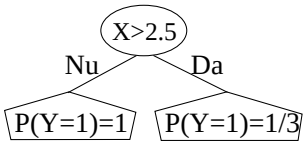
- **pentru un nod de test:**
  Acest arbore va avea probabilităţile: $P(Y = 1) = 1$ dacă $X < 2.5$, şi $P(Y = 1) = 1/3$ dacă $X \geq 2.5$. Prin urmare,

$$P(T_1 \mid D) \propto 1^2 \cdot \left(\frac{2}{3}\right)^2 \cdot \frac{1}{3} \cdot \left(\frac{1}{4}\right)^1 = \frac{1}{27} = 0.037$$

- **pentru două noduri de test:**
  Este vorba de arborele construit la punctul c. Acesta clasifică perfect datele, dar

$$P(T_2) \propto \left(\frac{1}{4}\right)^4 \Rightarrow P(T_2 \mid D) \propto 1 \cdot \left(\frac{1}{4}\right)^4 = \frac{1}{256} = 0.0039$$

Probabilitatea a posteriori maximă o are arborele T_1, deci acesta va fi arborele MAP. Reprezentarea grafică a acestui arbore este:



2.1.2 Algoritmii Bayes Naiv şi Bayes Optimal

5. **(Algoritmul Bayes Naiv: aplicare; comparaţie cu estimarea MLE)**

prelucrare de Liviu Ciortuz, după CMU, 2004 fall, T. Mitchell Z. Bar-Joseph, final exam, pr. 2

Fie setul de date alăturat, cu trei variabile booleene de intrare a, b, c şi o variabilă booleană de ieşire K.

a	b	c	K
1	0	1	1
1	1	1	1
0	1	1	0
1	1	0	0
1	0	1	0
0	0	0	1
0	0	0	1
0	0	1	0

a. Estimaţi probabilităţile $P(K = 1 \mid a = 1, b = 1)$ şi $P(K = 1 \mid a = 1, b = 1, c = 0)$ în sensul verosimilităţii maxime (engl., Maximum Likelihood Estimation, MLE).

b. Cum clasifică algoritmul Bayes Naiv instanţa $(a = 1, b = 1)$?
Dar instanţa $(a = 1, b = 1, c = 0)$?

Răspuns:

a. $P(K = 1 \mid a = 1, b = 1) = \frac{1}{2}$**, fiindcă avem două instanţe în care ambele atribute** ***a, b*** **sunt adevărate, dintre care una este clasificată** $K = 1$**, iar cealaltă** $K = 0$**.**

$P(K = 1 \mid a = 1, b = 1, c = 0) = 0$**, fiindcă există o singură instanţă cu** $a = 1, b = 1, c = 0$ **în setul de antrenament şi ea este clasificată** $K = 0$**.**

b. Cazul $(a = 1, b = 1)$**:**

$$\begin{aligned}
\hat{k}_{MAP} &= \operatorname*{argmax}_{k\in\{0,1\}} P(K = k|a = 1, b = 1) = \\
&= \operatorname*{argmax}_{k\in\{0,1\}} \frac{P(a = 1, b = 1|K = k) \cdot P(K = k)}{P(a = 1, b = 1)} = \\
&= \operatorname*{argmax}_{k\in\{0,1\}} P(a = 1, b = 1|K = k) \cdot P(K = k) = \\
&= \operatorname*{argmax}_{k\in\{0,1\}} P(a = 1|K = k) \cdot P(b = 1|K = k) \cdot P(K = k)
\end{aligned}$$

Avem:

$$\begin{aligned}
p_0 &= P(a = 1|K = 0) \cdot P(b = 1|K = 0) \cdot P(K = 0) = \frac{2}{4} \cdot \frac{2}{4} \cdot \frac{4}{8} = \frac{1}{8} \\
p_1 &= P(a = 1|K = 1) \cdot P(b = 1|K = 1) \cdot P(K = 1) = \frac{2}{4} \cdot \frac{1}{4} \cdot \frac{4}{8} = \frac{1}{16}
\end{aligned}$$

Prin urmare, $p_0 > p_1$**. Aşadar, clasificatorul Bayes Naiv va prezice** $K = 0$ **pentru instanţa** $(a = 1, b = 1)$ **cu probabilitatea**

$$\begin{aligned}
&P(K = 0 \mid a = 1, b = 1) \\
&= \frac{P(a = 1, b = 1 \mid K = 0) \cdot P(K = 0)}{P(a = 1, b = 1 \mid K = 0) \cdot P(K = 0) + P(a = 1, b = 1 \mid K = 1) \cdot P(K = 1)} \\
&= \frac{p_0}{p_0 + p_1} = \frac{\frac{1}{8}}{\frac{1}{8} + \frac{1}{16}} = \frac{2}{3}
\end{aligned}$$

Observaţie**: Se constată că probabilitatea** $P(K = 1|a = 1, b = 1)$ **estimată în sensul MLE (adică,** $1/2$**; vedeţi punctul** a**) diferă de probabilitatea (*a posteriori*) cu care algoritmul Bayes Naiv clasifică instanţa** $(a = 1, b = 1)$ **ca aparţinând clasei** $K = 1$ **(adică,** $1/3$**). Explicaţia rezidă — întrucât** $P(K = 0) = P(K = 1) = 1/2$ **— în faptul că presupoziţia de independenţă condiţională asumată de către algoritmul Bayes Naiv nu este validă. Într-adevăr,** $P(a = 0|K = 1) = 2/4$ **şi** $P(a = 0|b = 1, K = 1) = 0$**, deci** $P(a = 0|K = 1) \neq P(a = 0|b = 1, K = 1)$**.**

Cazul $(a = 1, b = 1, c = 0)$**:**

$$\begin{aligned}
\hat{k}_{MAP} &= \operatorname*{argmax}_{k\in\{0,1\}} P(K = k|a = 1, b = 1, c = 0) = \\
&= \operatorname*{argmax}_{k\in\{0,1\}} \frac{P(a = 1, b = 1, c = 0|K = k) \cdot P(K = k)}{P(a = 1, b = 1, c = 0)} = \\
&= \operatorname*{argmax}_{k\in\{0,1\}} P(a = 1, b = 1, c = 0|K = k) \cdot P(K = k) = \\
&= \operatorname*{argmax}_{k\in\{0,1\}} P(a = 1|K = k) \cdot P(b = 1|K = k) \cdot P(c = 0|K = k) \cdot P(K = k)
\end{aligned}$$

Avem:

$$\begin{aligned} p_0 &= P(a=1|K=0)\cdot P(b=1|K=0)\cdot P(c=0|K=0)\cdot P(K=0) = \\ &= \frac{2}{4}\cdot\frac{2}{4}\cdot\frac{1}{4}\cdot\frac{4}{8} = \frac{1}{32} \\ p_1 &= P(a=1|K=1)\cdot P(b=1|K=1)\cdot P(c=0|K=1)\cdot P(K=1) = \\ &= \frac{2}{4}\cdot\frac{1}{4}\cdot\frac{2}{4}\cdot\frac{4}{8} = \frac{1}{32} \end{aligned}$$

Întrucât $p_0 = p_1$, clasificatorul Bayes Naiv va prezice $K = 0$ sau $K = 1$ cu aceeaşi probabilitate $(1/2)$.

***Notă*: *Observaţia* de mai sus este valabilă şi pentru cazul $(a = 1, b = 1, c = 0)$.**

6. **(Calculul parametrilor pentru clasificatorul Bayes Naiv pornind de la distribuţia comună a variabilelor; comparaţie între algoritmii Bayes Naiv şi Bayes Optimal)**

prelucrare de L. Ciortuz, după

■ • ***CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, midterm, pr. 2.1***

Fie P o *distribuţie de probabilitate comună* peste variabilele aleatoare booleene x_1, x_2 şi y.

În cele ce urmează vom considera distribuţia P, definită conform tabelului alăturat.

x_1	x_2	y	$P(x_1, x_2, y)$
0	0	0	0.15
0	0	1	0.25
0	1	0	0.05
0	1	1	0.08
1	0	0	0.10
1	0	1	0.02
1	1	0	0.20
1	1	1	0.15

a. Exprimaţi $P(y = 0 \mid x_1, x_2)$ în funcţie de $P(x_1, x_2, y = 0)$ şi $P(x_1, x_2, y = 1)$.

b. Pornind de la această distribuţie comună, calculaţi probabilităţile necesare pentru clasificare bayesiană naivă. Completaţi tabelele de mai jos:

y	$P(y)$
$y=0$	
$y=1$	

$P(x_1 \mid y)$	$x_1=0$	$x_1=1$
$y=0$		
$y=1$		

$P(x_2 \mid y)$	$x_2=0$	$x_2=1$
$y=0$		
$y=1$		

c. Cât este probabilitatea $P(y = 1 \mid x_1 = 1, x_2 = 0)$ calculată de către clasificatorul Bayes Naiv?

d. Cât este probabilitatea $P(y = 1 \mid x_1 = 1, x_2 = 0)$ calculată de către clasificatorul Bayes Optimal? (Vă readucem aminte că acest algoritm este similar cu algoritmul Bayes Naiv, însă nu foloseşte presupoziţia de independenţă condiţională a atributelor.)

***Comentariu*:** Pentru cel de-al doilea algoritm, denumirea *Bayes Comun* se datorează faptului că acest clasificator lucrează cu distribuţia comună a variabilelor / atributelor de intrare, în vreme ce denumirea *Bayes Optimal* este justificată de faptul că nicio altă

distribuţie probabilistă asupra datelor de intrare nu poate conduce la o eroare medie la clasificare mai mică decât cea obţinută de către acest clasificator. (În cele ce urmează, vom opta pentru cea de-a doua denumire.)

e. Răspunsurile la precedentele două întrebări ar trebui să fie diferite. Care este motivul? Justificaţi.

Răspuns:

a. Aplicând definiţia probabilităţii condiţionate şi apoi proprietatea de aditivitate numărabilă din definiţia funcţiei de probabilitate, obţinem:

$$P(y=0 \mid x_1, x_2) = \frac{P(x_1, x_2, y=0)}{P(x_1, x_2)} = \frac{P(x_1, x_2, y=0)}{P(x_1, x_2, y=0) + P(x_1, x_2, y=1)}$$

b. Probabilităţile cerute în enunţ sunt $P(x_1 \mid y)$, $P(x_2 \mid y)$ şi $P(y)$.
$P(y)$ este o *probabilitate marginală* a distribuţiei comune, deci se calculează astfel:

$$\begin{aligned} P(y=0) &= P(x_1=0, x_2=0, y=0) + P(x_1=0, x_2=1, y=0) + \\ &\quad + P(x_1=1, x_2=0, y=0) + P(x_1=1, x_2=1, y=0) \\ &= 0.15 + 0.05 + 0.1 + 0.2 = 0.5 \\ P(y=1) &= 1 - P(y=0) = 0.5 \end{aligned}$$

$P(x_1 \mid y)$ se calculează folosind din nou definiţia probabilităţii condiţionate şi formula probabilităţii totale:

$$P(x_1=0 \mid y=0) = \frac{P(x_1=0, y=0)}{P(y=0)} = \frac{P(x_1=0, y=0)}{P(x_1=0, y=0) + P(x_1=1, y=0)}$$

Probabilităţile implicate în formulă sunt probabilităţi marginale ale distribuţiei comune şi se calculează astfel:

$$P(x_1=0, y=0) = P(x_1=0, x_2=0, y=0) + P(x_1=0, x_2=1, y=0) = 0.15 + 0.05 = 0.2$$
$$P(x_1=1, y=0) = P(x_1=1, x_2=0, y=0) + P(x_1=1, x_2=1, y=0) = 0.1 + 0.2 = 0.3$$

Prin urmare,

$$P(x_1=0 \mid y=0) = \frac{0.2}{0.2+0.3} = 0.4,$$

iar

$$P(x_1=1 \mid y=0) = 1 - P(x_1=0 \mid y=0) = 0.6$$

Analog se calculează şi celelalte probabilităţi corespunzătoare lui $P(x_1 \mid y)$:

$$P(x_1=0 \mid y=1) = \frac{P(x_1=0, y=1)}{P(y=1)} = \frac{0.25+0.08}{0.5} = 0.66$$
$$P(x_1=1 \mid y=1) = 1 - P(x_1=0 \mid y=1) = 0.34$$

$P(x_2 \mid y)$ se calculează în acelaşi mod.

Putem completa tabelele următoare cu valorile numerice ale probabilităţilor calculate la acest punct:

y	$P(y)$
$y=0$	0.5
$y=1$	0.5

$P(x_1 \mid y)$	$x_1=0$	$x_1=1$
$y=0$	0.40	0.60
$y=1$	0.66	0.34

$P(x_2 \mid y)$	$x_2=0$	$x_2=1$
$y=0$	0.50	0.50
$y=1$	0.54	0.46

c. Clasificatorul Bayes Naiv face presupunerea de *independenţă condiţională* a variabilelor, deci:

$$\begin{aligned}
P(y=1 \mid x_1=1, x_2=0) &\overset{Bayes}{=} \frac{P(x_1=1, x_2=0 \mid y=1) \cdot P(y=1)}{P(x_1=1, x_2=0)} \\
&= \frac{P(x_1=1, x_2=0 \mid y=1) \cdot P(y=1)}{P(x_1=1, x_2=0 \mid y=1)P(y=1) + P(x_1=1, x_2=0 \mid y=0)P(y=0)} \overset{indep.\ cdt.}{=} \\
&= \frac{P(x_1=1 \mid y=1) \cdot P(x_2=0 \mid y=1) \cdot P(y=1)}{P(x_1=1|y=1)P(x_2=0|y=1)P(y=1) + P(x_1=1|y=0)P(x_2=0|y=0)P(y=0)} \\
&= \frac{0.34 \cdot 0.54 \cdot 0.5}{0.34 \cdot 0.54 \cdot 0.5 + 0.6 \cdot 0.5 \cdot 0.5} \approx 0.3796
\end{aligned}$$

d. Clasificatorul Bayes Optimal nu face niciun fel de presupunere, deci folosind o formulă similară cu cea obţinută la punctul a, vom avea:

$$\begin{aligned}
P(y=1 \mid x_1=1, x_2=0) &= \frac{P(x_1=1, x_2=0, y=1)}{P(x_1=1, x_2=0, y=1) + P(x_1=1, x_2=0, y=0)} \\
&= \frac{0.02}{0.02+0.1} = 0.1(6).
\end{aligned}$$

e. Valorile calculate de clasificatorul Bayes Naiv şi de clasificatorul Bayes Optimal pentru $P(y=1 \mid x_1=1, x_2=0)$ sunt diferite deoarece presupunerea de independenţă condiţională făcută de clasificatorul Bayes Naiv nu este adevărată. Într-adevăr, se observă că variabilele x_1 şi x_2 nu sunt independente condiţional în raport cu variabila y:

$$P(x_1=1, x_2=0 \mid y=1) = \frac{P(x_1=1, x_2=0, y=1)}{P(y=1)} = \frac{0.02}{0.5} = 0.04$$
$$P(x_1=1 \mid y=1) \cdot P(x_2=0 \mid y=1) = 0.34 \cdot 0.54 = 0.1836 \neq 0.04$$

7. (Algoritmul Bayes Naiv şi algoritmul Bayes Optimal; comparaţie relativ la numărului de parametri)

■ • ***CMU, 2008 fall, Eric Xing, HW1, pr. 2***

Fie următoarea problemă de clasificare: X_1 şi X_2 sunt variabile aleatoare observabile, Y este eticheta clasei asignate fiecărei instanţe observate, conform tabelului alăturat.
În acest exerciţiu veţi compara rezultatele care se obţin în urma antrenării pe acest set de date de către doi algoritmi de clasificare: Bayes Naiv şi Bayes Comun (engl., Joint Bayes), care este numit adeseori şi *Bayes Optimal* (engl., Optimal Bayes).

X_1	X_2	Y	Nr. apariţii
0	0	0	2
0	0	1	18
1	0	0	4
1	0	1	1
0	1	0	4
0	1	1	1
1	1	0	2
1	1	1	18

a. Clasificaţi instanţa $X_1=0, X_2=0$ folosind clasificatorul Bayes Naiv.

b. Clasificaţi instanţa $X_1=0, X_2=0$ folosind clasificatorul Bayes Optimal.

c. Notăm cu P_{NB} şi respectiv P_{JB} valoarea probabilităţii $P(Y=1 \mid X_1=0, X_2=0)$ calculate pentru clasificatorul Bayes Naiv, respectiv pentru clasificatorul

Bayes Optimal. De ce diferă cele două valori? *Sugestie*: Calculaţi $P(X_1, X_2 \mid Y)$.

d. Care ar fi situaţia pentru P_{NB} şi P_{JB} de la întrebarea precedentă în situaţia în care datele observate ar proveni din tabelul alăturat?

X_1	X_2	Y	**Nr. apariţii**
0	0	0	3
0	0	1	9
1	0	0	3
1	0	1	9
0	1	0	3
0	1	1	9
1	1	0	3
1	1	1	9

e. De câţi parametri independenţi (i.e., probabilităţi estimate) este nevoie în total pentru a construi clasificatorul Bayes Naiv? Dar în cazul clasificatorului Bayes Optimal?

Răspundeţi la aceste întrebări şi în cazul general, când se folosesc n variabile binare observate. Comentaţi rezultatul.

Răspuns:

a. Clasificatorul Bayes Naiv face predicţia pentru valoarea lui Y după formula de mai jos:

$$\hat{y}_{NB} = \underset{y \in \{0,1\}}{\operatorname{argmax}} P(X_1 = 0|Y = y) \cdot P(X_2 = 0|Y = y) \cdot P(Y = y)$$

Avem:

$$\begin{aligned} p_0 &\overset{not.}{=} P(X_1 = 0|Y = 0) \cdot P(X_2 = 0|Y = 0) \cdot P(Y = 0) \\ &\overset{MLE}{=} \frac{6}{12} \cdot \frac{6}{12} \cdot \frac{12}{50} = \frac{3}{50} = \frac{6}{100} \end{aligned}$$

$$\begin{aligned} p_1 &\overset{not.}{=} P(X_1 = 0|Y = 1) \cdot P(X_2 = 0|Y = 1) \cdot P(Y = 1) \\ &\overset{MLE}{=} \frac{19}{38} \cdot \frac{19}{38} \cdot \frac{38}{50} = \frac{19}{100} \end{aligned}$$

Întrucât $p_0 < p_1$, clasificatorul Bayes Naiv va prezice $Y = 1$ pentru instanţa $(X_1 = 0, X_2 = 0)$.

b. Deoarece clasificatorul Bayes Optimal nu lucrează cu presupunerea de independenţă condiţională a atributelor de intrare în raport cu atributul de ieşire, el va face predicţia folosind formula de mai jos:

$$\hat{y}_{JB} = \underset{y \in \{0,1\}}{\operatorname{argmax}} P(X_1 = 0, X_2 = 0|Y = y) \cdot P(Y = y)$$

Probabilităţile din formulă sunt, ca şi în cazul clasificatorului Bayes Naiv, cele estimate din distribuţia datelor de antrenament.

Aşadar, avem:

$$p'_0 \overset{not.}{=} P(X_1 = 0, X_2 = 0 \mid Y = 0) \cdot P(Y = 0) \overset{MLE}{=} \frac{2}{12} \cdot \frac{12}{50} = \frac{2}{50}$$

$$p'_1 \overset{not.}{=} P(X_1 = 0, X_2 = 0 \mid Y = 1) \cdot P(Y = 1) \overset{MLE}{=} \frac{18}{38} \cdot \frac{38}{50} = \frac{18}{50}$$

Fiindcă $p'_0 < p'_1$, clasificatorul Bayes Optimal va prezice tot $Y = 1$.

c. Vom calcula cele două valori pentru $P(Y = 1 \mid X_1 = 0, X_2 = 0)$:

$$\begin{aligned}
P_{NB} &\overset{not.}{=} P(Y = 1 \mid X_1 = 0, X_2 = 0) \\
&\overset{F.\ Bayes}{=} \frac{P(X_1 = 0, X_2 = 0 \mid Y = 1) \cdot P(Y = 1)}{P(X_1 = 0, X_2 = 0 \mid Y = 1)P(Y = 1) + P(X_1 = 0, X_2 = 0 \mid Y = 0)P(Y = 0)} \\
&\overset{indep.\ cdt.}{=} \frac{P(X_1 = 0|Y = 1) \cdot P(X_2 = 0|Y = 1) \cdot P(Y = 1)}{\begin{array}{r} P(X_1 = 0|Y = 0) \cdot P(X_2 = 0|Y = 0) \cdot P(Y = 0)+ \\ P(X_1 = 0|Y = 1) \cdot P(X_2 = 0|Y = 1) \cdot P(Y = 1) \end{array}} \\
&= \frac{p_1}{p_0 + p_1} = \frac{\frac{19}{100}}{\frac{6}{100} + \frac{19}{100}} = \frac{19}{25}
\end{aligned}$$

$$\begin{aligned}
P_{JB} &\overset{not.}{=} P(Y = 1 \mid X_1 = 0, X_2 = 0) \\
&\overset{F.\ Bayes}{=} \frac{P(X_1 = 0, X_2 = 0 \mid Y = 1) \cdot P(Y = 1)}{P(X_1 = 0, X_2 = 0 \mid Y = 1)P(Y = 1) + P(X_1 = 0, X_2 = 0 \mid Y = 0)P(Y = 0)} \\
&= \frac{p'_1}{p'_0 + p'_1} = \frac{\frac{18}{50}}{\frac{2}{50} + \frac{18}{50}} = \frac{18}{20}
\end{aligned}$$

Cele două valori diferă deoarece presupunerea de independenţă condiţională a variabilelor X_1 şi X_2 în raport cu Y făcută de clasificatorul Bayes Naiv este falsă. Acest lucru se poate vedea uşor din valorile estimate pentru $P(X_1 = 0, X_2 = 0 \mid Y = 0)$, $P(X_1 = 0 \mid Y = 0)$ şi $P(X_2 = 0 \mid Y = 0)$:

$$\left.\begin{array}{l} P(X_1 = 0, X_2 = 0 \mid Y = 0) \overset{MLE}{=} \frac{2}{12} \\ P(X_1 = 0 \mid Y = 0) \cdot P(X_2 = 0 \mid Y = 0) \overset{MLE}{=} \frac{6}{12} \cdot \frac{6}{12} = \frac{1}{4} \end{array}\right\} \Rightarrow$$

$$\Rightarrow P(X_1 = 0, X_2 = 0 \mid Y = 0) \neq P(X_1 = 0 \mid Y = 0) \cdot P(X_2 = 0 \mid Y = 0) \Rightarrow$$
$$\Rightarrow P(X_1, X_2 \mid Y) \neq P(X_1 \mid Y) \cdot P(X_2 \mid Y)$$

Aşadar, variabilele X_1 şi X_2 nu sunt independente condiţional în raport cu variabila de ieşire Y.

d. Vom calcula cele două valori pentru $P(Y = 1 \mid X_1 = 0, X_2 = 0)$ în cazul noilor date:

$$\begin{aligned}
P_{NB} &= P(Y = 1 \mid X_1 = 0, X_2 = 0) \\
&= \frac{P(X_1 = 0, X_2 = 0 \mid Y = 1) \cdot P(Y = 1)}{P(X_1 = 0, X_2 = 0 \mid Y = 1)P(Y = 1) + P(X_1 = 0, X_2 = 0 \mid Y = 0)P(Y = 0)} \\
&= \frac{P(X_1 = 0|Y = 1) \cdot P(X_2 = 0|Y = 1) \cdot P(Y = 1)}{\begin{array}{r} P(X_1 = 0|Y = 0) \cdot P(X_2 = 0|Y = 0) \cdot P(Y = 0)+ \\ P(X_1 = 0|Y = 1) \cdot P(X_2 = 0|Y = 1) \cdot P(Y = 1) \end{array}}
\end{aligned}$$

$$= \frac{\frac{18}{36}\cdot\frac{18}{36}\cdot\frac{36}{48}}{\frac{6}{12}\cdot\frac{6}{12}\cdot\frac{12}{48}+\frac{18}{36}\cdot\frac{18}{36}\cdot\frac{36}{48}} = \frac{\frac{9}{48}}{\frac{3}{48}+\frac{9}{48}} = \frac{9}{12} = \frac{3}{4}$$

$$\begin{aligned} P_{JB} &= P(Y=1 \mid X_1=0, X_2=0) \\ &= \frac{P(X_1=0, X_2=0 \mid Y=1)\cdot P(Y=1)}{P(X_1=0, X_2=0 \mid Y=1)P(Y=1)+P(X_1=0, X_2=0 \mid Y=0)P(Y=0)} \\ &= \frac{\frac{9}{36}\cdot\frac{36}{48}}{\frac{3}{12}\cdot\frac{12}{48}+\frac{9}{36}\cdot\frac{36}{48}} = \frac{\frac{9}{48}}{\frac{3}{48}+\frac{9}{48}} = \frac{9}{12} = \frac{3}{4} \end{aligned}$$

Aşadar, în acest caz avem $P_{NB} = P_{JB}$.

De fapt, se poate constata uşor că în cazul distribuţiei probabiliste date la acest punct al problemei se verifică independenţa condiţională a variabilelor X_1 şi X_2 în raport cu Y. Prin urmare, predicţiile făcute de cei doi clasificatori, Bayes Naiv şi Bayes Optimal, vor coincide întotdeauna.

e. În contextul problemei noastre, clasificatorul Bayes Naiv are nevoie de estimările următoarelor probabilităţi:

$$\begin{array}{lcl} P(Y=0) & \Rightarrow & P(Y=1) = 1 - P(Y=0) \\ P(X_1=0 \mid Y=0) & \Rightarrow & P(X_1=1 \mid Y=0) = 1 - P(X_1=0 \mid Y=0) \\ P(X_1=0 \mid Y=1) & \Rightarrow & P(X_1=1 \mid Y=1) = 1 - P(X_1=0 \mid Y=1) \\ P(X_2=0 \mid Y=0) & \Rightarrow & P(X_2=1 \mid Y=0) = 1 - P(X_2=0 \mid Y=0) \\ P(X_2=0 \mid Y=1) & \Rightarrow & P(X_2=1 \mid Y=1) = 1 - P(X_2=0 \mid Y=1) \end{array}$$

Avem nevoie, prin urmare, doar de 5 valori pentru a construi complet clasificatorul Bayes Naiv.

În cazul general, dacă avem n variabile de intrare, avem nevoie de estimări pentru probabilităţile $P(Y)$, $P(X_i \mid Y)$ şi $P(X_i \mid \neg Y)$ pentru $i = \overline{1,n}$, deci $2n+1$ valori.

Pentru clasificatorul Bayes Optimal avem nevoie de:

$P(Y=0)$	$P(Y=1) = 1 - P(Y=0)$
$P(X_1=0, X_2=0 \mid Y=0)$ $P(X_1=0, X_2=1 \mid Y=0)$ $P(X_1=1, X_2=0 \mid Y=0)$	$P(X_1=1, X_2=1 \mid Y=0)$ **se poate determina din celelalte 3 valori, aşa cum vom arăta mai jos.**
$P(X_1=0, X_2=0 \mid Y=1)$ $P(X_1=0, X_2=1 \mid Y=1)$ $P(X_1=1, X_2=0 \mid Y=1)$	**Similar,** $P(X_1=1, X_2=1 \mid Y=1)$ **se poate determina din celelalte 3 valori.**

Notăm evenimentul $X_1 = 0$ cu A, $X_2 = 0$ cu B şi $Y = 0$ cu C. Ştim că:

$$\begin{aligned} &\Omega = (A \wedge B) \vee (\neg A \wedge B) \vee (A \wedge \neg B) \vee (\neg A \wedge \neg B) \Rightarrow \\ &\Omega \wedge C = ((A \wedge B) \vee (\neg A \wedge B) \vee (A \wedge \neg B) \vee (\neg A \wedge \neg B)) \wedge C \Rightarrow \\ &C = ((A \wedge B) \wedge C) \vee ((\neg A \wedge B) \wedge C) \vee ((A \wedge \neg B) \wedge C) \vee ((\neg A \wedge \neg B) \wedge C) \end{aligned}$$

De asemenea, deoarece toate evenimentele din partea dreaptă a egalităţii sunt disjuncte două câte două, putem scrie egalitatea de mai sus şi cu probabilităţi:

$$\begin{aligned} &P(C) = P((A \wedge B) \wedge C) + P((\neg A \wedge B) \wedge C) + P((A \wedge \neg B) \wedge C) + P((\neg A \wedge \neg B) \wedge C) \\ &\Rightarrow 1 = \frac{P((A \wedge B) \wedge C)}{P(C)} + \frac{P((\neg A \wedge B) \wedge C)}{P(C)} + \frac{P((A \wedge \neg B) \wedge C)}{P(C)} + \frac{P((\neg A \wedge \neg B) \wedge C)}{P(C)} \end{aligned}$$

$$\Rightarrow 1 = P(A, B \mid C) + P(\neg A, B \mid C) + P(A, \neg B \mid C) + P(\neg A, \neg B \mid C)$$
$$\Rightarrow P(\neg A, \neg B \mid C) = 1 - (P(A, B \mid C) + P(\neg A, B \mid C) + P(A, \neg B \mid C))$$

Prin urmare, ştiind 3 valori o putem afla şi pe a patra. La fel şi pentru $\neg C$.

Pentru a avea un clasificator Bayes Optimal complet avem deci nevoie de 7 valori diferite.

În cazul general, pentru n variabile de intrare, avem nevoie de probabilităţile $P(Y)$, $P(\tilde{X}_1, \cdots, \tilde{X}_n \mid Y)$ şi $P(\tilde{X}_1, \cdots, \tilde{X}_n \mid \neg Y)$, unde

$$\tilde{X}_i \in \{X_i, \neg X_i\}\ \forall i \in \overline{1,n} \qquad \text{şi} \qquad (\tilde{X}_1, \cdots, \tilde{X}_n) \neq (\neg X_1, \cdots, \neg X_n).$$

Avem, deci, $2(2^n - 1) + 1 = 2^{n+1} - 1$ valori.

Se observă că algoritmul Bayes Naiv foloseşte un număr liniar de parametri (în raport cu n, numărul de atribute de intrare), în vreme ce algoritmul Bayes Optimal foloseşte un număr exponenţial de parametri (în raport cu acelaşi n).

8. **(Algoritmul Bayes Naiv: aplicaţie la filtrarea emailurilor spam)**

■ • ***CMU, 2009 spring, Ziv Bar-Joseph, midterm, pr. 2***

Circa $2/3$ dintre emailurile tale sunt spam, aşadar te-ai decis să descarci de pe internet un filtru spam open-source care utilizează un clasificator Bayes Naiv.

Presupunem că ai strâns următoarele emailuri spam şi non-spam (engl., regular), şi de asemenea că doar trei cuvinte sunt discriminative pentru această clasificare, deci fiecare email este reprezentat ca un vector de 3 componente binare, fiecare dintre ele indicând dacă respectivul cuvânt este conţinut (sau nu) în email.

'study'	'free'	'money'	Category	count
1	0	0	**Regular**	1
0	0	1	**Regular**	1
1	0	0	**Regular**	1
1	1	0	**Regular**	1
0	1	0	**Spam**	4
0	1	1	**Spam**	4

a. Descoperi că filtrul spam open-source foloseşte o probabilitate a priori $P(\text{Spam}) = 0.1$. Explică în mod succint de ce crezi că această alegere este rezonabilă.

b. Calculează următorii parametri ai modelului prin metoda estimării de verosimilitate maximă (MLE), folosind netezire (engl., smoothing) de tip "add-one" (regula lui Laplace).

$P(\text{study}|\text{spam}) =$

$P(\text{study}|\text{regular}) =$

$P(\text{free}|\text{spam}) =$

$P(\text{free}|\text{regular}) =$

$P(\text{money}|\text{spam}) =$

$P(\text{money}|\text{regular}) =$

c. Folosind probabilitatea a priori şi probabilităţile condiţionate de mai sus, calculează probabilitatea ca mesajul $s =$ "money for psychology study" să fie spam, adică $P(\text{spam}|\ s)$.

d. Care ar trebui să fie valoarea probabilităţii a priori $P(\text{spam})$ în cazul în care dorim ca mesajul de mai sus să aibă aceeaşi probabilitate de fi spam respectiv non-spam (i.e., el va fi clasificat ca spam cu probabilitatea 0.5)?

Răspuns:

a. Diferenţa dintre $P_{MLE}(\text{Category}|\text{Spam}) = 2/3$ şi probabilitatea a priori indicată în enunţ (0.1) se explică prin faptul că se preferă trecerea prin filtru a unor emailuri spam, decât să fie marcate ca spam unele emailuri non-spam şi astfel să nu ajungă în Inbox.

b. Dacă nu am folosi regula de tip "add-one" a lui Laplace (pentru „netezirea" probabilităţilor), parametrii modelului ar avea valorile (obţinute prin metoda estimării de verosimilitate maximă – MLE) care apar mai jos în partea stângă. Folosind regula lui Laplace, parametrii primesc valorile indicate în partea dreaptă. Observaţi că apariţiile lui 2 de la numitorul fracţiilor corespund numărului de valori pentru fiecare dintre atributele / variabilele de intrare.

$$P(\textbf{study}|\textbf{spam}) = \frac{0}{8} = 0 \qquad P(\textbf{study}|\textbf{spam}) \overset{Laplace}{=} \frac{0+1}{8+2} = \frac{1}{10}$$

$$P(\textbf{study}|\textbf{regular}) = \frac{3}{4} \qquad P(\textbf{study}|\textbf{regular}) \overset{Laplace}{=} \frac{3+1}{4+2} = \frac{2}{3}$$

$$P(\textbf{free}|\textbf{spam}) = \frac{8}{8} = 1 \qquad P(\textbf{free}|\textbf{spam}) \overset{Laplace}{=} \frac{8+1}{8+2} = \frac{9}{10}$$

$$P(\textbf{free}|\textbf{regular}) = \frac{1}{4} \qquad P(\textbf{free}|\textbf{regular}) \overset{Laplace}{=} \frac{1+1}{4+2} = \frac{1}{3}$$

$$P(\textbf{money}|\textbf{spam}) = \frac{4}{8} = \frac{1}{2} \qquad P(\textbf{money}|\textbf{spam}) \overset{Laplace}{=} \frac{4+1}{8+2} = \frac{1}{2}$$

$$P(\textbf{money}|\textbf{regular}) = \frac{1}{4} \qquad P(\textbf{money}|\textbf{regular}) \overset{Laplace}{=} \frac{1+1}{4+2} = \frac{1}{3}$$

c. Avem mesajul $s =$ "money for psychology study", deci trebuie să calculăm $P(\text{spam} \mid s) = P(\text{spam} \mid \text{study}, \neg\text{free}, \text{money})$.

$$P(\textbf{spam} \mid s) \overset{F.\ Bayes}{=}$$

$$= \frac{P(\textbf{study, } \neg\textbf{free, money} \mid \textbf{spam}) \cdot P(\textbf{spam})}{P(\textbf{study,}\neg\textbf{free,money} \mid \textbf{spam})P(\textbf{spam}) + P(\textbf{study,}\neg\textbf{free,money} \mid \textbf{reg})P(\textbf{reg})}$$

Calculăm probabilităţile folosind ipoteza de independenţă condiţională:

$$P(\textbf{study, } \neg\textbf{free, money}|\textbf{spam}) = P(\textbf{study}|\textbf{spam}) \cdot P(\neg\textbf{free}|\textbf{spam}) \cdot P(\textbf{money}|\textbf{spam})$$
$$= \frac{1}{10} \cdot \frac{1}{10} \cdot \frac{1}{2} = \frac{1}{200}$$
$$P(\textbf{study, } \neg\textbf{free, money}|\textbf{reg}) = P(\textbf{study}|\textbf{reg}) \cdot P(\neg\textbf{free}|\textbf{reg}) \cdot P(\textbf{money}|\textbf{reg})$$
$$= \frac{2}{3} \cdot \frac{2}{3} \cdot \frac{1}{3} = \frac{4}{27}$$

Înlocuind valorile în formulă, obţinem: $P(\text{spam}|\ s) = \dfrac{\frac{1}{200} \cdot \frac{1}{10}}{\frac{1}{200} \cdot \frac{1}{10} + \frac{4}{27} \cdot \frac{9}{10}} \approx 0.0037$

Aceasta este o probabilitate mică. Se observă însă că dacă nu am fi folosit regula lui Laplace, probabilitatea ca emailul s să fie spam ar fi fost 0. Aceasta se datorează faptului că niciunul dintre emailurile spam din datele de antrenament nu conţine cuvântul *study*, care apare însă în emailul s.

d. Dacă notăm cu p probabilitatea a priori cerută, P(spam), ştiind că P(spam$\mid s) = 0.5$, putem scrie:

$$0.5 = \frac{\frac{1}{200} \cdot p}{\frac{1}{200} \cdot p + \frac{4}{27} \cdot (1-p)} \Leftrightarrow \frac{1}{2} = \frac{\frac{p}{200}}{\frac{p}{200} + \frac{4}{27} - \frac{4p}{27}} \Leftrightarrow \frac{2p}{200} = \frac{p}{200} + \frac{4}{27} - \frac{4p}{27}$$
$$\Leftrightarrow 54p = 27p + 800 - 800p \Leftrightarrow p = \frac{800}{827} \approx 0.9673$$

9. (Clasificare bayesiană: un caz particular)

CMU, 2010 spring, E. Xing, A. Singh, T. Mitchell, midterm, pr. 2.2

Se consideră variabilele aleatoare $X_1,\ X_2,\ X_3$ şi X_4. Aceste variabile sunt independente condiţional două câte două în raport cu variabila Y, cu excepţia perechii $X_3,\ X_4$. (Aşadar, dacă am aplica algoritmul Bayes Naiv, acesta ar produce erori de clasificare.)

Cum am putea modifica *regula de decizie* a algoritmului Bayes Naiv pentru a ţine cont de această particularitate a datelor?

Răspuns:

Întrucât are loc egalitatea

$$\operatorname*{argmax}_y P(Y = y \mid X_1, X_2, X_3, X_4) =$$
$$\operatorname*{argmax}_y\ P(X_1 \mid Y = y) \cdot P(X_2 \mid Y = y) \cdot P(X_3, X_4 \mid Y = y) \cdot P(Y = y),$$

urmează că regula de decizie a algoritmului Bayes Naiv în cazul dat este:

$$\operatorname*{argmax}_y\ P_{MLE}(X_1 \mid Y = y) \cdot P_{MLE}(X_2 \mid Y = y) \cdot P_{MLE}(X_3, X_4 \mid Y = y) \cdot P_{MLE}(Y = y).$$

Pentru a justifica egalitatea de mai sus, se foloseşte regula de înlănţuire condiţională:

$$P(A_1, A_2, A_3 \mid B) = P(A_3 \mid B) \cdot P(A_2 \mid A_3, B) \cdot P(A_1 \mid A_2, A_3, B),$$

care se demonstrează uşor. Varianta necondiţională a regulii de înlănţuire este:

$$P(A_1, A_2, A_3) = P(A_3) \cdot P(A_2 \mid A_3) \cdot P(A_1 \mid A_2, A_3).$$

Apoi, regula de înlănţuire condiţională se particularizează pentru evenimentele $A_1 = (X_1 = x_1)$, $A_2 = (X_2 = x_2)$, $A_3 = (X_3 = x_3, X_4 = x_4)$ şi $B = (Y = y)$. În fine, se va ţine cont că $P(X_1 = x_1 \mid X_2 = x_2, X_3 = x_3, X_4 = x_4, Y = y) = P(X_1 = x_1 \mid Y = y)$ şi $P(X_2 = x_2 \mid X_3 = x_3, X_4 = x_4, Y = y) = P(X_2 = x_2 \mid Y = y)$ datorită proprietăţii de independenţă condiţională din enunţ.

Observaţie*: Regula de decizie obţinută mai sus constituie fundamentul unui clasificator bayesian care poate fi situat din punct de vedere conceptual *între (engl., in between) **clasificatorii Bayes Naiv şi Bayes Optimal, fiindcă el combină avantajele celorlalţi doi clasificatori: produce o eroare la antrenare identică cu cea a lui Bayes Optimal şi foloseşte o proprietate de independenţă condiţională similară cu cea a lui Bayes Naiv (ceea ce conduce la un număr mai restrâns de parametri decât are Bayes Optimal). Pentru un exemplu similar, vedeţi ex. 35.e.**

10. **(Algoritmul Bayes Naiv: calculul ratei medii a erorii la antrenare)**

CMU, 2006 fall, T. Mitchell, E. Xing, midterm, pr. 6

Considerăm o problemă de clasificare binară în care fiecare exemplu X are două atribute binare $X_1, X_2 \in \{0, 1\}$ şi eticheta $Y \in \{0, 1\}$. Vom presupune că X_1 şi X_2 sunt independente condiţional în raport cu Y,[425] şi că $P(Y = 0) = P(Y = 1) = 0,5$. De asemenea, probabilităţile condiţionate sunt date în tabelele următoare:

$P(X_1 \mid Y)$	$Y = 0$	$Y = 1$
$X_1 = 0$	$0,7$	$0,2$
$X_1 = 1$	$0,3$	$0,8$

$P(X_2 \mid Y)$	$Y = 0$	$Y = 1$
$X_2 = 0$	$0,9$	$0,5$
$X_2 = 1$	$0,1$	$0,5$

a. Calculaţi predicţia $\hat{Y}$ făcută de clasificatorul Bayes Naiv pentru fiecare din cele patru combinaţii posibile de valori ale variabilelor X_1 şi X_2. Completaţi următorul tabel:

X_1	X_2	$P(X_1, X_2, Y = 0)$	$P(X_1, X_2, Y = 1)$	$\hat{Y}(X_1, X_2)$
0	0	$0,7 \cdot 0,9 \cdot 0,5$	$0,2 \cdot 0,5 \cdot 0,5$	0
0	1			
1	0			
1	1			

b. Presupunând că se folosesc o infinitate de exemple, calculaţi *rata medie a erorii* (engl., the expected error rate) **făcute de acest clasificator *la antrenare*, folosind formula:**

$$P(Y \neq \hat{Y}(X_1, X_2)) = \Sigma^1_{X_1=0}\Sigma^1_{X_2=0} P(X_1, X_2, Y = 1 - \hat{Y}(X_1, X_2)).$$

c. Care din următorii doi clasificatori are rata medie a erorii la antrenare mai mică:

- **clasificatorul Bayes Naiv care prezice Y având ca input doar X_1;**
- **clasificatorul Bayes Naiv care prezice Y având ca input doar X_2.**

d. Presupunem că definim un nou atribut X_3, care este o copie a lui X_2. Care este rata medie a erorii la antrenare a clasificatorului Bayes Naiv care prezice Y folosind toate atributele X_1, X_2, X_3? (Se presupune că datele de antrenament sunt în număr infinit.)

[425] Aşadar, în această situaţie rezultatele algoritmilor Bayes Naiv şi Bayes Optimal vor coincide.

e. Explicaţi de ce rata erorii de la punctul d diferă faţă de cea de la punctul a.

Răspuns:

a. Predicţia $\hat{Y}$ făcută de clasificatorul Bayes Naiv pentru fiecare din cele patru combinaţii posibile de valori ale variabilelor X_1 şi X_2 este înregistrată în ultima coloană a tabelului de mai jos. Calculele necesare pentru justificare — folosind formula $P(X_1, X_2, Y) = P(X_1, X_2|Y) \cdot P(Y) = P(X_1|Y) \cdot P(X_2|Y) \cdot P(Y)$ — sunt conţinute în coloanele a treia şi a patra.

X_1	X_2	$P(X_1, X_2, Y = 0)$	$P(X_1, X_2, Y = 1)$	$\hat{Y}(X_1, X_2)$
0	0	$0,7 \cdot 0,9 \cdot 0,5 = 0,315$	$0,2 \cdot 0,5 \cdot 0,5 = \mathbf{0,05}$	0
0	1	$0,7 \cdot 0,1 \cdot 0,5 = \mathbf{0,035}$	$0,2 \cdot 0,5 \cdot 0,5 = 0,05$	1
1	0	$0,3 \cdot 0,9 \cdot 0,5 = \mathbf{0,135}$	$0,8 \cdot 0,5 \cdot 0,5 = 0,2$	1
1	1	$0,3 \cdot 0,1 \cdot 0,5 = \mathbf{0,015}$	$0,8 \cdot 0,5 \cdot 0,5 = 0,2$	1

***Observaţie* (1):**
În acest tabel putem vedea valorile distribuţiei comune $P(X_1, X_2, Y)$. Spre deosebire de problema 6, unde distribuţia comună era dată, iar distribuţiile marginale condiţionale erau calculate pornind de la aceasta, aici se procedează invers, ţinând cont [şi] de presupoziţia de independenţă condiţională.

b. Rata medie a erorii este:

$$\begin{aligned} P(Y \neq \hat{Y}(X_1, X_2)) &= \sum_{X_1=0}^{1} \sum_{X_2=0}^{1} P(X_1, X_2, Y = 1 - \hat{Y}(X_1, X_2)) \\ &= P(X_1 = 0, X_2 = 0, Y = 1 - 0) + P(X_1 = 0, X_2 = 1, Y = 1 - 1) \\ &\quad + P(X_1 = 1, X_2 = 0, Y = 1 - 1) + P(X_1 = 1, X_2 = 1, Y = 1 - 1) \\ &= 0,05 + 0,035 + 0,135 + 0,015 = 0,235. \end{aligned}$$

Pentru justificarea penultimei egalităţi, vedeţi tabelul de la punctul precedent.

c. Făcând prezicerea doar cu X_1 ca atribut de intrare — folosind regula de înmulţire $P(X_1, Y) = P(X_1|Y) \cdot P(Y)$ —, obţinem:

X_1	$P(X_1, Y = 0)$	$P(X_1, Y = 1)$	$\hat{Y}_1(X_1, X_2)$
0	$0,7 \cdot 0,5 = 0,35$	$0,2 \cdot 0,5 = \mathbf{0,1}$	0
1	$0,3 \cdot 0,5 = \mathbf{0,15}$	$0,8 \cdot 0,5 = 0,4$	1

Rata medie a erorii în acest caz va fi:

$$\begin{aligned} P(Y \neq \hat{Y}_1(X_1, X_2)) &= \sum_{X_1=0}^{1} \sum_{X_2=0}^{1} P(X_1, X_2, Y = 1 - \hat{Y}_1(X_1, X_2)) \\ &= P(X_1 = 0, X_2 = 0, Y = 1 - 0) + P(X_1 = 0, X_2 = 1, Y = 1 - 0) \\ &\quad + P(X_1 = 1, X_2 = 0, Y = 1 - 1) + P(X_1 = 1, X_2 = 1, Y = 1 - 1) \\ &= 0,05 + 0,05 + 0,135 + 0,015 = 0,1 + 0,15 = 0,25. \end{aligned}$$

Similar, dacă luăm în considerare doar variabila X_2, avem:

X_2	$P(X_2, Y = 0)$	$P(X_2, Y = 1)$	$\hat{Y}_2(X_1, X_2)$
0	$0,9 \cdot 0,5 = 0,45$	$0,5 \cdot 0,5 = \mathbf{0,25}$	0
1	$0,1 \cdot 0,5 = \mathbf{0,05}$	$0,5 \cdot 0,5 = 0,25$	1

Acum, rata medie a erorii va fi:

$$\begin{aligned} P(Y \neq \hat{Y}_2(X_1, X_2)) &= \sum_{X_1=0}^{1} \sum_{X_2=0}^{1} P(X_1, X_2, Y = 1 - \hat{Y}_2(X_1, X_2)) \\ &= P(X_1 = 0, X_2 = 0, Y = 1 - 0) + P(X_1 = 0, X_2 = 1, Y = 1 - 1) \\ &\quad + P(X_1 = 1, X_2 = 0, Y = 1 - 0) + P(X_1 = 1, X_2 = 1, Y = 1 - 1) \\ &= 0,05 + 0,035 + 0,2 + 0,015 = 0,25 + 0,05 = 0,3. \end{aligned}$$

Prin urmare, rata medie a erorii la antrenare este mai mică pentru clasificatorul Bayes Naiv care prezice Y având ca input doar X_1 (decât pornind doar de la X_2).[426]

***Observaţie* (2):**
Atât în cazul lui X_1 cât şi în cazul lui X_2, rata medie a erorii (pentru algoritmul Bayes Naiv) putea fi calculată folosind direct probabilităţile din cele două tabele de mai sus. Justificarea ţine de faptul că distribuţiile calculate de Bayes Naiv în aceste două tabele sunt distribuţii marginale în raport cu distribuţia comună (reală!) din tabelul de la punctul a. La punctul d veţi vedea că acolo trebuie procedat altfel, fiindcă în cazul respectiv distribuţia calculată de către Bayes Naiv *nu* mai coincide cu distribuţia reală a datelor!

d. Clasificatorul Bayes Naiv care prezice valoarea lui Y în funcţie de toate cele trei variabilele $X_1, X_2, X_3 = X_2$ va lua deciziile conform tabelului următor:[427]

X_1	X_2	X_3	$P(X_1, X_2, X_3, Y = 0)$	$P(X_1, X_2, X_3, Y = 1)$	$\hat{Y}_3(X_1, X_2)$
0	0	0	$0,7 \cdot 0,9 \cdot 0,9 \cdot 0,5 = 0,2835$	$0,2 \cdot 0,5 \cdot 0,5 \cdot 0,5 = \mathbf{0,025}$	0
0	1	1	$0,7 \cdot 0,1 \cdot 0,1 \cdot 0,5 = \mathbf{0,0035}$	$0,2 \cdot 0,5 \cdot 0,5 \cdot 0,5 = 0,025$	1
1	0	0	$0,3 \cdot 0,9 \cdot 0,9 \cdot 0,5 = 0,1215$	$0,8 \cdot 0,5 \cdot 0,5 \cdot 0,5 = \mathbf{0,1}$	0
1	1	1	$0,3 \cdot 0,1 \cdot 0,1 \cdot 0,5 = \mathbf{0,0015}$	$0,8 \cdot 0,5 \cdot 0,5 \cdot 0,5 = 0,1$	1

***Observaţie* (3):**
Este util să observaţi că distribuţia de probabilitate comună calculată aici de către algoritmul Bayes Naiv nu mai coincide cu distribuţia „reală“ (vedeţi tabelul de la punctul a). Rata erorii se calculează în raport cu distribuţia „reală“ a datelor, nu cu cea calculată de către Bayes Naiv (deşi pentru a identifica situaţiile în care $\hat{Y} \neq Y$ se foloseşte ultimul tabel)!

Rata medie a erorii produse la antrenare va fi:

$$\begin{aligned} P(Y \neq \hat{Y}_3(X_1, X_2)) &= \sum_{X_1=0}^{1} \sum_{X_2=0}^{1} P(X_1, X_2, Y = 1 - \hat{Y}_3(X_1, X_2)) \\ &= P(X_1 = 0, X_2 = 0, Y = 1 - 0) + P(X_1 = 0, X_2 = 1, Y = 1 - 1) \\ &\quad + P(X_1 = 1, X_2 = 0, Y = 1 - 0) + P(X_1 = 1, X_2 = 1, Y = 1 - 1) \\ &= 0,05 + 0,035 + 0,2 + 0,015 = 0,3. \end{aligned}$$

e. Diferenţa dintre cele două rate medii ale erorilor care au fost calculate la punctele a şi d se datorează faptului că presupunerea de independenţă condiţională a variabilelor nu este adevărată. Într-adevăr, X_2 nu este independent

[426] Însă ambii clasificatori au rata medie a erorii mai mare decât clasificatorul Bayes Naiv care foloseşte atât X_1 cât şi X_2, ceea ce era de aşteptat întrucât în condiţiile date Bayes Naiv are acelaşi comportament ca şi Bayes Optimal.

[427] Este bine de observat că în această situaţie presupoziţia de independenţă condiţională este încălcată. Aşadar, Bayes Naiv nu va mai furniza aceleaşi rezultate ca Bayes Optimal.

condiţional faţă de X_3 deoarece cele două variabile au tot timpul valori identice.

***Observaţii importante*:**
1. Este imediat că atunci când datele satisfac presupoziţia de independenţă condiţională, algoritmii Bayes Naiv şi Bayes Optimal produc aceleaşi rezultate şi au aceeaşi rată medie a erorilor.
2. Se poate arăta uşor că algoritmul Bayes Optimal nu produce în mod neapărat o rată medie a erorilor nulă, chiar dacă lucrează cu distribuţia reală a datelor. De *exemplu*, în condiţiile definite iniţial de problema noastră, algoritmul Bayes Optimal produce rata medie 0.235, ca şi algoritmul Bayes Naiv (vedeţi punctul b). Erorile produse de către algoritmul Bayes Optimal se datorează faptului că el aplică operatorul $\arg\max$, echivalent cu luarea unui vot majoritar (impus deci minorităţii).

11. (Cât de naiv / prost este algoritmul Bayes Naiv?)

■ • ***CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, HW2, pr. 1.2***

În mod evident, clasificatorul Bayes Naiv lucrează cu o presupoziţie foarte restrictivă (engl., strong presupposition). Însă ne putem întreba dacă acest clasificator nu este totuşi destul de folositor chiar şi în cazul în care respectiva presupoziţie nu este satisfăcută.

În consecinţă, în acest exerciţiu ne propunem să folosim un exemplu simplu pentru a explora limitările algoritmului Bayes Naiv.

Fie X_1 şi X_2 variabile aleatoare binare de tip Bernoulli de parametru $p = 0.5$, iar Y o funcţie determinstă în raport cu valorile lui X_1 şi X_2, luând valori în mulţimea $\{1, 2\}$.

a. Definiţi Y astfel încât (pe setul de date respectiv) algoritmul Bayes Naiv să aibă rata medie a erorii de 50% (adică maximă!).

Pe acest caz, observaţi cum se corelează valorile lui X_1 şi X_2 când valorea lui Y este fixată. (Altfel spus, observaţi cât de (in)dependente sunt în acest context valorile lui X_1 şi X_2, dată fiind valoarea lui Y.)

X_1	X_2	Y
0	0	
0	1	
1	0	
1	1	

b. Există în total $2^4 = 16$ moduri în care poate fi definită funcţia Y. Însă, datorită simetriei (relativ la valorile lui Y), problema se reduce la doar 4 cazuri,

X_1	X_2	Y
0	0	1
0	1	1
1	0	1
1	1	1

X_1	X_2	Y
0	0	1
0	1	1
1	0	1
1	1	2

X_1	X_2	Y
0	0	1
0	1	2
1	0	1
1	1	2

X_1	X_2	Y
0	0	1
0	1	2
1	0	2
1	1	1

dintre care un caz corespunde punctului a de mai sus. În fiecare din acele trei cazuri rămase după rezolvarea de la punctul a, arătaţi că rata erorii produsă de algoritmul Bayes Naiv este 0.

Răspuns:

a. Considerăm Y definit conform tabelului de mai jos.

X_1	X_2	Y
0	0	1
0	1	2
1	0	2
1	1	1

***Observaţie*: Dacă se consideră valoarea lui Y fixată (fie 1, fie 2), atunci putem să stabilim o regulă astfel încât dacă îl cunoaştem pe X_1 să-l determinăm pe X_2 (şi invers).[428] Altfel spus, X_1 este unic determinat de X_2 (şi invers), dată fiind o valoare fixată a lui Y. Deci condiţia de independenţă condiţională este încălcată. Mai mult, în acest caz avem maximul posibil de „dependenţă“ între cele două variabile (în raport cu Y).**

Dorim să calculăm rata erorii înregistrate de algoritmul Bayes Naiv pe datele din tabelul de mai sus. Bayes Naiv estimează valoarea lui Y astfel:

$$\hat{y} = \underset{y \in \{1,2\}}{\operatorname{argmax}} P(X_1 \mid Y = y) \cdot P(X_2 \mid Y = y) \cdot P(Y = y)$$

Pentru $X_1 = 0, X_2 = 0$, algoritmul compară următoarele două valori:

$$\begin{aligned} p_1 &= P(X_1 = 0 \mid Y = 1) \cdot P(X_2 = 0 \mid Y = 1) \cdot P(Y = 1) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8} \\ p_2 &= P(X_1 = 0 \mid Y = 2) \cdot P(X_2 = 0 \mid Y = 2) \cdot P(Y = 2) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{8} \end{aligned}$$

Cum $p_1 = p_2$, algoritmul va alege una dintre ele cu o probabilitate de 0.5. Deoarece valoarea lui Y din tabel este 1, înseamnă că algoritmul va alege greşit cu o probabilitate de 0.5.

Pentru celelalte 3 cazuri, $(X_1 = 0, X_2 = 1)$, $(X_1 = 1, X_2 = 0)$ şi $(X_1 = 1, X_2 = 1)$, se observă uşor că se obţin de asemenea valori egale, iar algoritmul va alege pentru Y una dintre valorile 1 sau 2 cu o probabilitate de 0.5.

Deci pentru această definiţie a lui Y rata erorii este de 50%.

b. Vom calcula rata erorii pentru fiecare dintre cele 3 moduri de definire a lui Y care nu a fost studiat.

***Cazul 1*:**

X_1	X_2	Y
0	0	1
0	1	1
1	0	1
1	1	1

Este similar cu cazul:

Y
2
2
2
2

- **Pentru $X_1 = 0, X_2 = 0$, algoritmul compară:**

$$\begin{aligned} p_1 &= P(X_1 = 0 \mid Y = 1) \cdot P(X_2 = 0 \mid Y = 1) \cdot P(Y = 1) = \frac{2}{4} \cdot \frac{2}{4} \cdot 1 = \frac{1}{4} \\ p_2 &= P(X_1 = 0 \mid Y = 2) \cdot P(X_2 = 0 \mid Y = 2) \cdot P(Y = 2) = 0 \cdot 0 \cdot 0 = 0 \end{aligned}$$

  Cum $p_1 > p_2$ algoritmul alege pentru Y valoarea 1, ceea ce este corect.

- **Pentru celelalte 3 cazuri, $(X_1 = 0, X_2 = 1)$, $(X_1 = 1, X_2 = 0)$ şi $(X_1 = 1, X_2 = 1)$, se observă că se obţin aceleaşi valori pentru p_1 şi p_2 ca mai sus, deci algoritmul alege (în mod corect) pentru Y valoarea 1.**

[428] Pentru $Y = 1$, regula este: X_2 are aceeaşi valoare ca şi X_1. Pentru $Y = 2$, regula este: X_1 şi X_2 au valori complementare.

Aşadar, am obţinut că rata erorii este în acest caz 0.

***Cazul 2*:**

X_1	X_2	Y
0	0	1
0	1	1
1	0	1
1	1	2

Cazuri similare:

Y	Y	Y	Y	Y	Y	Y
1	1	2	2	2	2	1
1	2	1	2	2	1	2
2	1	1	2	1	2	2
1	1	1	1	2	2	2

- **Pentru $X_1 = 0, X_2 = 0$:**

$$\left.\begin{aligned} p_1 &= P(X_1 = 0 \mid Y = 1) \cdot P(X_2 = 0 \mid Y = 1) \cdot P(Y = 1) = \frac{2}{3} \cdot \frac{2}{3} \cdot \frac{3}{4} = \frac{1}{3} \\ p_2 &= P(X_1 = 0 \mid Y = 2) \cdot P(X_2 = 0 \mid Y = 2) \cdot P(Y = 2) = 0 \cdot 0 \cdot \frac{1}{4} = 0 \end{aligned}\right\} \Rightarrow \hat{y} = 1$$

- **Pentru $X_1 = 0, X_2 = 1$:**

$$\left.\begin{aligned} p_1 &= P(X_1 = 0 \mid Y = 1) \cdot P(X_2 = 1 \mid Y = 1) \cdot P(Y = 1) = \frac{2}{3} \cdot \frac{1}{3} \cdot \frac{3}{4} = \frac{1}{6} \\ p_2 &= P(X_1 = 0 \mid Y = 2) \cdot P(X_2 = 1 \mid Y = 2) \cdot P(Y = 2) = 0 \cdot 1 \cdot \frac{1}{4} = 0 \end{aligned}\right\} \Rightarrow \hat{y} = 1$$

- **Pentru $X_1 = 1, X_2 = 0$:**

$$\left.\begin{aligned} p_1 &= P(X_1 = 1 \mid Y = 1) \cdot P(X_2 = 0 \mid Y = 1) \cdot P(Y = 1) = \frac{1}{3} \cdot \frac{2}{3} \cdot \frac{3}{4} = \frac{1}{6} \\ p_2 &= P(X_1 = 1 \mid Y = 2) \cdot P(X_2 = 0 \mid Y = 2) \cdot P(Y = 2) = 1 \cdot 0 \cdot \frac{1}{4} = 0 \end{aligned}\right\} \Rightarrow \hat{y} = 1$$

- **Pentru $X_1 = 1, X_2 = 1$:**

$$\left.\begin{aligned} p_1 &= P(X_1 = 1 \mid Y = 1) \cdot P(X_2 = 1 \mid Y = 1) \cdot P(Y = 1) = \frac{1}{3} \cdot \frac{1}{3} \cdot \frac{3}{4} = \frac{1}{12} \\ p_2 &= P(X_1 = 1 \mid Y = 2) \cdot P(X_2 = 1 \mid Y = 2) \cdot P(Y = 2) = 1 \cdot 1 \cdot \frac{1}{4} = \frac{1}{4} \end{aligned}\right\} \Rightarrow \hat{y} = 2$$

Deci rata erorii este 0 pentru această definiţie a lui Y.

***Cazul 3*:**

X_1	X_2	Y
0	0	1
0	1	2
1	0	1
1	1	2

Cazuri similare:

Y	Y	Y
2	1	2
1	1	2
2	2	1
1	2	1

- **Pentru $X_1 = 0, X_2 = 0$:**

$$\left.\begin{aligned} p_1 &= \frac{1}{2} \cdot 1 \cdot \frac{1}{2} = \frac{1}{4} \\ p_2 &= \frac{1}{2} \cdot 0 \cdot \frac{1}{2} = 0 \end{aligned}\right\} \Rightarrow p_1 > p_2 \Rightarrow \hat{y} = 1 \textbf{ (corect)}$$

- **Pentru $X_1 = 0, X_2 = 1$:**

$$\left.\begin{aligned} p_1 &= \frac{1}{2} \cdot 0 \cdot \frac{1}{2} = 0 \\ p_2 &= \frac{1}{2} \cdot 1 \cdot \frac{1}{2} = \frac{1}{4} \end{aligned}\right\} \Rightarrow p_1 < p_2 \Rightarrow \hat{y} = 2 \textbf{ (corect)}$$

- **Pentru $X_1 = 1, X_2 = 0$:**

$$\left.\begin{aligned} p_1 &= \frac{1}{2} \cdot 1 \cdot \frac{1}{2} = \frac{1}{4} \\ p_2 &= \frac{1}{2} \cdot 0 \cdot \frac{1}{2} = 0 \end{aligned}\right\} \Rightarrow p_1 > p_2 \Rightarrow \hat{y} = 1 \textbf{ (corect)}$$

- **Pentru** $X_1 = 1, X_2 = 1$**:**

$$\left.\begin{aligned} p_1 &= \frac{1}{2} \cdot 0 \cdot \frac{1}{2} = 0 \\ p_2 &= \frac{1}{2} \cdot 1 \cdot \frac{1}{2} = \frac{1}{4} \end{aligned}\right\} \Rightarrow p_1 < p_2 \Rightarrow \hat{y} = 2 \textbf{ (corect)}$$

Prin urmare, rata erorii este 0 **şi în acest caz.**

***Cazul 4*: (cel de la punctul** a**)**

X_1	X_2	Y
0	0	1
0	1	2
1	0	2
1	1	1

Este similar cu cazul:

Y
2
1
1
2

***Concluzie* şi... o *observaţie interesantă*:**
Se constată că doar pentru 2 moduri (şi anume, cazul 4) de definire a lui Y **rata erorii este de** 50%**; pentru celelalte 14 moduri (cazurile 1, 2, 3) rata erorii este** 0**. Este imediat că aceste două moduri de definire a funcţiei** Y (XOR **şi respectiv** $\neg$XOR**) corespund exact acelor moduri / funcţii — dintre toate cele 16 moduri posibile — care au reprezentări care sunt neseparabile liniar în spaţiul** $\mathbb{R}^2$**. Acest fapt este în concordanţă cu rezultatul demonstrat la problema 14. Acolo se arată că regulii de decizie a algoritmului Bayes Naiv îi corespunde (în general) un separator liniar, similar cu cel din cazul regresiei logistice.**

12. **(O reprezentare grafică a neconcordanţei deciziilor luate de algoritmii Bayes Naiv şi Bayes Optimal)**

CMU, 2009 fall, Geoff Gordon, HW4, pr. 1
CMU, 2009 fall, Carlos Guestrin, HW1, pr. 4.1.5

Pentru un task de clasificare se consideră atributele X_1**,** X_2 **şi** X_3 **şi eticheta** Y**. Toate acestea sunt variabile aleatoare binare.** X_1 **şi** X_2 **sunt independente condiţional în raport cu** Y**, iar** X_3 **este o copie a lui** X_2 **(aşadar, întotdeauna** $X_2 = X_3$**).**

Sunt date următoarele probabilităţi condiţionate:

$$\begin{aligned} &P(X_1 = T \mid Y = T) = p, \quad P(X_1 = T \mid Y = F) = 1 - p, \\ &P(X_2 = F \mid Y = T) = q, \quad P(X_2 = F \mid Y = F) = 1 - q, \\ &P(Y = T) = 0.5. \end{aligned}$$

Se dă instanţa de test $X_1 = T$**,** $X_2 = X_3 = F$**. Vrem să clasificăm această instanţă, adică să prezicem valoarea lui** Y **pentru ea.**

a. Arătaţi că dacă se foloseşte clasificatorul Bayes Naiv, atunci eticheta pentru instanţa aceasta de test este T **— ceea ce revine la** $P(Y = T \mid X_1 = T, X_2 = F, X_3 = F) \geq 0.5$ **— dacă** $p \geq \frac{(1-q)^2}{q^2+(1-q)^2}$**.**

b. Ce devine inegalitatea de la punctul precedent dacă în schimbul clasificatorului Bayes Naiv se utilizează clasificatorul Bayes Optimal?

c. Desenaţi cele două curbe de decizie obţinute la punctele a **şi** b**. Pe axa *Ox* marcaţi valorile lui** q**, iar pe axa *Oy* marcaţi valorile lui** p**. Atât** p **cât şi** q

variază în intervalul $[0,1]$. Indicaţi pe grafic zona în care clasificatorul Bayes Naiv produce un output (Y) diferit de cel al algoritmului Bayes Optimal.

***Atenţie*! Acest exerciţiu *nu* studiază în ce condiţii cei doi algoritmi clasifică corect (ori dimpotrivă, eronat) instanţa $X_1 = T, X_2 = X_3 = F$, ci doar când anume (în funcţie de p şi q) produc ei clasificări diferite pentru această instanţă. Se va vedea (grafic!) că algoritmul Bayes Naiv nu se comportă deloc rău în comparaţie cu algoritmul Bayes Optimal.**

Răspuns:

Se lucrează cu variabile aleatoare binare, iar pentru claritatea calculelor vom nota prin X faptul că valoarea variabilei aleatoare X este T, iar prin $\neg X$ faptul că $X = F$.

Folosind aceste notaţii, putem transcrie datele din enunţ astfel:

$$\begin{array}{ll} P(X_1 \mid Y) = p, & P(X_1 \mid \neg Y) = 1-p, \\ P(\neg X_2 \mid Y) = q, & P(\neg X_2 \mid \neg Y) = 1-q, \\ P(Y) = 0.5. & \end{array}$$

a. Eticheta pentru instanţa aceasta de test este T dacă $P(Y \mid X_1, \neg X_2, \neg X_3) \geq 0.5$. Vom calcula această probabilitate utilizând regula lui Bayes precum şi ipoteza de independenţă condiţională făcută de algoritmul Bayes Naiv:

$$\begin{aligned} P(Y \mid X_1, \neg X_2, \neg X_3) &\overset{form.\ Bayes}{=} \\ &= \frac{P(X_1, \neg X_2, \neg X_3 \mid Y)P(Y)}{P(X_1, \neg X_2, \neg X_3 \mid Y)P(Y) + P(X_1, \neg X_2, \neg X_3 \mid \neg Y)P(\neg Y)} \overset{indep.\ cdt.}{=} \\ &= \frac{P(X_1 \mid Y)P(\neg X_2 \mid Y)P(\neg X_3 \mid Y)P(Y)}{P(X_1|Y)P(\neg X_2|Y)P(\neg X_3|Y)P(Y) + P(X_1|\neg Y)P(\neg X_2|\neg Y)P(\neg X_3|\neg Y)P(\neg Y)} \\ &= \frac{p \cdot q \cdot q \cdot 0.5}{p \cdot q \cdot q \cdot 0.5 + (1-p) \cdot (1-q) \cdot (1-q) \cdot 0.5} = \frac{pq^2}{pq^2 + (1-p)(1-q)^2} \end{aligned}$$

Deci eticheta este T dacă $\dfrac{pq^2}{pq^2 + (1-p)(1-q)^2} \geq 0.5$, ceea ce înseamnă că

$$pq^2 \geq 0.5(pq^2 + (1-p)(1-q)^2) \Leftrightarrow pq^2 - 0.5pq^2 \geq 0.5(1-p)(1-q)^2$$

$$\Leftrightarrow pq^2 \geq (1-q)^2 - p(1-q)^2 \Leftrightarrow p(q^2 + (1-q)^2) \geq (1-q)^2 \Leftrightarrow p \geq \frac{(1-q)^2}{q^2 + (1-q)^2}$$

b. Dacă în locul clasificatorului Bayes Naiv se foloseşte clasificatorul Bayes Optimal, nu se mai foloseşte presupunerea de independenţă condiţională. În locul acesteia se folosesc informaţiile furnizate în enunţ, şi anume că X_1 şi X_2 sunt independente condiţional în raport cu Y, iar X_3 este o copie a lui X_2.

$$\begin{aligned} P(Y \mid X_1, \neg X_2, \neg X_3) &\overset{form.\ Bayes}{=} \\ &= \frac{P(X_1, \neg X_2, \neg X_3 \mid Y)P(Y)}{P(X_1, \neg X_2, \neg X_3 \mid Y)P(Y) + P(X_1, \neg X_2, \neg X_3 \mid \neg Y)P(\neg Y)} \\ &= \frac{P(X_1 \mid Y)P(\neg X_2, \neg X_3 \mid Y)P(Y)}{P(X_1|Y)P(\neg X_2, \neg X_3|Y)P(Y) + P(X_1|\neg Y)P(\neg X_2, \neg X_3|\neg Y)P(\neg Y)} \\ &= \frac{P(X_1 \mid Y)P(\neg X_2 \mid Y)P(Y)}{P(X_1|Y)P(\neg X_2|Y)P(Y) + P(X_1|\neg Y)P(\neg X_2|\neg Y)P(\neg Y)} \end{aligned}$$

$$= \frac{p \cdot q \cdot 0.5}{p \cdot q \cdot 0.5 + (1-p) \cdot (1-q) \cdot 0.5} = \frac{pq}{pq + (1-p)(1-q)}$$

S-a folosit egalitatea

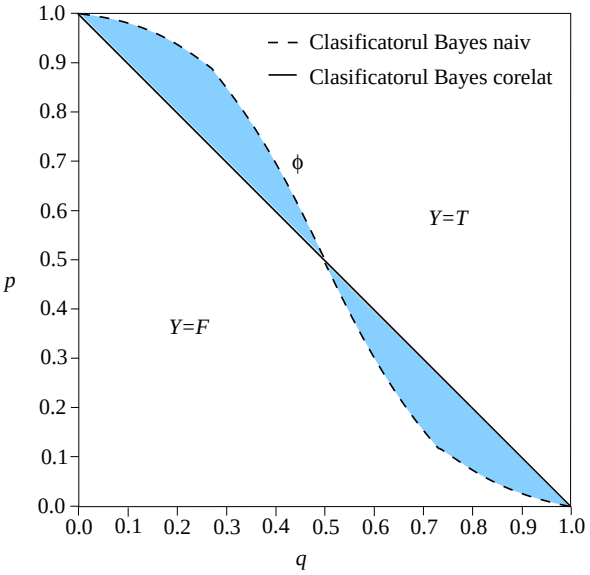
$$P(X_1, \neg X_2, \neg X_3 \,|\, Y) = P(X_1 \,|\, \neg X_2, \neg X_3, Y) \cdot P(\neg X_2, \neg X_3 \,|\, Y) = P(X_1 \,|\, Y) \cdot P(\neg X_2 \,|\, Y).$$

În acest caz eticheta este T dacă $\frac{pq}{pq + (1-p)(1-q)} \geq 0.5$, adică

$$pq \geq 0.5(pq + (1-p)(1-q)) \Leftrightarrow pq - 0.5pq \geq 0.5(1 - p - q + pq)$$
$$\Leftrightarrow pq \geq 1 - p - q + pq \Leftrightarrow p \geq 1 - q.$$

c. Dreapta de ecuaţie $p = 1 - q$ este uşor de reprezentat.

Notând $\phi(q) = \frac{(1-q)^2}{q^2 + (1-q)^2}$, se observă imediat că $\phi(q) + \phi(1-q) = 1 \Leftrightarrow \phi(1-q) = 1 - \phi(q)$. De aici, cu ajutorul unui raţionament geometric simplu, se ajunge imediat la concluzia că graficul funcţiei ϕ este simetric faţă de punctul de coordonate $(1/2, 1/2)$. Aşadar, va fi suficient să studiem graficul lui ϕ pe intervalul $[0, 1/2]$. Proprietăţile funcţiei ϕ necesare elaborării graficului sunt uşor de studiat. Adiţional, se poate arăta imediat că $\phi(q) \geq 1 - q$ pentru orice $q \in [0, 1/2]$ şi $\phi(q) \leq 1 - q$ pentru orice $q \in [1/2, 1]$.



În figura de mai sus am reprezentat curbele $p = 1 - q$ şi $p = \phi(q)$, precum şi zonele de decizie pentru cei doi clasificatori obţinuţi la punctele a şi b. Se observă uşor zona în care rezultatul produs de clasificatorul Bayes Naiv (Y) este diferit / „eronat“ în raport cu algoritmul Bayes Optimal.

13. **(Cât de multe date de antrenament necesită algoritmul Bayes Naiv vs. algoritmul Bayes Optimal? [LC: complexitatea la eşantionare])**

■ • ***CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, HW2, pr. 1.1***

Unul dintre motivele pentru care folosim clasificatorul Bayes Naiv este faptul că el necesită mult mai puţine date de antrenament (în vederea estimării parametrilor) decât clasificatorul Bayes Optimal.

Acest exerciţiu te va ajuta să înţelegi cât de importantă este această diferenţă dintre cei doi algoritmi.

Presupunem că o *observaţie / instanţă* este o valoare generată în mod aleatoriu de către vectorul de variabile aleatoare $\bar{X} = (X_1, \ldots, X_{d-1}, X_d)$, unde fiecare

X_i este o variabilă aleatoare urmând distribuţia probabilistă Bernoulli de parametru $p = 0.5$. Considerăm $X_1, \ldots, X_{d-1}$ variabilele de intrare, iar $X_d = Y$ variabila de ieşire.

Pentru a estima în sensul verosimilităţii maxime (MLE) parametrii clasificatorului Bayes Optimal, avem nevoie să *observăm* / întâlnim fiecare valoare a lui $\bar{X}$ de un număr rezonabil de ori. Similar, pentru a antrena clasificatorul Bayes Naiv avem nevoie să întâlnim fiecare valoare a fiecărei variabile X_i ($i = \overline{1,d}$) de un număr rezonabil de ori.

Ne întrebăm cât de multe observaţii sunt necesare (a fi generate) pentru ca fiecare valoare a variabilei comune $\bar{X}$ în cazul algoritmului Bayes Optimal, şi respectiv fiecare valoare a variabilelor X_i ($i = \overline{1,d}$) în cazul Bayes Naiv să fie întâlnită cel puţin o dată. (În practică este nevoie de mult mai multe observaţii, dar în acest exerciţiu ne limităm la câte o singură observaţie pentru fiecare valoare în parte.)

***Indicaţie*: La rezolvarea punctelor de mai jos vă sugerăm să folosiţi următoarele două inegalităţi:**

- **pentru orice evenimente $E_1, \ldots, E_n$, avem $P(E_1 \cup \ldots \cup E_n) \leq \sum_{i=1}^{n} P(E_i)$.**[429]
- **$(1 - \frac{1}{k})^k \leq \frac{1}{e}$ pentru orice $k \geq 1$, unde $e \approx 2.71828$ este baza logaritmului natural.**

a. Începem cu algoritmul Bayes Naiv. Fie $i \in \{1, \ldots, d\}$ fixat. Arătaţi că dacă s-au făcut N observaţii (având forma $\bar{x}_j = (x_1^j, \ldots, x_{d-1}^j, x_d^j)$ cu $j = 1, \ldots, N$), atunci probabilitatea să nu fi întâlnit ambele valori ale variabilei X_i este $\frac{1}{2^{N-1}}$. (Observaţi că această fracţie reprezintă un număr foarte mic atunci când N este suficient de mare.)

b. Fie $\varepsilon > 0$ fixat. Folosind prima inegalitate din *indicaţia* de mai sus, arătaţi că dacă au fost făcute câte $N_{NB} = 1 + \log_2 \frac{d}{\varepsilon}$ observaţii de aceeaşi formă ca mai sus, atunci probabilitatea să nu fi întâlnit ambele valori pentru fiecare dintre variabilele X_i ($i = \overline{1,d}$) este mai mică sau egală cu ε.

c. Acum trecem la algoritmul Bayes Optimal. Fie $\bar{x}$ o instanţă (fixată) a variabilei comune $\bar{X}$. Folosind a doua inegalitate din *indicaţia* de mai sus, arătaţi că dacă s-au făcut N observaţii (fiecare observaţie implicând simultan toate variabilele X_i cu $i = \overline{1,d}$), atunci probabilitatea ca să nu se fi întâlnit niciodată $\bar{x}$ este mai mică sau egală cu $e^{-\frac{N}{2^d}}$.

d. Arătaţi că dacă au fost făcute cel puţin $N_{JB} = 2^d \ln \frac{2^d}{\varepsilon}$ observaţii, atunci probabilitatea ca să nu se fi întâlnit toate instanţele variabilei comune $\bar{X}$ este mai mică sau egală cu ε.

e. Dacă se fixează $\varepsilon = 0.1$, calculaţi valorile N_{NB} şi N_{JB} pentru $d = 2$, $d = 5$ şi $d = 10$. Ce concluzie puteţi trage? (Altfel spus, cum interpretaţi rezultatele?)

***Observaţii*:**

1. Ştim că pentru clasificatorul Bayes Naiv, este necesar să estimăm din date

[429] Aceasta se numeşte *proprietatea de subadditivitate* a probabilităţilor.

probabilităţile $P(Y = y)$ — adică, $P(X_d = x_d)$ — şi $P(X_i = x_i|Y = y)$, pentru $i = 1, \ldots, d-1$. Putem considera că este suficient să întâlnim cu o probabilitate de cel puţin $1 - \varepsilon$ toate valorile y, precum şi perechile de forma (x_i, y), cu $i = 1, \ldots, d-1$. În mod implicit, problema noastră simplifică şi mai mult cerinţele, considerând că este suficient să găsim cu probabilitate de cel puţin $1 - \varepsilon$ toate valorile x_i, pentru $i = 1, \ldots, d$ (subînţelegând că atunci vor apărea în date, cu probabilităţi semnificative, atât valorile y cât şi fiecare dintre combinaţiile (x_i, y), cu $i = 1, \ldots, d-1$).

2. Pentru clasificatorul Bayes Optimal, în mod similar, este necesar să estimăm probabilităţile $P(Y = y)$ şi $P(X_1 = x_1, \ldots, X_{d-1} = x_{d-1}|Y = y)$, ceea ce, evident, va permite calcularea probabilităţilor de forma $P(x_1, \ldots, x_{d-1}, y)$. Problema noastră consideră în mod implicit că este suficient să găsim cu probabilitate de cel puţin $1 - \varepsilon$ toate combinaţiile $(x_1, \ldots, x_{d-1}, y)$.

Răspuns:

a. Dacă s-au facut N observaţii şi nu s-au întâlnit ambele valori ale variabilei X_i, înseamnă că ea are aceeaşi valoare în toate aceste observaţii (adică ea este fie 0 în toate observaţiile, fie 1 în toate observaţiile). Pentru fiecare dintre aceste două cazuri probabilitatea este $1/2^N$, deci:

$$P(\text{doar una dintre valorile variabilei } X_i \text{ a apărut în } N \text{ observaţii}) = \left(\frac{1}{2}\right)^N + \left(\frac{1}{2}\right)^N = \frac{2}{2^N} = \frac{1}{2^{N-1}}$$

b. Se cere să se calculeze probabilitatea să nu fi întâlnit ambele valori pentru fiecare dintre variabilele $X_1, \ldots, X_{d-1}, X_d$. Pentru acesta vom folosi prima inegalitate din indicaţia dată în enunţ:

$$\begin{aligned} &P(\text{nu toate valorile variabilelor } X_i,\ i = \overline{1,d},\ \text{au apărut în } N_{NB} \text{ observaţii}) \\ &\leq \sum_{i=1}^{d} P(\text{numai una dintre valorile variabilei } X_i \text{ a apărut în } N_{NB} \text{ observaţii}) \\ &= \sum_{i=1}^{d} \frac{1}{2^{N_{NB}-1}} = d \cdot \frac{1}{2^{N_{NB}-1}} = d \cdot \frac{1}{2^{1+\log_2 \frac{d}{\varepsilon}-1}} = d \cdot \frac{1}{2^{\log_2 \frac{d}{\varepsilon}}} = d \cdot \frac{1}{\frac{d}{\varepsilon}} = d \cdot \frac{\varepsilon}{d} = \varepsilon. \end{aligned}$$

c. Pentru algoritmul Bayes Optimal s-au făcut N observaţii. Trebuie să calculăm probabilitatea ca să nu se fi întâlnit niciodată instanţa $\bar{x}$.

Cum există în total 2^d posibile observaţii, probabilitatea ca instanţa $\bar{x}$ să nu fie obţinută la *una* dintre observaţii este $1 - \frac{1}{2^d}$. Observaţiile fiind independente, probabilitatea ca $\bar{x}$ să nu fie obţinută *după* N observaţii este $\left(1 - \frac{1}{2^d}\right)^N$. Aşadar,

$$\begin{aligned} P(\text{instanţa } \bar{x} \text{ n-a fost întâlnită în } N \text{ observaţii}) &= \left(1 - \frac{1}{2^d}\right)^N \\ &= \left[\left(1 - \frac{1}{2^d}\right)^{2^d}\right]^{N/2^d} \leq \left(\frac{1}{e}\right)^{N/2^d} = e^{-N/2^d} \end{aligned}$$

d. Vom calcula probabilitatea ca să nu se fi întâlnit toate instanţele variabilei $\bar{X}$ utilizând din nou prima inegalitate din *indicaţie*:

$$\begin{aligned} &P(\textbf{nu toate instanţele variabilei } \bar{X} \textbf{ au fost întâlnite în } N_{JB} \textbf{ observaţii}) \\ &\leq \sum_{\bar{x}} P(\textbf{instanţa } \bar{x} \textbf{ n-a fost întâlnită în } N_{JB} \textbf{ observaţii}) \\ &\leq \sum_{\bar{x}} e^{-N_{JB}/2^d} = 2^d \cdot e^{-N_{JB}/2^d} = 2^d \cdot e^{-\ln \frac{2^d}{\varepsilon}} = 2^d \cdot \frac{1}{e^{\ln \frac{2^d}{\varepsilon}}} = \frac{2^d}{\frac{2^d}{\varepsilon}} = \varepsilon. \end{aligned}$$

e. Vom înlocui datele numerice în formulele $N_{NB} = 1 + \log_2 \frac{d}{\varepsilon}$, $N_{JB} = 2^d \ln \frac{2^d}{\varepsilon}$.

$$\varepsilon = 0.1,\ d = 2 \Rightarrow \begin{cases} N_{NB} = 1 + \log_2 \frac{2}{0.1} = 1 + \log_2 20 \approx 5.32 \\ N_{JB} = 2^2 \cdot \ln \frac{2^2}{0.1} = 4 \cdot \ln 40 \approx 14.75 \end{cases}$$

$$\varepsilon = 0.1,\ d = 5 \Rightarrow \begin{cases} N_{NB} = 1 + \log_2 \frac{5}{0.1} = 1 + \log_2 50 \approx 6.64 \\ N_{JB} = 2^5 \cdot \ln \frac{2^5}{0.1} = 32 \cdot \ln 320 \approx 184.58 \end{cases}$$

$$\varepsilon = 0.1,\ d = 10 \Rightarrow \begin{cases} N_{NB} = 1 + \log_2 \frac{10}{0.1} = 1 + \log_2 100 \approx 7.64 \\ N_{JB} = 2^{10} \cdot \ln \frac{2^{10}}{0.1} = 1024 \cdot \ln 10240 \approx 9455.67 \end{cases}$$

Acum se observă uşor [diferenţa dintre] numărul de date de antrenament necesare pentru cei doi algoritmi: de ordin logaritmic pentru Bayes Naiv, respectiv de ordin exponenţial pentru Bayes Optimal.

14. (Algoritmul Bayes Naiv: raportul cu regresia logistică şi natura separatorului decizional; cazul când variabilele de intrare sunt de tip boolean)

■ □ • ○ ***CMU, 2005 fall, T. Mitchell, A. Moore, HW2, pr. 2***
CMU, 2009 fall, Carlos Guestrin, HW1, pr. 4.1.2
CMU, 2009 fall, Geoff Gordon, HW4, pr. 1.2-3
CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, HW2, pr. 3.a

a. [Bayes Naiv şi Regresia Logistică: relaţia dintre regulile de decizie][430]

Fie Y o variabilă aleatoare cu valori booleene, iar $X = (X_1, \ldots, X_d)$ un vector de variabile aleatoare cu valori booleene. Demonstraţi că distribuţia condiţională $P(Y|X)$ are forma funcţiei logistice de argument $z = -(w_0 + w_1 X_1 \ldots + w_d X_d)$,

[430] Pentru o introducere în chestiunea regresiei logistice, vedeţi Tom Mitchell, *Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression*, draft pentru un capitol suplimentar pentru o nouă ediţie a cărţii *Machine Learning*, 2016. Puteţi vedea de asemenea problema 13 de la capitolul *Metode de regresie* din prezenta culegere.

cu parametrii $w_0, w_1, \ldots, w_d \in \mathbb{R}$,[431] adică

$$P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + \sum_{i=1}^{d} w_i X_i)}$$

şi, prin urmare

$$P(Y = 0|X) = \frac{\exp(w_0 + \sum_{i=1}^{d} w_i X_i)}{1 + \exp(w_0 + \sum_{i=1}^{d} w_i X_i)}.$$

Vă reamintim că *funcţia logistică* (sau *sigmoidală*) este definită prin expresia $\sigma(z) = 1/(1 + e^{-z})$ pentru orice $z \in \mathbb{R}$. (Vedeţi problema 33 de la capitolul *Metode de regresie*.)

***Comentariu*:**
Regresia logistică (şi, mai general, *clasificatorii „discriminativi“*) învaţă [în mod] direct parametrii distribuţiei $P(Y|X)$,[432] pe când algoritmul Bayes Naiv (şi, mai general, *clasificatorii „generativi“*) învaţă [parametrii pentru] distribuţiile $P(X|Y)$ şi $P(Y)$, cu ajutorul cărora va calcula apoi $P(Y|X)$ şi cea mai probabilă valoare pentru Y (atunci când X are o valoare fixată / dată). Vom spune că regresia logistică este corespondentul „discriminativ“ al clasificatorului „generativ“ Bayes Naiv.

***Indicaţii*:**
1. Vom introduce o *notaţie* simplă, care ne va fi de folos în continuare. Întrucât variabilele X_i sunt sunt booleene, odată fixată o valoare y_k pentru variabila Y, vom avea nevoie de un singur parametru pentru a defini distribuţia condiţională $P(X_i|Y = y_k)$, pentru fiecare $i = 1, \ldots, d$. Aşadar, vom desemna cu θ_{i1} probabilitatea $P(X_i = 1|Y = 1)$ şi, prin urmare $P(X_i = 0|Y = 1) = 1 - \theta_{i1}$. În mod similar, vom desemna cu θ_{i0} probabilitatea $P(X_i = 1|Y = 0)$.
2. Remarcaţi că odată ce am introdus notaţiile de mai sus, vom putea scrie $P(X_i|Y = 1)$ după cum urmează:

$$P(X_i|Y = 1) = \theta_{i1}^{X_i}(1 - \theta_{i1})^{(1-X_i)}, \tag{230}$$

bineînţeles, cu excepţia cazurilor când $\theta_{i1} = 0$ şi $X_i = 0$, respectiv $\theta_{i1} = 1$ şi $X_i = 1$. Observaţi că atunci când X_i are valoarea 1, cel de-al doilea factor din partea dreaptă a egalităţii (230) este 1, pentru că exponentul lui este zero. Deci $P(X_i|Y = 1) = \theta_{i1}^{X_i} = \theta_{i1}$ pentru $X_i = 1$. În mod similar, atunci când $X_i = 0$ primul factor este egal cu 1, pentru că exponentul lui este zero. Deci $P(X_i|Y = 1) = (1 - \theta_{i1})^{1-X_i} = 1 - \theta_{i1}$ pentru $X_i = 0$.

b. [Relaxarea presupoziţiei de independenţă condiţională][433]

Pentru a putea exprima interacţiunile dintre trăsături, modelul regresiei logistice poate fi extins cu nişte termeni suplimentari. De exemplu, putem adăuga un termen care să exprime dependenţa dintre trăsăturile X_1 şi X_2:

$$P(Y = 1|X) = \frac{1}{1 + \exp(w_0 + w_{1,2}X_1X_2 + \sum_{i=1}^{d} w_i X_i)}.$$

[431] LC: Prin urmare, *separatorul decizional* (sau, *graniţa de decizie*) pentru algoritmul Bayes Naiv este — într-o astfel de situaţie — liniar (în funcţie de argumentele $X_1, \ldots, X_d$). Ecuaţia separatorului decizional va fi $w_0 + w_1X_1 \ldots + w_dX_d = 0$.

[432] LC: Parametrii distribuţiei $P(Y|X)$ sunt în acest caz $w_i \in \mathbb{R}$, cu $i = 0, 1, \ldots, d$, iar învăţarea lor se face prin maximizarea funcţiei de verosimilitate $\mathcal{L}(w) \stackrel{not.}{=} P(D|w)$, unde D este setul de date de antrenament. La rândul ei, maximizarea aceasta se realizează prin aplicarea unei metode de optimizare, de exemplu metoda gradientului ascendent sau metoda lui Newton. Vedeţi de exemplu problemele 13 şi 14 de la capitolul *Metode de regresie*.

[433] Vedeţi de exemplu problema 9.

În mod similar, presupoziţia de independenţă condiţională asumată de către algoritmul Bayes Naiv poate fi relaxată astfel încât trăsăturile X_1 şi X_2 să nu mai trebuiască să satisfacă independenţa condiţională. Aşadar, vom putea scrie:

$$P(Y|X) = \frac{P(Y)\,P(X_1, X_2|Y)\,\prod_{i=3}^{d} P(X_i|Y)}{P(X)}.$$

Demonstraţi că în acest caz distribuţia $P(Y|X)$ are aceeaşi formă ca şi modelul de regresie logistică augmentat cu un termen suplimentar, care exprimă dependenţa dintre X_1 şi X_2 (şi, în acest fel, modelul extins al regresiei logistice rămâne corespondentul discriminativ al clasificatorului nostru generativ).

***Indicaţii*:**
3. De data aceasta o altă notaţie simplă ne va ajuta. Vom avea nevoie de mai mulţi parametri decât la punctul a pentru a defini distribuţia comună $P(X_1, X_2|Y)$. Aşa că vom nota $\beta_{ijk} = P(X_1 = i, X_2 = j|Y = k)$, pentru fiecare combinaţie posibilă de valori pentru indicii i, j şi k.
4. Această nouă notaţie poate fi folosită acum pentru a exprima probabilitatea $P(X_1, X_2|Y = k)$ după cum urmează:

$$P(X_1, X_2|Y = k) = (\beta_{11k})^{X_1 X_2}\,(\beta_{10k})^{X_1(1-X_2)}\,(\beta_{01k})^{(1-X_1)X_2}\,(\beta_{00k})^{(1-X_1)(1-X_2)} \quad (231)$$

pentru $k \in \{0, 1\}$, cu excepţia următoarelor cazuri: *i.* $\beta_{11k} = 0$ şi $X_1X_2 = 0$, *ii.* $\beta_{10k} = 0$ şi $X_1(1 - X_2) = 0$, *iii.* $\beta_{01k} = 0$ şi $(1 - X_1)X_2 = 0$ şi *iv.* $\beta_{00k} = 0$ şi $(1 - X_1)(1 - X_2) = 0$.

Răspuns:

a. Mai întâi vom rescrie probabilitatea $P(Y = 1|X = x)$ ca o fracţie, folosind formula lui Bayes, compusă cu formula probabilităţii totale, apoi vom împărţi atât numărătorul cât şi numitorul fracţiei astfel obţinute cu expresia de la numărător:[434]

$$\begin{aligned} P(Y = 1|X = x) &\overset{FB}{=} \frac{P(X = x|Y = 1)\,P(Y = 1)}{\sum_{y' \in \{0,1\}} P(X = x|Y = y')\,P(Y = y')} \\ &= \frac{1}{1 + \dfrac{P(X = x|Y = 0)P(Y = 0)}{P(X = x|Y = 1)P(Y = 1)}}. \end{aligned}$$

După aceea, folosind formula $e^{\ln a} = a$ (valabilă pentru orice $a > 0$), vom forţa punerea fracţiei de mai sus într-o formă apropiată de cea a funcţiei sigmoidale ($\sigma(x) = 1/(1 + e^{-x})$):[435]

$$\begin{aligned} P(Y = 1|X = x) &= \frac{1}{1 + \exp\left(\ln \dfrac{P(X = x|Y = 0)P(Y = 0)}{P(X = x|Y = 1)P(Y = 1)}\right)} \\ &= \frac{1}{1 + \exp\left(\ln \dfrac{P(X_1 = x_1, \ldots, X_d = x_d|Y = 0)P(Y = 0)}{P(X_1 = x_1, \ldots, X_d = x_d|Y = 1)P(Y = 1)}\right)}. \end{aligned}$$

[434] Cu excepţia cazului când $P(X = x|Y = 1)\,P(Y = 1) = 0$.
[435] Cu excepţia cazului când $P(X = x|Y = 0)\,P(Y = 0) = 0$.

Mai departe, ţinând cont de presupoziţia de independenţă condiţională şi de proprietăţile funcţiei logaritm, obţinem:[436]

$$P(Y=1|X=x) = \frac{1}{1+\exp\left(\ln\frac{P(Y=0)}{P(Y=1)}+\sum_{i=1}^{d}\ln\frac{P(X_i=x_i|Y=0)}{P(X_i=x_i|Y=1)}\right)}.$$

Vom nota probabilităţile a priori $P(Y=1)$ şi $P(Y=0)$ cu π şi respectiv $1-\pi$. Apoi, conform *indicaţiei* 2, vom scrie $P(X_i|Y=1)$ ca $\theta_{i1}^{X_i}(1-\theta_{i1})^{(1-X_i)}$ şi $P(X_i|Y=0)$ ca $\theta_{i0}^{X_i}(1-\theta_{i0})^{(1-X_i)}$. În consecinţă,

$$\begin{aligned}
P(Y=1|X=x) &= \frac{1}{1+\exp\left(\ln\frac{1-\pi}{\pi}+\sum_{i=1}^{d}\ln\frac{\theta_{i0}^{X_i}(1-\theta_{i0})^{(1-X_i)}}{\theta_{i1}^{X_i}(1-\theta_{i1})^{(1-X_i)}}\right)} \\
&= \frac{1}{1+\exp\left(\ln\frac{1-\pi}{\pi}+\sum_{i=1}^{d}\left(X_i\ln\frac{\theta_{i0}}{\theta_{i1}}+(1-X_i)\ln\frac{1-\theta_{i0}}{1-\theta_{i1}}\right)\right)} \\
&= \frac{1}{1+\exp\left(\ln\frac{1-\pi}{\pi}+\sum_{i=1}^{d}\ln\frac{1-\theta_{i0}}{1-\theta_{i1}}+\sum_{i=1}^{d}X_i\left(\ln\frac{\theta_{i0}}{\theta_{i1}}-\ln\frac{1-\theta_{i0}}{1-\theta_{i1}}\right)\right)}.
\end{aligned}$$

Pentru a pune această ultimă expresie sub forma dorită, adică

$$P(Y=1|X=x) = 1/(1+\exp(w_0+\sum_{i=1}^{d}w_iX_i),$$

vom alege valorile parametrilor w_i în mod natural:

$$w_0 = \ln\frac{1-\pi}{\pi}+\sum_{i=1}^{d}\ln\frac{1-\theta_{i0}}{1-\theta_{i1}} \quad \text{şi} \quad w_i = \ln\frac{\theta_{i0}}{\theta_{i1}}-\ln\frac{1-\theta_{i0}}{1-\theta_{i1}} \text{ pentru } i=1,\ldots,d.$$

b. Vom începe ca şi la punctul precedent, prin a pune probabilitatea $P(Y=1|X=x)$ sub o formă apropiată de cea a funcţiei sigmoidale:[437]

$$\begin{aligned}
P(Y=1|X) &\overset{FB}{=} \frac{P(X|Y=1)P(Y=1)}{P(X|Y=1)P(Y=1)+P(X|Y=0)P(Y=0)} \\
&= \frac{1}{1+\frac{P(X|Y=0)P(Y=0)}{P(X|Y=1)P(Y=1)}} = \frac{1}{1+\exp\left(\ln\frac{P(X|Y=0)P(Y=0)}{P(X|Y=1)P(Y=1)}\right)}.
\end{aligned}$$

Până aici nu avem încă nicio diferenţă în raport cu calculul de la punctul a. Însă acum vom ţine cont că toate variabilele X_i ci $i=1,\ldots,d$ sunt independente condiţional două câte două în raport cu Y, cu excepţia perechii X_1, X_2:[438]

[436] Cu excepţia cazurilor când $P(X=x_i|Y=0)=0$ sau $P(X=x_i|Y=1)=0$ pentru $i=1,\ldots,d$.
[437] Cu excepţia cazurilor când $P(X|Y=1)P(Y=1)=0$ sau $P(X|Y=0)P(Y=0)=0$.
[438] Cu excepţia cazurilor când $P(Y=0)=0$ sau $P(Y=1)=0$, respectiv $P(X_1X_2|Y=0)=0$ sau $P(X_1X_2|Y=1)=0$ şi încă $P(X_i|Y=0)=0$ sau $P(X_i|Y=1)=0$ pentru $i=3,\ldots,d$.

$$P(Y=1|X) = \frac{1}{1+\exp\left(\ln\frac{P(X_1,X_2|Y=0)\ \prod_{i=3}^{d}P(X_i|Y=0)P(Y=0)}{P(X_1,X_2|Y=1)\ \prod_{i=3}^{d}P(X_i|Y=1)P(Y=1)}\right)}$$
$$= \frac{1}{1+\exp\left(\ln\frac{1-\pi}{\pi}+\sum_{i=3}^{d}\ln\frac{P(X_i|Y=0)}{P(X_i|Y=1)}+\ln\frac{P(X_1,X_2|Y=0)}{P(X_1,X_2|Y=1)}\right)}.$$

Ca şi la punctul a, vom ţine cont că

$$\ln\frac{P(X_i|Y=0)}{P(X_i|Y=1)} = \ln\frac{\theta_{i0}^{X_i}(1-\theta_{i0})^{(1-X_i)}}{\theta_{i1}^{X_i}(1-\theta_{i1})^{(1-X_i)}}$$

atunci când condiţiile asociate cu relaţia (230) sunt satisfăcute. Mai departe, folosind *indicaţia* 4, vom putea înlocui $P(X_1,X_2|Y=0)$ şi $P(X_1,X_2|Y=1)$ în funcţie de β_{ijk}, obţinând (atunci când condiţiile asociate cu relaţia (231) sunt satisfăcute):

$$\ln\frac{P(X_1,X_2|Y=0)}{P(X_1,X_2|Y=1)} = \ln\frac{(\beta_{110})^{X_1X_2}\,(\beta_{100})^{X_1(1-X_2)}\,(\beta_{010})^{(1-X_1)X_2}\,(\beta_{000})^{(1-X_1)(1-X_2)}}{(\beta_{111})^{X_1X_2}\,(\beta_{101})^{X_1(1-X_2)}\,(\beta_{011})^{(1-X_1)X_2}\,(\beta_{001})^{(1-X_1)(1-X_2)}}.$$

Aşadar, va rezulta:

$$P(Y=1|X) = \frac{1}{1+\exp\left(w_0+\sum_{i=3}^{d}w_iX_i+w_1X_1+w_2X_2+w_{1,2}X_1X_2\right)},$$

unde

$$\begin{aligned}
w_0 &= \ln\frac{1-\pi}{\pi}+\sum_{i=3}^{d}\ln\frac{1-\theta_{i1}}{1-\theta_{i0}}+\ln\frac{\beta_{000}}{\beta_{001}}\\
w_1 &= \ln\frac{\beta_{100}}{\beta_{101}}+\ln\frac{\beta_{001}}{\beta_{000}}\\
w_2 &= \ln\frac{\beta_{010}}{\beta_{011}}+\ln\frac{\beta_{001}}{\beta_{000}}\\
w_{1,2} &= \ln\frac{\beta_{110}}{\beta_{111}}+\ln\frac{\beta_{101}}{\beta_{100}}+\ln\frac{\beta_{011}}{\beta_{010}}+\ln\frac{\beta_{000}}{\beta_{001}}\\
w_i &= \ln\frac{\theta_{i0}}{\theta_{i1}}+\ln\frac{1-\theta_{i1}}{1-\theta_{i0}}\ \text{pentru } i=3,\ldots,d.
\end{aligned}$$

2.1.3 Clasificare bayesiană [cu atribute de intrare] de tip gaussian

15. **(Algoritmul Bayes [Naiv] gaussian: aplicare pe date din $\mathbb{R}$)**

prelucrare de Liviu Ciortuz, după
■ • ***CMU, 2001 fall, Andrew Moore, midterm, pr. 3.a***

X	Y
0	A
2	A
3	B
4	B
5	B
6	B
7	B

Presupunem că dispunem de setul de date de antrenament din tabelul alăturat; singurul atribut de intrare (X) ia valori reale, iar atributul de ieşire (Y) este de tip Bernoulli, deci ia două valori, notate cu A şi respectiv B.

a. Pornind de la acest set de date, va trebui mai întâi să învăţaţi *parametrii* clasificatorului Bayes gaussian, prin metoda estimării de verosimilitate maximă (MLE).[439] Centralizaţi rezultatele, completând tabelul următor:

$\mu_A =$	$\sigma_A^2 =$	$P(Y = A) =$
$\mu_B =$	$\sigma_B^2 =$	$P(Y = B) =$

b. Notăm $\alpha = p(X = 2|Y = A)$ şi $\beta = p(X = 2|Y = B)$.

- **Cât este $p(X = 2, Y = A)$ în funcţie de α?**
- **Cât este $p(X = 2, Y = B)$ în funcţie de β?**
- **Cât este $p(X = 2)$ în funcţie de α şi β?**
- **Cât este $p(Y = A|X = 2)$ în funcţie de α şi β?**

c. Cum va clasifica algoritmul Bayes [Naiv] gaussian punctul $X = 2$? Puteţi exprima răspunsul fie în funcţie de α şi β, fie — mai bine! — calculând în prealabil valorile lui α şi β în funcţie de parametrii calculaţi / estimaţi la punctul precedent.

Răspuns:

a. Pentru a estima mediile μ_A şi μ_B, vom folosi formula demonstrată la problema 50.a de la capitolul de *Fundamente*:

$$\mu_{MLE} = \frac{\sum_{i=1}^{n} x_i}{n},$$

unde n este numărul instanţelor de antrenament. Aşadar, $\mu_A = \frac{\sum_{i=1}^{2} X_i}{2} = \frac{0+2}{2} = 1$, iar $\mu_B = \frac{\sum_{i=3}^{7} X_i}{5} = \frac{3+4+5+6+7}{5} = 5$.

[439] Vedeţi secţiunea corespunzătoare din capitolul de *Fundamente*, în speţă (pentru acest caz) problemele 50.a şi 51.a.

Similar, pentru calculul varianţelor σ_A şi σ_B, vom folosi formula care a fost demonstrată la problema 51.a de la capitolul de *Fundamente*:

$$\sigma^2_{MLE} = \frac{\sum_{i=1}^{n}(x_i - \mu_{MLE})^2}{n}.$$

Aşadar, $\sigma^2_A = \frac{1}{2}[(0-1)^2 + (2-1)^2] = 1$**, iar** $\sigma^2_B = \frac{1}{5}[(3-5)^2 + (4-5)^2 + 0^2 + (6-5)^2 + (7-5)^2] = \frac{1}{5} \cdot 2 \cdot [4+1] = 2$**.**

Pentru calculul probabilităţilor $P(Y = A)$ **şi** $P(Y = B)$ **se foloseşte formula clasică (numărul de cazuri favorabile împărţit la numărul de cazuri posibile; vedeţi fie problema 43.c fie problema 125.a, ambele de la capitolul de *Fundamente*), fiindcă** Y **este variabilă de tip Bernoulli. Aşadar,** $P(Y = A) = 2/7$ **şi** $P(Y = B) = 5/7$**.**

Centralizând aceste estimări, obţinem:

$\mu_A = 1$	$\sigma^2_A = 1$	$P(Y = A) = 2/7$
$\mu_B = 5$	$\sigma^2_B = 2$	$P(Y = B) = 5/7$

b. Folosind regula de înmulţire a probabilităţilor, calculăm $p(X = 2, Y = A) = p(X = 2|Y = A) \cdot P(Y = A) = \frac{2\alpha}{7}$ **şi** $p(X = 2, Y = B) = p(X = 2|Y = B) \cdot P(Y = B) = \frac{5\beta}{7}$**.**

Probabilitatea $p(X = 2)$ **se poate obţine aplicând formula probabilităţii totale:** $p(X = 2) = p(X = 2|Y = A) \cdot P(Y = A) + p(X = 2|Y = B) \cdot P(Y = B) = \frac{1}{7}(2\alpha + 5\beta)$**.**

Probabilitatea condiţionată $p(Y = A|X = 2)$ **se calculează folosind definiţia:** $p(Y = A|X = 2) = \frac{p(Y = A, X = 2)}{p(X = 2)} = \frac{2\alpha}{2\alpha + 5\beta}$**.**

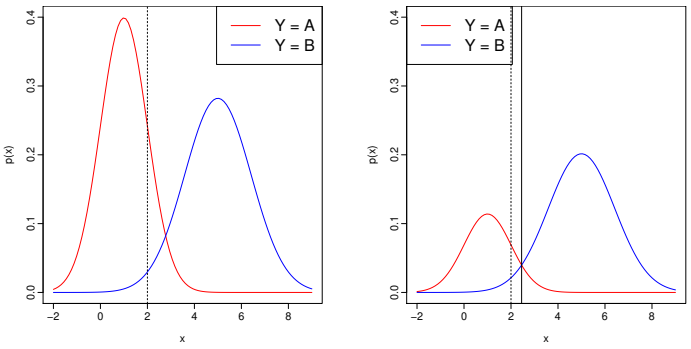
c. Algoritmul Bayes [Naiv] gaussian va asocia punctului $X = 2$ **eticheta** $Y = A$ **dacă** $p(Y = A|X = 2) \geq p(Y = B|X = 2) \Leftrightarrow \frac{2}{7}\alpha \geq \frac{5}{7}\beta$**.**

Folosind valorile estimate pentru parametrii μ_A**,** μ_B**,** σ_A **şi** σ_B **la punctul** a**, vom putea scrie:** $\alpha = \frac{1}{\sqrt{2\pi}} e^{-\frac{(2-1)^2}{2}} = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}} = 0.24197$ **şi** $\beta = \frac{1}{\sqrt{2\pi} \cdot \sqrt{2}} e^{-\frac{(2-5)^2}{2 \cdot 2}} = \frac{1}{2\sqrt{\pi}} e^{-\frac{9}{4}} = 0.029732$**. Deci,**

$$\frac{2}{7}\alpha \geq \frac{5}{7}\beta \Leftrightarrow \frac{2}{7} \cdot 0.24197 \geq \frac{5}{7} \cdot 0.029732 \Leftrightarrow 0.06913 \geq 0.02123 \text{ (adevărat).}$$

Prin urmare, algoritmul Bayes [Naiv] gaussian va asocia punctului $X = 2$ **eticheta** $Y = A$**, cu probabilitatea** $\frac{0.06913}{0.06913 + 0.02123} = 0.76499$**.**

***Observaţie*: În cele două grafice alăturate am reprezentat $\mathcal{N}(\mu_A, \sigma_A^2)$ şi $\mathcal{N}(\mu_B, \sigma_B^2)$, p.d.f.-urile gaussienelor asociate claselor A şi respectiv B (în partea stângă), precum şi funcţiile obţinute din aceste două p.d.f.-uri prin înmulţirea cu *factorii de selecţie* $2/7$ şi respectiv $5/7$ (în partea dreaptă).**



Se poate constata relativ uşor că există două puncte de intersecţie ($x_1 = -8.451$ şi $x_2 = 2.451$) pentru graficele funcţiilor $\frac{2}{7}\mathcal{N}(\mu_A, \sigma_A^2)$ şi $\frac{5}{7}\mathcal{N}(\mu_B, \sigma_B^2)$. Toate instanţele de test x situate între aceste puncte de intersecţie ($x_1 < x < x_2$) vor aparţine clasei A (acolo curba roşie este situată deasupra celei albastre). Instanţele situate fie la stânga lui x_1 fie la dreapta lui x_2 vor aparţine clasei B (acolo curba albastră este situată deasupra celei roşii). *Separatorul decizional* este de tip pătratic, fiind constituit din punctele x_1 şi x_2.[440]

16. **(Clasificatorul Bayes [Naiv] gaussian, cazul când se foloseşte un singur atribut de intrare: zone de decizie şi graniţe de decizie; analiza diferitelor *cazuri specifice*)**

□ ***Sergiu Dinu, Liviu Ciortuz, 2020***

În acest exerciţiu ne propunem să identificăm *zonele de decizie* şi *graniţele de decizie* determinate de clasificatorul Bayes [Naiv] gaussian (abreviat *GB*), atunci când folosim un singur atribut de intrare X, iar atributul de ieşire, pe care îl vom nota cu Y, este binar, deci poate lua două valori, desemnate în continuare cu A şi B. Vom desemna prin $p_A = p$ *probabilitatea de selecţie* pentru clasa A. Prin urmare, probabilitatea de selecţie pentru clasa B este $p_B = 1 - p$. Vom presupune că $p \in (0, 1)$ şi $X|(Y = A) \sim \mathcal{N}(x|\mu_A, \sigma_A^2)$, iar $X|(Y = B) \sim \mathcal{N}(x|\mu_B, \sigma_B^2)$.

a. Arătaţi că *regula de decizie* a acestui clasificator Bayes [Naiv] gaussian

$$\hat{Y}_{GB}(X = x) = A \Leftrightarrow p \cdot \mathcal{N}(x|\mu_A, \sigma_A^2) \geq (1 - p) \cdot \mathcal{N}(x|\mu_B, \sigma_B^2) \tag{232}$$

devine echivalentă cu

$$\left(\sigma_A^2 - \sigma_B^2\right)(x - x_1)(x - x_2) \geq 0,$$

unde

$$x_1 = \frac{\sigma_A^2\mu_B - \sigma_B^2\mu_A - \sqrt{\Delta'}}{\sigma_A^2 - \sigma_B^2} \quad \text{şi} \quad x_2 = \frac{\sigma_A^2\mu_B - \sigma_B^2\mu_A + \sqrt{\Delta'}}{\sigma_A^2 - \sigma_B^2},$$

[440] LC: Mulţumesc studentului MSc Sergiu Dinu pentru această observaţie.

cu

$$\Delta' \stackrel{not.}{=} \sigma_A^2\sigma_B^2\left[(\mu_A-\mu_B)^2+(\sigma_A^2-\sigma_B^2)\ln\left(\frac{1-p}{p}\cdot\frac{\sigma_A}{\sigma_B}\right)^2\right],$$

în *condiţiile* în care $\sigma_A^2 \neq \sigma_B^2$ şi $\Delta' \geq 0$.

b. Arătaţi că atunci când $\sigma_A^2 = \sigma_B^2 \stackrel{not.}{=} \sigma^2$ şi $\mu_A \neq \mu_B$, regula de decizie (232) **este echivalentă cu**

$$x \geq x_0 \text{ dacă } \mu_A > \mu_B$$

şi, respectiv

$$x \leq x_0 \text{ dacă } \mu_A < \mu_B,$$

unde

$$x_0 = \frac{\mu_A+\mu_B}{2} + \frac{\sigma^2}{\mu_A-\mu_B}\cdot\ln\frac{1-p}{p}.$$

c. Arătaţi că este posibil ca în anumite cazuri să avem $\Delta' < 0$, adică inegalitatea (232) **să fie ori adevărată pentru orice $x \in \mathbb{R}$, ori falsă pentru orice $x \in \mathbb{R}$. Altfel spus, arătaţi că pot exista combinaţii de valori pentru parametrii σ_A şi σ_B (ambii strict pozitivi), μ_A şi μ_B din $\mathbb{R}$, precum şi $p \in (0,1)$, astfel încât**

$$(\mu_A-\mu_B)^2+(\sigma_A^2-\sigma_B^2)\ln\left(\frac{1-p}{p}\cdot\frac{\sigma_A}{\sigma_B}\right)^2 < 0. \tag{233}$$

Răspuns:

a. Putem scrie următoarea succesiune de echivalenţe:

$$\begin{gathered}
p\cdot\mathcal{N}(x|\mu_A,\sigma_A^2) \geq (1-p)\cdot\mathcal{N}(\mu_B,\sigma_B^2) \Leftrightarrow \\
p\cdot\frac{1}{\sqrt{2\pi}\sigma_A}\exp\left(-\frac{(x-\mu_A)^2}{2\sigma_A^2}\right) \geq (1-p)\cdot\frac{1}{\sqrt{2\pi}\sigma_B}\exp\left(-\frac{(x-\mu_B)^2}{2\sigma_B^2}\right) \Leftrightarrow \\
\exp\left(\frac{1}{2}\left[\left(\frac{x-\mu_B}{\sigma_B}\right)^2-\left(\frac{x-\mu_A}{\sigma_A}\right)^2\right]\right) \geq \frac{1-p}{p}\cdot\frac{\sigma_A}{\sigma_B} \Leftrightarrow \\
\left(\frac{x-\mu_B}{\sigma_B}\right)^2-\left(\frac{x-\mu_A}{\sigma_A}\right)^2 \geq \ln\left(\frac{1-p}{p}\cdot\frac{\sigma_A}{\sigma_B}\right)^2 \Leftrightarrow \\
(\sigma_A^2-\sigma_B^2)x^2+2(\sigma_B^2\mu_A-\sigma_A^2\mu_B)x+\sigma_A^2\mu_B^2-\sigma_B^2\mu_A^2-\sigma_A^2\sigma_B^2\ln\left(\frac{1-p}{p}\cdot\frac{\sigma_A}{\sigma_B}\right)^2 \geq 0.
\end{gathered} \tag{234}$$

Membrul stâng al acestei ultime inegalităţi este o funcţie polinomială de gradul al doilea. Prin urmare, discriminantul ei „pe jumătate“ (adică, $\Delta' \stackrel{not.}{=} \Delta/2$) este:

$$\Delta' = (\sigma_B^2\mu_A-\sigma_A^2\mu_B)^2-(\sigma_A^2-\sigma_B^2)\left[\sigma_A^2\mu_B^2-\sigma_B^2\mu_A^2-\sigma_A^2\sigma_B^2\ln\left(\frac{1-p}{p}\cdot\frac{\sigma_A}{\sigma_B}\right)^2\right]$$

Însă,

$$\begin{aligned}
&(\sigma_B^2\mu_A-\sigma_A^2\mu_B)^2-(\sigma_A^2-\sigma_B^2)(\sigma_A^2\mu_B^2-\sigma_B^2\mu_A^2) \\
&= \cancel{\sigma_B^4\mu_A^2}+\cancel{\sigma_A^4\mu_B^2}-2\sigma_A^2\sigma_B^2\mu_A\mu_B-\cancel{\sigma_A^4\mu_B^2}+\sigma_A^2\sigma_B^2\mu_A^2+\sigma_A^2\sigma_B^2\mu_B^2-\cancel{\sigma_B^4\mu_A^2} \\
&= \sigma_A^2\sigma_B^2(\mu_A-\mu_B)^2.
\end{aligned}$$

Aşadar,

$$\Delta' = \sigma_A^2\sigma_B^2\left[(\mu_A-\mu_B)^2 + (\sigma_A^2-\sigma_B^2)\ln\left(\frac{1-p}{p}\cdot\frac{\sigma_A}{\sigma_B}\right)^2\right],$$

ceea ce ne conduce imediat la concluzia din enunţ.

***Consecinţă*: În acest caz (a), separatorul decizional este format din punctele x_1 şi x_2. Dacă $\sigma_A^2 > \sigma_B^2$, atunci zona de decizie corespunzătoare clasei A este $(-\infty, x_1]\cup[x_2, +\infty)$, iar zona de decizie corespunzătoare clasei B este intervalul (x_1, x_2). Similar, dacă $\sigma_A^2 < \sigma_B^2$, atunci zona de decizie corespunzătoare clasei A este (x_1, x_2), iar zona de decizie corespunzătoare clasei B este intervalul $(-\infty, x_1]\cup[x_2, +\infty)$.**

b. Dacă $\sigma_A^2 = \sigma_B^2 \overset{not.}{=} \sigma^2$, atunci este uşor de observat că inegalitatea (234) devine

$$2(\mu_A-\mu_B)x + \mu_B^2 - \mu_A^2 - \sigma^2\ln\left(\frac{1-p}{p}\right)^2 \geq 0,$$

ceea ce fundamentează concluzia din enunţ.

***Consecinţă*: În acest caz (b), separatorul decizional este punctul x_0. Zonele de decizie se stabilesc imediat conform relaţiilor din enunţ.**

c. Observăm că dacă vom considera cazul particular $\mu_A = \mu_B$, relaţia (233) va căpăta o formă mai simplă:

$$(\sigma_A^2-\sigma_B^2)\ln\left(\frac{1-p}{p}\cdot\frac{\sigma_A}{\sigma_B}\right) < 0. \tag{235}$$

Această inegalitate este satisfăcută, de pildă atunci când σ_A şi σ_B (care se consideră, prin convenţie, strict pozitive) sunt în relaţia $\sigma_A > \sigma_B$, iar $\frac{1-p}{p} < \frac{\sigma_B}{\sigma_A}$. Se poate constata imediat că a doua inegalitate este echivalentă cu $p > \frac{\sigma_A}{\sigma_A+\sigma_B}$. Se observă că această ultimă fracţie are valori în intervalul $(1/2, 1)$, fiindcă $\sigma_A > \sigma_B$.

Similar, dacă $\sigma_A < \sigma_B$, atunci pentru a satisface inegalitatea (235) se impune condiţia $\frac{1-p}{p} > \frac{\sigma_B}{\sigma_A}$, care revine la $p < \frac{\sigma_A}{\sigma_A+\sigma_B}$. Această ultimă fracţie ia valori în intervalul $(0, 1/2)$, întrucât $\sigma_A < \sigma_B$.

***Consecinţă*: În acest caz (c), nu există separator decizional. Una din cele două zone de decizie este vidă, iar cealaltă zonă coincide cu mulţimea tuturor numerelor reale, $\mathbb{R}$.**

17. **(Algoritmul Bayes Naiv gaussian: deducerea [formei liniare a] regulii de decizie în cazul matricelor de covarianţă diagonale şi identice, i.e., $\sigma_{i0} = \sigma_{i1}$, pentru $i = 1, \ldots, d$)**

■ ● ○ ***CMU, 2009 spring, Ziv Bar-Joseph, HW2, pr. 2***

Considerăm un model de tip Bayes Naiv cu două clase ($Y \in \{0, 1\}$), definit peste spaţiul real $\mathbb{R}^d$ al atributelor de intrare $X_1, \ldots, X_d$. Presupunem că în acest

model distribuţia [comună] condiţionată $X|Y = 0$, unde $X = (X_1, \ldots, X_d) \in \mathbb{R}^d$, poate fi definită ca un vector de distribuţii gaussiene unidimensionale independente şi-l vom desemna prin notaţia

$$Gaussian(\mu_0 =< \mu_{10}, \ldots, \mu_{d0} >,\ \sigma =< \sigma_1, \ldots, \sigma_d >)$$

şi analog pentru $X|Y = 1$:

$$Gaussian(\mu_1 =< \mu_{11}, \ldots, \mu_{d1} >,\ \sigma =< \sigma_1, \ldots, \sigma_d >).$$

Observaţi că intrările $X_1, \ldots, X_d$ au — la condiţionare în raport cu clasa — medii diferite dar varianţe (de fapt, matrice de covarianţă, diagonale) identice pentru ambele clase.

În acest exerciţiu vă vom arăta că, în modelul specificat mai sus, probabilitatea condiţionată $P(Y = 1|X = x)$, unde $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$, se poate rescrie ca valoare a unei funcţii sigmoidale / „logistice“ de o formă particulară, $\frac{1}{1 + e^{-(w_0 + w \cdot x)}}$, cu parametrii $w_0 \in \mathbb{R}$ şi $w = (w_1, \ldots, w_d) \in \mathbb{R}^d$ aleşi în mod convenabil.[441]

a. Folosiţi regula lui Bayes (compusă cu formula probabilităţii totale) pentru a rescrie $P(Y = 1|X = x)$ sub forma unei fracţii. Împărţiţi atât numărătorul cât şi numitorul fracţiei astfel obţinute cu expresia de la numărător.

b. La punctul a ar fi trebuit să ajungeţi la un rezultat de forma

$$\frac{1}{1 + f(x, X, Y)},$$

unde f este o anumită funcţie având argumentele x, X şi Y. Folosind formula $e^{\ln a} = a$ (valabilă pentru orice $a > 0$), putem forţa punerea acestei fracţii într-o formă apropiată de funcţia sigmoidală:

$$\frac{1}{1 + e^{\ln f(x, X, Y)}}.$$

Vă cerem să rescrieţi sub o astfel de formă rezultatul de la punctul a.

c. Explicitaţi presupoziţia Bayes „naivă“ în cadrul modelului probabilist dat. Apoi folosiţi-o pentru a converti exponentul care apare în fracţia rezultată la punctul b la o sumă de forma

$$\ln g(Y) + \sum_{i=1}^{d} \ln h(x_i, X_i, Y).$$

Precizaţi expresiile funcţiilor g şi h.

d. Acum folosiţi specificul modelului dat — şi anume, de tip gaussian, având pentru componentele condiţionate (şi anume, variabilele aleatoare condiţionate $X_i|Y = 1$, respectiv $X_i|Y = 0$, pentru $i = 1, \ldots, n$) medii diferite dar varianţe egale —, pentru a aduce expresia de la punctul c la o formă mai convenabilă.

[441] În consecinţă, [se poate arăta imediat că] regula de decizie a clasificatorului Bayes Naiv pentru acest model este de tip *liniar*.

e. Rescriind $P(Y = 1|X = x)$ conform expresiilor obţinute la punctele b şi d, rezultatul ar trebui să semene cu un alt model pe care l-am întâlnit la curs. Care anume? Exprimaţi *parametrii* acelui model în raport cu $P(Y = 1)$, μ_{i0}, μ_{i1} şi σ_i, cu $i = 1, \ldots, d$.

Răspuns:

a. Procedând conform cerinţelor, vom scrie:

$$\begin{aligned} P(Y = 1|X = x) &= \frac{P(X = x|Y = 1)\, P(Y = 1)}{\sum_{y\in\{0,1\}} P(X = x|Y = y)\, P(Y = y)} \\ &= \frac{1}{1 + \dfrac{P(X = x|Y = 0)P(Y = 0)}{P(X = x|Y = 1)P(Y = 1)}}. \end{aligned}$$

Considerând $f(x, X, Y) = \dfrac{P(X = x|Y = 0)P(Y = 0)}{P(X = x|Y = 1)P(Y = 1)}$, se observă că ultima expresie obţinută are într-adevăr forma $\dfrac{1}{1 + f(x, X, Y)}$.

b. Folosind formula $a = e^{\ln a}$, obţinem:

$$P(Y = 1|X = x) = \frac{1}{1 + \exp\left(\ln \dfrac{P(X = x|Y = 0)P(Y = 0)}{P(X = x|Y = 1)P(Y = 1)}\right)}.$$

c. Conform presupoziţiei specifice algoritmului Naive Bayes, vom considera că atributele X_i sunt independente condiţional două câte două în raport cu variabila de ieşire. Aşadar, vom avea egalităţile următoare:

$$\begin{aligned} P(X = x|Y = 1) &= \prod_{i=1}^{d} P(X_i = x_i|Y = 1) \\ P(X = x|Y = 0) &= \prod_{i=1}^{d} P(X_i = x_i|Y = 0) \end{aligned}$$

Prin urmare, vom scrie argumentul funcţiei $\exp(\ )$ **din fracţia de la punctul b astfel:**

$$\ln \frac{P(X = x|Y = 0)P(Y = 0)}{P(X = x|Y = 1)P(Y = 1)} = \ln \frac{P(Y = 0)}{P(Y = 1)} + \sum_{i=1}^{d} \ln \frac{P(X_i = x_i|Y = 0)}{P(X_i = x_i|Y = 1)}.$$

În consecinţă, funcţiile g şi h care au fost cerute în enunţ vor fi:

$$g(Y) = \frac{P(Y = 0)}{P(Y = 1)} \qquad h(x_i, X_i, Y) = \frac{P(X_i = x_i|Y = 0)}{P(X_i = x_i|Y = 1)} \text{ pentru } i = 1, \ldots, d.$$

d. Ţinând cont de specificul gaussian al modelului din enunţ, vom putea scrie rezultatul de la punctul c astfel:

$$\begin{aligned} &\ln \frac{P(X = x|Y = 0)P(Y = 0)}{P(X = x|Y = 1)P(Y = 1)} \\ &= \ln \frac{P(Y = 0)}{P(Y = 1)} + \sum_{i=1}^{d} \ln \left(\frac{\dfrac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\dfrac{(x_i - \mu_{i0})^2}{2\sigma_i^2}\right)}{\dfrac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\dfrac{(x_i - \mu_{i1})^2}{2\sigma_i^2}\right)} \right) \end{aligned}$$

$$
\begin{aligned}
&= \ln\frac{P(Y=0)}{P(Y=1)} + \sum_{i=1}^{d}\left(\frac{(x_i-\mu_{i1})^2}{2\sigma_i^2} - \frac{(x_i-\mu_{i0})^2}{2\sigma_i^2}\right)\\
&= \ln\frac{P(Y=0)}{P(Y=1)} + \sum_{i=1}^{d}\frac{2x_i(\mu_{i0}-\mu_{i1}) + (\mu_{i1}^2-\mu_{i0}^2)}{2\sigma_i^2}\\
&= \ln\frac{P(Y=0)}{P(Y=1)} + \sum_{i=1}^{d}\left(\frac{x_i(\mu_{i0}-\mu_{i1})}{\sigma_i^2} + \frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2}\right)\\
&= \ln\frac{P(Y=0)}{P(Y=1)} + \sum_{i=1}^{d}\frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2} + \sum_{i=1}^{d}\frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2}x_i.
\end{aligned}
$$

e. Notând în expresia obţinută la punctul d

$$w_0 = \ln\frac{P(Y=0)}{P(Y=1)} + \sum_{i=1}^{d}\frac{\mu_{i1}^2-\mu_{i0}^2}{2\sigma_i^2} \quad \textbf{şi} \quad w_i = \frac{\mu_{i0}-\mu_{i1}}{\sigma_i^2} \ \textbf{pentru}\ i = 1,\ldots,d$$

şi apoi revenind la expresia de la punctul b, vom putea scrie:

$$P(Y=1|X=x) = \frac{1}{1+\exp(w_o + \sum_{i=1}^{d} w_i x_i)}.$$

Aşadar, expresia probabilităţii a posteriori este exact cea a funcţiei sigmoidale, $\frac{1}{1+e^{-z}}$. Rezultatul seamănă foarte bine cu modelul de *regresie logistică*.[442]

18. **(Algoritmul Bayes Optimal gaussian: raportul faţă de regresia logistică în cazul $\Sigma_0 = \Sigma_1$)**

■ □ • ○ ***CMU, 2011 spring, Tom Mitchell, HW2, pr. 2.2***

În cele ce urmează, vom considera:
1. Y, o variabilă booleană care urmează o distribuţie de tip Bernoulli, cu parametrul $\pi = P(Y=1)$, ceea ce implică $P(Y=0) = 1-\pi$;
2. $X = (X_1, X_2, \ldots, X_d)^\top$, un vector de variabile aleatoare care *nu* sunt independente condiţional în raport cu variabila Y, probabilitatea [comună şi] condiţionată $P(X|Y=k)$ urmând o *distribuţie gaussiană multidimensională*, $\mathcal{N}(\mu_k, \Sigma)$, unde $k \in \{0,1\}$.

Reţineţi faptul că μ_k, media acestei distribuţii multidimensionale, este un vector-coloană (altfel spus, o matrice $d \times 1$) şi există două astfel de medii, câte una pentru fiecare dintre cele două valori ale variabilei Y. De asemenea, Σ, matricea de covarianţă are dimensiunea $d \times d$, iar ea nu depinde de valorile variabilei Y.[443]

În rezolvarea problemei, veţi folosi funcţia de densitate [de probabilitate] a distribuţiei gaussiene multidimensionale în notaţie matriceală:[444]

$$\mathcal{N}(x;\mu,\Sigma) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}\exp\left(-\frac{1}{2}(x-\mu)^\top\Sigma^{-1}(x-\mu)\right),$$

[442] Pentru o introducere la modelul regresiei logistice, vedeţi problema 13 de la capitolul *Metode de regresie*.
[443] Altfel spus, presupunând că Σ_k, pentru $k \in \{0,1\}$, desemnează matricea de covarianţă a distribuţiei gaussiene multidimensionale $\mathcal{N}(\mu_k, \Sigma_k)$, atunci considerăm că $\Sigma_0 = \Sigma_1$.
[444] Vedeţi problema 37 de la capitolul de *Fundamente*.

unde simbolul $\top$ desemnează operaţia de transpunere a matricelor.

Răspundeţi la următoarea *întrebare*:
Este oare distribuţia $P(Y|X)$ corespunzătoare acestui clasificator de tip Bayes Optimal gaussian (nu naiv!) de aceeaşi formă cu cea a *regresiei logistice*?

***Sugestie*: Rafinaţi expresia distribuţiei probabiliste $P(Y|X)$.**

***Observaţie*: La problema 17 am arătat că în cazul (particular!) în care intrările X_1, X_2, ..., X_d sunt independente condiţional în raport cu ieşirea Y — ceea ce, conform problemei 34 de la capitolul de *Fundamente*, este echivalent cu a spune că matricea Σ este diagonală —, răspunsul la întrebarea pusă în enunţ este pozitiv.**

Răspuns:

Vom demara calculele în maniera standard (adică, similar cu prima parte a rezolvării problemelor 14 şi 17):

$$\begin{aligned} P(Y=1|X) &= \frac{P(X|Y=1)\,P(Y=1)}{P(X|Y=1)\,P(Y=1)+P(X|Y=0)\,P(Y=0)} \\ &= \frac{1}{1+\dfrac{P(Y=0)\,P(X|Y=0)}{P(Y=1)\,P(X|Y=1)}} = \frac{1}{1+\exp\left(\ln\dfrac{P(Y=0)\,P(X|Y=0)}{P(Y=1)\,P(X|Y=1)}\right)} \\ &= \frac{1}{1+\exp\left(\ln\dfrac{P(Y=0)}{P(Y=1)}+\ln\dfrac{P(X|Y=0)}{P(X|Y=1)}\right)}. \end{aligned}$$

Acum ne vom concentra atenţia asupra termenului $\ln\dfrac{P(X|Y=0)}{P(X|Y=1)}$, ţinând cont de faptul că $X|Y=0\sim\mathcal{N}(\mu_0,\Sigma)$ şi $X|Y=1\sim\mathcal{N}(\mu_1,\Sigma)$:

$$\begin{aligned} &\ln\frac{P(X|Y=0)}{P(X|Y=1)} \\ &= \ln\frac{\dfrac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}}{\dfrac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}}}+\ln\exp\left(\frac{1}{2}\left[(X-\mu_1)^\top\Sigma^{-1}(X-\mu_1)-(X-\mu_0)^\top\Sigma^{-1}(X-\mu_0)\right]\right) \\ &= \frac{1}{2}\left[(X-\mu_1)^\top\Sigma^{-1}(X-\mu_1)-(X-\mu_0)^\top\Sigma^{-1}(X-\mu_0)\right] \\ &= \frac{1}{2}\left[-X^\top\Sigma^{-1}\mu_1-\mu_1^\top\Sigma^{-1}X+\mu_1^\top\Sigma^{-1}\mu_1+X^\top\Sigma^{-1}\mu_0+\mu_0^\top\Sigma^{-1}X-\mu_0^\top\Sigma^{-1}\mu_0\right] \\ &= \frac{1}{2}\left[\mu_1^\top\Sigma^{-1}\mu_1-\mu_0^\top\Sigma^{-1}\mu_0+X^\top\Sigma^{-1}(\mu_0-\mu_1)+(\mu_0^\top-\mu_1^\top)\Sigma^{-1}X\right] \\ &= \frac{1}{2}\mu_1^\top\Sigma^{-1}\mu_1-\frac{1}{2}\mu_0^\top\Sigma^{-1}\mu_0+(\mu_0-\mu_1)^\top\Sigma^{-1}X. \end{aligned}$$

Remarcaţi faptul că $((\mu_0^\top-\mu_1^\top)\Sigma^{-1}X)^\top=((\mu_0-\mu_1)^\top\Sigma^{-1}X)^\top=X^\top(\Sigma^{-1})^\top(\mu_0-\mu_1)=X^\top(\Sigma^\top)^{-1}(\mu_0-\mu_1)=X^\top\Sigma^{-1}(\mu_0-\mu_1)$, întrucât matricea Σ^{-1} este simetrică (ca urmare a faptului că Σ însăşi, ca matrice de covarianţă, este simetrică; vedeţi problema 20 de la capitolul de *Fundamente*).

Prin urmare,

$$\begin{aligned} P(Y=1|X) &= \frac{1}{1+\exp\Big(\ln\frac{1-\pi}{\pi}+\frac{1}{2}\mu_1^\top\Sigma^{-1}\mu_1-\frac{1}{2}\mu_0^\top\Sigma^{-1}\mu_0+(\mu_0-\mu_1)^\top\Sigma^{-1}X\Big)} \\ &= \frac{1}{1+\exp(w_0+w^\top X)}, \end{aligned}$$

cu $w_0=\ln\frac{1-\pi}{\pi}+\frac{1}{2}\mu_1^\top\Sigma^{-1}\mu_1-\frac{1}{2}\mu_0^\top\Sigma^{-1}\mu_0$ **şi** $w=\Sigma^{-1}(\mu_0-\mu_1)$**. Evident,** w_0 **este un număr real (constant), iar** w **un vector-coloană (mai precis, o matrice de dimensiune** $d\times 1$**).**

În concluzie, distribuţia probabilistă $P(Y|X)$ **are (şi în acest caz!) aceeaşi formă cu cea din modelul regresiei logistice.**

19. **(Clasificatorul Bayes Optimal Gaussian: natura separatorului decizional în cazul general (în speţă** $\Sigma_0\neq\Sigma_1$**))**

■ ● ○ ***Stanford, 2014 fall, Andrew Ng, midterm, pr. 2.b***

Fie setul de date de antrenament $\{(x_1,y_1),\ldots,(x_n,y_n)\}$**, cu** x_i **vectori-coloană din** $\mathbb{R}^d$ **şi** $y_i\in\{0,1\}$ **pentru orice** i**. Algoritmul Bayes Optimal gaussian estimează următorii parametri:** $\phi\in(0,1)$**, vectorii-coloană** μ_0 **şi** μ_1 **din** $\mathbb{R}^d$**, precum şi matricele de covarianţă** Σ_0 **şi** Σ_1 **de dimensiune** $d\times d$**, corespunzători următoarelor distribuţii:**

$$\begin{aligned} p(y) &= \phi^y(1-\phi)^{1-y}, \text{ unde } \phi=p(y=1) \\ p(x|y=0) &= \frac{1}{(2\pi)^{d/2}|\Sigma_0|^{1/2}}\exp\Big(-\frac{1}{2}(x-\mu_0)^\top\Sigma_0^{-1}(x-\mu_0)\Big) \\ p(x|y=1) &= \frac{1}{(2\pi)^{d/2}|\Sigma_1|^{1/2}}\exp\Big(-\frac{1}{2}(x-\mu_1)^\top\Sigma_1^{-1}(x-\mu_1)\Big) \end{aligned}$$

Vă readucem aminte ***regula de decizie*** **a acestui clasificator:**

$$y=1 \text{ dacă } p(y=1|x)\geq p(y=0|x) \text{ şi } y=0 \text{ în caz contrar.} \tag{236}$$

Arătaţi că atunci când $\Sigma_0\neq\Sigma_1$**,** ***separatorul decizional*** **determinat de algoritmul Bayes Optimal gaussian este de ordin pătratic, adică inegalitatea dintre probabilităţile condiţionate a posteriori,** $p(y=1|x)\geq p(y=0|x)$**, este echivalentă cu o inegalitate numerică de forma**

$$x^\top Ax+B^\top x+C\geq 0, \tag{237}$$

unde A **este o anumită matrice de dimensiune** $d\times d$**, cu** $A\neq 0$ **(matricea nulă de dimensiune** $d\times d$**),** $B\in\mathbb{R}^d$ **(un anumit vector-coloană) şi** $C\in\mathbb{R}$ **(o anumită constantă). Veţi specifica în mod clar valorile pentru** A**,** B **şi** C**.**

Răspuns:

Elaborând relaţia dintre probabilităţile care determină decizia luată de către clasificatorul Bayes Optimal gaussian pentru o instanţă oarecare de test x **(vedeţi relaţia** (236)**), putem obţine următoarele echivalenţe:**

$$
\begin{aligned}
& p(y=1|x) \geq p(y=0|x) \Leftrightarrow \ln p(y=1|x) \geq \ln p(y=0|x) \\
\Leftrightarrow \quad & \ln p(y=1|x) - \ln p(y=0|x) \geq 0 \Leftrightarrow \ln \frac{p(y=1|x)}{p(y=0|x)} \geq 0 \\
\overset{F.\ Bayes}{\Leftrightarrow} \quad & \ln \frac{p(x|y=1)\,p(y=1)}{p(x|y=0)\,p(y=0)} \geq 0 \Leftrightarrow \ln \frac{p(y=1)}{p(y=0)} + \ln \frac{p(x|y=1)}{p(x|y=0)} \geq 0 \\
\Leftrightarrow \quad & \ln \frac{\phi}{1-\phi} - \ln \frac{|\Sigma_1|^{1/2}}{|\Sigma_0|^{1/2}} - \frac{1}{2}\Big((x-\mu_1)^\top \Sigma_1^{-1}(x-\mu_1) - (x-\mu_0)^\top \Sigma_0^{-1}(x-\mu_0)\Big) \geq 0 \\
\Leftrightarrow \quad & -\frac{1}{2}\Big(x^\top(\Sigma_1^{-1} - \Sigma_0^{-1})x - 2(\mu_1^\top \Sigma_1^{-1} - \mu_0^\top \Sigma_0^{-1})x + \mu_1^\top \Sigma_1^{-1}\mu_1 - \mu_0^\top \Sigma_0^{-1}\mu_0\Big) \\
& + \ln \frac{\phi}{1-\phi} - \ln \frac{|\Sigma_1|^{1/2}}{|\Sigma_0|^{1/2}} \geq 0 \\
\Leftrightarrow \quad & x^\top\Big(\frac{1}{2}(\Sigma_0^{-1} - \Sigma_1^{-1})\Big)x + \Big(\mu_1^\top \Sigma_1^{-1} - \mu_0^\top \Sigma_0^{-1}\Big)x \\
& + \ln \frac{\phi}{1-\phi} + \ln \frac{|\Sigma_0|^{1/2}}{|\Sigma_1|^{1/2}} + \frac{1}{2}\Big(\mu_0^\top \Sigma_0^{-1}\mu_0 - \mu_1^\top \Sigma_1^{-1}\mu_1\Big) \geq 0. \qquad (238)
\end{aligned}
$$

Coroborând rezultatul pe care tocmai l-am obţinut cu relaţia (236) **din enunţ, observăm că putem considera**

$$
\begin{aligned}
A &= \frac{1}{2}(\Sigma_0^{-1} - \Sigma_1^{-1}) \\
B^\top &= \mu_1^\top \Sigma_1^{-1} - \mu_0^\top \Sigma_0^{-1} \Leftrightarrow \\
B &= (\mu_1^\top \Sigma_1^{-1} - \mu_0^\top \Sigma_0^{-1})^\top = (\Sigma_1^{-1})^\top \mu_1 - (\Sigma_0^{-1})^\top \mu_0 = \Sigma_1^{-1}\mu_1 - \Sigma_0^{-1}\mu_0 \\
C &= \ln \frac{\phi}{1-\phi} + \ln \frac{|\Sigma_0|^{1/2}}{|\Sigma_1|^{1/2}} + \frac{1}{2}\Big(\mu_0^\top \Sigma_0^{-1}\mu_0 - \mu_1^\top \Sigma_1^{-1}\mu_1\Big). \qquad (239)
\end{aligned}
$$

Se poate constata imediat că ipoteza $\Sigma_0 \neq \Sigma_1$ din enunţ implică faptul că $\Sigma_0^{-1} \neq \Sigma_1^{-1}$, deci $A \neq 0$. Prin urmare, graniţa de separare (engl., decision boundary) determinată în acest caz de către algoritmul Bayes Optimal gaussian este de ordin pătratic.

20. **(Clasificatorul Bayes Optimal gaussian: aplicare pe date din $\mathbb{R}^2$; raportul cu regresia logistică)**

prelucrare de Liviu Ciortuz, după
• ***MIT, 2001 fall, Tommi Jaakkola, HW2, pr. 2.abe***

a. Găsiţi ecuaţiile separatorilor decizionali determinaţi de algoritmul Bayes Optimal gaussian pentru fiecare din cele 6 cazuri de mai jos. (În fiecare dintre aceste cazuri, P_i reprezintă probabilităţi de selecţie; μ_i reprezintă medii, iar Σ_i sunt matricele de covarianţă ale distribuţiilor gaussiene respective.) Veţi menţiona în mod explicit, în fiecare caz în parte, tipul — liniar, pătratic (cerc, elipsă, parabolă, hiperbolă etc.), etc. — acestor separatori decizionali.

Pentru fiecare dintre aceste cazuri veţi face câte o reprezentare grafică într-un sistem de axe de coordonate bidimensional, în care să apară separatorul decizional, mediile μ (reprezentate ca puncte) şi curbe de izocontur care corespund gaussienelor respective.

$(i.)$

$P_0 = 0.5,\ P_1 = 0.5$
$\mu_0 = (1,1)^\top$
$\mu_1 = (-1,-1)^\top$
$\Sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

$(ii.)$

$P_0 = 0.005,\ P_1 = 0.995$
$\mu_0 = (1,1)^\top$
$\mu_1 = (-1,-1)^\top$
$\Sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

$(iii.)$

$P_0 = 0.5,\ P_1 = 0.5$
$\mu_0 = (1,1)^\top$
$\mu_1 = (-1,-1)^\top$
$\Sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$\Sigma_1 = \begin{pmatrix} 2 & 0 \\ 0 & 1 \end{pmatrix}$

$(iv.)$

$P_0 = 0.5,\ P_1 = 0.5$
$\mu_0 = (0,0)^\top$
$\mu_1 = (0,0)^\top$
$\Sigma_0 = \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}$
$\Sigma_1 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$

$(v.)$

$P_0 = 0.5,\ P_1 = 0.5$
$\mu_0 = (1,1)^\top$
$\mu_1 = (1,1)^\top$
$\Sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$
$\Sigma_1 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$

$(vi.)$

$P_0 = 0.5,\ P_1 = 0.5$
$\mu_0 = (-2,1)^\top$
$\mu_1 = (1,1)^\top$
$\Sigma_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1/2 \end{pmatrix}$
$\Sigma_1 = \begin{pmatrix} 2 & 0 \\ 0 & 4 \end{pmatrix}$

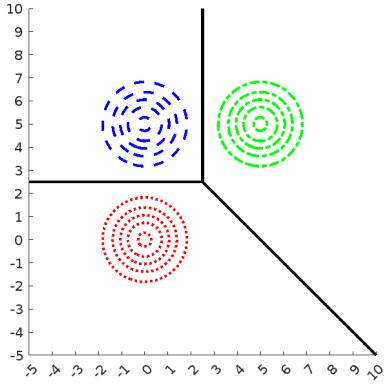
***Sugestie*: În cazul general, ecuaţia separatorului decizional determinat de clasificatorul Bayes Optimal gaussian este $x^\top Ax + B^\top x + C = 0$ (vedeţi realţia (237) de la problema 19), unde valorile pentru A, B şi C au fost determinate prin relaţiile (239).**

b. Arătaţi că este posibil ca întreg planul euclidian să fie alocat de către algoritmul Bayes Optimal gaussian[445] unei singure clase, situaţie în care, practic, nu avem separator decizional.

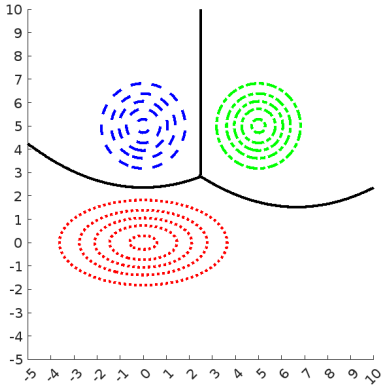
c. Care dintre *separatorii decizionali* de la punctul a poate corespunde *şi* unui model de tip *regresie logistică*? Justificaţi răspunsul dumneavoastră.

d. Care dintre *zonele de decizie* din figurile de mai jos poate să corespundă unui model de tip *regresie logistică multinomială* (*softmax*)? Justificaţi răspunsul dumneavoastră.[446]

(A)



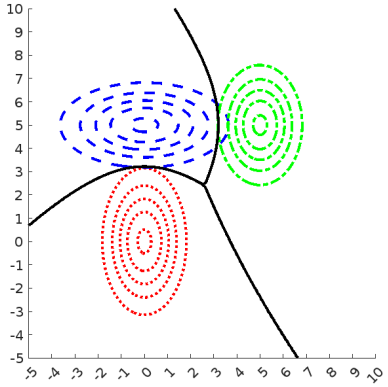
(B)



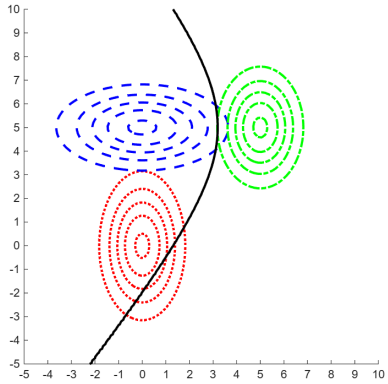
[445] LC: De fapt, este suficient să ne gândim la cazul $\Sigma_0 = \Sigma_1$. (Menţionăm că acest caz particular de clasificare bayesiană gaussiană este numit în literatura de specialitate *analiză gaussiană discriminativă*.)

[446] Pentru o introducere în regresia logistică multinomială, vedeţi problema 19 de la capitolul *Metode de regresie*.

(C)



(D)



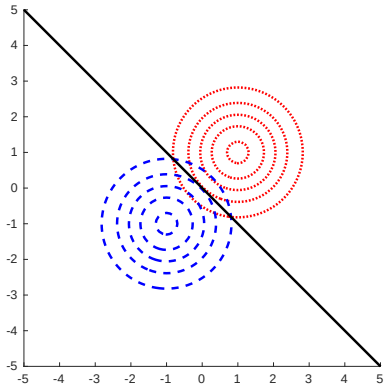
Răspuns:

a. Folosind formulele (239), se calculează coeficienţii A, B şi C din ecuaţia $x^\top Ax + B^\top x + C = 0$, pentru fiecare dintre cazurile date:[447]

i. $A = 0$ **– matricea nulă de dimensiune** 2×2, $B^\top = (2, 2)$, $C = 0$.
Ecuaţia separatorului decizional este

$$2x_1 + 2x_2 = 0 \Leftrightarrow x_2 = -x_1.$$

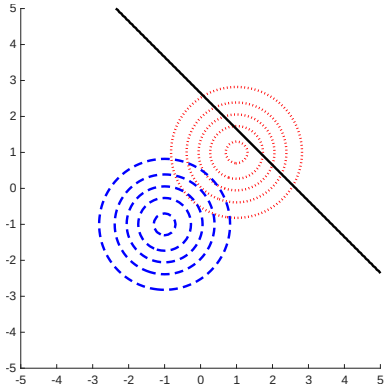
Această ecuaţie reprezintă o dreaptă care trece prin originea sistemului de coordonate. Observaţi că această dreaptă este mediatoarea segmentului de dreaptă care uneşte punctele reprezentate de μ_1 şi μ_2 (mediile celor două gaussiene).



ii. $A = 0$, $B^\top = (-2, -2)$, $C = \ln 199$.
Ecuaţia separatorului decizional este

$$-2x_1 - 2x_2 + \ln 199 = 0 \Leftrightarrow x_2 = -x_1 + \underbrace{\ln \sqrt{199}}_{\approx 2.6466}.$$

Această ecuaţie reprezintă o dreaptă paralelă cu cea de la punctul precedent (i). Ea este situată acum mult mai aproape de punctul μ_1 — datorită faptului că în acest caz probabilitatea P_0 este mult mai mare decât probabilitatea P_1.



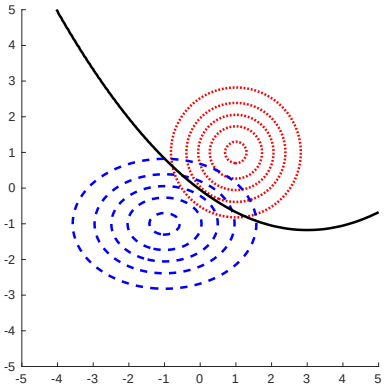
[447] Am omis calculele efective, fiindcă nu sunt dificile. Invităm studentul să verifice că este capabil să obţină el însuşi aceste rezultate.

iii. $A = \begin{pmatrix} 1/4 & 0 \\ 0 & 0 \end{pmatrix}$, $B^\top = (-3/2, -2)$, $C = 1/4 - \ln\sqrt{2}$.

În acest caz, spre deosebire de cazurile precedente, ecuaţia separatorului decizional este de ordin pătratic:

$$\frac{1}{4}x_1^2 - \frac{3}{2}x_1 - 2x_2 + \frac{1}{4} = 0$$
$$\Leftrightarrow x_2 = \frac{1}{8}x_1^2 - \frac{3}{4}x_1 + \frac{1}{8} - \frac{1}{2}\ln\sqrt{2}.$$

Această ecuaţie reprezintă o parabolă, al cărei vârf are coordonatele $(3, -1)$.



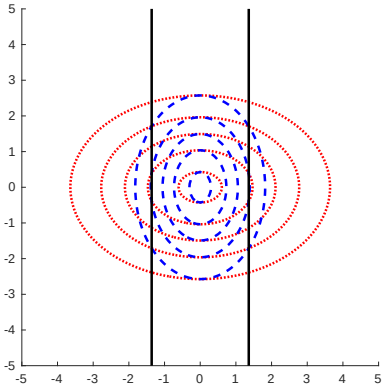
iv. $A = \begin{pmatrix} -3/8 & 0 \\ 0 & 0 \end{pmatrix}$, $B^\top = 0$ – **vectorul nul din** $\mathbb{R}^2$, $C = \ln 2$.

Ecuaţia separatorului decizional este

$$-\frac{3}{8}x_1^2 + \ln 2 = 0 \Leftrightarrow x_1^2 = \frac{8}{3}\ln 2.$$

Aşadar, în acest caz, separatorul decizional este tot de ordin pătratic, dar este reprezentat de două drepte, având ecuaţiile $x_1 = -\sqrt{\frac{8}{3}\ln 2} \approx -1.3595$ **şi respectiv** $x_1 = \sqrt{\frac{8}{3}\ln 2} \approx 1.3595$.

Ţinând cont de relaţia (238) **care a fost demonstrată la problema 19, deducem că zona de decizie corespunzătoare clasei** 1 **este situată între aceste două drepte, iar zona de decizie corespunzătoare clasei** 0 **este în exteriorul lor.**

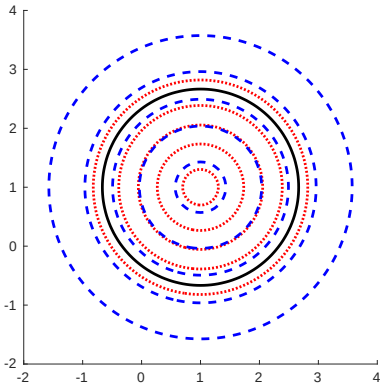


v. $A = \begin{pmatrix} 1/4 & 0 \\ 0 & 1/4 \end{pmatrix}$, $B^\top = (-1/2, -1/2)$, $C = -\ln 2 + 1/2$.

Ecuaţia separatorului decizional este

$$\frac{1}{4}(x_1^2 + x_2^2) - \frac{1}{2}(x_1 + x_2) - \ln 2 + \frac{1}{2} = 0 \Leftrightarrow$$
$$\frac{1}{4}(x_1^2 - 2x_1 + x_2^2 - 2x_2) = \ln 2 - \frac{1}{2} \Leftrightarrow$$
$$(x_1 - 1)^2 + (x_2 - 1)^2 = 4\big(\ln 2 - \frac{1}{2}\big) + 2.$$

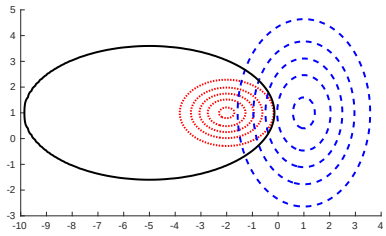
Această ecuaţie — tot de ordin pătratic — reprezintă un cerc cu centrul în punctul $(1, 1)$ **şi raza de** $2\sqrt{\ln 2} \approx 1.3862$.



vi. $A = \begin{pmatrix} 1/4 & 0 \\ 0 & 7/8 \end{pmatrix}$, $B^\top = (5/2, -7/4)$, $C = -\ln(1/4) + 21/8$.

Ecuaţia separatorului decizional este

$$\frac{1}{4}x_1^2+\frac{7}{8}x_2^2+\frac{5}{2}x_1-\frac{7}{4}x_2+2\ln 2+\frac{21}{8}=0 \Leftrightarrow$$
$$\frac{1}{4}(x_1^2+10x_1)+\frac{7}{8}(x_2^2-2x_2)=-2\ln 2-\frac{21}{8} \Leftrightarrow$$
$$\frac{(x_1+5)^2}{4}+\frac{(x_2-1)^2}{\frac{8}{7}}=-2\ln 2-\frac{21}{8}+\frac{25}{4}+\frac{7}{8}.$$



Această ecuaţie — tot de ordin pătratic — reprezintă o elipsă care are centrul de simetrie în punctul $(-5,1)$.

b. Pornind de la regula de decizie $y_{GNB}=\text{argmax}_{y\in\{0,1\}}P(Y=y|X=x)$, putem raţiona astfel:

$$P(Y=1|X=x)\geq P(Y=0|X=x) \overset{F.\ Bayes}{\Leftrightarrow}$$
$$P(X=x|Y=1)P(Y=1)\geq P(X=x|Y=0)P(Y=0).$$

Dacă alegem ca distribuţiile gaussiene corespunzătoare claselor $Y=1$ şi $Y=0$ să fie identice — adică să aibă aceiaşi parametri μ şi Σ —, în inegalitatea de mai sus factorii $P(X=x|Y=1)$ şi $P(X=x|Y=0)$ vor fi egali. Prin urmare, atunci când $P(Y=1)\neq P(Y=0)$, va rezulta că întreg planul / spaţiul va fi alocat unei singure clase, şi anume acelei clase $y\in\{0,1\}$ care are probabilitatea a priori ($P(Y=y)$) mai mare.

c. Se ştie că regresia logistică determină separatori decizionali liniari.[448] Prin urmare, doar separatorii din cazurile i şi ii pot corespunde (şi) unor modele de regresie logistică. (Vedeţi şi rezultatul teoretic de la problema 18.)

d. Justificarea este similară cu cea de la punctul c: doar în cazul (A) separatorii decizionali pot corespunde (şi) unui model de regresie logistică multinomială / softmax.

21. (Clasificatori de tip Bayes gaussian, cazul datelor cu atribute de intrare corelate: exemplificare în $\mathbb{R}^2$; comparaţie)

■ □ • ○ ***CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, HW2, pr. 5.c***

În cazul bidimensional, putem vizualiza modul în care se comportă algoritmul Bayes Naiv gaussian atunci când atributele de intrare (adică, trăsăturile; engl., features) sunt corelate (engl., correlated).[449]

Fie setul de date din figura (A) de mai jos, în care instanţele roşii sunt din clasa 0, iar instanţele albastre din clasa 1. Distribuţiile comune condiţionale

[448] Vedeţi problema 13 de la capitolul *Metode de regresie*, în special relaţiile (182), din care rezultă:

$$P(Y=1|X=x)\geq P(Y=0|X=x)\Leftrightarrow\sigma(z)\geq 1/2\Leftrightarrow z\geq 0,$$

unde $\sigma(z)\overset{def.}{=}\frac{1}{1+e^{-z}}$, $z\overset{not.}{=}w_0+\sum_{i=1}^d w_ix_i$ şi $x\overset{not.}{=}(x_1,\ldots,x_d)$. Aşadar, în cazul regresiei logistice separatorul decizional are ecuaţia de forma $w_0+\sum_{i=1}^d w_ix_i=0$, deci este de tip liniar.

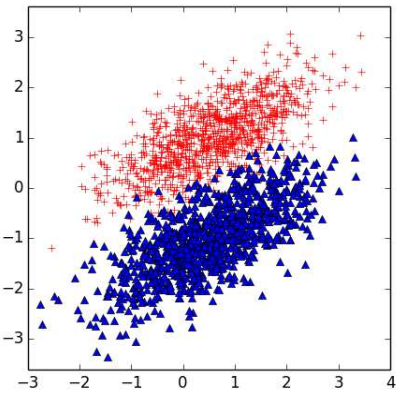
[449] Pentru definiţia noţiunii de *corelare* pentru două variabile aleatoare, vedeţi problema 19 de la capitolul de *Fundamente*.

($(X_1, X_2)|Y = 0$ şi $(X_1, X_2)|Y = 1$) sunt de tip gaussian bidimensional. Elipsele din figurile (B), (C) şi (D) reprezintă curbe de izocontur pentru [diverse] distribuţii condiţionale asociate celor două clase. Centrele elipselor corespund mediilor, iar curbele de izocontur sunt situate la o distanţă de două deviaţii standard faţă de medii.[450]

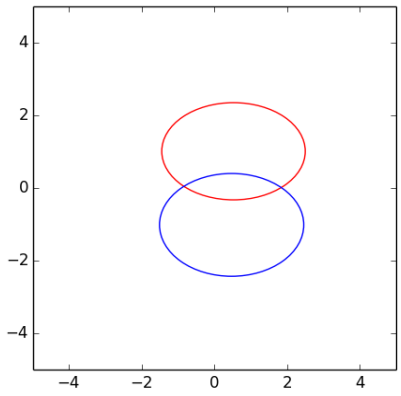
a. Care anume dintre perechile de elipse din figurile (B), (C) şi (D) corespunde cel mai probabil distribuţiilor condiţionale care au generat datele din figura (A)?

b. Care anume dintre aceste elipse corespunde [cel mai probabil] estimărilor de parametri făcute de către algoritmul Bayes Naiv gaussian?

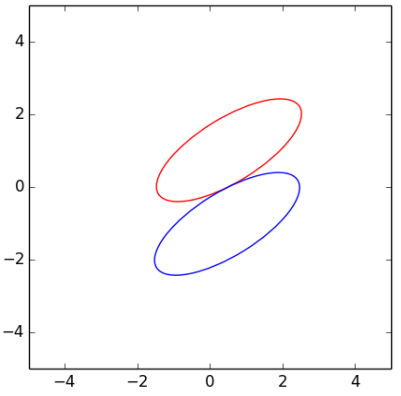
c. Dacă presupunem că probabilităţile a priori pentru cele două clase sunt egale, care dintre modelele (B), (C) şi (D) va obţine o acurateţe mai mare pe datele de antrenament? Care va fi natura separatorului decizional în acest caz?



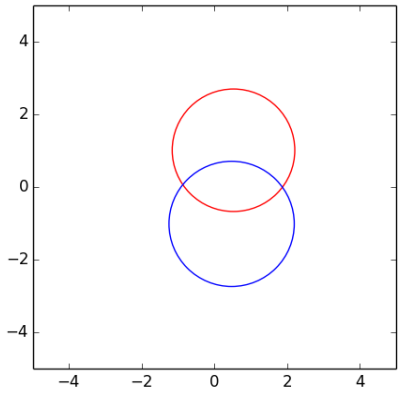
(A) Date



(B)



(C)



(D)

Răspuns:

[450] Mai exact, o astfel de curbă de izocontur este constituită din punctele x din planul euclidian pentru care $\Sigma^{-1/2}(x - \mu) = 2 \Leftrightarrow (x - \mu)^\top \Sigma^{-1}(x - \mu) = 4$, unde μ este media distribuţiei gaussiene considerate. Vedeţi *Pattern Classification*, R. Duda, P. Hart, D. Stork, 2nd ed. (Wiley-Interscience, 2000), Appendix A, pag. 625.

a. Se observă în figura (A) că datele au fost generate de distribuţii gaussiene bidimensionale având matrice de covarianţă nediagonale şi identice, însă având medii diferite. În plus, dacă notăm o instanţă oarecare cu $x = (x_1, x_2)$ este evident că x_1 şi x_2 nu sunt independente, ci există o anumită corelare între ele (mai precis, x_1 şi x_2 sunt într-o relaţie de dependenţă de tip liniar). În consecinţă, curbele de izocontur pentru aceste distribuţii sunt elipse identice ca mărime, având axele de simetrie neparalele cu axele sistemului de coordonate. Evident, doar desenul (C) corespunde acestei situaţii.

b. Clasificatorul Bayes Naiv gaussian presupune independenţa condiţională a celor două atribute (X_1 şi X_2) în raport cu eticheta / variabila de ieşire (Y). Această independenţă corespunde unor matrice de covarianţă diagonale, respectiv unor curbe de izocontur reprezentate de elipse ale căror axe de simetrie sunt paralele cu axele sistemului de coordonate. Atât elipsele din desenul (B) cât şi cele din desenul (D) satisfac aceste specificaţii, însă în cazul (D) elipsele sunt chiar cercuri, în vreme ce datele din figura (A) sunt dispuse în elipse având deviaţii standard diferite pe cele două axe de simetrie. Aşadar, cazul (B) corespunde estimărilor făcute de clasificatorul Bayes Naiv gaussian pe aceste date.

c. Evident, cazul (C) furnizează cea mai mică eroare la antrenare, deci cea mai mare acurateţe. În acest caz, se lucrează cu algoritmul Bayes Optimal gaussian. Întrucât matricele de covarianţă sunt egale, separatorul decizional este de tip liniar. (Pentru justificare riguroasă, vedeţi rezultatul teoretic demonstrat la problema 18.)

22. **(Algoritmul Bayes Naiv [gaussian] vs. regresia logistică — comparaţii)**

□ • ○ ***CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, HW2, pr. 1.f, 3.b-d***

Fie un set de date caracterizate de atributele $X_1, \ldots, X_n$ (pe care le vom considera ca fiind fie toate Bernoulli, fie toate gaussiene) şi de eticheta Y.[451] Modelul Bayes Naiv [eventual de tip gaussian] va fi identificat în continuare cu abrevierea NB, iar modelul regresiei logistice cu LR.

a. Presupunem că datele satisfac presupoziţia de independenţă condiţională de tip Bayes Naiv. Atunci când numărul de exemple de antrenament tinde la infinit, care dintre cei doi clasificatori va produce rezultate mai bune, NB sau LR? Justificaţi.

b. Presupunem acum că datele *nu* satisfac presupoziţia de independenţă condiţională de tip Bayes Naiv. Ne punem aceeaşi întrebare ca mai sus: atunci când numărul de exemple de antrenament tinde la infinit, care dintre cei doi clasificatori va produce rezultate mai bune, NB sau LR? Justificaţi.

c. Este oare posibil să calculăm distribuţia $P(X)$ cu ajutorul parametrilor estimaţi de către algoritmul Bayes Naiv? Explicaţi în mod succint.

[451] Aşa este cazul problemei 14, în care variabilele condiţionate $X_i|Y$ sunt de tip Bernoulli şi sunt independente condiţional două câte două, sau cel al problemei 17, în care toate variabilele condiţionate, $X_i|Y$ pentru $i = 1, \ldots, n$ urmează distribuţii gaussiene — având varianţele $\sigma_{i0} = \sigma_{i1}$ — şi sunt, de asemenea, independente condiţional două câte două. (Similar este şi cazul problemei 51, care constituie o combinaţie a precedentelor două tipuri.)

d. Este oare posibil să calculăm distribuţia $P(X)$ cu ajutorul parametrilor w calculaţi de către regresia logistică? Explicaţi în mod succint.

Răspuns:

a. Regresia logistică este un clasificator probabilist de tip *discriminativ*, adică aproximează / modelează $P(Y|X)$ cu ajutorul funcţiei logistice de argument $w \cdot X$ (unde w este vectorul de parametri, $w \in \mathbb{R}^d$, sau $w \in \mathbb{R}^{d+1}$ dacă extindem fiecare instanţă X cu componenta $X_0 = 1$). În contrast cu acesta, NB este un clasificator probabilist de tip *generativ*, deci calculează distribuţiile $P(X|Y = y)$ (care în cazul GNB sunt de tip gaussian multidimensional) şi de asemenea $P(Y)$, estimând parametrii acestor distribuţii.

Atunci când numărul de instanţe tinde la infinit, pe de o parte aproximaţia calculată de regresia logistică pentru distribuţia $P(Y|X)$ tinde la distribuţia reală $P(Y|X)$, iar pe de altă parte estimările făcute de clasificatorul Bayes Naiv pentru distribuţiile $P(Y)$ şi $P(X|Y)$ vor tinde la distribuţiile reale corespunzătoare, dat fiind (în cazul lui $P(X|Y)$) că datele de antrenament satisfac presupoziţia de independenţă condiţională. Corespondenţa dintre distribuţiile reale $P(Y|X)$ (pe de o parte) şi $P(Y)$ şi $P(X|Y)$ (pe de altă parte) este dată de formula lui Bayes. Prin urmare, în aceste condiţii cei doi clasificatori vor produce rezultate echivalente.

b. Regresia logistică va produce rezultate mai bune, fiindcă ea nu lucrează cu presupoziţia de independenţă condiţională. (Vedeţi pe de o parte problema 37 de la capitolul *Metode de regresie*, iar pe de altă parte problema 10 de la prezentul capitol.)

c. Da, algoritmul Bayes Naiv este un clasificator de tip *generativ* (engl., generative classifier). Putem calcula $P(X)$ prin „marginalizarea“ distribuţiei condiţionate $P(X|Y)$ în raport cu eticheta / clasa Y, şi anume: $P(X) = \sum_y P(X|Y = y) \cdot P(Y = y)$.

d. Nu, nu este posibil. Aşa cum am precizat la punctul a, regresia logistică estimează $P(Y|X)$ (nu $P(X|Y)$ şi $P(Y)$, cum calculează clasificatorii de tip Bayes Naiv).

23. **(Clasificarea bayesiană gaussiană vs. regresia logistică: Adevărat sau Fals?)**

◦ CMU, 2010 fall, Aarti Singh, midterm, pr. 1.2
CMU, (?) 15-781, midterm example questions, pr. 1.d

a. Corespondenţa dintre regresia logistică şi clasificatorul Bayes Naiv de tip gaussian înseamnă — în cazul în care matricele de covarianţă corepunzătoare claselor sunt diagonale şi egale, ca în problema 17 — că există o corespondenţă 1-la-1 între parametrii celor doi clasificatori.

b. Presupunând că lucrăm cu un număr fix de atribute, putem învăţa un clasificator Bayes Optimal de tip gaussian în timp liniar în raport cu numărul de instanţe din setul de date de antrenament.

Răspuns:

a. Fals. Se poate preciza de la început că deşi cei doi clasificatori învaţă separatori decizionali care au aceeaşi formă (şi anume, o formă liniară, vedeţi problema 17) nu rezultă în mod neapărat că pe un acelaşi set de date cei doi separatori decizionali învăţaţi coincid. Faptul că matricele de covarianţă sunt, toate, matrice identitate / diagonale înseamnă că presupoziţia de independenţă condiţională este satisfăcută.[452] Suntem, aşadar, în condiţii similare cu cele de la problema 22.a, însă aici nu avem neapărat satisfăcută ipoteza că numărul de instanţe de antrenament tinde la infinit, caz în care rezultatele de clasificare furnizate de LR şi NB ar fi echivalente. Aşadar, nu există (în general) o corespondenţă de tip 1-la-1 între parametrii w_{LR} calculaţi de regresia logistică[453] şi parametrii w_{NB} corespunzători clasificatorului Bayes Naiv gaussian.[454]

***Observaţie*: Rezultatul acesta este valabil şi în cazul algoritmului Bayes Naiv cu variabile categoriale.**

b. Adevărat. În cazul clasificatorului Bayes Optimal de tip gaussian, regula de calcul pentru clasificarea unei instanţe noi $x \in \mathbb{R}^d$ este următoarea:

$$\begin{aligned} y_{GNB} &\overset{def.}{=} \underset{y \in Val(Y)}{\operatorname{argmax}} P(Y = y | X = x) = \underset{y \in Val(Y)}{\operatorname{argmax}} \frac{P(X = x | Y = y) P(Y = y)}{P(X = x)} \\ &= \underset{y \in Val(Y)}{\operatorname{argmax}} P(X = x | Y = y) P(Y = y) = \underset{y \in Val(Y)}{\operatorname{argmax}} \mathcal{N}(x; \mu_y, \Sigma_y) P(Y = y). \end{aligned}$$

Aşadar, pentru fiecare $y \in Val(Y)$ vom avea de estimat câte o pereche de parametri, care caracterizează o distribuţie gaussiană multidimensională: vectorul de medii μ_y şi matricea de covarianţă Σ_y. Conform problemei 53 de la capitolul de *Fundamente*, estimările celor doi parametri sunt media la eşantionare şi respectiv matricea de covarianţă la eşantionare, deci se calculează în timp liniar în raport cu numărul de instanţe de antrenament.

[452] Vedeţi problema 34 de la capitolul de *Fundamente*.

[453] Vedeţi problema 13 de la capitolul *Metode de regresie*.

[454] În primul rând, *din punctul de vedere al terminologiei*, dacă ne referim la *parametrii* calculaţi de NB ca fiind parametrii distribuţiilor $P(X|Y = y)$ şi $P(Y)$, este imediat că nu există o corespondenţă 1-la-1 între aceştia şi parametrii w_{LR} calculaţi de LR, care servesc la estimarea / aproximarea distribuţiei $P(Y|X)$.

2.2 Clasificare bayesiană — Probleme propuse

2.2.1 Ipoteze de probabilitate maximă a posteriori (MAP)

24. **(Formula lui Bayes; inferenţe statistice; ilustrarea noţiunii de ipoteză MAP)**

∗ ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW1, pr. 5***

Imaginează-ţi că te afli în faţa a trei cutii, care sunt etichetate cu literele A, B, C. Două dintre ele sunt goale, iar una conţine un premiu. Tu nu ştii în care dintre ele se află premiul; trebuie să ghiceşti. Procedezi în felul următor:

Mai întâi alegi la întâmplare o cutie X (să zicem că $X = A$). Totuşi, chiar înainte de a deschide cutia X observi că altcineva, înaintea ta, a deschis cutia Y (să zicem că $Y = B$). Ai dreptul să priveşti în cutia Y, ca să vedeţi dacă ea conţine sau nu premiul. Dacă ea nu conţine premiul, vei avea dreptul să schimbi alegerea pe care ai făcut-o iniţial.

În vederea luării unei decizii cât mai bune, ţi se comunică *strategia* după care a fost aleasă cutia Y, şi anume, folosind una dintre următoarele trei *variante*:

a. **Dacă cutia pe care ai ales-o iniţial conţine premiul, atunci cutia Y este aleasă cu probabilitate de $1/2$ una dintre cele două cutii goale (diferite de cutia X). Dacă X este vidă, atunci Y se alege ca fiind cutia goală diferită de X.**

b. **Se alege aleatoriu cu probabilitate de $1/2$ una dintre cutiile diferite de cea pe care ai ales-o tu iniţial, X. (În consecinţă, cutia Y poate sau nu să conţină premiul. Dacă Y conţine premiul, ai pierdut jocul.)**

c. **Se alege în mod aleatoriu cu probabilitate de $1/2$ una dintre cutiile goale. (Deci este posibil ca $Y = X$. Aşadar, în cazul $Y = X$ observi că a fost deschisă anterior chiar cutia X. În continuare vei putea alege una dintre celelalte două cutii.)**

Considerând (pentru simplitate) că $X = A$, $Y = B$, iar cutia B este vidă, pentru fiecare dintre cele trei *variante* de mai sus decide ce cutie ar trebui să alegi în final pentru a-ţi maximiza şansele de a obţine premiul. Justifică-ţi decizia, elaborând calculul probabilistic aferent.

25. **(Probabilităţi condiţionate; formula lui Bayes; formula probabilităţii totale; ilustrarea noţiunii de ipoteză MAP)**

□ • ○ ∗ ***CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, HW1, pr. 3***

Trei deţinuţi din SUA — desemnaţi aici în mod simplu prin literele a, b, şi c — au fost condamnaţi la moarte şi aşteptau să fie executaţi. Guvernatorul statului în care se afla închisoarea respectivă a decis să-l graţieze pe unul dintre acesti trei deţinuţi (adică, să-i comute pedeapsa cu moartea) şi a ales în mod uniform aleatoriu pe care anume dintre ei să îl graţieze. Guvernatorul l-a

informat pe gardianul închisorii despre decizia sa, precum şi despre rezultatul alegerii sale aleatorii, însă i-a cerut să păstreze secretul.

După ce în închisoare au început să circule zvonuri despre graţiere, deţinutul a a încercat să-l convingă pe gardian să-i spună care este soarta lui. Gardianul a refuzat. Atunci deţinutul a i-a cerut gardianului să-i spună cine — mai precis, *unul* — dintre deţinuţii b sau c care va fi executat. Gardianul s-a gândit un pic şi apoi i-a spus că deţinutul b va fi executat.

***Indicaţie importantă*: La rezolvarea exerciţiului va trebui să presupunem că, răspunzând la întrebarea deţinului a, gardianul a ales în mod aleatoriu — şi uniform, dacă a fost cazul să aleagă între mai multe posibilităţi —, şi că el a respectat totuşi atât adevărul cât şi cerinţa formulată de guvernator (înţeleasă într-un sens mai lax, şi anume că *gardianul nu are voie să comunice niciunui deţinut soarta sa, în mod direct*).**

a. Fie $X = a$, respectiv $X = b$ sau $X = c$ evenimentul care reprezintă faptul că deţinutul a, respectiv b sau c a fost graţiat. Precizaţi care sunt valorile numerice pentru probabilităţile a priori $P(X = a)$, $P(X = b)$ şi $P(X = c)$.

Notăm cu $Y = b$ evenimentul (comunicat de gardian) că deţinutul b urmează să moară (adică, să fie executat). Calculaţi $P(X = a|Y = b)$. În urma obţinerii de către deţinutul a informaţiei adiţionale (de la gardian) că deţinutul b va muri, a crescut oare probabilitatea ca el (deţinutul a) să supravieţuiască?

***Indicaţie*: Comparaţi probabilitatea a posteriori $P(X = a|Y = b)$ cu probabilitatea a priori $P(X = a)$. În prealabil veţi completa un tabel în care veţi specifica valorile tuturor probabilităţilor condiţionate $P(Y = y|X = x)$, cu x şi $y \in \{a, b, c\}$.**

	$P(Y = a\|X)$	$P(Y = b\|X)$	$P(Y = c\|X)$
$X = a$			
$X = b$			
$X = c$			

b. Presupunem că deţinutul a a comunicat toate cele de mai sus deţinutului c. Calculaţi cât devine acum probabilitatea deţinutului c de a supravieţui. (Aşadar, calculaţi $P(X = c|Y = b)$.) Care dintre probabilităţile a posteriori $P(X = a|Y = b)$, $P(X = b|Y = b)$ şi $P(X = c|Y = b)$ este cea mai mare?

26. **(Adevărat sau Fals?)**

CMU, 2002 fall, Andrew Moore, final exam, pr. 11.a

Fie D un set de exemple (date de antrenament), iar H o mulţime de ipoteze pentru (un algoritm de) învăţare automată pe datele D. Precizaţi care este *valoarea de adevăr* a următoarelor afirmaţii:

$\text{argmax}_{h \in H} P(D|h)$ **este ipoteză de probabilitate maximă a posteriori, [iar]**
$\text{argmax}_{h \in H} P(h|D)$ **este ipoteză de verosimilitate maximă.**

2.2.2 Algoritmii Bayes Naiv şi Bayes Optimal

27. **(Algoritmii Bayes Naiv şi Bayes Optimal; aplicare; numărul minimal de parametri de estimat)**

• ○ ***Liviu Ciortuz, 2017, pornind de la setul de date din Machine Learning, Tom Mitchell, 1997, ch. Decision Trees, page 59***

Considerăm următorul set de date de antrenament, în care variabila de ieşire este *EnjoyTennis*:

Day	Outlook	Temperature	Humidity	Wind	EnjoyTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

a. Determinaţi decizia luată de către algoritmul Bayes Naiv pentru instanţa de test

$$X = \langle Outlook = sunny,\ Temp = cool,\ Humidity = high,\ Wind = strong \rangle,$$

precum şi probabilitatea cu care este luată această decizie.

b. Care este numărul *minim* de parametri pe care trebuie să-l estimeze algoritmul Bayes Naiv pe aceste date [pentru a face apoi predicţii pe un set oarecare de instanţe de test]? Dar în cazul clasificatorului Bayes Optimal?

c. Implementaţi algoritmul Bayes Naiv, iar apoi cu ajutorul acestei implementări calculaţi eroarea la antrenare şi eroarea la CVLOO pe acest set de date.

28. **(Algoritmul Bayes Naiv: aplicare)**

∗ ***CMU, 2004 fall, T. Mitchell Z. Bar-Joseph, midterm, pr. 6.a***

Se dă setul de date din tabelul alăturat, în care A, B, C sunt atribute (de intrare) binare, iar Y este atribut de ieşire. Care va fi răspunsul algoritmului de clasificare Bayes Naiv pentru intrarea $A = 0$, $B = 0$, $C = 1$?

A	B	C	Y
0	0	1	0
0	1	0	0
1	1	0	0
0	0	1	1
1	1	1	1
1	0	0	1
1	1	0	1

29. **(Algoritmul Bayes Naiv şi algoritmul Bayes Optimal: aplicare)**

prelucrare de Liviu Ciortuz, după

• ∗ ***CMU, 2002 fall, Andrew Moore, final exam, pr. 4.b-e***

Se dă setul de date alăturat, cu A şi B variabile de intrare, iar C variabilă de ieşire.

A	B	C	nr. apariţii
0	0	1	3
0	1	0	1
0	1	1	4
1	0	0	5
1	1	0	2
1	1	1	1

a. Care este numărul minim de probabilităţi ce trebuie estimate pentru a putea construi după aceea (pe acest set de date) un clasificator de tip Bayes Naiv? Justificaţi.

b. Similar, pentru clasificatorul Bayes Optimal. Justificaţi.

c. Care este decizia clasificatorului Bayes Naiv pentru $A = 0, B = 1$? Precizaţi cu ce probabilitate este luată această decizie.

d. Care este decizia clasificatorului Bayes Optimal pentru $A = 0, B = 1$? Precizaţi cu ce probabilitate este luată această decizie.

e. Dacă rezultatele obţinute la punctele c şi d diferă (fie şi numai în privinţa *probabilităţilor* cu care sunt luate deciziile), care este explicaţia? Justificaţi în mod riguros.

30. **(Aplicarea algoritmului Bayes Naiv la clasificarea de texte)**

• ∗ ***Edinburgh, 2009 fall, C. Williams, V. Lavrenko, tutorial 2, pr. 2***

Firma Whizzco decide să implementeze un clasificator de texte. Pentru început, ei vor să clasifice documente aparţinând fie clasei *sport* fie clasei *politică*. Ei decid să reprezinte fiecare document ca un vector de atribute descriind prezenţa ori absenţa unor cuvinte-cheie:

goal, football, golf, defence, offence, wicket, office, strategy.

Datele de antrenament sunt reprezentate folosind o matrice în care fiecare linie este un vector de valori (0 sau 1) pentru cele 8 atribute.

```
xP=[1 0 1 1 1 0 1 1; % Politica
    0 0 0 1 0 0 1 1;
    1 0 0 1 1 0 1 0;
    0 1 0 0 1 1 0 1;
    0 0 0 1 1 0 1 1;
    0 0 0 1 1 0 0 1]
```

```
xS=[1 1 0 0 0 0 0 0; % Sport
    0 0 1 0 0 0 0 0;
    1 1 0 1 0 0 0 0;
    1 1 0 1 0 0 0 1;
    1 1 0 1 1 0 0 0;
    0 0 0 1 0 1 0 0;
    1 1 1 1 1 0 1 0]
```

Folosind algoritmul Bayes Naiv, care este probabilitatea cu care documentul $x = (1, 0, 0, 1, 1, 1, 1, 0)$ va fi clasificat ca aparţinând clasei *politică*?

31. **(Aplicarea algoritmului Bayes Naiv: chestiunea valorilor lipsă (engl., missing values) în datele de antrenament)**

□ • ***CMU, 2013 fall, A. Smola, G. Gordon, midterm practice, pr. 9***

Fie setul de date de antrenament (x, y) şi datele de test z:

$$\begin{array}{ll} x_1 = (0,0,0,1,0,0,1) & y_1 = 1 \\ x_2 = (0,0,1,1,0,0,0) & y_2 = 1 \\ x_3 = (1,1,0,0,0,1,0) & y_3 = -1 \\ x_4 = (1,0,0,0,1,1,0) & y_4 = -1 \end{array}$$

$$\begin{array}{l} z_1 = (1,0,0,0,0,1,0) \\ z_2 = (0,1,1,0,0,1,1) \end{array}$$

Ce *problemă* va întâmpina clasificatorul Bayes Naiv pe aceste date? (*Indicaţie*: Pentru ca răspunsul dumneavoastră să fie cât mai bine justificat, veţi estima toţi parametrii necesari şi veţi aplica algoritmul pe cele două instanţe de test. Veţi nota atributele cu $A1, A2, \ldots$.)

La curs am prezentat un „remediu“ standard pentru o astfel de *problemă*. Precizaţi cum se numeşte „tehnica“ respectivă şi aplicaţi-o pe aceste date. După aceea, veţi aplica algoritmul Bayes Naiv pentru a clasifica instanţele de test z_1 şi z_2.

32. **(Algoritmul Bayes Naiv: calculul ratei medii a erorii – exemplificare)**

• ○ ***CMU, 2011 spring, Tom Mitchell, midterm, pr. 5.1-2***

Considerăm o problemă de clasificare binară în care se foloseşte o variabilă $X_1 \in \{0,1\}$ şi eticheta $Y \in \{0,1\}$. Distribuţia generativă „adevărată“ $P(X_1, Y) = P(Y) \cdot P(X_1|Y)$ este determinată conform tabelelor următoare:

Y	0	1
$P(Y)$	0.8	0.2

$P(X_1\|Y)$	$Y=0$	$Y=1$
$X_1=0$	0.7	0.3
$X_1=1$	0.3	0.7

a. Presupunem că am antrenat un clasificator Bayes Naiv folosind o infinitate de *date de antrenament* generate conform celor două tabele de mai sus. Scrieţi în tabelul următor *predicţiile* făcute de algoritmul Bayes Naiv pentru diferitele valori ale lui X_1. Remarcaţi faptul că $\hat{Y}(X_1)$ din acest tabel este decizia [lui Bayes Naiv] cu privire la valoarea lui Y dat fiind X_1. În coloanele care corespund probabilităţilor, veţi scrie atât valorile concrete ale acestor probabilităţi, cât şi modul cum au fost ele calculate (de exemplu, $0.8 \cdot 0.7 = 0.56$), iar în coloana care corespunde deciziei, veţi scrie [pe fiecare linie] fie $\hat{Y} = 0$ fie $\hat{Y} = 1$.

	$P(X_1, Y=0)$	$P(X_1, Y=1)$	$\hat{Y}(X_1)$
$X_1=0$			
$X_1=1$			

b. Cât este *rata medie a erorilor* (engl., expected error rate) produsă de acest clasificator Bayes Naiv pe *instanţele de test* care sunt generate conform primelor două tabele de mai sus? Cu alte cuvinte, calculaţi $P(\hat{Y}(X_1) \neq Y)$ unde perechile (X_1, Y) sunt generate conform celor două tabele.

***Indicaţie*: $P(\hat{Y}(X_1) \neq Y) = P(\hat{Y}(X_1) \neq Y, X_1 = 0) + P(\hat{Y}(X_1) \neq Y, X_1 = 1)$.**

Pentru următoarele trei puncte ale acestui exerciţiu vom considera două variabile, şi anume $X_1 \in \{0, 1\}$ şi $X_2 \in \{0, 1\}$, precum şi eticheta $Y \in \{0, 1\}$. Y şi X_1 sunt şi de data aceasta generate conform primelor două tabele de mai sus, iar apoi X_2 este creat ca o *copie exactă* (adică, duplicat) după X_1.

c. Acum vom presupune că am antrenat un clasificator Bayes Naiv folosind o infinitate de *exemple de antrenament* care au fost generate în conformitate cu primele două tabele de mai sus *şi* cu regula de duplicare. Scrieţi în tabelul următor *predicţiile* făcute de către acest clasificator Bayes Naiv pentru diferitele valori ale perechii (X_1, X_2). În privinţa probabilităţilor din tabel, puteţi să scrieţi doar cum anume sunt ele calculate (de exemplu, în loc de $0.8 \cdot 0.3 \cdot 0.3 = 0.072$ veţi putea scrie doar $0.8 \cdot 0.3 \cdot 0.3$ pentru a economisi un pic de timp).

	$\hat{P}(X_1, X_2, Y = 0)$	$\hat{P}(X_1, X_2, Y = 1)$	$\hat{Y}(X_1, X_2)$
$X_1 = 0, X_2 = 0$			
$X_1 = 0, X_2 = 1$			
$X_1 = 1, X_2 = 0$			
$X_1 = 1, X_2 = 1$			

d. Cât este *rata medie a erorilor* pentru acest clasificator Bayes Naiv pe instanţe de test care sunt generate în conformitate cu primele două tabele de mai sus *şi* cu regula de duplicare?

e. Comparativ cu cazul precedent (adică, fără X_2), cum s-a schimbat rata medie a erorilor (adică, a crescut ori a scăzut)? În tabelul de la punctul c, *ce* linie este responsabilă pentru această schimbare? Cum explicaţi ce s-a întâmplat?

33. **(Algoritmului Bayes Naiv: calculul ratei medii a erorilor)**

■ • ○ ***CMU, 2010 fall, Aarti Singh, HW1, pr. 4.2***

Considerăm următoarea problemă de clasificare:
Fie variabila aleatoare $Y : \boldsymbol{Hike} \in \{T, F\}$ care denotă faptul că Alice şi Bob merg sau nu în drumeţie în funcţie de condiţiile vremii: $X_1 : \boldsymbol{Sunny} \in \{T, F\}$ şi $X_2 : \boldsymbol{Windy} \in \{T, F\}$.

Se presupune că au fost estimaţi următorii parametri:

$$P(\boldsymbol{Hike}) = 0.5$$
$$P(\boldsymbol{Sunny} \mid \boldsymbol{Hike}) = 0.8, \quad P(\boldsymbol{Sunny} \mid \neg\boldsymbol{Hike}) = 0.7$$
$$P(\boldsymbol{Windy} \mid \boldsymbol{Hike}) = 0.4, \quad P(\boldsymbol{Windy} \mid \neg\boldsymbol{Hike}) = 0.5$$

De asemenea, se consideră că este satisfăcută presupoziţia de independenţă condiţională a algoritmului Bayes Naiv.

a. Care este probabilitatea (comună) ca Alice şi Bob să meargă în drumeţie atunci când vremea este însorită şi bate vântul, adică

$$P(\textbf{\textit{Sunny}} = T, \textbf{\textit{Windy}} = T, \textbf{\textit{Hike}} = T) = ?$$

Care este decizia luată de algoritmul Bayes Naiv în acest caz?

b. Completaţi tabelul următor:

X_1	X_2	Y	$P(X_1, X_2, Y)$	$P_{NB}(Y \mid X_1, X_2)$	***decizia algoritmului Bayes Naiv***
F	F	F			
F	F	T			
F	T	F			
F	T	T			
T	F	F			
T	F	T			
T	T	F			
T	T	T			

***Observaţie*: Calculele de la punctul a corespund ultimei linii din tabelul de mai sus.**

c. Care este rata medie a erorilor (engl., expected error rate) produse de algoritmul Bayes Naiv? Vă reamintim că această (rată) medie este definită ca fiind suma probabilităţilor $P(X_1, X_2, Y)$ pentru acele (triplete de) valori ale variabilelor X_1, X_2, Y pentru care decizia luată de algoritmul Bayes Naiv diferă de valoarea variabilei Y.

În cele ce urmează se presupune că se obţin mai multe informaţii despre vreme. Se introduce o nouă trăsătură X_3:*Rainy* $\in \{T, F\}$. Se presupune că în fiecare zi vremea poate fi fie *Rainy* fie *Sunny*, dar nu şi *Rainy* şi *Sunny*. Similar, se presupune că vremea nu poate fi într-o zi $\neg$*Rainy* şi $\neg$*Sunny*.

d. În noile condiţii, presupoziţia de independenţă condiţională rămâne oare adevărată? Justificaţi.

e. Calculaţi $P(\textit{Sunny} = T, \textit{Windy} = T, \textit{Rainy} = F, \textit{Hike} = T)$.

f. Care este rata medie a erorilor produse de clasificatorul Bayes Naiv când se folosesc toate cele 3 atribute de intrare?

g. S-a îmbunătăţit performanţa algoritmului Bayes Naiv prin adăugarea atributului *Rainy*? Explicaţi de ce.

34. **(Algoritmul Bayes Naiv: calculul ratei medii a erorii – exemplificare; comparaţie cu regresia logistică)**

prelucrare de Liviu Ciortuz, după

• ∗ ***CMU, 2009 fall, Carlos Guestrin, HW1, pr. 4.1.4***

Considerăm o problemă de clasificare binară în care fiecare exemplu de antrenament are două atribute binare $X_1, X_2 \in \{T, F\}$ şi eticheta / clasa $Y \in \{T, F\}$.

Presupunem că $P(Y = T) = 0.5$, iar $P(X_1 = T|Y = T) = 0.8$, $P(X_1 = F|Y = F) = 0.7$, $P(X_2 = T|Y = T) = 0.5$ şi $P(X_2 = F|Y = F) = 0.9$. (Se poate observa că atributul X_1 furnizează / constituie un indiciu întrucâtva mai puternic decât atributul X_2 în ce priveşte determinarea clasei unei instanţe oarecare.)

În cele ce urmează vom presupune că X_1 şi X_2 sunt independente în raport cu Y.

a. Calculaţi probabilităţile $P(X_1 = F|Y = T)$, $P(X_1 = T|Y = F)$, $P(X_2 = F|Y = T)$ şi $P(X_2 = T|Y = F)$. Asociaţi răspunsului dumneavoastră o justificare generală, sub forma unei formule din teoria probabilităţilor:

$$P(\neg A|B) = \ldots, \text{ unde } A \text{ şi } B \text{ sunt evenimente aleatoare oarecare.}$$

b. Scrieţi regula de decizie a algoritmului Bayes Naiv pentru $X_1 = x_1$ şi $X_2 = x_2$, justificând în mod succint obţinerea ei.

c. Calculaţi rata medie a erorii produse de algoritmul Bayes Naiv, atunci când se folosesc ambele atribute, X_1 şi X_2. (Veţi da în prealabil definiţia ratei medii a erorii.) Este oare această rată mai bună decât în cazul în care se foloseşte un singur atribut (X_1 sau X_2)? De ce?

d. Să presupunem acum că se crează un nou atribut, X_3, care este o copie exactă a lui X_2. Aşadar, pentru fiecare exemplu de antrenament, atributele X_2 şi X_3 au aceeaşi valoare, $X_2 = X_3$. Răspundeţi la următoarele întrebări:

– Sunt X_2 şi X_3 independente condiţional în raport cu Y?

– Cât este rata medie a erorii pentru Bayes Naiv acum? (Atenţie! Distribuţia „adevărată“ a datelor nu s-a modificat.)

– Explicaţi ce se întâmplă cu algoritmul Bayes Naiv. Oare *regresia logistică* are aceeaşi problemă? Explicaţi de ce.

35. **(Clasificare bayesiană: calculul ratei medii a erorilor pentru diverşi clasificatori bayesieni)**

prelucrare de L. Ciortuz, după
■ • ○ ***CMU, 2004 fall, T. Mitchell Z. Bar-Joseph, HW3, pr. 1.2***

Fie funcţia $Y = (A \wedge B) \vee \neg(B \vee C)$, unde A, B şi C sunt variabile aleatoare binare independente, fiecare dintre ele având posibilitatea să ia valoarea 0 cu probabilitate de 50%.

a. Câţi parametri trebuie să estimeze clasificatorul Bayes Naiv pentru a învăţa funcţia Y? Enumeraţi aceşti parametri. *Atenţie*: $P(\neg x)$ nu va fi socotit ca parametru dacă $P(x)$ a fost deja estimat ca parametru.

b. Care este rata medie a erorii la antrenare pentru clasificatorul Bayes Naiv la învăţarea conceptului Y, presupunând că avem o infinitate de date de antrenament?

***Indicaţie*: Scrieţi mai întâi tabela de adevăr a funcţiei Y, apoi estimaţi valorile parametrilor (în sensul verosimilităţii maxime, MLE). Pentru convenienţă, centralizaţi toate calculele făcute de către algoritmul Bayes Naiv într-un tabel.**

***Convenţie*: În cazul în care, pentru o setare oarecare a variabilelor A, B şi C, cele două probabilităţi calculate de către algoritmul Bayes Naiv în vederea determinării valorii y_{NB} sunt egale, convenim că algoritmul va lua decizia $y_{NB} = 1$.**

c. Câţi parametri trebuie să estimeze clasificatorul Bayes Optimal pentru a „învăţa" funcţia Y? Justificaţi în detaliu.

d. Care este rata medie a erorii la antrenare pentru clasificatorul Bayes Optimal la învăţarea conceptului Y, presupunând acelaşi lucru ca mai sus? *Atenţie*: Nu este nevoie să calculaţi efectiv această rată; este suficient să indicaţi valoarea ei şi să o justificaţi printr-un *raţionament calitativ*.[455]

e. Considerăm un alt clasificator de tip Bayes, care presupune că A este independent de C, condiţionat de B şi Y — în contrast cu clasificatorul Bayes Naiv, care presupune că variabilele A, B şi C sunt independente două câte două în raport cu Y.

Arătaţi că acest clasificator Bayes va avea nevoie să estimeze mai puţini parametri decât clasificatorul Bayes Optimal la învăţarea conceptului Y, şi totuşi va obţine aceeaşi rată medie a erorii la antrenare (considerând că este valabilă aceeaşi presupoziţie în legătură cu datele de antrenament).

***Indicaţii*:**

– Folosind tabelă de adevăr a funcţiei Y, dovediţi că într-adevăr proprietatea de independenţă condiţională formulată mai sus este satisfăcută. Şi anume: pentru fiecare pereche de valori b şi y ale variabilelor aleatoare B şi respectiv Y, arătaţi că

$$P(A = a, C = c|B = b, Y = y) = P(A = a|B = b, Y = y) \cdot P(C = c|B = b, Y = y),$$

pentru $\forall a \in Val(A)$ şi $\forall c \in Val(C)$.

– Scrieţi regula de decizie pentru acest nou clasificator de tip Bayes şi arătaţi că ea este echivalentă cu regula de decizie a clasificatorului Bayes Optimal.

– Calculaţi numărul minim de parametri de estimat de către noul nostru clasificator Bayes şi enumeraţi-i.

36. **(Algoritmul Bayes Naiv: independenţa [condiţională a] atributelor de intrare; calculul ratei medii a erorii)**

• ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW3, pr. 1.1***

Presupunem că A şi B sunt variabile aleatoare binare independente, fiecare dintre ele având posibilitatea de a lua valoarea 0 cu o probabilitate de 50%.

Definiţi o funcţie booleană $y = f(A, B)$ în aşa fel încât variabila A să *nu* fie independentă de variabila B în raport cu y (văzut şi el ca variabilă aleatoare), însă clasificatorul Bayes Naiv să producă o rată medie a erorii de 0% (presupunând că datele de antrenament sunt în număr infinit).

[455] LC: Totuşi, este recomandabil să procedaţi aşa după ce în prealabil aţi văzut ce valoare produce algoritmul Bayes Optimal pentru [măcar] una dintre combinaţiile de valori ale variabilelor, de exemplu, $A = B = C = 0$.

Demonstraţi că acest clasificator are într-adevăr rata erorii de 0%.

***Indicaţie*: Puteţi să identificaţi o astfel de funcţie făcând căutare exhaustivă între cele 16 (de fapt, suficient, 14) funcţii de la problema 11.**

37. **(Algoritmii Bayes Naiv şi Bayes Optimal: complexitatea modelului)**

prelucrare de L. Ciortuz, după

□ • ***CMU, 2020 fall, E. Xing, Z. Bar-Joseph, HW1, pr. 4***

Fie $X = (x_1, x_2, \ldots, x_n)$ o instanţă oarecare, cu $x_1, x_2, \ldots, x_n$ desemnând valorile trăsăturilor / atributelor, iar $y \in \{0, 1\}$ eticheta asociată respectivei instanţe.

Vă readucem aminte că în orice variantă / model de *clasificare de tip generativ*, probabilitatea a posteriori a etichetei (şi anume, $P(y|X)$) este exprimată cu ajutorul probabilităţii condiţionate a trăsăturilor în raport cu eticheta, $P(X|y)$:

$$P(y|X) \propto P(X|y)P(y), \tag{240}$$

unde semnul $\propto$ înseamnă „proporţional cu".

a. Scrieţi expresia probabilităţii condiţionate $P(X|y)$ din partea dreaptă a relaţiei (240) în condiţiile în care se foloseşte presupoziţia [algoritmului Bayes Naiv] că trăsăturile sunt independente condiţional în raport cu eticheta.

b. Să presupunem că pentru fiecare trăsătură / atribut de intrare, valorile x_i sunt luate în mulţimea $\{1, 2, \ldots, K\}$. De asemenea, presupunem că distribuţia probabilistă asociată etichetei y este de tip Bernoulli, iar distribuţia probabilistă condiţionată a fiecărei trăsături x_i în raport cu eticheta y (adică, $P(X|y)$) este de tip categorial. Vă cerem să elaboraţi [în mod riguros] răspunsuri la următoarele întrebări.

***i*. Cât este numărul total de parametri ai acestui model dacă se foloseşte presupoziţia de independenţă condiţionată a algoritmului Bayes Naiv?**

***ii*. Cât este numărul total de parametri când nu se foloseşte presupoziţia de independenţă condiţionată a algoritmului Bayes Naiv?**

***iii*. Presupunând acum că eticheta y ia valori în mulţimea $\{0, 1, \ldots, M-1\}$, cum se vor schimba răspunsurile date la întrebările precedente (cu, şi respectiv fără presupoziţia de independenţă condiţionată a algoritmului Bayes Naiv)?**

38. **(Legătura dintre calculul parametrilor algoritmului Bayes Naiv şi estimarea [lor] în sensul verosimilităţii maxime (MLE); caracterul liniar al separatorului decizional determinat de Bayes Naiv, în cazul variabilelor binare)**

■ • ○ ***Stanford, 2007 fall, Andrew Ng, HW1, pr. 4***

În această problemă vom face estimarea parametrilor algoritmului Bayes Naiv în sensul verosimilităţii maxime (MLE) folosind, bineînţeles, presupoziţia de independenţă condiţională.

Considerăm că în modelul nostru atributele de intrare x_j, cu $j = 1, \ldots, n$, sunt variabile aleatoare discrete, luând valori binare: $x_j \in \{0, 1\}$. Vom numi

$x = (x_1, x_2, \ldots, x_n)^\top$ ***vector de intrare*** **(engl., input vector). Pentru fiecare exemplu de antrenament, *ieşirea* asociată lui (engl., output target) este o valoare binară:** $y \in \{0, 1\}$**. Ca urmare, *modelul* nostru va fi descris cu ajutorul *parametrilor*** $\theta_{j0} = P(x_j = 1|y = 0)$**,** $\theta_{j1} = P(x_j = 1|y = 1)$ **şi** $\theta_y = P(y = 1)$**.**

Vom modela *distribuţia probabilistă comună* (engl., joint distribution) a perechii (x, y) **în felul următor:**

$$\begin{aligned} P(y) &= (\theta_y)^y (1-\theta_y)^{1-y} \\ P(x|y=0) &= \prod_{j=1}^{n} P(x_j|y=0) = \prod_{j=1}^{n} ((\theta_{j0})^{x_j}(1-\theta_{j0})^{1-x_j}) \\ P(x|y=1) &= \prod_{j=1}^{n} P(x_j|y=1) = \prod_{j=1}^{n} ((\theta_{j1})^{x_j}(1-\theta_{j1})^{1-x_j}). \end{aligned}$$

a. Notând cu θ **întregul set de parametri** $\{\theta_y, \theta_{j0}, \theta_{j1}, j = 1, \ldots, n\}$**, determinaţi expresia funcţiei de verosimilitate comună (engl., joint likelihood function),** $\ell(\theta) = \ln \prod_{i=1}^{M} P(x^{(i)}, y^{(i)}; \theta)$**, în raport cu parametrii modelului care au fost precizaţi mai sus.**

b. Arătaţi că valorile parametrilor pentru care funcţia de verosimilitate îşi atinge maximul sunt următoarele:

$$\begin{aligned} \theta_{j0} &= \frac{\sum_{i=1}^{m} 1_{\{x_j^{(i)}=1 \wedge y^{(i)}=0\}}}{\sum_{i=1}^{m} 1_{\{y^{(i)}=0\}}} \\ \theta_{j1} &= \frac{\sum_{i=1}^{m} 1_{\{x_j^{(i)}=1 \wedge y^{(i)}=1\}}}{\sum_{i=1}^{m} 1_{\{y^{(i)}=1\}}} \\ \theta_y &= \frac{\sum_{i=1}^{m} 1_{\{y^{(i)}=1\}}}{m}. \end{aligned}$$

Observaţie **(1):** Pentru o perspectivă mai generală asupra maximizării verosimilităţii datelor de antrenament în cazul folosirii clasificatorului Bayes Naiv cu variabile Bernoulli, vedeţi *Comentariul* de la problema 149 de la capitolul de *Fundamente*.

Observaţie **(2):** În cazul în care atributele / variabilele de intrare $X_1, \ldots, X_n$ urmează distribuţii categoriale, punctele a şi b din enunţul prezentei probleme se generalizează foarte uşor. Practic, estimarea în sensul MLE a parametrilor necesari pentru a face predicţii cu algoritmul Bayes Naiv revine — după scrierea funcţiei de log-verosimilitate a datelor de antrenament — la a face estimarea în sensul MLE pentru parametrii unor distribuţii categoriale (câte două astfel de distribuţii pentru fiecare atribut, dacă variabila de ieşire este binară). Pentru ideea demonstraţiei, vedeţi problema 44 de la capitolul de *Fundamente*.

c. Ştim că modul în care algoritmul Bayes Naiv face predicţie pentru o instanţă [nouă] oarecare x **constă în a determina cea mai probabilă clasă pentru** x**.**

Demonstraţi că *ipoteza* produsă de către Bayes Naiv este un clasificator liniar — adică, presupunând că $P(y = 0|x)$ **şi** $P(y = 1|x)$ **sunt probabilităţile a posteriori ale claselor care sunt calculate de Bayes Naiv pentru** x**, atunci există un anumit vector-coloană** $w \in \mathbb{R}^{n+1}$ **astfel încât**

$$P(y = 1|x) \geq P(y = 0|x) \text{ dacă şi numai dacă } w^\top \begin{bmatrix} 1 \\ x \end{bmatrix} \geq 0 \ (\forall x).$$

(Veţi considera că w_0 este termenul liber (engl., intercept term).)

39. **(Caracterul liniar al regulii de decizie a clasificatorului Bayes Naiv: exemplificare, folosind datele de la problema 2 de la capitolul *Arbori de decizie*)**

■ □ • ○ ***Liviu Ciortuz, 2024***

Considerăm setul de date alăturat, pe care l-am preluat de la problema 2 de la capitolul *Arbori de decizie*, cu următoarele schimbări referitoare la *notaţii*:[456] Atributele de intrare au fost redenumite $X_1, \ldots, X_4$, iar variabila de ieşire a fost renotată cu Y.

	X_1	X_2	X_3	X_4	Y
A	1	0	0	0	1
B	1	0	1	0	1
C	0	1	0	1	1
D	0	0	0	1	0
E	1	1	1	0	0
F	1	0	1	1	0
G	1	0	0	1	0
H	0	1	0	0	0

În acest exerciţiu veţi crea un *clasificator discriminativ* pentru acest set de date (desemnat în cele ce urmează cu $\mathcal{D}$), aplicând raţionamentul care a fost descris la problema 38.

a. Conform problemei 38, pentru modelarea dataset-ului nostru vom folosi parametrii următori:

$$\theta_{10} = P(X_1 = 1|Y = 0),\ \theta_{11} = P(X_1 = 1|Y = 1),$$
$$\theta_{20} = P(X_2 = 1|Y = 0),\ \theta_{21} = P(X_2 = 1|Y = 1),$$
$$\theta_{30} = P(X_3 = 1|Y = 0),\ \theta_{31} = P(X_3 = 1|Y = 1),$$
$$\theta_{40} = P(X_4 = 1|Y = 0),\ \theta_{41} = P(X_4 = 1|Y = 1),$$
$$\theta_y = P(Y = 1).$$

Scrieţi funcţia de verosimilitate a setului de date $\mathcal{D}$ în raport cu parametrii de mai sus (desemnaţi în cele ce urmează cu vectorul θ), iar apoi calculaţi estimările de verosimilitate maximă (engl., Maximul Likelihood, MLE) pentru aceşti parametri.

b. Considerăm $\ell(\theta)$ funcţia de log-verosimilitate pentru setul de date $\mathcal{D}$. Pornind de la expresia funcţiei $\ell(\theta)$, demonstraţi că aceasta este o funcţie strict concavă (deci admite un singur punct de maxim, reprezentat chiar de estimările în sens MLE pe care le-aţi calculat la punctul a).

***Indicaţie*: Arătaţi că matricea hessiană a funcţiei $\ell(\theta)$ (notată aici cu $H(\theta)$) este negativ definită. Pentru aceasta, este suficient să arătaţi că sunt satisfăcute următoarele două *condiţii suficiente*: i. matricea $H(\theta)$ este diagonală (adică toate elementele ei sunt 0, cu excepţia eventuală a elementelor de pe diagonala principală) şi ii. toate elementele de diagonala principală a matricei $H(\theta)$ sunt strict negative.**

c. Fie următorul set de date de test:

[456] Am făcut aceste schimbări de notaţie pentru a face mai uşor legătura cu rezultatele de la problema 38. În rezolvarea problemei 38 se demonstrează i. faptul că antrenarea clasificatorului Bayes Naiv corespnde maximizării unei funcţiei de verosimilitate a datelor şi ii. regula de decizie a acestui clasificator este echivalentă cu un clasificator liniar.

	X_1	X_2	X_3	X_4	Y
U	**0**	**1**	**1**	**1**	**?**
V	**1**	**1**	**0**	**1**	**?**
W	**1**	**1**	**0**	**0**	**?**

Folosind estimările de verosimilitate maximă $\hat{\theta}$ care au fost obţinute la punctul a, aplicaţi regula de decizie a algoritmului [de tip generativ] Bayes Naiv pentru a clasifica instanţele de test U, V şi W.

d. Folosiţi formulele care au fost deduse la problema 38.c, şi anume

$$w_0 = \sum_{j=1}^{n} \ln \frac{1-\theta_{j1}}{1-\theta_{j0}} + \ln \frac{\theta_y}{1-\theta_y}, \quad w_j = \ln \frac{\theta_{j1}(1-\theta_{j0})}{\theta_{j0}(1-\theta_{j1})},\ j = 1, \ldots, n, \tag{241}$$

unde n este numărul de atribute de intrare, pentru a calcula ponderile w_0, w_1, w_2, w_3 şi w_4 corespunzătoare datasetului $\mathcal{D}$, care a fost dat în enunţ. Apoi folosiţi aceste ponderi pentru a scrie separatorul liniar $w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 = 0$. (Regula de decizie sub forma discriminativă va fi $P(y = 1|x) \geq P(y = 0|x) \Leftrightarrow w_0 + w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 \geq 0$.)

e. Arătaţi cum sunt clasificate instanţele de test U, V şi W, folosind regula de decizie discriminativă care a fost dedusă la punctul d. Verificaţi că rezultatele obţinute coincid cu cele de la punctul c.

40. **(Modele probabiliste [generative] corespunzătoare algoritmilor Bayes Naiv şi Bayes Optimal; estimarea parametrilor)**

prelucrare de Liviu Ciortuz, după
□ • ○ ***CMU, 2009 spring, Tom Mitchell, HW2, pr. 2***

***Precizare*: Pe tot parcursul acestui exerciţiu, atunci când vom folosi termenul de estimare, se va considera că este vorba despre estimare în sensul verosimilităţii maxime (engl., maximum likelihood estimation, MLE), nu despre estimare în sensul probabilităţii maxime a posteriori (engl., maximum a posteriori probability, MAP).**

Presupunem că dispunem de un set de date de antrenament S, format din exemple pozitive (etichetate cu $y = 1$) şi exemple negative (etichetate cu $y = 0$), fiecare instanţă având $n = 10$ atribute cu valori binare, generate conform următorului model, M_{indep}, care foloseşte independenţa condiţională a acestor atributes, în raport cu eticheta / clasa:

$$M_{indep} : \forall i : 1 \leq i \leq n, \forall x, y \in \{0, 1\} \ \ P(X_i = x|Y = y) = p^{indep}_{i,x,y}.$$

Cu alte cuvinte, fiecare exemplu $((x_1, x_2, \ldots, x_n), y)$ este generat alegând mai întâi o valoare y pentru clasa Y şi, după aceea, câte o valoare x_i pentru fiecare atribut X_i, conform probabilităţii $p^{indep}_{i,x_i,y}$. Fiecare atribut x_i este astfel determinat în mod independent faţă de celelalte atribute. Vom presupune de asemenea că probabilitatea de a alege clasa $Y = 1$ este de 0.5, adică $P(Y = 1) = P(Y = 0) = 0.5$.

Partea A.

a. Care este numărul parametrilor „liberi“ (engl., free) din acest model, $p_{i,x,y}^{indep}$? (Menţionaţi în mod explicit care sunt aceşti parametri.)

b. Vom presupune acum că ni se dă o „variantă“ particulară a acestui model, în care parametrii au următoarele valori: $\forall i : p_{i,1,1}^{indep} = 0.8$ şi $p_{i,1,0}^{indep} = 0.6$. Aşadar, probabilitatea ca un atribut oarecare să aibă valoarea 1 este 0.8 pentru exemple pozitive, şi respectiv 0.6 pentru exemple negative. Presupunem de asemenea că ni se dă un singur exemplu din clasa pozitivă, $(\bar{x}_{test}, y_{test}) = ((1,1,0,0,1,1,0,1,1,1),1)$.

Care este probabilitatea ca instanţa neetichetată $\bar{x}_{test} = (1,1,0,0,1,1,0,1,1,1)$ să fie generată de către (sau: în) clasa pozitivă? Cu alte cuvinte, cât este $P(\bar{x}_{test}|y = 1, M_{indep})$?

c. Conform modelului definit la punctul b, cât este $P(y = 1|\bar{x}_{test}, M_{indep})$? Aşadar, cât este probabilitatea a posteriori ca instanţa $\bar{x}_{test}$ să aparţină clasei 1?[457]

d. Vom folosi acum setul de date de antrenament S (vedeţi precizările de mai sus). Referindu-ne mai întâi la parametrul $p_{i,x,1}^{indep}$ al modelului, vă cerem să scrieţi regula de calcul pentru estimarea sa de verosimilitate maximă, $\hat{p}_{i,x,1}^{indep}$, în funcţie de *datele de antrenament* (nu în funcţie de parametrii „variantei“ modelului care a fost definită la punctul b de mai sus).
Similar, pentru parametrul $p_{i,x,0}^{indep}$ al modelului, scrieţi regula de calcul pentru estimarea de verosimilitate maximă, $\hat{p}_{i,x,0}^{indep}$, în funcţie de *datele de antrenament*.

Partea B.

Considerăm următorul set de date de antrenament (S'):

X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	Y
0	1	0	1	1	1	0	1	1	0	0
1	1	0	1	1	1	0	1	1	0	0
1	1	1	0	1	1	1	0	1	0	0
1	1	1	1	0	1	0	0	1	0	1
0	1	0	0	0	1	1	1	0	0	1
1	1	0	0	0	1	1	0	0	0	1

e. Bazat pe acest set de date de antrenament, calculaţi estimările de verosimilitate maximă $\hat{p}_{i,x,y}^{indep}$ pentru parametrii modelului M_{indep}.

f. Distribuţia de probabilitate Beta[458] **este adeseori folosită ca distribuţie a priori pentru a evita ca estimările parametrilor să aibă valoarea zero. Dacă presupunem că folosim probabilităţi a priori Beta pentru fiecare dintre parametrii modelului M_{indep} şi considerăm că parametrii distribuţiei Beta sunt**

[457] Odată calculată probabilitatea $P(y = 1|\bar{x}_{test}, M_{indep})$, vom putea spune cum va clasifica algoritmul Bayes Naiv instanţa $\bar{x}_{test}$ (în modelul M_{indep}).

[458] Vedeţi problema 130.d — şi, ca aplicaţie pentru estimare în sens MAP, problema 41.B — ambele de la capitolul de *Fundamente*.

$\alpha = \beta = 2$, cât sunt estimările în sens MAP pentru parametrii modelului M_{indep}?[459]

Partea C.

Vom considera acum un nou model, M_{dep}, pentru care nu se face nicio presupunere privitoare la posibilele dependenţe dintre atribute:

$$M_{dep} : \forall \bar{x} : \bar{x} \in \{0,1\}^n, \forall y \in \{0,1\} \;\; P(\bar{X} = \bar{x} | Y = y) = p^{dep}_{\bar{x},y}.$$

Cu alte cuvinte, fiecare exemplu $((x_1, x_2, \ldots, x_n), y)$ este generat alegând mai întâi o valoare y pentru clasa Y şi, după aceea, alegând un întreg vector $\bar{x} = (x_1, x_2, \ldots, x_n)$. Probabilitatea de a alege acest vector este dată de parametrul $p^{dep}_{\bar{x},y}$. Ca şi mai sus, vom presupune că probabilitatea de a alege clasa $Y = 1$ este de 0.5. Aşadar, $P(Y = 1) = P(Y = 0) = 0.5$.

g. Cât este numărul parametrilor „liberi", $p^{dep}_{\bar{x},y}$, din acest model? Cum este acest număr, comparat cu numărul parametrilor „liberi" din modelul M_{indep}?

h. Prin $\bar{x}_{test}$ ne vom referi aici la instanţa de test de la punctul b. În raport cu noul model, M_{dep}, pentru a calcula probabilitatea a posteriori $P(Y|\bar{x}_{test})$ — aşa cum se face atunci când se aplică algoritmul Bayes Optimal — avem nevoie ca mai întâi să estimăm parametrii $p^{dep}_{\bar{x}_{test},1}$ şi $p^{dep}_{\bar{x}_{test},0}$.

Vom presupune că sunt date câte 500 de exemple de antrenament în fiecare dintre cele două clase ($y = 0$ şi respectiv $y = 1$), care sunt generate conform modelului M_{indep}. Când vi se va cere să calculaţi estimările $\hat{p}^{dep}_{\bar{x}_{test},1}$ şi $\hat{p}^{dep}_{\bar{x}_{test},0}$ în modelul M_{dep} — vedeţi mai jos — veţi folosi acest set de date de antrenament (S'').

(*i.*) Pentru parametrul $p^{dep}_{\bar{x}_{test},1}$, scrieţi regula de calcul pentru estimarea de verosimilitate maximă, $\hat{p}^{dep}_{\bar{x}_{test},1}$, în funcţie de datele de antrenament (ca şi la punctul d).[460]

(*ii.*) Considerând că datele de antrenament au fost generate conform modelului M_{indep} folosind parametrii daţi la punctul b, care este probabilitatea ca estimarea $\hat{p}^{dep}_{\bar{x}_{test},1}$ [făcută conform punctului i] să fie zero? Altfel spus, determinaţi $P(\hat{p}^{dep}_{\bar{x}_{test},1} = 0)$, probabilitatea aceasta calculându-se în raport cu diferitele rezultate ale „experimentului" care se referă la generarea datelor de antrenament conform modelului M_{indep}. (Aşadar, se cere să se calculeze probabilitatea ca instanţa $\bar{x}_{test}$ să nu fie prezentă între cele 500 de exemple pozitive care există în setul de date de antrenament.)

(*iii.*) Cât este $P(\hat{p}^{dep}_{\bar{x}_{test},0} = 0)$, presupunând din nou că datele au fost generate folosind modelul M_{indep} de la punctul b? (Cu alte cuvinte, se cere să se calculeze probabilitatea ca instanţa $\bar{x}_{test}$ să nu fie prezentă între cele 500 de exemple negative din setul de date de antrenament.)

[459] *Observaţie*: Dacă o variabilă X_i nu ar fi fost binară şi, deci, în locul distribuţiei Bernoulli ar fi urmat distribuţia categorială, pe post de distribuţie a priori am fi putut folosi o distribuţie Dirichlet. Vedeţi problema 129 de la capitolul de *Fundamente*.

[460] *Observaţie*: Întrucât datele de antrenament sunt generate aleatoriu, putem vedea estimarea / estimatorul $\hat{p}^{dep}_{\bar{x}_{test},1}$ ca fiind o variabilă aleatoare. La punctul *ii* de mai jos se va cere să se calculeze probabilitatea ca această variabilă aleatoare să ia o anumită valoare (şi similar pentru $\hat{p}^{dep}_{\bar{x}_{test},0}$, la punctul *iii*).

i. Formulaţi în una sau două fraze cum anume se leagă problema aceasta de discuţia pe care am făcut-o la curs în legătură cu algoritmul Bayes Naiv şi independenţa condiţională.

41. **(Algoritmul Bayes Naiv: comparaţie cu alţi clasificatori)**

prelucrare de L. Ciortuz, după
∗ ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, midterm, pr. 6.b***

Considerăm un clasificator Bayes Naiv care lucrează pe un set de date descrise de atributele de intrare A şi B şi atributul de ieşire Y. (Exemplu: $Y = A \text{ XOR } B$). A şi B sunt variabile aleatoare independente între ele.

a. Este posibil ca, în situaţia aceasta, vreun alt clasificator — de exemplu, ID3, regresia logistică (eventual kernel-izată) sau SVM — să lucreze mai bine decît clasificatorul Bayes Naiv?

b. Care este motivul?

42. **(Comparaţie între clasificatorul Bayes Naiv şi algoritmul ID3)**

□ • ○ ***CMU, 2010 fall, Ziv Bar-Joseph, midterm, pr. 5.b***

Care dintre afirmaţiile de mai jos sunt adevărate atât pentru clasificatorul Bayes Naiv cât şi pentru algoritmul ID3 pentru învăţarea de arbori de decizie? (Veţi putea alege nu neapărat una singură dintre aceste afirmaţii.)

1. În cazul ambilor clasificatori se presupune că orice pereche de atribute X_i şi X_j cu $i \neq j$ — văzute ca variabile aleatoare — sunt independente.
2. În cazul ambilor clasificatori se presupune că orice pereche de atribute X_i şi X_j cu $i \neq j$ sunt dependente.
3. În cazul ambilor clasificatori se presupune că orice pereche de atribute sunt independente în raport cu eticheta (adică variabila care reprezintă clasa).
4. În cazul ambilor clasificatori se presupune că orice pereche de atribute sunt dependente în raport cu eticheta.

43. **(Algoritmul Bayes Naiv – clasificator de tip MAP; o condiţie suficientă pentru echivalenţa cu clasificarea de tip ML)**

□ • ○ ***CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, HW3, pr. 4.2***

Algoritmul Bayes Naiv asociază unui exemplu x clasa c dacă aceasta maximizează probabilitatea $P(c|x)$.
Formulaţi o condiţie suficientă pentru ca această „regulă" [din formularea algoritmului Bayes Naiv] echivalentă cu selecţionarea acelei clase c care maximizează probabilitatea $P(x|c)$?

44. **(Algoritmii Bayes Naiv şi Bayes Optimal: Adevărat sau Fals?)**

• ○ ***CMU, 2005 spring, C. Guestrin, T. Mitchell, midterm, pr. 2.b.5***
CMU, 2011 spring, Tom Mitchell, midterm, pr. 1.1.ab

a. Clasificatorul Bayes Optimal poate să obţină rata de eroare 0 [la antrenare] pentru orice set de date. Justificaţi.

b. Dacă antrenăm un clasificator Bayes Naiv folosind un număr infinit de date de antrenament care satisfac toate presupoziţiile luate în calcul de acest tip de modelare (de exemplu, independenţa condiţională), atunci acest clasificator va produce *eroare* zero pe setul de exemple de antrenament considerat.

c. Dacă antrenăm un clasificator Bayes Naiv folosind un număr infinit de date de antrenament care satisfac toate presupoziţiile luate în calcul de acest tip de modelare (de exemplu, independenţa condiţională), atunci acest clasificator va produce *eroare „adevărată“* zero pentru exemple de test generate conform aceleiaşi distribuţii.

2.2.3 Clasificare bayesiană [cu atribute de intrare] de tip gaussian

45. **(Distribuţia gaussiană unidimensională: exemplificare pentru estimarea mediei în sens MLE şi calcularea unei probabilităţi a posteriori)**

• ○ ∗ ***U. Edinburgh, 2009 fall, C. Williams, V. Lavrenko, HW2, pr. 1***

Considerăm un set de date [de antrenament] care constă din *exemple* pentru două *clase*. Exemplele pentru clasa 1 sunt 0.5, 0.1, 0.2, 0.4, 0.3, 0.2, 0.2, 0.1, 0.35, 0.25, **iar exemplele pentru clasa 2 sunt** 0.9, 0.8, 0.75, 1.0**.**

a. Vă cerem să antrenaţi (engl., fit) câte o distribuţie gaussiană unidimensională pentru fiecare din cele două clase, folosind metoda estimării în sensul verosimilităţii maxime (engl., Maximum Likelihood estimation, MLE). Veţi presupune că varianţa distribuţiei corespunzătoare clasei 1 este 0.0149**, iar varianţa distribuţiei pentru clasa 2 este** 0.0092**.**

b. Estimaţi de asemenea în sens MLE probabilităţile *a priori* de selecţie pentru [generarea de exemple din] cele două clase.

c. Care este probabilitatea *a posteriori* ca punctul $x = 0.6$ să aparţină clasei 1?

46. **(Algoritmul Bayes [Naiv] gaussian: exemplificare pe date din $\mathbb{R}$, graniţe de decizie)**

□ • ○ ***CMU, 2009 spring, Tom Mitchell, midterm, pr. 5***

În acest exerciţiu vom considera mai mulţi clasificatori de tip Bayes Naiv gaussian (GNB) pentru un set de date având un singur atribut x şi două

clase, 0 şi 1.[461] **Ca de obicei pentru clasificatori bayesieni, vom clasifica o instanţă x ca aparţinând clasei 1 dacă**

$$P(y=1|x) \geq P(y=0|x) \Leftrightarrow \ln \frac{P(y=1|x)}{P(y=0|x)} \geq 0.$$

Vom folosi notaţia $\mathcal{N}(\mu, \sigma^2)$ pentru a desemna o distribuţie normală / gaussiană de medie μ şi varianţă σ^2. Vi se va cere să scrieţi expresia analitică a separatorului decizional (sau, a graniţei de decizie) pentru fiecare model de tip GNB prezentat mai jos.

a. Fie următorul clasificator Bayes [Naiv] gaussian:

$$\begin{aligned} x|y=0 &\sim \mathcal{N}(0,1) \\ x|y=1 &\sim \mathcal{N}(2,1) \\ P(y=1) &= 0.5 \end{aligned}$$

Este oare graniţa de decizie a acestui clasificator GNB liniară? Altfel spus, puteţi scrie o expresie de forma $w_0 + w_1 x \geq 0$ care reprezintă graniţa de decizie a acestui model GNB? În cazul afirmativ, calculaţi valorile w_0 şi w_1.[462] **În cazul negativ, justificaţi.**

b. Acum vom considera un alt clasificator de tip GNB. Noii parametri pentru cele două distribuţii gaussiene sunt:

$$\begin{aligned} x|y=0 &\sim \mathcal{N}(0,\, 1/4) \\ x|y=1 &\sim \mathcal{N}(0,1) \\ P(y=1) &= 0.5 \end{aligned}$$

Este graniţa de decizie a acestui model GNB liniară? Dacă da, marcaţi care dintre opţiunile de mai jos constituie graniţa de decizie corectă. Dacă alegeţi opţiunea (v), daţi o scurtă explicaţie.

(i) **Alege clasa 1 dacă $x \geq 1/2$.**

(ii) **Alege clasa 1 dacă $x \leq -1/2$.**

(iii) **Alege clasa 1 dacă $x \leq 1$.**

(iv) **Alege clasa 1 dacă $x \geq -1$.**

(v) **Graniţa de decizie nu este liniară.**

c. Acum vom considera o graniţă de decizie *pătratică*. La o graniţă de decizie pătratică adăugăm o nouă trăsătură, x^2, instanţei de antrenament $< x, y >$. Astfel, instanţa $< x, y >$ va fi transformată în $< x^2, x, y >$. O graniţă de decizie *liniară* pentru acest set de date modificat — aşadar, determinată de $w_0, w_1, w_2 \in \mathbb{R}$ cu proprietatea $w_0 + w_1 x + w_2 x^2 \geq 0 \Leftrightarrow P(y=1|x) \geq P(y=0|x)$ pentru $\forall x \in \mathbb{R}$ — produce o graniţă de decizie *pătratică* pentru setul de date original.

Este oare posibil să găsim o graniţă de decizie pătratică care corespunde exact graniţei de decizie a modelului GNB de la punctul b? Dacă da, marcaţi care dintre opţiunile de mai jos constituie graniţa de decizie corectă. Dacă alegeţi opţiunea (iv), daţi o scurtă explicaţie.

[461] Fiind dat un singur atribut de intrare, putem renunţa la termenul „Naiv“ din expresia care desemnează tipul clasificatorului.

[462] Adică, găsiţi w_0 şi $w_1 \in \mathbb{R}$ astfel încât $P(y=1|x) \geq P(y=0|x) \Leftrightarrow w_o + w_1 x \geq 0$ pentru $\forall x \in \mathbb{R}$.

(*i*) **Alege clasa 1 dacă** $x \leq -0.68$ **or** $x \geq 0.68$**.**

(*ii*) **Alege clasa 1 dacă** $-0.95 \leq x \leq 0.95$**.**

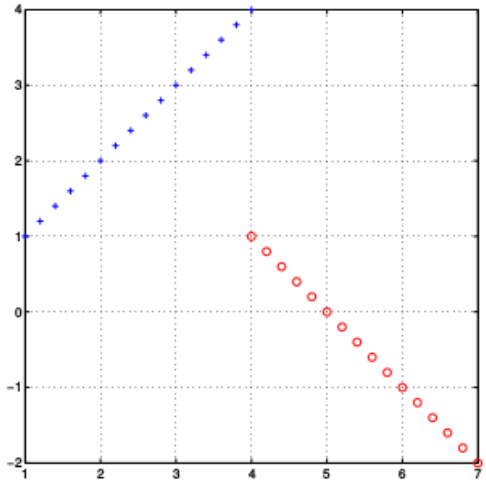
(*iii*) **Alege clasa 1 dacă** $-0.68 \leq x \leq 0.68$**.**

(*iv*) **Graniţa de decizie nu este pătratică.**

47. **(Algoritmul Bayes Naiv gaussian: aplicare pe date din $\mathbb{R}^2$)**

prelucrare de L. Ciortuz, după
• ∗ CMU, 2007 spring, midterm, pr. 4
CMU, 2017 fall, Nina Balcan, midterm, pr. 4.7

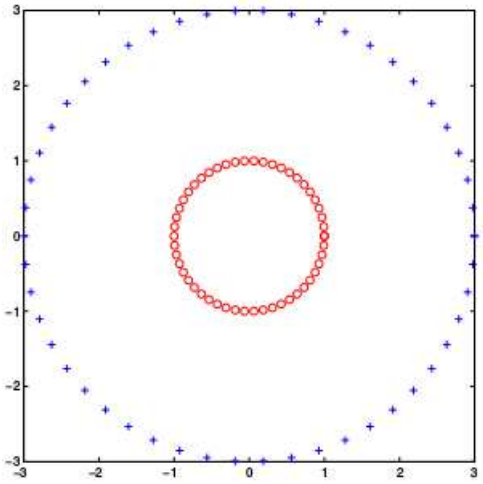
a. Considerăm setul de date din figura alăturată, în care sunt două clase, desemnate cu + şi respectiv ∘. Număr de instanţe din cele două clase este acelaşi. Fiecare instanţă are două trăsături, care corespund coordonatelor X_1 şi X_2, ambele având valori numere reale.



Determinaţi în mod analitic şi apoi trasaţi *graniţa de decizie* pe care o va învăţa clasificatorul Bayes Naiv gaussian pe acest set de date.

Ce s-ar întâmpla dacă am încerca să folosim clasificatorul Bayes Optimal gaussian în locul clasificatorului Bayes Naiv gaussian pe aceste date?

b. Desenaţi graniţa de decizie determinată de algoritmul Bayes Naiv gaussian pentru setul de instanţe din figura alăturată. (Presupunem că probabilităţile de selecţie ale celor două clase sunt egale.) Justificaţi cât se poate de riguros.



48. **(Algoritmul Bayes Naiv gaussian: chestiuni legate de estimarea parametrilor distribuţiilor condiţionale; algoritmul Bayes Optimal gaussian: numărul parametrilor necesari de estimat)**

prelucrare de Liviu Ciortuz, după

■ □ • ○ ***CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, HW2, pr. 5.ab***

Fie un set de date de antrenament pentru care $Y \in \{0, 1\}$ reprezintă clasele / etichetele, iar $X \in \mathbb{R}^d$ reprezintă vectorii de trăsături (d-dimensionali).

La punctele a şi b ale acestui exerciţiu vom lucra cu modelul Bayes Naiv gaussian, în care distribuţia probabilistă condiţională pentru fiecare trăsătură este o distribuţie gaussiană unidimensională / univariată.

a. Considerând n instanţe de antrenament generate în mod independent, $\{(X^{(1)}, Y^{(1)}), \cdots, (X^{(n)}, Y^{(n)})\}$, calculaţi estimările de verosimilitate maximă (MLE) pentru parametrii distribuţiilor probabiliste urmate de variabilele condiţionale $X^{(j)}|Y$, pentru $j = 1, \ldots, d$.

b. Câţi parametri trebuie estimaţi pentru modelul Bayes Naiv gaussian? Veţi presupune că parametrul pentru distribuţia a priori a lui Y este deja cunoscut.

c. În modelul Bayes Optimal [complet] Gaussian, presupunem că distribuţia probabilistă condiţională $Pr(X|Y)$ este o gaussiană multidimensională / multivariată, $X|Y \sim \mathcal{N}(\mu_Y, \Sigma_Y)$, unde μ_Y este vectorul medie, iar $\Sigma_Y \in \mathbb{R}^{d\times d}$ este matricea de covarianţă.

Câţi parametri trebuie estimaţi pentru modelul Bayes complet gaussian? Veţi presupune, ca şi la punctul b, că parametrul pentru distribuţia a priori a lui Y este deja cunoscut.

***Observaţie*: Prin numărul de *parametri* se va înţelege aici *nu* numărul de vectori de medii — din $\mathbb{R}^d$ — şi numărul de matrice de covarianţă — din $\mathbb{R}^{d\times d}$ — de estimat, *ci* numărul total de componente (adică, numere reale) ale tuturor acestor vectori de medii şi matrice de covarianţă, care sunt de estimat.**

49. **(Algoritmul Bayes Naiv gaussian: legătura dintre estimarea parametrilor şi maximizarea log-verosimilităţii datelor)**

□ • ***CMU, 2017 fall, Nina Balcan, Practice questions, pr. 1.2***

După cum ştiţi, în clasificarea binară obiectivul este să putem prezice un „target“ binar $Y \in \{0, 1\}$ pentru o instanţă oarecare de test $X = (X_1, \ldots, X_d)$. Aici vom presupune că $X \in \mathbb{R}^d$. Pentru a prezice Y, este suficient să estimăm funcţia $P(Y|X)$. Algoritmul Bayes Naiv calculează $P(Y|X)$ astfel:

$$P(Y|X) = \frac{P(Y, X)}{P(X)} = \frac{P(Y)\prod_{j=1}^{d} P(X_j|Y)}{P(X)}.$$

Aici vom presupune că toate distribuţiile condiţionate în raport cu clasa sunt de tip gaussian.[463] Aceasta înseamnă:

$$P(X_j|Y = k) = \mathcal{N}(X_j|\mu_{j,k}, \sigma^2_{j,k}),$$

[463] LC: De fapt, concluzia pe care o vom formula mai jos se menţine şi atunci când în locul distribuţiilor gaussiene lucrăm cu distribuţii Bernoulli (sau, mai general, cu distribuţii categoriale).

unde $\mu_{j,k}$ şi $\sigma^2_{j,k}$ sunt media şi respectiv varianţa. Să presupunem că avem n instanţe de antrenament $\{X^{(i)}, Y^{(i)}\}_{i=1}^n$. Pentru a estima parametrii modelului pornind de la aceste date, vom învăţa (engl., fit) câte o distribuţie gaussiană univariată (sau, unidimensională) pentru fiecare trăsătură / atribut de intrare (X_j) şi pentru fiecare clasă ($Y = k$).

Arătaţi că acest estimator de tip Bayes Naiv gaussian de fapt rezolvă următoarea *problemă* de estimare în sensul verosimilităţii maxime (engl., maximum likelihood estimation, MLE):

$$\max \frac{1}{n} \sum_{i=1}^{n} \ln P(X^{(i)}, Y^{(i)}).$$

50. **(Algoritmul Bayes Naiv gaussian, cazul $\sigma_{i0} \neq \sigma_{i1}$: raportul cu regresia logistică)**

□ • · ***CMU, 2011 spring, Tom Mitchell, HW2, pr. 2.1***

În această problemă vom modifica clasificatorul Bayes Naiv gausian care a fost prezentat la problema 17 şi-l vom face ceva mai general, eliminând presupunerea că σ_i, deviaţia standard pentru distribuţia urmată de $P(X_i|Y = k)$ nu depinde de eticheta k. Prin urmare, pentru fiecare X_i şi fiecare k, unde $i \in \{1, 2, \ldots, n\}$ şi $k \in \{0, 1\}$, distribuţia urmată de $P(X_i|Y = k)$ este de tipul $\mathcal{N}(\mu_{ik}, \sigma^2_{ik})$.

Este oare şi noua formă a probabilităţii condiţionate $P(Y|X)$, care este implicată de acest clasificator Bayes Naiv gausian [mai general], la fel cu forma regresiei logistice? Pentru a justifica răspunsul dumneavoastră, calculaţi noua formă a probabilităţii condiţionate $P(Y|X)$.

51. **(Algoritmul Bayes Naiv: deducerea regulii de decizie pentru cazul când toate atributele sunt gaussiene, în afară de unul singur, care este de tip boolean)**

□ • · ***CMU, 2010 fall, Aarti Singh, HW1, pr. 4.1***

Considerăm funcţia de învăţare $X \to Y$, unde $Y \in \{T, F\}$ reprezintă eticheta clasei, iar X este n-uplul $(X_1, X_2, \ldots, X_n)$, cu X_1 variabilă booleană şi $X_2, \ldots, X_n$ variabile continue. Presupunem că în cazul fiecărei variabile continue X_i, distribuţia $P(X_i|Y = y)$ este de tip gaussian de medie $\mu_{i,y}$ şi varianţă $\sigma^2_{i,y}$. Vrem ca în acest cadru să antrenăm un clasificator de tip Bayes Naiv, deci care să lucreze cu presupoziţia de independenţă condiţională a variabilelor X_i (cu $i = 1, \ldots, n$) în raport cu variabila Y.

a. Enumeraţi toţi parametrii care trebuie estimaţi pentru a putea clasifica o instanţă nouă. Care este numărul total al acestor parametri?

b. Elaboraţi formula de calcul pentru $P(Y|X)$ în funcţie de aceşti parametri şi de atributele / trăsăturile X_i. Trataţi cazul particular când $\sigma^2_{i,T} = \sigma^2_{i,F}$.

52. **(Algoritmul Bayes Optimal gaussian: câteva chestiuni de bază)**

□ • ○ ***CMU, 2017 fall, Nina Balcan, midterm, pr. 4.1-3,6***

Fie două variabile aleatoare, $X \in \mathbb{R}^d$ şi $Y \in \{0, 1\}$. Vom considera că atât $p_0(X = x) \overset{not.}{=} p(x|Y = 0)$ cât şi $p_1(X = x) \overset{not.}{=} p(x|Y = 1)$ sunt distribuţii gaussiene multidimensionale:

$$p_y(x) = \frac{1}{(2\pi)^d |\Sigma_y|^{1/2}} \exp\Big(-\frac{1}{2}(x - \mu_y)^\top \Sigma_y^{-1}(x - \mu_y)\Big), \text{ pentru } y \in \{0, 1\}.$$

Aşadar, putem scrie $X|(Y = y) \sim \mathcal{N}(\mu_y, \Sigma_y)$. Mai departe, vom nota *probabilităţile de selecţie* ale celor două clase cu $\pi_y \overset{not.}{=} P(Y = y)$ pentru $y \in \{0, 1\}$. Vom lucra cu un set de n „realizări“ i.i.d. ale cuplului de variabile (X, Y), şi anume $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$.

a. Clasificatorul Bayes Optimal gaussian va prezice eticheta / clasa unei noi instanţe x conform următoarei *regule de decizie*: $\hat{y} = I(\pi_1\, p_1(x) > \pi_0\, p_0(x))$, unde $I(.)$ este funcţie-indicator.[464] Demonstraţi că această regulă se poate scrie în mod echivalent astfel:

$$\hat{y} = \tag{242}$$
$$I((x-\mu_1)^\top \Sigma_1^{-1}(x-\mu_1) - 2\ln\pi_1 + \ln(|\Sigma_1|) < (x-\mu_0)^\top \Sigma_0^{-1}(x-\mu_0) - 2\ln\pi_0 + \ln(|\Sigma_0|)).$$

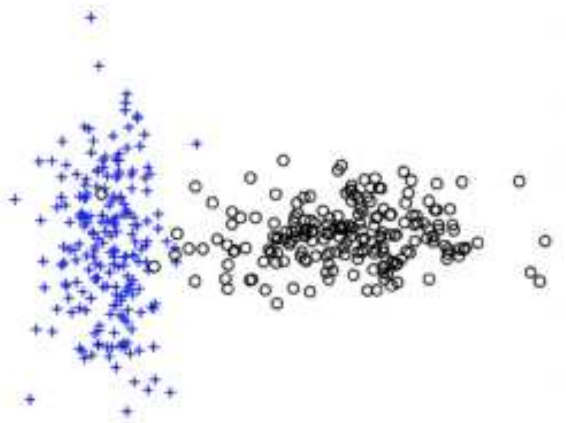
b. Este oare *graniţa de decizie* determinată de algoritmul Bayes Optimal gaussian liniară? Dacă da, explicaţi de ce este aşa. Dacă nu, precizaţi în ce condiţii / cazuri această graniţă de decizie este liniară.

c. Dat fiind setul de date de antrenament S, precizaţi care este estimarea de verosimilitate maximă (MLE) pentru vectorii de medii μ_y, precum şi pentru probabilităţile de selecţie π_y, pentru $y \in \{0, 1\}$. Justificaţi.

***Sugestie*: Pentru estimarea mediilor μ_y, vă sugerăm să faceţi referire la problema 53.a de la capitolul *Fundamente*, iar pentru estimarea probabilităţilor de selecţie π_y, vedeţi problema 125.a (sau problema 43.c) de la capitolul *Fundamente*.**

d. Desenaţi graniţa de decizie determinată de algoritmul Bayes Optimal gaussian pentru setul de instanţe din figura alăturată. (Presupunem că $\pi_0 = \pi_1 = 0.5$.) Justificaţi cât se poate de riguros.

Care dintre cele două trăsături (corespunzătoare celor două coordonate din planul euclidian) are o putere discriminativă mai mare?



[464] Aceasta înseamnă că $\hat{y} = 1$ dacă x satisface condiţia care a fost scrisă ca argument al funcţiei $I(.)$, şi $\hat{y} = 0$ în cazul contrar.

53. **(Algoritmul Bayes Optimal gaussian: aplicare pe setul de date XOR)**

□ • ○ ∗ ***CMU, (?) 15-781, midterm example questions, pr. 3.2.b***

În acest exerciţiu ne vom referi la învăţarea conceptului XOR. Aşadar, vom considera că avem două atribute de intrare, x şi y, care iau valorile -1 şi $+1$, iar outputul va fi pozitiv dacă şi numai dacă $x \neq y$.

Ce se întâmplă dacă încercăm să aplicăm clasificatorul Bayes Optimal gaussian pe setul de date corespunzător conceptului XOR? (Vom presupune că acest clasificator este capabil să *estimeze* orice matrice de covarianţă.)

54. **(Clasificatorul Bayes Optimal gaussian: aplicare în $\mathbb{R}^2$; graniţe de decizie, în cazul $\Sigma_0 \neq \Sigma_1$)**

• ∗ ***CMU, 2010 fall, Aarti Singh, midterm, pr. 2.4***

Fie următoarea problemă de clasificare în $\mathbb{R}^2$:
Mai întâi presupunem că $P(y = 0) = P(y = 1) = 1/2$. De asemenea, presupunem că funcţiile densitate de probabilitate (p.d.f.) condiţionale în raport cu clasa / eticheta sunt de tip gaussian, cu media $\mu_0 \in \mathbb{R}^2$ şi matricea de covarianţă Σ_0 pentru clasa 0, respectiv media $\mu_1 \in \mathbb{R}^2$ şi matricea de covarianţă Σ_1 pentru clasa 1. Mai mult, presupunem că $\mu_0 = \mu_1 \overset{not.}{=} (\mu^1, \mu^2) \in \mathbb{R}^2$.

Pentru cazul

$$\Sigma_0 = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \qquad \Sigma_1 = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}$$

trasaţi într-un sistem de coordonate cartezian *curbe de izocontur* corespunzătoare celor două p.d.f-uri condiţionale, pe care le veţi nota cu $p(x|y = 0)$ şi respectiv $p(x|y = 1)$.

Apoi trasaţi *graniţele de decizie* (engl., decision boundaries) determinate de clasificatorul Bayes Optimal gaussian. Determinaţi în mod riguros — adică, plecând de la *regula de decizie* a acestui clasificator — ecuaţia satisfăcută de aceste graniţe de decizie.

În final, indicaţi regiunile în care clasificatorul va prezice clasa 0, respectiv clasa 1.

***Indicaţie*: Vă reamintim că pentru o variabilă aleatoare $X : \Omega \to \mathbb{R}^n$, care este reprezentată pe componente ca vector-coloană $X = (X_1, \ldots, X_n)^\top$ şi care urmează o distribuţie gaussiană având media $\mu \in \mathbb{R}^n$ şi matricea de covarianţă Σ, funcţia densitate de probabilitate are forma analitică următoare:**

$$p(x; \mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right),$$

unde $\top$ reprezintă operaţia de transpunere.[465]

[465] Observaţi că notaţia $(x-\mu)^\top \Sigma^{-1}(x-\mu)$ este matriceală. În urma efectuării operaţiilor de înmulţire rezultă o matrice de tip 1×1. În expresia $\exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right)$, matricea menţionată mai sus este de fapt substituită cu un număr real, şi anume cu unicul ei element.

55. **(Clasificare bayesiană gaussiană: câteva chestiuni teoretice)**

□ ***Liviu Ciortuz, 2026***

a. Demonstraţi că atunci când la exerciţiul 17 considerăm $d = 1$ şi $\mu_{10} = \mu_{11}$, rezultatul de la punctul e al acestui exerciţiu coincide cu rezultatul de la exerciţiul 16.b.

***Observaţie*: În exerciţiul 17 am presupus de la început că $\sigma_0 = \sigma_1$.**

b. Demonstrţi că atunci când la exerciţiul 18 presupunem că matricele $\Sigma_0 = \Sigma_1$ sunt diagonale, rezultatul de la acest exerciţiu coincide cu rezultatul de la exerciţiul 17.e (referitor la forma regulii de decizie a algoritmului Bayes Naiv gaussian).

c. Demonstraţi că atunci când la exerciţiul 19 avem $\Sigma_0 = \Sigma_1$, rezultatul de la acest exerciţiu coincide cu rezultatul de la exerciţiul 18.

d. Considerăm că vrem să facem clasificare bayesiană gaussiană binară. Demonstraţi că pentru cazul în care avem $d = 2$ dimensiuni, probabilităţile de selecţie sunt egale ($P_1 = P_2 = 1/2$) şi matricele de covarianţă sunt egale şi de forma $\sigma^2 I_2$, algoritmul Bayes Gaussian (Naiv sau Optimal, e acelaşi lucru în acest caz) obţine ca separator decizional mediatoarea segmentului determinat de μ_1 si μ_2, punctele care reprezintă mediile celor două gaussine bidimensionale. (Se consideră $\mu_1 \neq \mu_2$.)

Este oare acest rezultat valabil şi în cazul în care matricele de covarianţă sunt diagonale şi egale, dar nu de forma indicată mai sus?

Dar în cazul $\mu_1 = \mu_2$, $\Sigma = \sigma_1^2 I_2$ şi $\Sigma = \sigma_2^2 I_2$ cu σ_1 diferit de σ_2?

56. **(Clasificarea bayesiană gaussiană vs. regresia logistică şi regresia liniară: Da sau nu?)**

• ○ ***CMU, (?) 15-781, midterm example questions, pr. 3.3***
CMU, 2009 spring, Tom Mitchell, midterm, pr. 1.b

a. Ne întrebăm dacă există vreun clasificator de tip Bayes gaussian pentru [date cu] un singur atribut de intrare x astfel încât, atunci când va fi folosit, să facă următoarele predicţii:

clasa 1 dacă $x < -1$;

clasa 2 dacă $-1 < x < 1$;

clasa 1 dacă $x > 1$.

În cazul afirmativ, precizaţi cum anume se poate construi un astfel de clasificator.

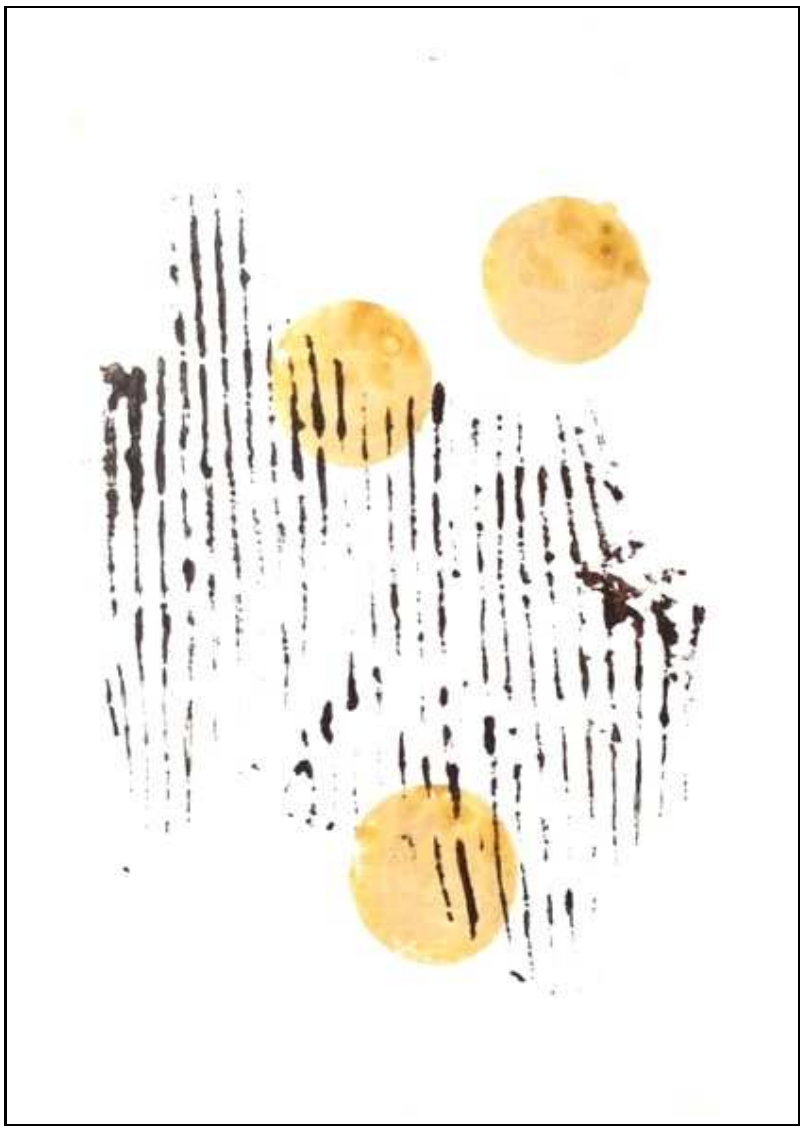
b. Presupunem că antrenăm mai mulţi clasificatori pentru a învăţa funcţia $f : X \to Y$, unde $X =< X_1, X_2, X_3 >$ este vectorul de trăsături. Pentru fiecare dintre următorii clasificatori,

***i.* algoritmul Bayes Naiv [gaussian]**

***ii.* regresia logistică**

iii. **regresia liniară**

indicaţi dacă respectivul clasificator oferă suficiente informaţii ca să putem calcula probabilităţile $P(X_1, X_2, X_3, Y)$. Atunci când răspundeţi cu *da*, precizaţi detaliile. Invers, atunci când răspundeţi cu *nu*, spuneţi [şi] ce anume lipseşte pentru a calcula $P(X_1, X_2, X_3, Y)$.



© *M. Romanică*

3 Învăţare bazată pe memorare

Sumar

Noţiuni preliminare

– **măsuri de distanţă, măsuri de similaritate: ex. 2;**

– **normă într-un spaţiu vectorial; [măsura de] distanţă indusă de către o normă: ex. 7;**

– **k-NN vecinătate a unui punct din $\mathbb{R}^d$.**

Algoritmul k-NN

– **pseudo-cod (cf. cartea ML, pag. 232):**

Training:
Store all training examples.

Classification:
Given a query/test instance x_q,
first locate the k nearest training examples $x_1, \ldots, x_k$,
then take a vote among its k nearest neighbors

$$\underset{v \in V}{\operatorname{argmax}} \sum_{i=1}^{k} 1_{\{f(x_i)=v\}}$$

unde $1_{\{\cdot\}}$ este bine-cunoscuta *funcţie-indicator*, iar V este mulţimea de valori pentru variabila de ieşire.

– **bias-ul inductiv: „Cine se aseamănă se adună“ (sau: „Spune-mi cu cine te împrieteneşti, ca să-ţi spun cine eşti“): ex. 16.a;**

– **exemple (simple) de aplicare: ex. 1, ex. 2;**

– **complexitate de spaţiu: $\mathcal{O}(d\,n)$**
complexitate de timp:

la antrenare: $\mathcal{O}(d\,n)$
la testare: $\mathcal{O}(d\,n \log n)$
[LC: $\mathcal{O}(d\,n\,k \log k)$ pt. $k > 1$ (worst case) şi $\mathcal{O}(d\,n)$ pt. $k = 1$],

unde d este numărul de atribute, iar n este numărul de exemple;

– **arbori kd (engl., kd-trees): *Statistical Pattern Recognition*, Andrew R. Webb, 3rd ed., 2011, Willey, pag. 163-173;**

– **k-NN ca algoritm ML “lazy” (vs. “eager”):**
suprafeţe de decizie şi graniţe de decizie:
diagrame Voronoi pentru 1-NN: ex. 4, ex. 11.a, ex. 20, ex. 21, ex. 22.a, ex. 23.ac;

– **analiza erorilor:**

- **1-NN pe date consistente: eroarea la antrenare este 0: ex. 2, ex. 12.a;**

  - **CVLOO: ex. 3, ex. 12.b, ex. 16.bc, ex. 18, ex. 24.a, ex. 22.b, ex. 23.b;**
  - **sensibilitatea / robusteţea la „zgomote“: ex. 5, ex. 16;**
  - **eroarea asimptotică: ex. 10, ex. 27.**
- **efectul trăsăturilor redundante sau irelevante;**
- **alegerea valorii convenabile pentru k: ex. 25;**
  variaţia numărului de erori (la antrenare şi respectiv testare) în funcţie de valorile lui k: ex. 26.ab;
  cross-validarea ca metodă neparametrică pentru alegerea lui k: ex. 26.c.

Proprietăţi ale algoritmului k-NN

- **(P0) output-ul algoritmului k-NN pentru o instanţă oarecare de test x_q depinde de valoarea lui k: ex. 1, ex. 17;**
- **(P1) pe seturi de date de antrenament *consistente*, eroarea la antrenare produsă de algoritmul 1-NN este 0, indiferent de *măsura de distanţă* folosită: ex. 2, ex. 12.a, ex. 16.b;**
- **(P2) output-ul algoritmului k-NN, precum şi suprafeţele de decizie şi separatorii decizionali depind de *măsura de distanţă* folosită: ex. 7;**
- **(P3): Pe orice set de exemple *consistente* din $\mathbb{R}$ (deci cu un singur atribut de intrare, care este numeric şi continuu), atât algoritmul 1-NN cât şi algoritmul ID3 produc aceleaşi rezultate la testare / generalizare: ex. 14.a.**
- **(P4) „blestemul marilor dimensiuni“ (engl., the curse of dimensionality): în anumite condiţii, numărul de instanţe de antrenament necesare pentru a avea un *cel mai apropiat vecin* situat la distanţă *rezonabilă* faţă de instanţa de test x_q creşte exponenţial în funcţie de numărul de atribute folosite: ex. 9;**
- **(P5) în anumite condiţii, rata medie a *erorii asimptotice* a algoritmului 1-NN este mărginită superior de dublul ratei medii a erorii algoritmului Bayes Optimal: ex. 10, ex. 27;**

Comparaţii cu alţi algoritmi de clasificare automată

- **ID3: ex.11.b, ex.12.c, ex. 13, ex. 14, ex. 23.d, ex. 24.b;**
- **SVM: ex.12.d, ex. 24.b;**
- **regresia logistică: ex. 24.b;**
- **1-NN cu mapare cu RBF: ex. 15.**

Variante ale algoritmului k-NN

- **k-NN folosind alte măsuri de distanţă (decât distanţa euclidiană): ex. 7;**
- **k-NN cu *ponderarea distanţelor* (engl., distance-weighted k-NN): cartea ML, pag. 236-238 (formulele 8.2, 8.3, 8.4);**[466]
- **algoritmul lui Shepard: ex.8.**

[466] Sectiunea 8.3 din cartea ML (pag. 236-238) se referă la regresia [liniară] local-ponderată ca o formă mai generală de aproximare a [valorilor] funcţiilor, în raport cu cele calculate de către algoritmul k-NN atunci când se foloseşte ponderarea distanţelor.

Alte metode de tip IBL

– **reţele RBF: cartea ML, pag. 238-240;**

– **raţionare bazată pe cazuri (engl., case-based reasoning): cartea ML, pag. 240-244.**

3.1 Învăţare bazată pe memorare — Probleme rezolvate

1. **(Algoritmul k-NN: aplicare în $\mathbb{R}^2$ pentru diferite valori ale lui k)**

■ • ***CMU, 2006 fall, E. Xing, T. Mitchell, final exam, pr. 2***

Fie setul de instanţe de antrenament din tabelul alăturat:

x	y	
-1	1	$-$
0	1	$+$
0	2	$-$
1	-1	$-$
1	0	$+$
1	2	$+$
2	2	$-$
2	3	$+$

a. Vizualizaţi datele într-un sistem de axe din $\mathbb{R}^2$.

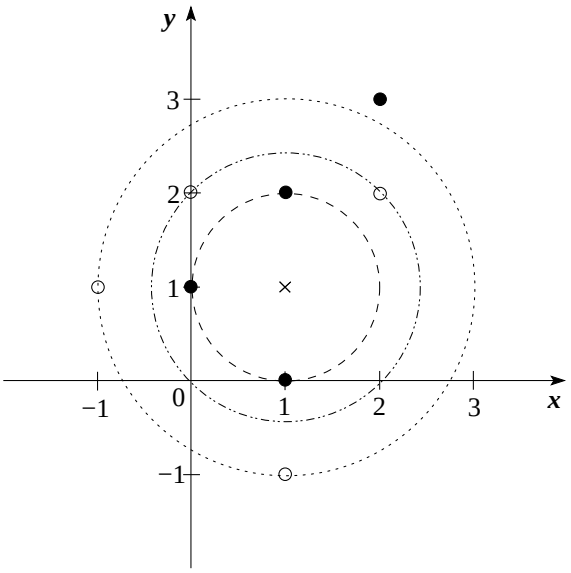
b. Presupunând că se foloseşte distanţa euclidiană, care va fi predicţia făcută pentru punctul (1,1) de către

– clasificatorul 3-NN?

– clasificatorul 5-NN?

– clasificatorul 7-NN?

Răspuns:



În figura alăturată am reprezentat:

– datele de antrenament, folosind cerculeţe albe pentru cele clasificate cu $-$ şi cerculeţe negre pentru cele clasificate cu $+$;

– punctul care trebuie clasificat, marcat cu $\times$;

– vecinătăţile luate în considerare de către cei trei clasificatori precizaţi în enunţ; aceste vecinătăţi sunt reprezentate prin cele trei cercuri concentrice având centrul în punctul (1,1).

Analizând pe rând etichetele instanţelor din aceste trei vecinătăţi, ajungem la concluzia că rezultatele obţinute de cei trei clasificatori vor fi următoarele:

- **3-NN: $+$**
- **5-NN: $+$**
- **7-NN: $-$**

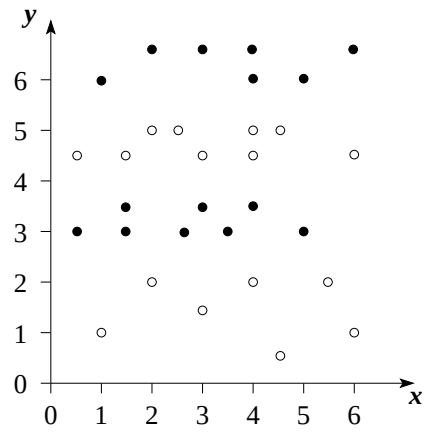
***Observaţie*: Acest exerciţiu pune în evidenţă faptul că la clasificarea cu algoritmul k-NN rezultatul poate diferi în funcţie de diversele valori ale lui k, adică în funcţie de cum se lărgeşte (sau se restrânge) vecinătatea punctului de test considerat.**

2. **(Algoritmul 1-NN: eroarea la antrenare)**

CMU, 2002 fall, Andrew Moore, final exam, pr. 6.e

Figura alăturată prezintă un set de date având două atribute de intrare x şi y, şi un atribut de ieşire, ale cărui valori sunt reprezentate prin culoarea punctului (alb sau negru).

Putem alege o *metrică* (adică, *măsură* sau *funcţie de distanţă*) astfel încât, folosind algoritmul de învăţare 1-NN (engl., [one] nearest neighbour), să obţinem eroare 0 la antrenare pe setul acesta de date?



Răspuns:

Fie $d : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ o metrică oarecare. Conform definiţiei matematice, d îndeplineşte următoarele *condiţii*:

$d(x, y) \geq 0, \forall x, y \in \mathbb{R}^2$	**(nenegativitatea)**
$d(x, y) = 0 \Leftrightarrow x = y$	**(identitatea indiscernabililor)**
$d(x, y) = d(y, x)$	**(simetria)**
$d(x, y) \leq d(x, z) + d(z, y), \forall x, y, z \in \mathbb{R}^2$	**(inegalitatea triunghiului)**

În particular, d poate fi metrica euclidiană, definită astfel:

$$d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2},$$

pentru orice $(x_1, y_1), (x_2, y_2) \in \mathbb{R}^2$.

În general, când algoritmul 1-NN primeşte o instanţă de test (x, y), el caută în setul de antrenament punctul (x', y') cu proprietatea că distanţa $d((x, y), (x', y'))$ este minimă.

În cazul particular în care punctul (x, y) însuşi aparţine datelor de antrenament, urmează că $(x', y') = (x, y)$, datorită primelor două proprietăţi ale lui d enunţate mai sus.

Cum setul de date din enunţ este *consistent* — adică nu există nicio instanţă care să apară de două sau mai multe ori, însă cu etichete diferite —, vom avea de luat în considerare o singură valoare pentru calculul atributului de ieşire pentru punctul de test (x, y). Evident, algoritmul 1-NN o va folosi pe aceasta ca rezultat al clasificării. Concluzia acestui raţionament este că eroarea la antrenare produsă de algoritmul 1-NN pe acest set de date este 0.

Facem *observaţia* că nu doar pe acest set de date ci pe orice set de date de antrenament fără zgomote / inconsistenţe în ce priveşte etichetarea, algoritmul 1-NN va avea eroarea la antrenare 0, indiferent de metrica folosită (bineînţeles, dacă există o metrică în spaţiul instanţelor respective).

3. **(Algoritmul k-NN: calculul erorii la CVLOO pentru diferite valori ale lui k)**

CMU, 2003 fall, T. Mitchell, A. Moore, final exam, pr. 5.ab

X	Y
-0.1	$-$
0.7	$+$
1.0	$+$
1.6	$-$
2.0	$+$
2.5	$+$
3.2	$-$
3.5	$-$
4.1	$+$
4.9	$+$

Fie setul de date de antrenament din tabelul alăturat.
Vom folosi algoritmul k-NN cu distanţa euclidiană (neponderată) pentru a prezice valorile atributului de ieşire Y (de tip boolean) plecând de la atributul de intrare X, care ia valori reale.

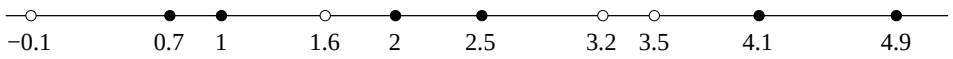
Care este eroarea produsă de algoritmul k-NN la cross-validare cu metoda "Leave-One-Out" în cazurile următoare:

a. $k = 1$.
b. $k = 3$.

Exprimaţi răspunsul sub forma numărului de instanţe clasificate eronat.

Răspuns:

Figura următoare vizualizează pe o axă datele de antrenament, precum şi clasificările acestora ($\circ$ pentru instanţe negative şi $\bullet$ pentru instanţe pozitive):



Comportamentul celor doi clasificatori la cross-validare prin metoda "Leave-One-Out" este cel explicat mai jos:

• Clasificatorul 1-NN:

Data	Eticheta	Vecinătate	Clasificare la CVLOO	Eroare?
-0.1	$-$	$\{0.7\}$	$+$	**da**
0.7	$+$	$\{1.0\}$	$+$	**nu**
1.0	$+$	$\{0.7\}$	$+$	**nu**
1.6	$-$	$\{2.0\}$	$+$	**da**
2.0	$+$	$\{1.6\}$	$-$	**da**
2.5	$+$	$\{2.0\}$	$+$	**nu**
3.2	$-$	$\{3.5\}$	$-$	**nu**
3.5	$-$	$\{3.2\}$	$-$	**nu**
4.1	$+$	$\{3.5\}$	$-$	**da**
4.9	$+$	$\{4.1\}$	$+$	**nu**

• Clasificatorul 3-NN:

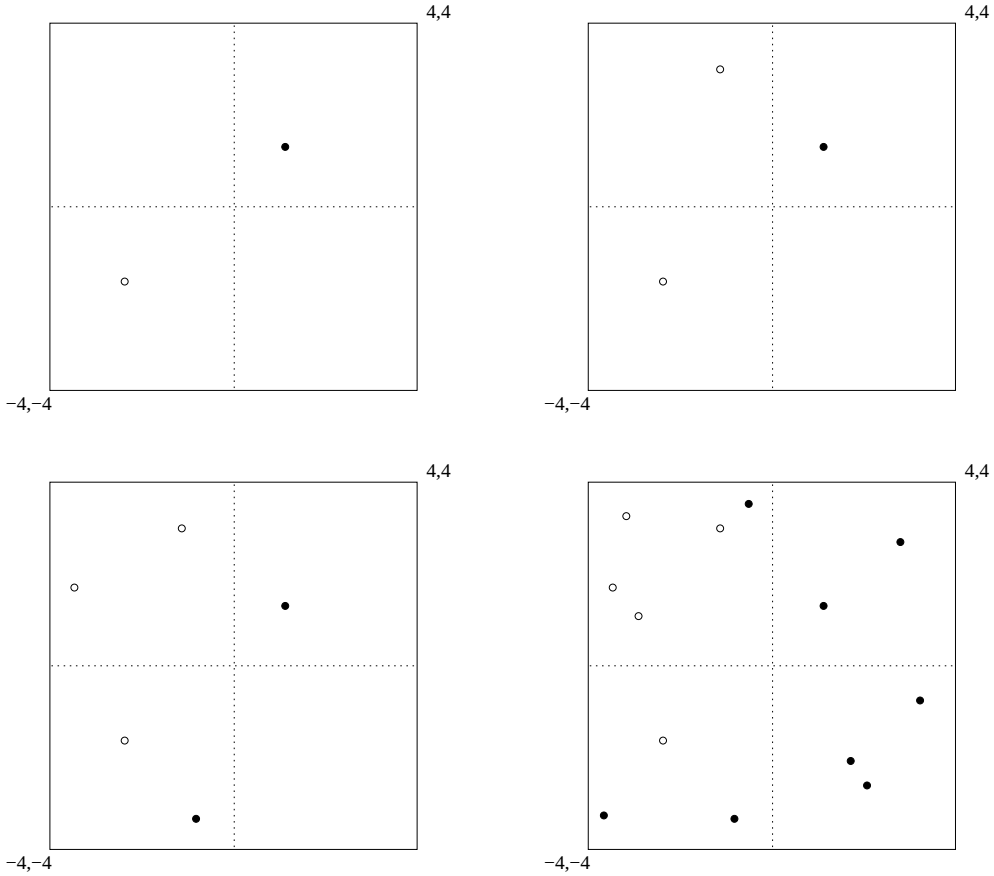
Data	Eticheta	Vecinătate	Clasificare la CVLOO	Eroare?
-0.1	$-$	$\{0.7; 1.0; 1.6\}$	$+$	**da**
0.7	$+$	$\{-0.1; 1.0; 1.6\}$	$-$	**da**
1.0	$+$	$\{0.7; 1.6; 2.0\}$	$+$	**nu**
1.6	$-$	$\{1.0; 2.0; 0.7/2.5\}$	$+$	**da**
2.0	$+$	$\{1.0; 1.6; 2.5\}$	$+$	**nu**
2.5	$+$	$\{1.6; 2.0; 3.2\}$	$-$	**da**
3.2	$-$	$\{2.5; 3.5; 4.1\}$	$+$	**da**
3.5	$-$	$\{2.5; 3.2; 4.1\}$	$+$	**da**
4.1	$+$	$\{3.2; 3.5; 4.9\}$	$-$	**da**
4.9	$+$	$\{3.2; 3.5; 4.1\}$	$-$	**da**

În concluzie, la cross-validare cu metoda "Leave-One-Out" avem 4 erori la clasificarea cu algoritmul 1-NN şi 8 erori la clasificarea cu algoritmul 3-NN. Putem concluziona că rezultatul algoritmului 3-NN este foarte afectat de către puternica „mixare" (adică, de frecventele schimbări de clasă, de la o instanţă oarecare la vecinii ei) din setul de antrenament.

4. **(Algoritmul 1-NN: graniţe / suprafeţe de decizie; diagrame Voronoi ca modalitate de învăţare rapidă / "eager")**

■ • ***CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, HW1, pr. 3.1-2***

În fiecare din figurile următoare se dau câteva puncte în spaţiul euclidian bidimensional. Fiecare dintre aceste puncte este etichetat fie pozitiv (cerc plin) fie negativ (cerc simplu).



a. Presupunând că folosim ca metrică distanţa euclidiană, desenaţi suprafeţele de decizie corespunzătoare clasificatorului 1-NN, în fiecare din aceste patru cazuri.

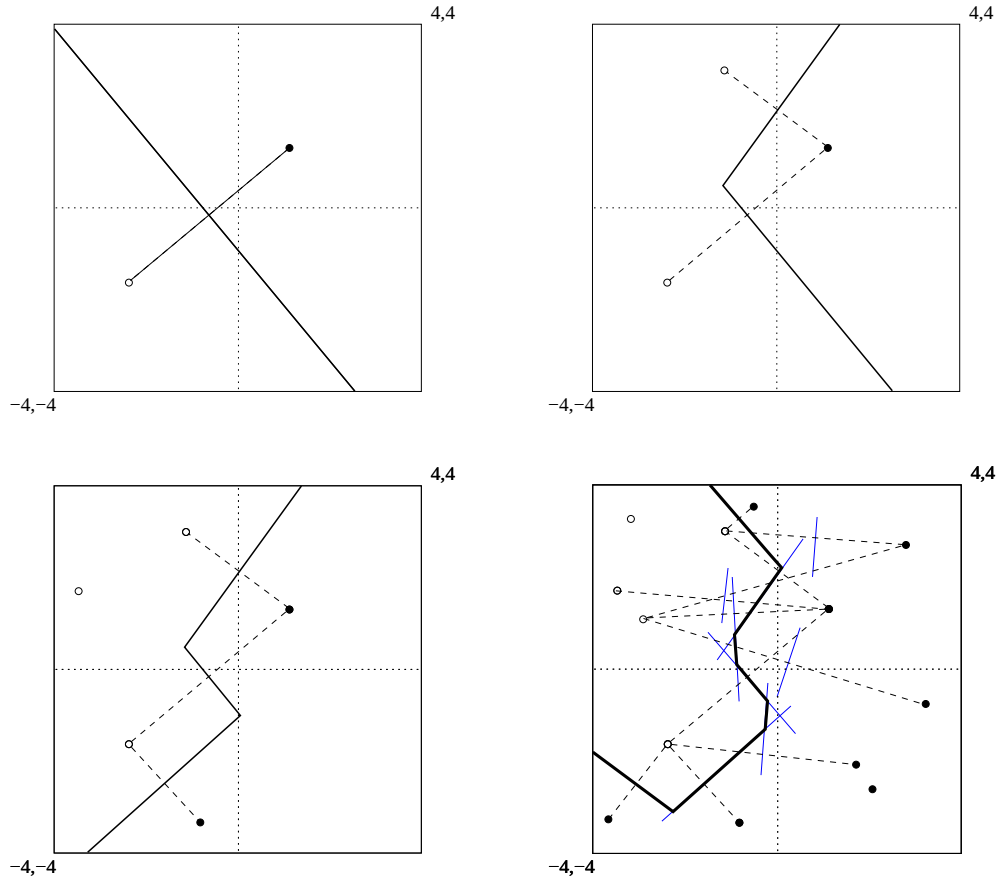
b. La curs am afirmat despre k-NN că este un clasificator lent (engl., lazy), care doar memorează toate instanţele de antrenament până ajunge la faza de test.

Totuşi, la punctul precedent am văzut că putem să trasăm suprafaţele de decizie pentru clasificatorul 1-NN, înainte de a intra în faza de test / generalizare. Atunci, în această fază, în loc să calculăm diverse distanţe şi apoi să determinăm care sunt cei mai apropiaţi vecini faţă de punctul de test dat (notat x_q), pur şi simplu i se va asigna lui x_q clasa / eticheta corespunzătoare suprafeţei de decizie în care se plasează.

Dacă am decide să memorăm toate aceste suprafeţe de decizie (ca linii poligonale) în loc să memorăm toate datele de antrenament, am obţine *întotdeauna* o îmbunătăţire în ceea ce priveşte consumul de memorie necesar pentru acest clasificator?

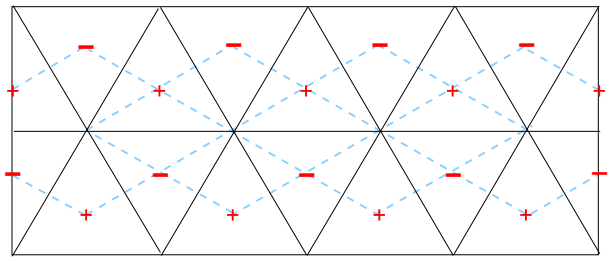
Răspuns:

a. Suprafeţele de decizie corespunzătoare clasificatorului 1-NN pentru cele patru cazuri sunt:



b. Răspunsul este negativ, adică: nu întotdeauna memorarea separatorului decizional este mai convenabilă decât memorarea datelor de antrenament. Vom justifica dând un exemplu, care reprezintă o situaţie-limită, şi anume, cazul care apare atunci când se creează tot atâtea suprafeţe de decizie câte instanţe sunt în setul de antrenament.

În figura alăturată, pentru cele n instanţe de antrenament avem nevoie să memorăm $3\frac{n}{2}$ puncte care determină triunghiurile – suprafeţele de decizie pentru clasa $+$. (Clasa $-$ va fi determinată prin excluziune / complementaritate.)

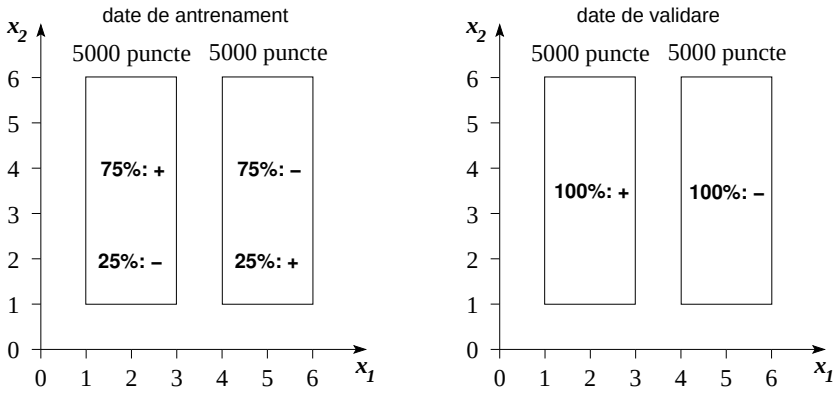


Se poate demonstra uşor următoarea proprietate: Chiar dacă am memora o singură dată cele $3\frac{n}{2}$ puncte — fiecare punct având câte 2 coordonate — care determină contururile suprafeţelor de decizie şi apoi am folosi indecşi pentru a indica ce puncte determină fiecare suprafaţă de decizie, consumul de memorie ar fi mai mare decât dacă am memora doar instanţele de antrenament.

5. **(Algoritmul k-NN: rata medie a erorii la antrenare, la CVLOO şi la testare în prezenţa unor „zgomote“ în datele de antrenament)**

• ○ ***CMU, 2002 fall, Andrew Moore, final exam, pr. 7***

Se constituie un set de date de antrenament şi un set de date de validare, alegând în mod uniform aleatoriu puncte situate în anumite regiuni dreptunghiulare, conform imaginii următoare:



Se observă că datele de antrenament sunt „bruiate“ (engl., noisy): în fiecare regiune, 25% din date provin din clasa adversă. Datele de validare nu sunt bruiate.

Pentru fiecare dintre întrebările următoare, încercuiţi fracţia care este *cea mai apropiată de* media / rata erorii în cazul respectiv. Justificaţi în mod riguros alegerea făcută.

a. Care este rata medie a erorii la antrenare pentru clasificatorul 1-NN ?

`0 1/8 1/4 3/8 1/3 1/2 5/8 2/3 3/4 7/8 1`

b. Care este rata medie a erorii la cross-validare cu metoda “Leave-One-Out” pentru clasificatorul 1-NN pe setul de antrenare?

0 1/8 1/4 3/8 1/3 1/2 5/8 2/3 3/4 7/8 1

c. Care este rata medie a erorii la testare pentru clasificatorul 1-NN pe setul de validare? (Antrenarea se face pe setul de antrenare.)

0 1/8 1/4 3/8 1/3 1/2 5/8 2/3 3/4 7/8 1

d. Care este rata medie a erorii la antrenare pentru clasificatorul 21-NN ?

0 1/8 1/4 3/8 1/3 1/2 5/8 2/3 3/4 7/8 1

e. Care este rata medie a erorii la cross-validare cu metoda "Leave-One-Out" pentru clasificatorul 21-NN pe setul de antrenare?

0 1/8 1/4 3/8 1/3 1/2 5/8 2/3 3/4 7/8 1

f. Care este rata medie a erorii la testare pentru clasificatorul 21-NN pe setul de validare? (Antrenarea se face pe setul de antrenare.)

0 1/8 1/4 3/8 1/3 1/2 5/8 2/3 3/4 7/8 1

Răspuns:

a. În acest caz, rata medie a erorii este 0, întrucât orice punct din datele de antrenament este cel mai apropiat vecin în raport cu el însuşi, iar probabilitatea ca două puncte care au fost generate în mod aleatoriu (în oricare din cele două dreptunghiuri) să aibă exact aceleaşi coordonate şi etichete diferite este foarte mică (este practic 0).

b. $3/8$.
Să analizăm cazul primului dreptunghi. Constituie eroare cazul când un exemplu pozitiv (care apare cu probabilitatea „aşteptată" de $3/4$) este clasificat negativ (iar aceasta se întâmplă cu probabilitatea $1/4$) sau, invers, când un exemplu negativ (probabilitate: $1/4$) este clasificat pozitiv (probabilitate: $3/4$). Aşadar, probabilitatea de a clasifica greşit un exemplu din primul dreptunghi o aflăm astfel: $3/4 \cdot 1/4 + 1/4 \cdot 3/4 = 3/8$. Pentru dreptunghiul din dreapta raţionamentul este similar, iar selecţia exemplelor din cele două dreptunghiuri se face cu aceeaşi probabilitate ($1/2$), deci rezultatul final este $3/8$.

c. $1/4$.
Probabilitatea de eroare este $1/4$ pentru fiecare dreptunghi, fiindcă (în general) pentru un punct (x_1, x_2) selectat în mod aleatoriu din datele de validare din dreptunghiul respectiv aceasta (adică $1/4$) este probabilitatea ca vecinul cel mai apropiat de (x_1, x_2) în setul de date de antrenament să aibă semnul opus (faţă de semnul lui (x_1, x_2)). Aşadar, media erorii la testare cu clasificatorul 1-NN pe setul de validare este $1/2 \cdot 1/4 + 1/2 \cdot 1/4 = 1/4$.

d. $1/4$.
Folosind algoritmul 21-NN, în dreptunghiul din stânga $3/4$ din datele de antrenament sunt (în general) clasificate corect, iar restul de $1/4$ (şi anume, cele având semnul $-$) sunt clasificate eronat. Într-adevăr, la testarea unui punct oarecare (x_1, x_2) din acel dreptunghi, în 21-NN vecinătatea lui (x_1, x_2), unul (cel mai apropiat vecin) este însuşi (x_1, x_2), iar dintre ceilalţi 20 cei mai apropiaţi vecini $3/4$ sunt (în general) de semn $+$, iar $1/4$ de semn $-$. Analog se raţionează pentru dreptunghiul din dreapta. Deci rata medie a erorii la antrenare este:

$$\frac{1}{2}\left(\frac{3}{4}\cdot 0 + \frac{1}{4}\cdot 1\right) + \frac{1}{2}\left(\frac{3}{4}\cdot 0 + \frac{1}{4}\cdot 1\right) = \frac{1}{4}.$$

e. $1/4$.
În cazul primului dreptunghi, apare eroare atunci când un exemplu pozitiv (probabilitate de selectare: $3/4$**) este clasificat negativ (probabilitate:** 0**, fiind vorba de 21-NN) sau când un exemplu negativ (probabilitate de selectare:** $1/4$**) este clasificat pozitiv (probabilitate:** 1**).**[467] **Aşadar, probabilitatea de a clasifica greşit un exemplu din primul dreptunghi este** $3/4 \cdot 0 + 1/4 \cdot 1 = 1/4$**. Cazul dreptunghiului din dreapta este similar, prin urmare rezultatul final este** $1/4$**.**

f. 0.
Datorită distribuţiei uniforme a datelor de antrenament, fiecare instanţă (x_1, x_2) **din setul de validare va avea majoritatea vecinilor — dintre cei mai apropiaţi, selectaţi de către algoritmul 21-NN din setul de date de antrenament — de acelaşi semn cu semnul lui** (x_1, x_2)**. (Şi anume, în medie de 3 ori mai mulţi vecini decât cei de semn contrar.) Prin urmare, fiecare punct din setul de validare va fi clasificat corect.**

***Observaţie* (1): Comparând rezultatele obţinute la punctele** a **şi** c **pe de o parte cu cele de la punctele** d **şi** f **(sau chiar** b **şi** e**) pe de altă parte, se observă că are loc o relaţie exact de acelaşi gen cu cea din definiţia fenomenului de *overfitting* (sau, *supra-specializare*):**[468] **erorile produse de algoritmii** 1**-NN şi** 21**-NN la antrenare sunt în relaţia** $0 < 1/4$ **dar în relaţie inversă** $(1/4 > 0)$ **la testare (respectiv** $3/8 > 1/4$ **la cross-validare). Această manifestare a fenomenului de overfitting va fi pusă în evidenţă foarte clar în graficul de la finalul *Observaţiei* (2); vedeţi curba roşie (ascendentă) şi curba neagră (descendentă), în special în zona corespunzătoare valorilor** $k \in \{1, \ldots, 22\}$**.**

***Observaţie* (2): Remarcaţi „sublinierea“ din expresia „fracţia *cea mai apropiată de* media / rata erorii“ din enunţ. Generarea aleatorie a datelor poate conduce la rezultate uşor diferite faţă de cele pe care le-am obţinut mai sus. În urma realizării unei implementări,**[469] **au fost obţinute următoarele rezultate de tip *eroare medie*, calculată în urma repetării de 100 de ori a generării datelor şi aplicării algoritmilor** k**-NN, conform cerinţelor din enunţ:**

a. 0, **b.** 0.374022, **c.** 0.250472, **d.** 0.249342, **e.** 0.253088, **f.** 0.006436.

Constatăm că se verifică „previziunile“ noastre din rezolvarea de mai sus.

[467] În 21-NN vecinătatea punctului considerat (pentru testare), spre deosebire de cazul erorii la antrenare, la CVLOO nu se mai include punctul respectiv.

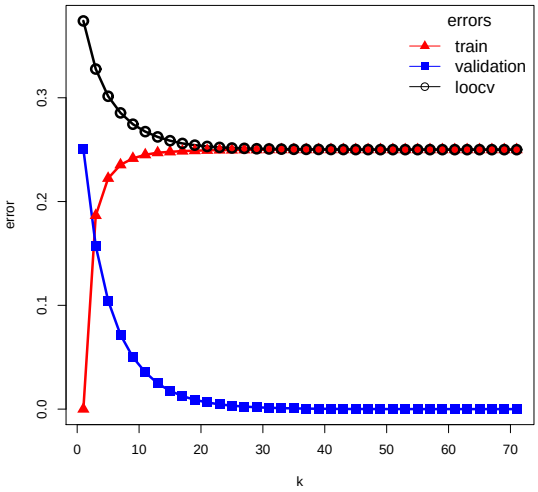
[468] Vă readucem aminte definiţia lui Tom Mitchell pentru fenomenul de *overfitting* (vedeţi cartea *Machine Learning*, pag. 67): două ipoteze h şi h' obţinute (eventual cu un acelaşi algoritm de clasificare automată) pe un set de date sunt în raport de overfitting dacă

$$error_{train}(h) < error_{train}(h'), \quad \text{dar} \quad error_{\mathcal{D}}(h) > error_{\mathcal{D}}(h'),$$

unde $\mathcal{D}$ este distribuţia reală a datelor.

[469] Implementarea a fost făcută de către studentul Sebastian Ciobanu de la Facultatea de Informatică a Universităţii „Al. I. Cuza“ din Iaşi în semestrul I al anului universitar 2016-2017.

Evoluţia celor trei tipuri de eroare (la antrenare, la validare şi cross-validare cu metoda "leave-one-out"), pentru $k = 1, 3, \ldots, 69, 71$ este prezentată în figura alăturată. Se observă convergenţa la aceeaşi valoare (aprox., 0.25) pentru eroarea la antrenare şi eroarea CV-LOO (pe setul de date de antrenare) începând din jurul valorii $k = 31$, precum şi convergenţa la valori foarte apropiate de 0 pentru eroarea la testare pe setul de date de validare, începând din jurul valorii $k = 35$.



6. **(O versiune ipotetică pentru algoritmul k-NN: selectarea celor mai apropiaţi vecini nu pe baza calculării funcţiei de distanţă, ci folosind un oracol / "black box")**

CMU, 2009 spring, Ziv Bar-Joseph, final exam, pr. 2.2-3

Se încearcă clasificarea unor puncte în spaţiul euclidian bidimensional. Sunt date n instanţe P_1, P_2, ..., P_n, precum şi etichetările corespunzătoare c_1, c_2, ..., c_n, unde c_1, c_2, ..., c_n reprezintă valori dintr-o mulţime C. În schema de clasificare k-NN, fiecare element nou Q este clasificat cu eticheta majoritară obţinută în cadrul vecinătăţii formate din cei mai apropiaţi k vecini.

Să presupunem că măsura de distanţă nu este dată în mod explicit. În locul acesteia, aveţi la dispoziţie un "black box". Dacă se introduc instanţele $P_{i_1}, P_{i_2}, \ldots, P_{i_l}$ (unde l este un număr natural oarecare) şi un punct Q, black box-ul returnează cel mai apropiat vecin al lui Q, adică un element $P_{i_0} \in \{P_{i_1}, P_{i_2}, \ldots, P_{i_l}\}$, precum şi clasificarea corespunzătoare (c_{i_0}).

a. Este posibil să se construiască un algoritm de tip k-NN bazat doar pe acest black box? Dacă da, explicaţi cum anume, iar dacă nu, explicaţi de ce nu este posibil.

b. Dacă, în schimb, acel black box returnează cei mai apropiaţi j vecini (şi etichetele corespunzătoare), iar $j \neq k$, este posibil să se construiască un algoritm de tip k-NN bazat doar pe acest black box? Dacă da, explicaţi cum anume, iar dacă nu, explicaţi de ce nu este posibil.

Răspuns:

a. Da. Se foloseşte black box-ul dându-i ca intrare mai întâi mulţimea de exemple $P_1, P_2, \ldots, P_n$; se obţine cel mai apropiat vecin al lui Q şi eticheta sa. Apoi se scoate instanţa / punctul returnat din mulţimea de exemple. Se repetă acest procedeu de k ori, iar la final se va alege pentru Q eticheta corespunzătoare majorităţii din mulţimea de k vecini care au fost identificaţi de către black box.

b. Dacă $j < k$, atunci se foloseşte black box-ul de $[k/j]$ ori, obţinându-se $j*[k/j]$ cei mai apropiaţi vecini şi clasificările acestora. Notăm cu V_1 mulţimea formată din aceşti vecini. În cazul în care $k \neq j*[k/j]$, pentru a obţine restul de vecini necesari pentru k-NN, adică încă $k - j*[k/j]$ vecini, vom folosi black box-ul încă o dată. Vom nota cu V_2 mulţimea acestor noi j vecini. Dintre aceştia va trebui să alegem doar $k - j*[k/j]$ instanţe. Vom proceda astfel: considerăm o nouă mulţime de instanţe alcătuită din elementele lui V_2 şi $j - (k - j*[k/j])$ dintre toţi ceilalţi vecini obţinuţi anterior (V_1). Aplicăm black box-ul pe acest nou set de date şi vom obţine cei mai apropiaţi j vecini pentru punctul Q. Printre aceştia se vor afla cei $k - j*[k/j]$ vecini căutaţi.

Dacă $j > k$, iar black box-ul nu acceptă intrări duplicate, atunci el nu poate fi folosit pentru a determina cei mai apropiaţi k vecini ai instanţei de test.

Dar dacă $j > k$ şi black box-ul acceptă intrări duplicate, este posibil să rezolvăm problema, cel puţin în situaţia în care cei mai apropiaţi j vecini ai lui Q se află toţi la distanţe diferite faţă de acesta.[470] De exemplu, pentru $k = 1$ şi $j = 3$ vom proceda astfel:

***Pasul 1*: Aplicăm black box-ul inputului $P_1, \ldots, P_n$. Vom obţine outputul $P_{i_1}, P_{i_2}, P_{i_3}$.**

***Pasul 2*: Corespunzător fiecărui punct $P_{i_1}, P_{i_2}, P_{i_3}$, vom aplica pe rând black box-ul inputului**

- **$P_{i_1}, P_{i_1}, P_{i_2}, P_{i_3}$,**
- **$P_{i_1}, P_{i_2}, P_{i_2}, P_{i_3}$ şi respectiv**
- **$P_{i_1}, P_{i_2}, P_{i_3}, P_{i_3}$.**

Se poate constata că într-unul singur din aceste cazuri black box-ul returnează outputul $P_{i_1}, P_{i_2}, P_{i_3}$.[471] De pildă, dacă outputurile în aceste trei cazuri sunt

- **$P_{i_1}, P_{i_1}, P_{i_2}$,**
- **$P_{i_1}, P_{i_2}, P_{i_2}$,**
- **$P_{i_1}, P_{i_2}, P_{i_3}$,**

rezultă că P_{i_3} este cel mai distant dintre cei trei vecini ai lui Q.

***Pasul 3*: Considerând că lucrurile stau ca la finalul pasului precedent, pentru fiecare dintre punctele P_{i_1}, P_{i_2} vom aplica black box-ului următorul input:**

- **$P_{i_1}, P_{i_1}, P_{i_1}, P_{i_2}$ şi respectiv**
- **$P_{i_1}, P_{i_2}, P_{i_2}, P_{i_2}$.**

Dacă vom obţine outputul $P_{i_1}, P_{i_1}, P_{i_1}$, va rezulta că P_{i_1} este mai apropiat de Q decât punctul P_{i_2}. Invers, dacă obţinem outputul $P_{i_2}, P_{i_2}, P_{i_2}$, va rezulta că P_{i_2} este mai apropiat de Q decât punctul P_{i_1}.

[470] Rămâne de analizat varianta contrară.

[471] Am presupus că black box-ul returnează exact j instanţe, chiar dacă între $P_1, \ldots, P_n$ există mai multe instanţe egal depărtate faţă de punctul Q, în speţă mai multe instanţe situate la maximumul distanţelor dintre fiecare din cele j pe de o parte şi punctul Q pe de altă parte.

7. **(Algoritmul 1-NN: suprafeţele de decizie [şi separatorii decizionali] depind de măsurile de distanţă folosite)**

CMU, 2008 fall, Eric Xing, HW1, pr. 3.1.2

Se dau două puncte din spaţiul euclidian bidimensional: punctul $(-1, 0)$ clasificat negativ şi punctul $(1, 0)$ clasificat pozitiv.

Clasificatorul 1-NN care foloseşte distanţa euclidiană şi are ca set de date de antrenament cele două puncte de mai sus are următoarea *formă analitică* (uşor de dedus):

- **dat fiind un punct arbitrar (x, y),**
  în cazul în care $x > 0$, eticheta asignată punctului respectiv este $+$, iar în cazul $x < 0$ eticheta asignată este $-$;
- **dreapta $x = 0$ este *graniţa de decizie* (engl., decision boundary) corespunzătoare acestui clasificator.**

a. Care va fi forma analitică a clasificatorului 1-NN dacă în locul distanţei euclidiene (indusă de norma L_2) se foloseşte distanţa Manhattan (indusă de norma L_1)? Vă reamintim că distanţa Manhattan dintre două puncte (x_1, y_1) şi (x_2, y_2) este $|x_1 - x_2| + |y_1 - y_2|$.

b. Dar dacă se foloseşte distanţa Chebyshev, care este indusă de norma L_∞ şi este definită în $\mathbb{R}^2$ prin relaţia

$$d((x_1, y_1), (x_2, y_2)) = \max\{|x_1 - x_2|, |y_1 - y_2|\} \ ?$$

Răspuns:

a. Putem exprima distanţa Manhattan dintre un punct oarecare (x, y) din plan şi de fiecare dintre cele două puncte date în enunţ — punctul $(1, 0)$ clasificat pozitiv şi $punctul(-1, 0)$ clasificat negativ — astfel:

$$\begin{aligned} d_+ &\overset{not.}{=} d((x, y), (1, 0)) = |x - 1| + |y| \\ d_- &\overset{not.}{=} d((x, y), (-1, 0)) = |x + 1| + |y| \end{aligned}$$

În consecinţă, a stabili care dintre cele două distanţe (d_+ şi d_-) este mai mică revine la a compara (doar) expresiile $|x - 1|$ şi $|x + 1|$.

Avem următoarele cazuri:

- **dacă $x > 1$, atunci**

$$\left.\begin{array}{l} d_+ = x - 1 + |y| \\ d_- = x + 1 + |y| \end{array}\right\} \Rightarrow d_+ < d_- \Rightarrow \textbf{ punctul } (x, y) \textbf{ va fi clasificat } +$$

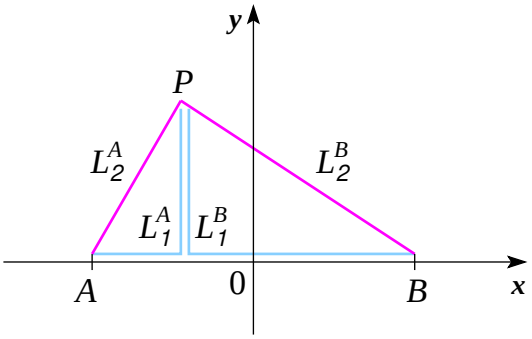
- **dacă $x < -1$:**

$$\left.\begin{array}{l} d_+ = -x + 1 + |y| \\ d_- = -x - 1 + |y| \end{array}\right\} \Rightarrow d_+ > d_- \Rightarrow \textbf{ punctul } (x, y) \textbf{ va fi clasificat } -$$

- **dacă $-1 \leq x \leq 1$:**

$$\left.\begin{array}{l} d_+ = -x + 1 + |y| \\ d_- = x + 1 + |y| \end{array}\right\} \Rightarrow \left\{\begin{array}{l} d_+ < d_- \textbf{ pentru } x > 0 \textbf{ deci } (x, y) \textbf{ va fi clasificat } + \\ d_+ > d_- \textbf{ pentru } x < 0 \textbf{ deci } (x, y) \textbf{ va fi clasificat } - \\ d_+ = d_- \textbf{ dacă şi numai dacă } x = 0. \end{array}\right.$$

Rezultă că graniţa de decizie a acestui clasificator este dreapta $x = 0$. Chiar mai mult: forma analitică a clasificatorului 1-NN care foloseşte distanţa Manhattan pe setul de date din enunţ este aceeaşi cu a clasificatorului 1-NN care foloseşte distanţa euclidiană.



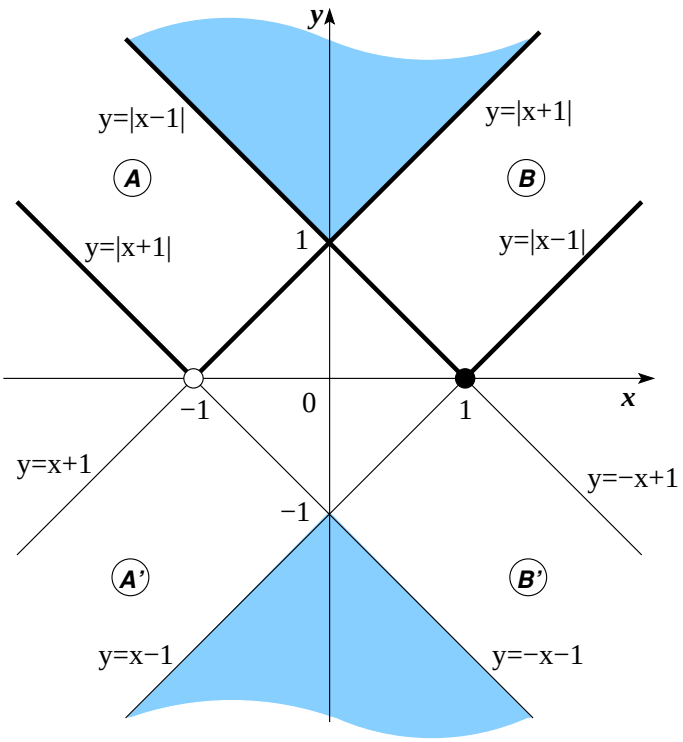
***Observaţie*: La concluzia de mai sus se putea ajunge mult mai uşor ţinând cont de următoarea *proprietate* (demonstrabilă imediat pe cale geometrică): pentru orice două puncte A şi B din $\mathbb{R}^2$ care au aceeaşi *ordonată* au loc următoarele două relaţii de echivalenţă:**

$$L_2(P,A) < L_2(P,B) \Leftrightarrow L_1(P,A) < L_1(P,B) \text{ pentru } \forall P \in \mathbb{R}^2 \text{ şi}$$
$$L_2(P,A) = L_2(P,B) \Leftrightarrow L_1(P,A) = L_1(P,B) \text{ pentru } \forall P \in \mathbb{R}^2.$$

b. Dacă se foloseşte distanţa Cebyshev, atunci conform definiţiei acestei metrici vom avea:

$$\begin{aligned} d_+ &= d((x,y),(1,0)) \\ &= \max\{|x-1|,|y|\} \\ d_- &= d((x,y),(-1,0)) \\ &= \max\{|x+1|,|y|\}. \end{aligned}$$

Mai întâi vom trasa graficele funcţiilor $y = |x-1|$ şi $y = |x+1|$, şi vom obţine rezultatul din figura alăturată.



***Cazul i*: Se observă că $|y| \geq |x-1|$ şi $|y| \geq |x+1|$ pentru toate punctele (x,y) situate în zonele unghiulare haşurate. În consecinţă, d_+ va fi egal cu d_- pentru orice punct din aceste zone. Aşadar, zonele haşurate vor aparţine graniţei / suprafeţei de decizie a clasificatorului nostru.**

***Cazul ii*: Considerăm acum zonele identificate prin literele A, A', B şi B' în figura de mai sus.[472] Se observă uşor că pentru zonele A şi A' avem $d_+ =$**

[472] Analitic, zona A este definită de punctele (x,y) care satisfac inecuaţiile $|x+1| < |y| < |x-1|$ şi $y > 0$; zona A': $|x+1| < |y| < |x-1|$ şi $y < 0$; zona B: $|x-1| < |y| < |x+1|$ şi $y > 0$; zona B': $|x-1| < |y| < |x+1|$ şi $y < 0$.

$\max\{|x-1|,|y|\} = |x-1|$ **şi** $d_- = \max\{|x+1|,|y|\} = |y|$**, iar pentru zonele** B **şi** B' **avem** $d_+ = \max\{|x-1|,|y|\} = |y|$ **şi** $d_- = \max\{|x+1|,|y|\} = |x+1|$**. În consecinţă,** $d_- = |y| < |x-1| = d_+$ **pentru zonele** A **şi** A'**, iar** $d_+ = |y| < |x+1| = d_-$ **pentru zonele** B **şi** B'**. Aşadar, zonele** A **şi** A' **vor fi clasificate negativ, iar zonele** B **şi** B' **vor fi clasificate pozitiv.**

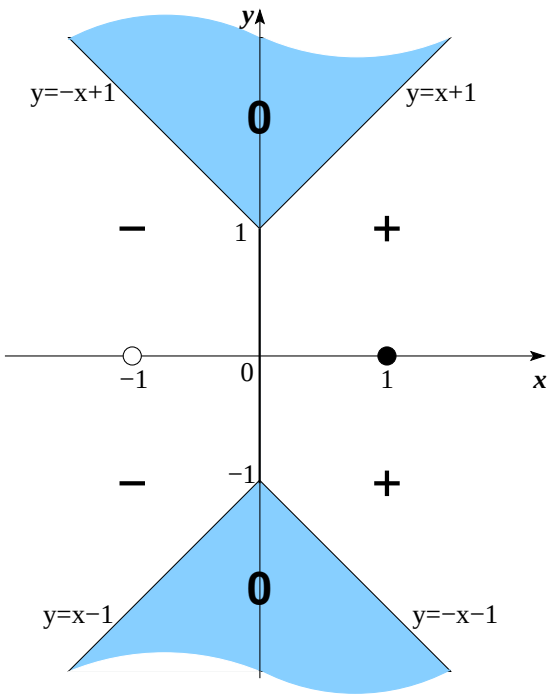
***Cazul iii*: Pentru toate celelalte zone rămase în discuţie — adică pentru orice punct** (x,y) **situat în afara celor două zone haşurate şi în afara zonelor** A, A', B, B' **—, vom avea** $d_+ = \max\{|x-1|,|y|\} = |x-1|$ **şi** $d_- = \max\{|x+1|,|y|\} = |x+1|$**. Din grafic se observă că** $|x+1| > |x-1|$ **pentru** $x > 0$ **şi** $|x-1| > |x+1|$ **pentru** $x < 0$**, iar** $|x+1| = |x-1|$ **pentru** $x = 0$**. Aşadar, sumarizând, în aceste zone avem:** $d_+ = \min\{d_+, d_-\}$ **pentru** $x > 0$ **şi** $d_- = \min\{d_+, d_-\}$ **pentru** $x < 0$**, iar** $d_+, = d_-$ **pentru** $x = 0$**.**

Concluzionând, forma analitică a clasificatorului 1-NN care foloseşte distanţa L_∞ **este următoarea:**

- (x,y) **va fi etichetat cu** $+$ **dacă** $x > 0$ **şi** $-x-1 < y < x+1$**;**
- (x,y) **va fi etichetat cu** $-$ **dacă** $x < 0$ **şi** $x-1 < y < -x+1$**;**
- **în rest este vorba de suprafaţa de separare, adică locul geometric al punctelor** (x,y) **pentru care distanţa faţă de cele două puncte din enunţ este egală.**

Reprezentarea grafică a suprafeţelor de decizie este dată în figura alăturată.

Este de remarcat faptul că pentru acest clasificator graniţa / suprafaţa de decizie nu este formată doar din drepte, ci este reuniunea unei drepte (axa Oy**)cu două intersecţii de (câte două) semiplane.**



8. **(Algoritmul / metoda lui Shepard – aplicare)**

CMU, 2002 fall, Andrew Moore, final exam, pr. 6.a-d

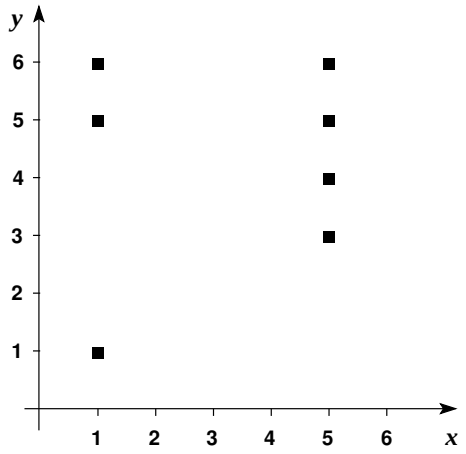
Figura de mai jos prezintă un set de date de antrenament cu un atribut de intrare $x \in \mathbb{R}$ **şi un atribut de ieşire** $y \in \mathbb{R}$**.**

Vom estima din aceste date câteva valori ale unei funcţii continue $f : \mathbb{R} \to \mathbb{R}$, folosind *metoda lui Shepard*. Aceasta este o variantă (de tip *regresie*) a algoritmului k-NN în care se iau în considerare toate punctele de antrenament, dar se aplică ponderi în funcţie de distanţă:

$$\hat{f}(x) \leftarrow \frac{\sum_i w(x, x_i) f(x_i)}{\sum_i w(x, x_i)}$$

Se va considera

$$w(x, x_i) = \begin{cases} 1, & \textbf{dacă } \mid x - x_i \mid \leq 3 \\ 0, & \textbf{în rest.} \end{cases}$$



Care va fi valoarea prezisă pentru funcţia f pentru

a. $x = 1$**?** **c.** $x = 5$**?**

b. $x = 3$**?** **d.** $x = 6$**?**

Răspuns:

Din modul cum au fost definite ponderile w ajungem la concluzia că valoarea lui f pentru un punct oarecare x va fi calculată ca medie aritmetică a valorilor / componentelor y din acele date de antrenament pentru care abscisa (x') este situată la distanţă de cel mult 3 unităţi de punctul x care ne interesează.

a. Avem $|1 - 1| = 0 \leq 3$ şi $|1 - 5| = 4 > 3$, prin urmare vor fi luate în considerare doar valorile învăţate pentru $x = 1$.

$\hat{f}(1) = \dfrac{1 + 5 + 6}{3} = \dfrac{12}{3} = 4.$

b. Avem $|3 - 1| = 2 \leq 3$ şi $|3 - 5| = 2 \leq 3$, prin urmare vor fi luate în considerare şi valorile învăţate pentru $x = 1$ şi cele pentru $x = 5$.

$\hat{f}(3) = \dfrac{1 + 5 + 6 + 3 + 4 + 5 + 6}{7} = \dfrac{30}{7}.$

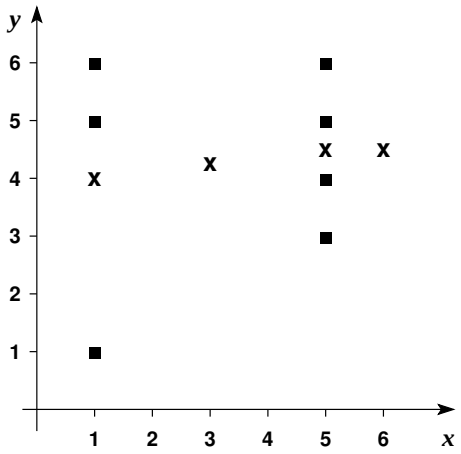
c. Avem $|5 - 1| = 4 > 3$ şi $|5 - 5| = 0 \leq 3$, prin urmare vor fi luate în considerare doar valorile învăţate pentru $x = 5$.

$\hat{f}(5) = \dfrac{3 + 4 + 5 + 6}{4} = \dfrac{18}{4} = \dfrac{9}{2}.$

d. Avem $|6 - 1| = 5 > 3$ şi $|6 - 5| = 1 \leq 3$, prin urmare vor fi luate în considerare doar valorile învăţate pentru $x = 5$, ca şi în cazul precedent.

$\hat{f}(6) = \hat{f}(5) = 4.5$

Dacă vom plasa rezultatele de mai sus pe grafic şi vom reprezenta punctele $(x, \hat{f}(x))$ sub forma unor cruciuliţe, vom obţine figura alăturată.



9. **(Asupra folosirii algoritmului k-NN în spaţii ($\mathbb{R}^p$) de dimensiune (p) mare: un avertisment: „blestemul marilor dimensiuni“)**

prelucrare de Liviu Ciortuz, după
■ □ • ***CMU, 2010 fall, Aarti Singh, HW2, pr. 2.2***
CMU, 2012 spring, Roni Rosenfeld, HW10, pr. 2

Noţiunea de hiper-sferă este o generalizare pentru noţiunile de cerc (în $\mathbb{R}^2$) şi sferă (în $\mathbb{R}^3$). Volumul unei hiper-sfere de rază r din $\mathbb{R}^p$ este dat de formula

$$V_p(r) = \frac{(r\sqrt{\pi})^p}{\Gamma\left(\frac{p}{2}+1\right)},$$

unde Γ reprezintă funcţia Gamma a lui Euler, care generalizează noţiunea de număr factorial ($\Gamma(n+1) = n!$ pentru orice $n \in \mathbb{N}^*$) şi are următoarele proprietăţi:

$$\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi},\ \Gamma(1) = 1, \text{ iar } \Gamma(x+1) = x\Gamma(x) \text{ pentru } x > 0.^{473}$$

Considerăm punctele $x_1, x_2, \ldots, x_n$ distribuite în mod independent şi uniform într-o hiper-sferă (notată cu B) care are raza egală cu unitatea[474] şi centrul în O, originea spaţiului $\mathbb{R}^p$. Aşadar, $B = \{x : ||x||^2 \leq 1\} \subset \mathbb{R}^p$, unde $||x|| = \sqrt{x \cdot x}$, iar operatorul $\cdot$ desemnează produsul scalar din $\mathbb{R}^p$.

A. În această secţiune a problemei vom pune în evidenţă o proprietate contra-intuitivă a hiper-sferei B, şi anume faptul că pe măsură ce p, numărul de dimensiuni ale spaţiului creşte, majoritatea instanţelor x_i tind să se plaseze din ce în ce mai departe de originea sistemului de coordonate, adică înspre marginea hiper-sferei. (Acest „comportament“ contravine *bias-ului inductiv* al algoritmului k-NN: „cine se aseamănă se adună“.)

[473] Se verifică uşor că pentru $p = 2$ se obţine volumul sferei ($V_2(r) = \pi r^2$), iar pentru $p = 3$ se obţine volumul sferei: $V_3(r) = \frac{(r\sqrt{\pi})^3}{\frac{3}{4}\sqrt{\pi}} = \frac{4\pi r^3}{3}$. Pentru demonstrarea proprietăţilor funcţiei Γ indicate mai sus, vedeţi problema 31.b de la capitolul de *Fundamente*.

[474] Termenul folosit în limba engleză pentru o astel de hiper-sferă este *unit ball*.

a. Determinaţi în funcţie de p cât anume (i.e., ce fracţie) din volumul hiper-sferei B îl reprezintă volumul hiper-sferei de rază $1/2$ care are centrul în O, originea sistemului de coordonate. Ce observaţi?

b. Determinaţi în funcţie de p cât anume este raza hiper-sferei cu centrul în O, originea sistemului de coordonate, care conţine jumătate din volumul hiper-sferei B. Ce observaţi?

B. În această secţiune a problemei veţi studia „mărimea" vecinătăţii de tip 1-NN pentru originea O a sistemului de coordonate şi cum anume variază această „mărime" în raport cu dimensiunea p. În acest fel, veţi putea vedea care sunt dezavantajele folosirii algoritmului k-NN într-un spaţiu de dimensiune mare.

Din punct de vedere formal, „mărimea" menţionată mai sus va fi identificată cu d^*, distanţa de la O la cel mai apropiat vecin din mulţimea $\{x_1, x_2, \ldots, x_n\}$:

$$d^* \stackrel{not.}{=} \min_{1 \le i \le n} ||\, x_i \,|| \,.$$

***Observaţie*: Din moment ce eşantionul $\{x_1, x_2, \ldots, x_n\}$ este generat în mod aleatoriu, distanţa d^* poate fi văzută ca fiind [produsă de către] o variabilă aleatoare.**

c. În cazul particular $p = 1$, calculaţi expresia *funcţiei de distribuţie cumulativă*[475] a lui d^* (văzută ca variabilă aleatoare), şi anume $P(d^* \le t)$ pentru $t \in [0,\, 1]$. Veţi nota această funcţie cu $F_{n,1}(t)$.

d. Determinaţi expresia *funcţiei de distribuţie cumulativă* (c.d.f.) a lui d^* în cazul general, adică pentru $p \in \{1, 2, 3, \ldots\}$. Veţi nota această funcţie cu $F_{n,p}(t)$.

e. Care este *mediana* variabilei aleatoare d^* (adică, valoarea lui t pentru care $P(d^* \le t) = 1/2$)? Va trebui ca răspunsul să fie formulat în funcţie de n şi p (dimensiunea eşantionului şi, respectiv, dimensiunea spaţiului din care se face extragerea instanţelor, $\mathbb{R}^p$).
Pentru $n = 100$, alcătuiţi un grafic cu valorile [funcţiei] mediane pentru $p = 1, 2, 3, \ldots, 100$. Valorile lui p vor fi plasate pe axa Ox, iar valorile [funcţiei] mediane pe axa Oy. Ce observaţi?

f. Folosind $F_{n,p}(t)$, funcţia de distribuţie cumulativă (c.d.f.) de la punctul d, determinaţi cât de mare ar trebui să fie n (mărimea eşantionului) astfel încât

$$P(d^* \le 0.5) \ge 0.9,$$

adică, cu probabilitate de cel puţin $9/10$, distanţa d^* de la originea O la cel mai apropiat vecin să fie mai mică decât $1/2$ (adică, jumătate din distanţa de la O la marginea hiper-sferei). Va trebui să formulaţi răspunsul ca expresie a unei funcţii în raport cu variabila p.
Reprezentaţi grafic valorile acestei funcţii pentru $p = 1, 2, \ldots, 20$, plasând valorile lui p pe axa Ox şi valorile funcţiei pe axa Oy. Ce observaţi?

***Sugestie*: Pentru $\ln(1-x)$, puteţi face apel la dezvoltarea sa sub formă de *serie Taylor*:**

$$\ln(1-x) = -\sum_{i=1}^{\infty} \frac{x^i}{i} \text{ pentru } -1 \le x < 1.$$

[475] Engl., cummulative distribution function, c.d.f.

g. În urma rezolvării punctelor $c-f$ de mai sus, ce puteţi spune despre dezavantajele algoritmului k-NN în raport cu [diferitele valori posibile pentru] n şi p?

Răspuns:

a. În expresia volumului hiper-sferei de rază r, raportul $\frac{(\sqrt{\pi})^p}{\Gamma\left(\frac{p}{2}+1\right)}$ nu depinde de r. În consecinţă,

$$\frac{V_p(1/2)}{V_p(1)} = \frac{(1/2)^p}{1^p} = \frac{1}{2^p}.$$

Se observă că acest raport tinde la 0. De exemplu, pentru $d = 10$, mai puţin de $1/1000$ din volumul hiper-sferei de rază 1 (B) se află în hiper-sfera de rază $1/2$.

b.

$$V_p(r) = \frac{1}{2}V_p(1) \Leftrightarrow \frac{(r\sqrt{\pi})^p}{\Gamma\left(\frac{p}{2}+1\right)} = \frac{1}{2}\cdot\frac{(\sqrt{\pi})^p}{\Gamma\left(\frac{p}{2}+1\right)} \Leftrightarrow r^p = \frac{1}{2} \Leftrightarrow r = \frac{1}{2^{\frac{1}{p}}} = 2^{-\frac{1}{p}}.$$

Se observă că pentru $p \to \infty$ vom avea $r \to 1/2^0 = 1$. De exemplu, pentru $p = 10000$, obţinem $r \approx 0.99993$. Aşadar, pentru valori suficient de mari ale lui p, cea mai mare parte din volumul hiper-sferei B se află în apropierea marginii ei.

c. Pentru $p = 1$, hiper-sfera de rază 1 este intervalul $[-1, 1]$, iar funcţia de distribuţie cumulativă va avea expresia:

$$F_{n,1}(t) \overset{not.}{=} P(d^* \leq t) = 1 - P(d^* > t) = 1 - P(||x_i|| > t,\ i = 1, 2, \ldots, n)$$

Ţinem cont de presupoziţia de independenţă la generarea punctelor x_i, rezultă:

$$F_{n,1}(t) = 1 - \prod_{i=1}^{n} P(||x_i|| > t) = 1 - (1-t)^n.$$

d. În cazul general, adică pentru p un număr natural oarecare nenul, fixat, vom exprima mai întâi $P(d^* \leq t)$ exact ca mai înainte:

$$\begin{aligned} F_{n,p}(t) \overset{not.}{=} P(d^* \leq t) &= 1 - P(d^* > t) = 1 - P(||x_i|| > t,\ i = 1, 2, \ldots, n) \\ &\overset{indep.\ cdt.}{=} 1 - \prod_{i=1}^{n} P(||x_i|| > t). \end{aligned}$$

Apoi, ţinând cont de presupoziţia de uniformitate la generarea punctelor x_i şi, folosind notaţia $V_p(t)$ pentru volumul hiper-sferei de rază t, obţinem:

$$F_{n,p}(t) = 1 - \left(\frac{V_p(1) - V_p(t)}{V_p(1)}\right)^n = 1 - \left(1 - \frac{V_p(t)}{V_p(1)}\right)^n.$$

În sfârşit, folosind formula *sugerată* în enunţ pentru V_p, rezultă imediat că $F_{n,p}(t) = 1 - (1-t^p)^n$.

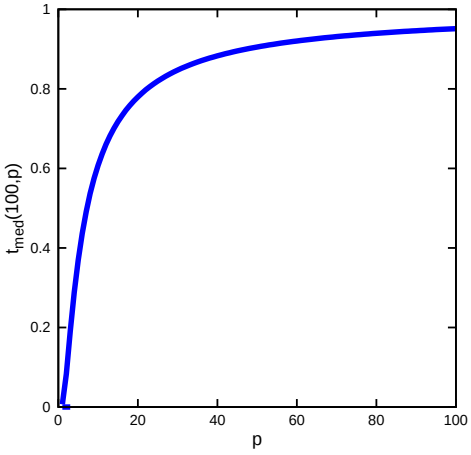
e. Pentru a afla valoarea mediană corespunzătoare variabilei aleatoare d^*, vom rezolva ecuaţia $P(d^* \leq t) = 1/2$ în funcţie de t:

$$\begin{aligned} P(d^* \leq t) = \frac{1}{2} &\Leftrightarrow F_{n,p}(t) = \frac{1}{2} \overset{b}{\Leftrightarrow} 1 - (1 - t^p)^n = \frac{1}{2} \\ &\Leftrightarrow \quad (1 - t^p)^n = \frac{1}{2} \Leftrightarrow 1 - t^p = \frac{1}{2^{1/n}} \\ &\Leftrightarrow \quad t^p = 1 - \frac{1}{2^{1/n}} \end{aligned}$$

Prin urmare,

$$t_{med}(n,p) = \left(1 - \frac{1}{2^{1/n}}\right)^{1/p}.$$

Graficul funcţiei $t_{med}(100, p)$ pentru $p = 1, 2, \ldots, 100$ este cel din figura alăturată. Se observă că hiper-sfera minimală care conţine (cu probabilitate de $1/2$) cel mai apropiat vecin al originii O se lărgeşte foarte repede pe măsură ce p creşte. Pentru valori ale lui p mai mari decât 10, majoritatea dintre cele 100 de instanţe de antrenament sunt mai aproape de conturul hiper-sferei de rază 1 decât de originea O (întrucât $P(d^* > t_{med}(n,p)) = 1/2$).



Cu probabilitate de $1/2$ cel mai apropiat vecin (un x_{i^*}, cu $i^* \in \{1, \ldots, n\}$) faţă de originea O *nu* se află în hiper-sfera de rază $\bar{t} \overset{not.}{=} t_{med}(n,p)$ şi cu centrul în O. Acest cel mai apropiat vecin se află cu probabilitate de $1/2$ dincolo de această hiper-sferă, adică în „coroana“ hiper-sferică delimitată de B şi hiper-sfera de rază $\bar{t}$ şi centru O. În acest caz, toate instanţele x_i cu $i = 1, \ldots, n$ se află în acea coroană hiper-sferică.

f. Putem scrie următorul şir de echivalenţe:

$$\begin{aligned} P(d^* \leq 0.5) \geq 0.9 \quad &\Leftrightarrow \quad F_{n,p}(0.5) \geq 0.9 \Leftrightarrow \\ &\overset{b}{\Leftrightarrow} \quad 1 - \left(1 - \frac{1}{2^p}\right)^n \geq \frac{9}{10} \;\Leftrightarrow\; \left(1 - \frac{1}{2^p}\right)^n \leq \frac{1}{10} \\ &\Leftrightarrow \quad n \cdot \ln\left(1 - \frac{1}{2^p}\right) \leq -\ln 10 \\ &\Leftrightarrow \quad n \geq \frac{\ln 10}{-\ln\left(1 - \frac{1}{2^p}\right)} \end{aligned}$$

Se poate vedea imediat că membrul din partea dreaptă a inegalităţii de mai sus tinde la $+\infty$ pentru $p \to +\infty$. Este necesar să vedem *cât de repede* are loc această tindere la infinit. Pentru aceasta, vom folosi descompunerea lui

$-\ln(1-1/2^p)$ sub forma unei serii Taylor (luând $x = 1/2^p$):[476]

$$P(d^* \leq 0.5) \geq 0.9 \quad \Rightarrow \quad n \geq (\ln 10)\, 2^p \frac{1}{1 + \frac{1}{2}\cdot\frac{1}{2^p} + \frac{1}{3}\cdot\frac{1}{2^{2p}} + \ldots + \frac{1}{n}\frac{1}{2^{(n-1)p}} + \ldots}$$

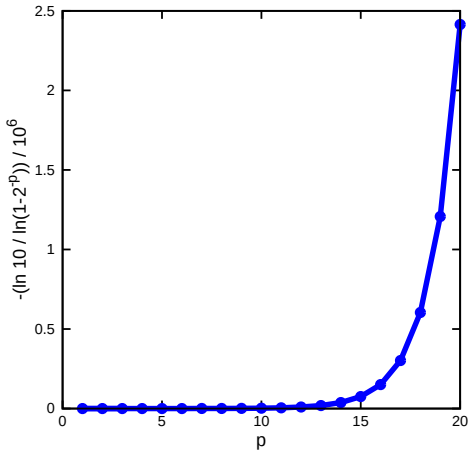
$$\Rightarrow \quad n > \frac{4}{3} 2^{p-1}\ \ln 10.$$

Pentru obţinerea ultimei inegalităţi de mai sus am ţinut cont de faptul că inegalitatea $\frac{1}{n\cdot 2^{(n-1)p}} \leq \frac{1}{2^n} \Leftrightarrow 2^n \leq n\cdot 2^{(n-1)p}$ are loc pentru orice p şi n cu $p \geq 1$ şi $n \geq 2$,[477] **deci**

$$\begin{aligned} & 1 + \frac{1}{2}\cdot\frac{1}{2^p} + \frac{1}{3}\cdot\frac{1}{2^{2p}} + \ldots + \frac{1}{n}\cdot\frac{1}{2^{(n-1)p}} + \ldots \\ \leq\ & 1 + \frac{1}{2^2} + \frac{1}{2^3} + \ldots + \frac{1}{2^n} + \ldots \\ <\ & \left[1 + \frac{1}{2} + \frac{1}{2^2} + \frac{1}{2^3} + \ldots + \frac{1}{2^n} + \ldots\right] - \frac{1}{2} \\ & \to \frac{1}{1-\frac{1}{2}} - \frac{1}{2} = 2 - \frac{1}{2} = \frac{3}{2}. \end{aligned}$$

Observaţi că limita pe care tocmai am calculat-o $(3/2)$ este o limită *superioară*, de aceea inegalitatea se păstrează până la final.

Aşadar, rezultă că n creşte în mod exponenţial în raport cu p.[478] **Graficul pentru marginea inferioară dedusă mai sus $(-\ln 10/\ln(1-2^{-p}))$ este cel din figura alăturată.**



Se observă că, într-adevăr, creşterea acestei margini inferioare (deci şi a lui n) este exponenţială.

g. Conform intuiţiei, clasificatorul k-NN se comportă bine atunci când instanţa de test x_q este situată într-o vecinătate densă de instanţe de antrenament. Totuşi, analiza teoretică de mai sus ne arată — în ipoteza că datele de antrenament sunt distribuite uniform — că pentru a ne asigura că punctul x_q are o vecinătate densă, numărul tuturor instanţelor de antrenament trebuie să

[476] $-ln(1-x) = x + \frac{1}{2}x^2 + \frac{1}{3}x^3 + \ldots + \frac{1}{n}x^n + \ldots$ pentru orice $x \in (-1, +1)$.

[477] Demonstraţia se poate face prin inducţie după valorile lui p.

[478] Mai detaliat: n, numărul de instanţe de antrenament necesare pentru a ne asigura că d^* (distanţa până la cel mai apropiat vecin al originii O) este cu o probabilitate mare (şi anume, $9/10$) mai mică decât 0.5 creşte în mod exponenţial în raport cu p.

crească exponenţial în raport cu p, ceea ce nu este fezabil pentru valori mari ale lui p. (Parametrul p este dimensiunea spaţiului în care se lucrează, adică numărul de trăsături ale instanţelor de antrenament şi, respectiv, de test).

În consecinţă, pentru aplicaţiile practice în care se folosesc date cu multe atribute, este recomandat ca execuţia algoritmului k-NN să fie precedată de efectuarea unei „selecţii de trăsături“ (engl., feature selection).

10. **(Algoritmul 1-NN [comparativ cu clasificatorul Bayes Optimal]: o margine superioară pentru *eroarea medie asimptotică* [la antrenare])**

■ □ • ***CMU, 2005 spring, C. Guestrin, T. Mitchell, HW3, pr. 1***

Un rezultat interesant obţinut de Cover şi Hart (1967) arată că, atunci când numărul datelor de antrenament tinde la infinit, iar datele de antrenament umplu spaţiul în mod dens, rata medie a erorii produsă de către clasificatorul 1-NN este mărginită superior de dublul ratei medii a erorii pentru clasificatorul Bayes Comun (engl., Joint Bayes), care este numit adeseori şi *Bayes Optimal* (engl., Optimal Bayes).

În acest exerciţiu vi se va arăta, pas cu pas, cum se demonstrează rezultatul lui Cover şi Hart în cazul particular al clasificării binare. Aşadar, fie $x_1, x_2, \ldots$ instanţele de antrenament, iar $y_1, y_2, \ldots$ etichetele corespunzătoare, cu $y_i \in \{0, 1\}$. Putem considera instanţele x_i ca fiind puncte într-un spaţiu euclidian d-dimensional.

Notăm $p_y(x) = P(X = x \mid Y = y)$ probabilitatea condiţionată care reprezintă distribuţia instanţelor din clasa y. Vom presupune că aceste probabilităţi condiţionate sunt continue în raport cu variabila x şi că $p_y(x) \in (0, 1)$ pentru orice x şi orice y. Notăm cu θ probabilitatea ca un exemplu de antrenament selectat în mod aleatoriu să fie din clasa 1, aşadar $\theta \stackrel{not.}{=} P(Y = 1)$. Din nou, presupunem că $\theta \in (0, 1)$.

a. Calculaţi probabilitatea a posteriori ca o instanţă oarecare x să aparţină clasei 1: $q(x) \stackrel{not.}{=} P(Y = 1 \mid X = x)$. Exprimaţi $q(x)$ în funcţie de $p_0(x)$, $p_1(x)$ şi θ.

b. Clasificatorul Bayes Optimal asignează unui punct dat x cea mai probabilă clasă, $\text{argmax}_y P(Y = y \mid X = x)$. (Aceasta implică faptul că algoritmul Bayes Optimal maximizează probabilitatea clasificării corecte a tuturor datelor.) Considerând o instanţă oarecare x, calculaţi probabilitatea ca x să fie clasificat greşit folosind clasificatorul Bayes Optimal, în funcţie de probabilitatea $q(x) \stackrel{not.}{=} P(Y = 1|X = x)$ care tocmai a fost calculată la punctul precedent. Veţi desemna această nouă probabilitate cu $Error_{Bayes}(x)$.

c. Acum considerăm clasificatorul 1-NN. Acesta îi asignează unei instanţe oarecare de test x eticheta celei mai apropiate instanţe de antrenament x'. Dată fiind o instanţă de test x (aleasă în mod arbitrar, dar fixată), calculaţi *eroarea „aşteptată“* (engl., expected error) produsă de către clasificatorul 1-NN, adică probabilitatea ca instanţa x să fie clasificată greşit. Notaţi această probabilitate cu $Error_{1\text{-}NN}(x)$ şi exprimaţi-o sub forma unei funcţii definită în raport cu probabilităţile $q(x)$ şi $q(x')$.

d. În *cazul asimptotic*, numărul de exemple de antrenament al fiecărei clase tinde la infinit, iar datele de antrenament umplu spaţiul în mod dens. Atunci

$q(x') \to q(x)$**, unde, ca şi mai sus,** x' **este cel mai apropiat vecin al lui** x**.**[479] **Făcând această substituţie în rezultatul obţinut la punctul anterior, deduceţi expresia** ***erorii asimptotice*** **pentru clasificatorul 1-NN în punctul** x**, adică** $\lim_{x' \to x} Error_{1\text{-}NN}(x)$**, în funcţie de probabilitatea** $q(x)$**.**

e. Arătaţi că eroarea asimptotică obţinută la punctul d **este mai mică decât dublul erorii clasificatorului Bayes Optimal obţinută la punctul** b**, adică:**

$$\lim_{x' \to x} Error_{1\text{-}NN}(x) \leq 2 Error_{Bayes}(x).$$

În final, din această inegalitate deduceţi relaţia corespunzătoare între ratele medii ale erorilor:[480]

$$E[\lim_{n \to \infty} Error_{1\text{-}NN}] \leq 2E[Error_{Bayes}].$$

Răspuns:

a. Conform enunţului, $p_1(x) \stackrel{not.}{=} P(X = x|Y = 1)$**,** $p_0(x) \stackrel{not.}{=} P(X = x|Y = 0)$**,** $q(x) \stackrel{not.}{=} P(Y = 1|X = x)$ **şi** $\theta \stackrel{not.}{=} P(Y = 1)$**. Putem calcula probabilitatea** $q(x)$ **în funcţie de** $p_1(x)$**,** $p_0(x)$ **şi** θ **folosind formula lui Bayes:**

$$\begin{aligned} q(x) &\stackrel{Bayes}{=} \frac{P(X = x|Y = 1)P(Y = 1)}{P(X = x)} \\ &= \frac{P(X = x|Y = 1)P(Y = 1)}{P(X = x|Y = 1)P(Y = 1) + P(X = x|Y = 0)P(Y = 0)} \\ &= \frac{p_1(x)\ \theta}{p_1(x)\ \theta + p_0(x)(1 - \theta)} \end{aligned}$$

b. Este imediat faptul următor: probabilitatea ca algoritmul Bayes Optimal să greşească este $P(Y = 0|X = x)$ **în cazul în care** $P(Y = 1|X = x) \geq P(Y = 0|X = x)$**, respectiv** $P(Y = 1|X = x)$ **atunci când** $P(Y = 0|X = x) \geq P(Y = 1|X = x)$**. Altfel spus,**

$$\begin{aligned} Error_{Bayes}(x) &= \min\{P(Y = 0|X = x), P(Y = 1|X = x)\} \\ &= \min\{1 - q(x), q(x)\} = \begin{cases} q(x) \text{ \textbf{în cazul} } q(x) \in [0\,,\, 1/2] \\ 1 - q(x) \text{ \textbf{în cazul} } q(x) \in (1/2\,,\, 1]. \end{cases} \end{aligned}$$

c. Algoritmul 1-NN greşeşte atunci când instanţa de test x **are eticheta** 1**, iar** x'**, cel mai apropiat vecin al lui** x**, are eticheta** 0**, sau invers, adică atunci când** x **are eticheta** 0 **iar** x' **are eticheta** 1**. În consecinţă, folosind algoritmul 1-NN, eroarea „aşteptată“ la clasificarea lui** x **este:**

$$\begin{aligned} Error_{1\text{-}NN}(x) &= P(Y = 1|X = x)P(Y = 0|X = x') + \\ &\quad P(Y = 0|X = x)P(Y = 1|X = x') \\ &= q(x)(1 - q(x')) + (1 - q(x))q(x'). \end{aligned}$$

[479] Adică, $P(Y = 1|X = x') \to P(Y = 1|X = x)$. Aceasta se justifică ţinând cont de continuitatea lui $p_y(x) \stackrel{not.}{=} P(X = x|Y = y)$ care a fost asumată în enunţ şi, de asemenea, de rezultatul obţinut la punctul a.

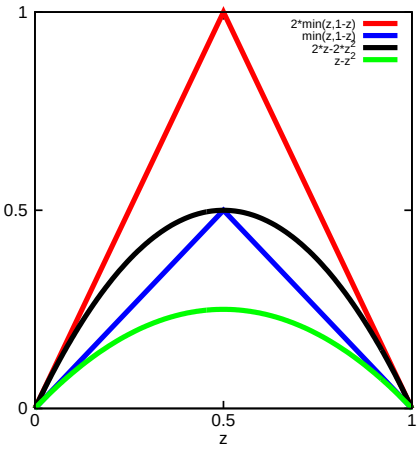
[480] Cititorul atent va remarca faptul că în expresia de mai jos ($E[\lim_{n \to \infty} Error_{1\text{-}NN}]$) s-a înlocuit $\lim_{x \to x'}$ (folosită anterior) cu $\lim_{n \to \infty}$, pentru că se face trecerea la medii. Când $n \to \infty$, conform presupoziţiilor din enunţ, rezultă $x' \to x$ pentru orice x.

d. Este imediat că $\lim_{x' \to x} Error_{1\text{-}NN}(x) = 2q(x)(1 - q(x))$**.**

e. Se arată imediat că $z - z^2 \leq z$ **pentru** $\forall z$**, deci şi pentru** $z \in [0\,,\,1/2]$**, iar** $z - z^2 \leq 1 - z$ **pentru** $\forall z$**, deci şi pentru** $z \in [1/2\,,\,1]$**. Aşadar, pentru orice** x**, vom avea:**

$$q(x)(1-q(x)) \leq \begin{cases} q(x) \text{ dacă } q(x) \in [0\,,\,1/2] \\ 1 - q(x) \text{ dacă } q(x) \in (1/2\,,\,1]. \end{cases}$$

Corelând cu rezultatul de la punctul b**, obţinem:** $2q(x)(1-q(x)) \leq 2Error_{Bayes}(x)$ **pentru orice** x**.**



Combinând acest rezultat cu egalitatea de la punctul d**, rezultă că inegalitatea**

$$\lim_{n \to \infty} Error_{1\text{-}NN}(x) = \lim_{x' \to x} Error_{1\text{-}NN}(x) \leq 2Error_{Bayes}(x)$$

este adevărată pentru orice x**. Înmulţind ambii membri ai acestei inegalităţi cu** $P(x)$ **şi însumând după toate valorile lui** x **— de fapt, integrând în raport cu** x **—, obţinem:**

$$E[\lim_{n \to \infty} Error_{1\text{-}NN}] \leq 2E[Error_{Bayes}].$$

Aşadar, am demonstrat că media (sau: rata medie a) erorii asimptotice a algoritmului 1-NN este cel mult dublul mediei (sau: ratei medii a) erorii algoritmului Bayes Optimal.

***Observaţia 1*: Această margine a erorii asimptotice nu se păstrează şi în cazul neasimptotic, unde numărul de exemple de antrenament este finit.**

***Observaţia 2*: La fel, se poate arăta că** $2z - 2z^2 \geq z$ **pentru** $\forall z \in [0\,,\,1/2]$ **şi** $2z - 2z^2 \geq 1 - z$ **pentru** $\forall z \in [1/2\,,\,1]$**. Luând din nou** $z = q(x)$ **şi ţinând cont de rezultatul de la punctul** b**, obţinem că** $2q(x)(1 - q(x)) \geq Error_{Bayes}(x)$**, pentru orice** x**. Combinând această inegalitate cu egalitatea de la punctul** d**, rezultă o nouă inegalitate:**

$$\lim_{n \to \infty} Error_{1\text{-}NN}(x) = \lim_{x' \to x} Error_{1\text{-}NN}(x) \geq Error_{Bayes}(x) \text{ pentru orice } x.$$

În final, trecând la medii, obţinem următoarea inegalitate (care era de altfel de aşteptat):

$$E[\lim_{n \to \infty} Error_{1\text{-}NN}] \geq E[Error_{Bayes}].$$

***Observaţia 3* (preluată din *An Elementary Introduction to Statistical Learning Theory*, de Sanjeev Kulkarni şi Gilbert Harman, 2011, pag. 69): În mod intuitiv, dacă mărim valoarea lui** k**, ar trebui ca eroarea medie a algoritmului** k**-NN să se reducă. Într-adevăr, în anumite condiţii (dar nu în orice condiţii!) se poate arăta că are loc următoarea inegalitate dublă:**

$$E[Error_{Bayes}] \leq E[\lim_{n \to \infty} Error_{k\text{-}NN}] \leq \left(1 + \frac{1}{k}\right) E[Error_{Bayes}].$$

Este de remarcat faptul că există distribuţii probabilistice ale datelor pentru care clasificatorul 1-NN se comportă mai bine decât k-NN pentru orice $k \neq 1$.

Observaţia 4 **(preluată din aceeaşi lucrare, *An Elementary Introduction to Statistical Learning Theory*, citată mai sus): Dacă lucrăm cu k_n-NN, adică îl fixăm pe k în funcţie de n (numărul instanţelor de antrenament), se poate demonstra că în cazul în care $\frac{k_n}{n} \to 0$ pentru $n \to \infty$ (de exemplu, $k_n = \sqrt{n}$), se obţine:**

$$E[\lim_{n\to\infty} Error_{k_n\text{-}NN}] = E[Error_{Bayes}].$$

Aceasta înseamnă că, la limită, algoritmul k_n-NN se comportă la fel de bine ca algoritmul Bayes Optimal!

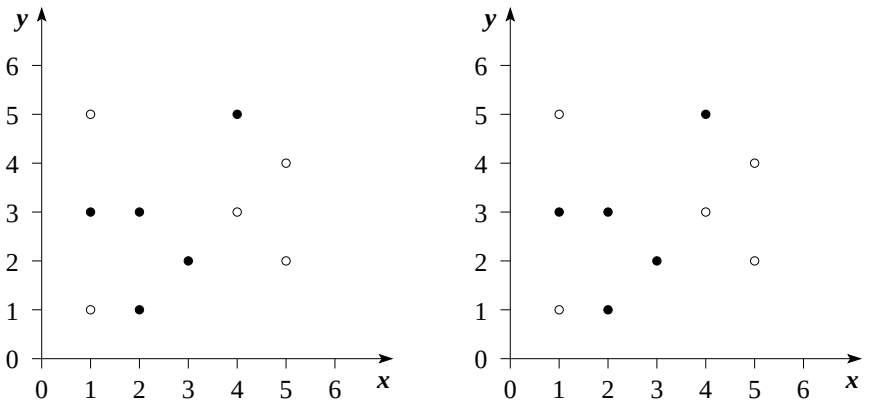
11. **(Comparaţie între algoritmii 1-NN şi ID3: zone de decizie şi separatori decizionali)**

prelucrare de Liviu Ciortuz, după
■ • ***CMU, 2007 fall, Carlos Guestrin, HW2, pr 1.4***

Pe setul de date de mai jos desenaţi *graniţele de decizie* şi apoi haşuraţi *suprafeţele de decizie* produse de

a. algoritmul 1-NN (veţi obţine deci diagrama Voronoi);

b. algoritmul ID3 extins cu capacitatea de a procesa atribute cu valori continue.



Răspuns:

a. *Algoritmul 1-NN*:

Pentru a defini suprafeţele de decizie în acest caz se procedează în felul următor:

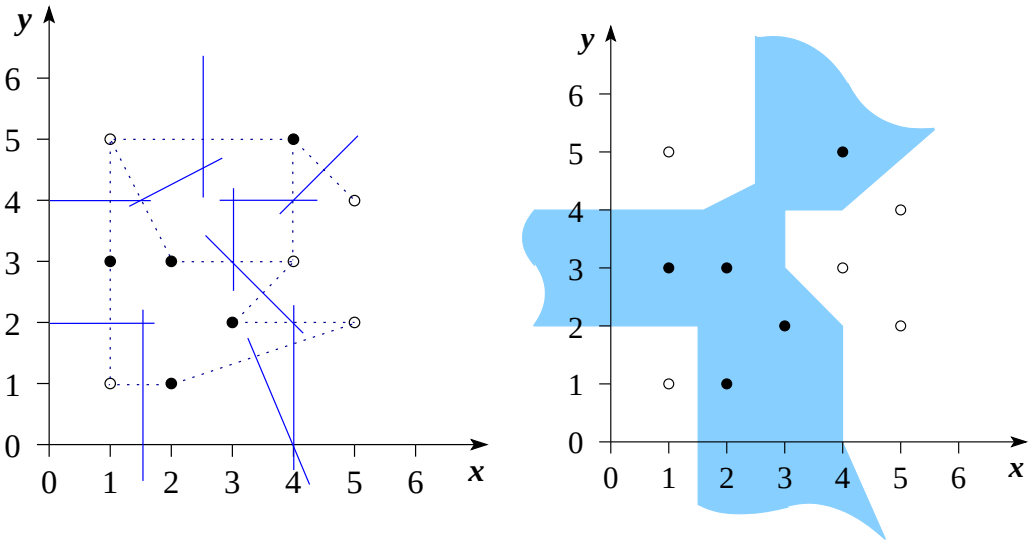
– se trasează mediatoarele segmentelor de dreaptă determinate de perechi de puncte din setul de antrenament care sunt etichetate în mod diferit;

– se stabilesc intersecţiile acestor mediatoare; acest lucru este reprezentat în figura de mai jos, partea stângă;

– apoi se marchează pe aceste mediatoare acele segmente (determinate de intersecţii) care determină zonele de decizie corespunzătoare clasificatorului 1-NN.

***Observaţii*:**
1. De fapt, întrucât nu este necesar să se lucreze cu toate perechile de instanţe cu etichete diferite sunt relevante pentru clasificarea unei instanţe / zone, în timpul „executării“ punctelor de mai sus este foarte util să se ţină cont de următoarea regulă / euristică de *ghidare*: alegerea perechilor de instanţe şi apoi a segmentelor de pe mediatoare se va face urmărind delimitarea zonelor corespunzătoare instanţelor negative (○) de zonele corespunzătoare instanţelor pozitive (•).
2. Suplimentar, în jurul fiecărui punct de antrenament A se poate identifica câte o zonă [convexă] care va constitui mulţimea punctelor mai apropiate de A decât de oricare alt punct din setul de date de antrenament. (Toate punctele din această zonă convexă vor avea aceeaşi clasificare / etichetă ca şi punctul A.) Aceasta este uşor de văzut pentru instanţele negative (○) din cazul de faţă.

În figura următoare, în partea dreaptă am haşurat zona corespunzătoare instanţelor pozitive (•).



b. *Algoritmul ID3*:

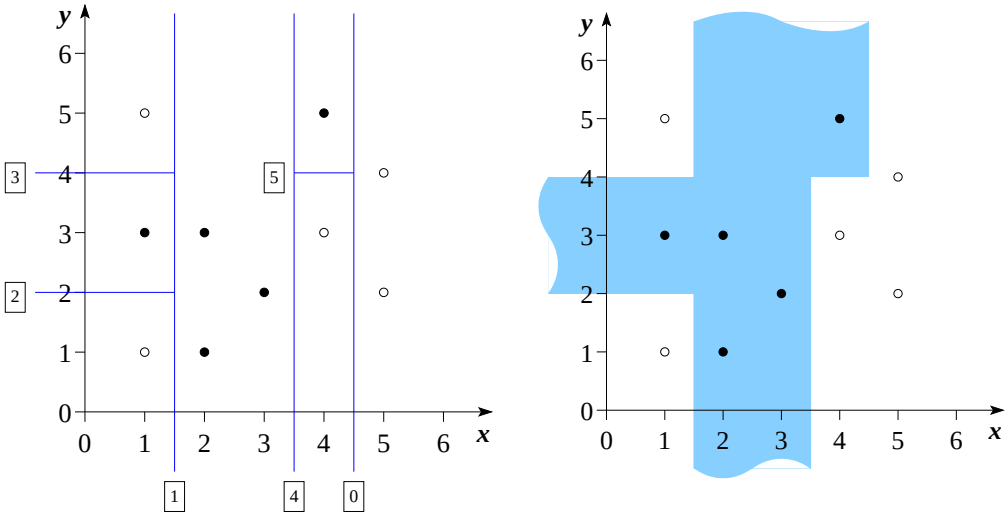
Pentru construirea suprafeţelor de decizie ale algoritmului ID3,

– mai întâi se determină valorile-prag pentru teste (adică, punctele de splitare de pe fiecare axă) şi apoi se alege testul corespunzător nodului rădăcină din arborele ID3; se trasează o dreaptă prin punctul respectiv, paralelă cu cealaltă axă. În cazul nostru, testul din nodul rădăcină va fi $x < 4.5$.

– pentru fiecare test / split ulterior se trasează o semidreaptă (sau un segment de dreaptă) mărginit(ă) la un capăt de dreapta corespunzătoare nodului-părinte.

În figura de mai jos, în partea stângă am vizualizat toate testele / split-urile realizate de algoritmul ID3 pentru a învăţa complet arborele de decizie, iar

în partea dreaptă, ca şi mai sus, am păstrat doar frontierele dintre zonele cu clasificări diferite.



Se observă că suprafeţele de decizie determinate de cei doi algoritmi nu sunt identice, dar sunt totuşi asemănătoare într-o anumită măsură (pentru că ambele sunt consistente cu datele de antrenament).

***Observaţii*:**
3. Este de reţinut faptul că suprafaţele de decizie produse de algoritmul ID3 nu sunt neapărat unic determinate, fiindcă sunt situaţii în care două teste diferite pot conduce la acelaşi câştig de informaţie. De exemplu, dacă în exerciţiul nostru am avea de partiţionat la un moment dat mulţimea formată din instanţele de antrenament $(1,1),(1,3),(2,1),(2,3),(3,2)$, atunci testele $x > 1.5$ şi $y > 1.5$ ar produce acelaşi câştig de informaţie, iar suprafeţele de decizie rezultate ar fi determinate (în mod diferit!) de ce anume alegem ca prim test.
4. De asemenea, trebuie să scoatem în evidenţă faptul că pragurile / spliturile care sunt calculate de algoritmul ID3 pentru un acelaşi atribut continuu pot diferi de la un nod de test la altul. De exemplu, la nodul rădăcină (nodul 0), atunci când se calculează câştigul de informaţie maxim, pentru atributul y se iau în calcul pragurile 1.5, 2.5, 3.5, şi 4.5, în vreme ce la nodurile 2 şi / sau 3 (vedeţi figura de mai sus, partea stângă) se analizează pragurile 2 şi 4, întrucât seturile / partiţiile de instanţe asignate acestor noduri sunt diferite!

12. **(Comparaţie între algoritmii 1-NN, ID3 cu atribute continue şi SVM: eroarea la antrenare, eroarea la CVLOO)**

prelucrare de Liviu Ciortuz, după
• ***CMU, 2003 fall, T. Mitchell, A. Moore, midterm exam, pr. 5***

Folosind metoda 1-NN cu distanţa euclidiană, învăţăm un clasificator cu două valori pentru atributul de ieşire, $Y = 0$ şi $Y = 1$, pornind de la datele de antrenament din tabelul de mai jos (X_1 şi X_2 sunt atribute de intrare).

X_1	X_2	Y
0	0	1
1	0	1
2	0	1
2.5	2	1
3	0	1
1	2	0
1.5	0	0
2	2	0
3	2	0
4	2	0

a. Care este eroarea la antrenare (exprimată ca număr de exemple clasificate eronat)?

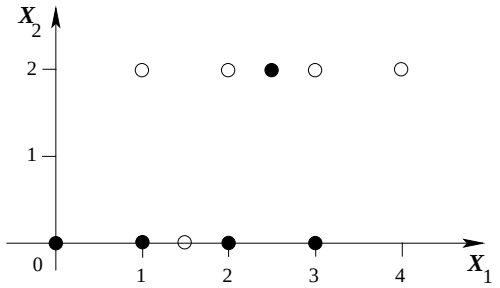
b. Care este eroarea la cross-validare folosind metoda "Leave-One-Out"?

c. Răspundeţi la întrebările de mai sus, considerând acum arbori de decizie cu atribute numerice continue în locul metodei 1-NN.

d. În sfârşit, răspundeţi la întrebările a şi b, considerând maşini cu vectori-suport în locul metodei 1-NN. (Se vor considera doar SVM-uri în cazul liniar cu margine "soft", cu un parametru C suficient de mare pentru a minimiza numărul de instanţe de antrenament clasificate eronat.)

Răspuns:

Reprezentarea datelor în planul euclidian este cea din figura alăturată.



a. După cum am explicat şi la exerciţiul 2, întrucât datele de antrenament nu conţin inconsistenţe, eroarea la antrenare produsă de algoritmul 1-NN va fi 0.

b. Comportamentul algoritmului la cross-validare cu metoda "Leave-One-Out" este cel descris în tabelul următor:

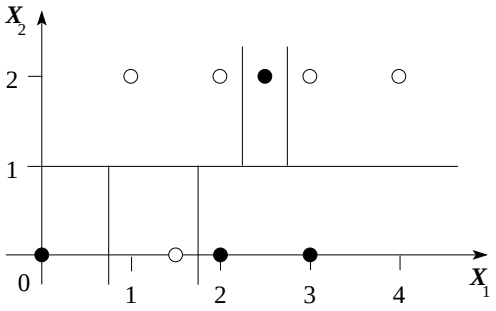
Data	Eticheta	Vecinătate	Clasificare la CVLOO	Eroare?
$(0;0)$	1	$(1;0)$	1	**nu**
$(1;0)$	1	$(1.5;0)$	0	**da**
$(2;0)$	1	$(1.5;0)$	0	**da**
$(2,5;2)$	1	$(2;2)/(3;2)$	0	**da**
$(3;0)$	1	$(2;0)$	1	**nu**
$(1;2)$	0	$(2;2)$	0	**nu**
$(1,5;0)$	0	$(1;0)/(2;0)$	1	**da**
$(2;2)$	0	$(2.5;2)$	1	**da**
$(3;2)$	0	$(2.5;2)$	1	**da**
$(4;2)$	0	$(3;2)$	0	**nu**

Deci în total avem 6 erori (din totalul de 10 instanţe), ceea ce indică faptul ca algoritmul 1-NN este foarte puţin adecvat pentru acest gen de date.

c. Eroarea la antrenare cu algoritmul ID3 pe acest set de date este 0, fiindcă instanţele sunt etichetate în mod consistent. Eroarea la CVLOO produsă de algoritmul ID3 pe acelaşi set de date se poate calcula destul de uşor, ţinând cont că este suficient să determinăm graniţele de decizie — deci şi zonele de decizie — corespunzătoare celor două clase.[481]

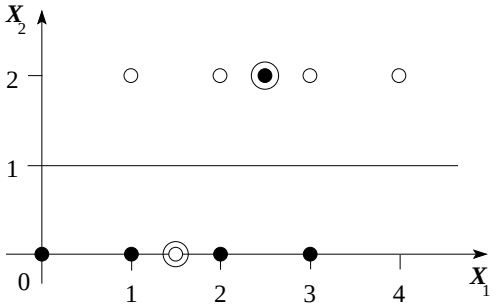
[481] Datorită mixării puternice a datelor în raport cu separatorii de forma $X_1 = t$, la fiecare din cele 10 cazuri se va obţine în nodul rădăcină acelaşi test: $X_2 < 1$.

De *exemplu*, atunci când se va elimina instanţa (1,0), se vor obţine graniţele de decizie ca în figura alăturată. Este imediat că *modelul* rezultat în acest caz va clasifica instanţa (1,0) în mod eronat.



Se constată că eroarea de tip CVLOO produsă de ID3 cu atribute continue este de 6 (din totalul de 10) instanţe. Ba chiar, — este încă o coincidenţă! — ID3 produce la CVLOO exact aceleaşi erori (mai precis, el clasifică în mod eronat exact aceleaşi instanţe) ca şi algoritmul 1-NN.

d. Se observă uşor că separatorul liniar care minimizează eroarea la antrenare este cel reprezentat în figura alăturată. Datele clasificate eronat de către acest separator sunt cele două puncte încercuite din figură; ele sunt de asemenea singurele puncte care generează eroare şi la testare cu metoda CVLOO.



Mai concret, în cazul CVLOO folosind SVM, deoarece separatorul optimal este „susţinut“ de mai mulţi vectori-suport pe fiecare parte, lipsa unuia singur dintre ei nu influenţează cu nimic construirea separatorului. În fiecare caz în parte se va învăţa ca separator dreapta paralelă cu Ox_1 care trece prin punctul $(0, 1)$.

Avem deci în ambele situaţii, adică atât la antrenare cât şi la cross-validare cu metoda "Leave-One-Out", (doar) două puncte clasificate eronat de către SVM.

Comparând cei trei clasificatori, 1-NN, ID3 cu atribute continue şi SVM, rezultă în mod clar că SVM este cel mai convenabil pe acest set de date (deşi la antrenare SVM produce două erori, iar 1-NN şi ID3 nicio eroare), fiindcă nu produce *overfitting* (aici!).

13. **(Comparaţii între algoritmii 1-NN şi ID3: Da sau nu?)**

• ○ ***CMU, 2009 spring, Ziv Bar-Joseph, final exam, pr. 2.1***
CMU, 2015 fall, Z. Bar-Joseph, E. Xing, HW1, pr. 3.a

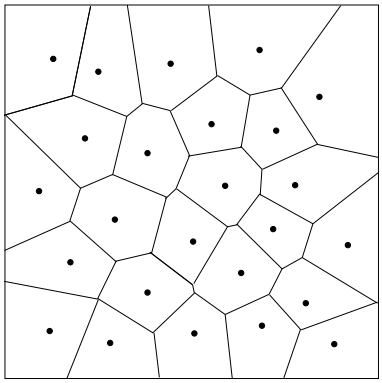
a. Este posibil să se construiască un arbore de decizie (având în fiecare nod intern teste de forma $x > a$, $x \leq b$, $y > c$, sau $y \leq d$, unde a, b, c şi d sunt numere reale oarecare) care să producă la clasificare aceleaşi rezultate ca şi algoritmul 1-NN folosind distanţa euclidiană? Justificaţi răspunsul.

(Învăţare rapidă / "eager" vs. învăţare lentă / "lazy"; k-NN vs. ID3)

CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, HW1, pr. 3.3

b. Algoritmul ID3 este o metodă de învăţare de tip "batch", care solicită ca toate datele de antrenament să-i fie puse la dispoziţie pentru a putea elabora arborele de decizie. Aşadar, în situaţia în care date de antrenament suplimentare ne sunt puse ulterior la dispoziţie, acestea trebuie tratate cu atenţie fiindcă ele pot modifica arborele de decizie rezultat în urma învăţării.[482] Algoritmul k-NN suferă şi el de această problemă? Justificaţi.

Răspuns:



a. Nu.
Suprafeţele de decizie pentru algoritmul 1-NN corespund diagramei Voronoi şi nu sunt neapărat paralele cu axele de coordonate, după cum se observă în figura alăturată.
Suprafeţele de decizie pentru un arbore de decizie cu atribute cu valori continue sunt întotdeauna paralele cu axele de coordonate, deoarece deciziile din fiecare nod sunt de forma $x > a$, $x \leq b$, $y > c$, sau $y \leq d$, $\forall a, b, c, d \in \mathbb{R}$.

Aşadar, răspunsul este negativ pentru cazul general, deşi există situaţii în care cei doi algoritmi produc exact aceeaşi clasificare.

b. Nu.
Răspunsul decurge din modul de lucru a algoritmului k-NN, care este un algoritm de învăţare de tip "lazy": k-NN estimează valoarea locală a unei funcţii-target $\hat{f}$ pentru una sau mai multe instanţe de test x_q. Valoarea $\hat{f}(x_q)$ nu depinde decât de cei mai apropiaţi vecini ai lui x_q; celelalte instanţe de antrenament nu sunt necesare pentru calculul lui $\hat{f}(x_q)$. Evident, dacă pe măsură ce se acumulează noi date de antrenament se modifică şi vecinătatea lui x_q, atunci se prea poate să se modifice şi $\hat{f}(x_q)$. Însă în sine, algoritmul procedează exact la fel ca mai înainte. Se poate modifica doar outputul lui. Spre deosebire de algoritmul k-NN, la învăţarea arborilor de decizie adăugarea de noi instanţe modifică în general şi modelul / arborele rezultat, nu doar decizia pentru o instanţă de test particulară.

14. **(Comparaţii între algoritmii 1-NN şi ID3: [o clasă de] seturi de date de antrenament pe care cei doi clasificatori obţin rezultate identice)**

prelucrare de Liviu Ciortuz, după
• ○ CMU, 2009 spring, Ziv Bar-Joseph, final exam, pr. 2.1
CMU, 2015 fall, Z. Bar-Joseph, E. Xing, HW1, pr. 3.a

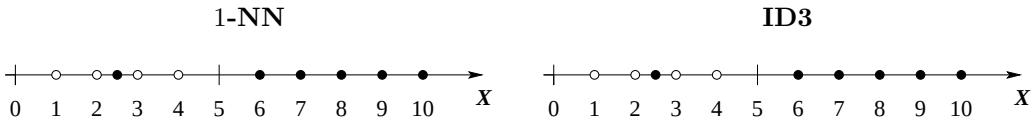
Considerăm instanţele de antrenament $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ având respectiv etichetele $y_1, y_2, \ldots, y_n$. Acest exerciţiu vă va ajuta să răspundeţi la întrebarea următoare:

[482] Vedeţi problema 16 de la capitolul *Arbori de decizie*.

Este oare posibil ca aplicând algoritmul ID3 cu atribute numerice continue să obţinem aceleaşi rezultate la testare / generalizare (deci şi aceleaşi zone de decizie) ca şi cele produse de clasificatorul 1-NN folosind distanţa euclidiană?

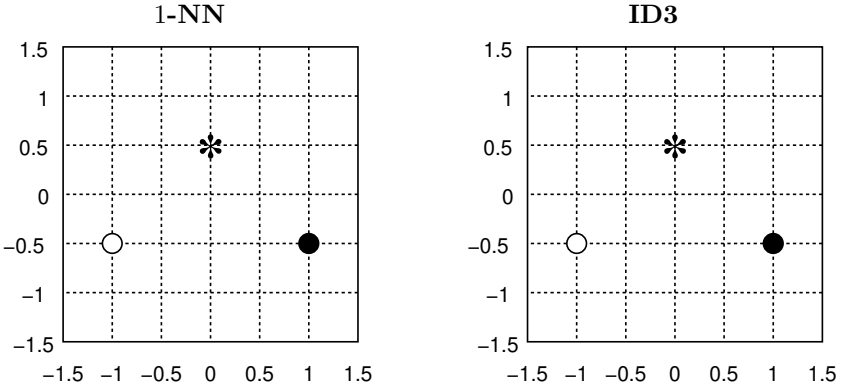
Veţi aborda separat cazurile $d = 1$ şi $d > 1$.

a. În cazul $d = 1$, pentru a înţelege mai bine cerinţa problemei, pentru setul de date de mai jos trasaţi diagrama Voronoi pe figura din stânga. Veţi identifica în mod clar separatorii decizionali.
Pe figura din dreapta, desenaţi graniţele de decizie şi zonele de decizie determinate de algoritmul ID3. (Vă readucem aminte convenţia noastră: semnul $\circ$ desemnează un exemplu negativ, iar semnul $\bullet$ desemnează un exemplu pozitiv.)



Ce concluzie puteţi trage referitor la calculul erorilor de tip CVLOO pentru cei doi clasificatori (1-NN şi ID3 cu atribute numerice continue) pe seturi de date din $\mathbb{R}$?

b. În cazul $d > 1$, procedaţi similar pentru setul de date reprezentat mai jos, unde simbolii $\circ$, $\bullet$ şi $*$ denotă trei clase diferite. Adică, trasaţi diagrama Voronoi pe figura din stânga, iar pe figura din dreapta desenaţi graniţele de decizie şi zonele de decizie determinate de algoritmul ID3 (justificaţi riguros!).

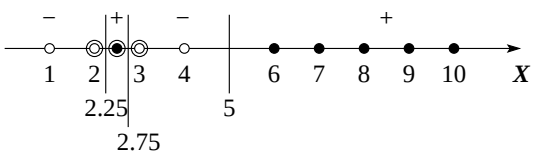


În cazul algoritmului ID3, rezultatul este unic determinat? Dacă da, explicaţi de ce. Dacă nu, arătaţi câte variante se pot obţine în total.

c. Ce răspuns puteţi formula acum referitor la *Întrebarea* de mai sus, din enunţ? Încercaţi să *generalizaţi*, pornind de la exemplele de la punctele a şi b.

Răspuns:

a. Atât pentru 1-NN cât şi pentru ID3 zonele de decizie ąi separatorii decizionali sunt conform figurii următoare:



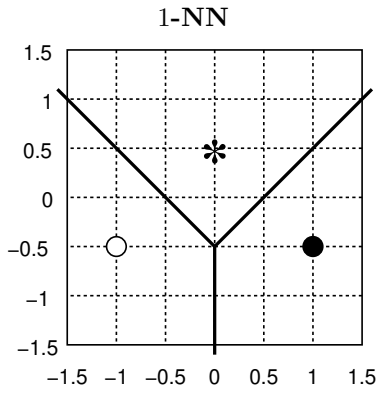
În această figură am încercuit exemplele care conduc la eroare la cross-validarea de tip "Leave-One-Out", atât pentru algoritmul 1-NN cât şi pentru algoritmul ID3.

Se constată că întotdeauna, pe orice set de exemple *consistente* din $\mathbb{R}$ (deci cu un singur atribut de intrare, care este numeric şi continuu), atât algoritmul 1-NN cât şi algoritmul ID3 produc aceleaşi rezultate la testare / generalizare.

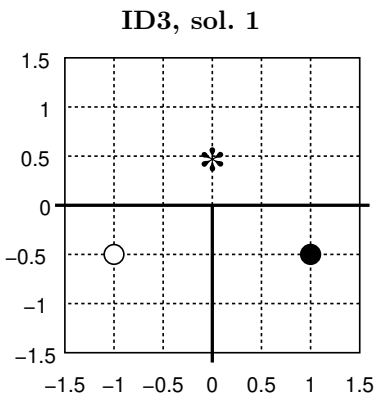
Explicaţia ţine de folosirea distanţei euclidiene la algoritmul 1-NN şi respectiv modul cum se stabilesc pragurile de splitare de către algoritmul ID3 (şi anume, la jumătatea distanţei dintre două instanţe consecutive pe axa reală, care au etichete diferite).

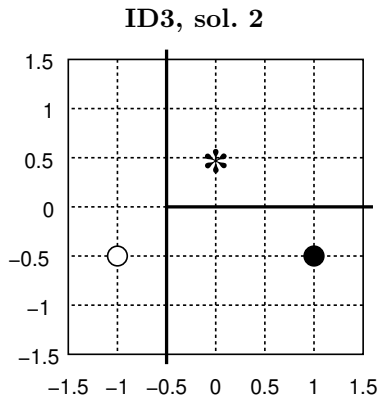
În consecinţă, pe seturi de date *consistente* din $\mathbb{R}$, cei doi algoritmi produc exact aceleaşi zone de decizie şi aceiaşi separatori decizionali, precum şi aceleaşi erori la CVLOO.

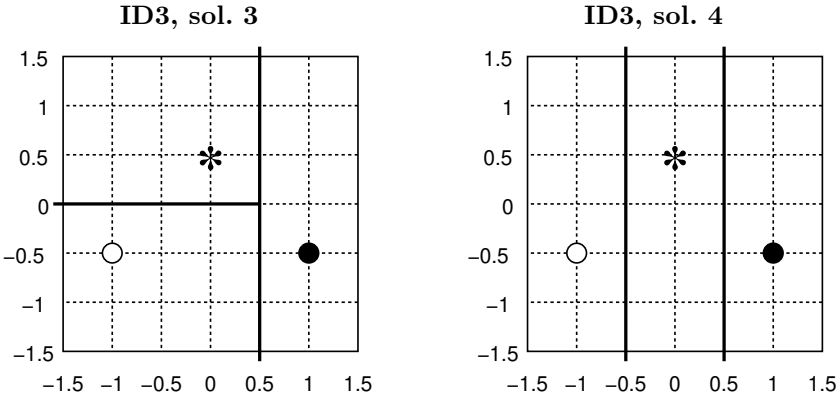
b. Diagrama Voronoi produsă de algoritmul 1-NN este cea din figura următoare:



În cazul algoritmului ID3, split-urile sunt $x = -0.5$, $x = +0.5$, şi $y = 0$, iar compaşii de decizie care se formează au toţi acelaşi tip de structură. (De exemplu: unul dintre ei are partiţia [1-,1+,1*] în rădăcină, iar descendenţii lui au partiţiile [1-,0+,0*] şi respectiv [0-,1+,1*]. Ceilalţi compaşi de decizie sunt uşor de „vizualizat".) Asta face să se obţină mai multe soluţii (a se vedea mai jos). Toate aceste soluţii comportă zone de decizie diferite de cele determinate de algoritmul 1-NN (vedeţi diagrama Voronoi de mai sus).







c. Spre deosebire de cazul $d = 1$, atunci când $d > 1$ se constată că în general algoritmul 1-NN şi algoritmul ID3 pot produce la testare / generalizare rezultate diferite (deci şi zone de decizie diferite). Există însă şi cazuri în care rezultatele celor doi algoritmi sunt identice.

15. **(1-NN cu mapare cu RBF: Adevărat sau Fals?)**

■ ● ○ ***CMU, 2003 fall, T. Mitchell, A. Moore, final exam, pr. 7.f***

Algoritmul 1-NN folosind distanţa euclidiană neponderată este capabil să obţină rezultate mai bune dacă în prealabil intrările sale sunt mapate într-un „spaţiu de trăsături“ folosind o funcţie-nucleu cu baza radială (RBF).[483] Adevărat sau Fals?

Răspuns:

Fals.
Fie $\phi : \mathbb{R}^d \to \mathbb{R}^n$ funcţia de mapare în spaţiul de trăsături, astfel încât să avem $K(x,y) \stackrel{not.}{=} e^{-\frac{||x-y||^2}{2\sigma^2}} = \phi(x) \cdot \phi(y), \forall x, y \in \mathbb{R}^d$. ($\mathbb{R}^d$ reprezintă spaţiul iniţial, $\mathbb{R}^n$ spaţiul de trăsături în care se face maparea, iar $e^{-\frac{||x-y||^2}{2\sigma^2}}$ este funcţia-nucleu cu baza radială.) Avem:

$$||\phi(x) - \phi(y)||^2 = (\phi(x) - \phi(y)) \cdot (\phi(x) - \phi(y))$$
$$= \phi(x) \cdot \phi(x) + \phi(y) \cdot \phi(y) - 2\phi(x) \cdot \phi(y) = e^{-\frac{||x-x||^2}{2\sigma^2}} + e^{-\frac{||y-y||^2}{2\sigma^2}} - 2e^{-\frac{||x-y||^2}{2\sigma^2}}$$
$$= e^0 + e^0 - 2e^{-\frac{||x-y||^2}{2\sigma^2}} = 2 - 2e^{-\frac{||x-y||^2}{2\sigma^2}} = 2 - 2K(x,y).$$

Prin urmare, pentru orice $x, x_i, x_j \in \mathbb{R}^d$ vom avea:

$$||\phi(x) - \phi(x_i)|| \leq ||\phi(x) - \phi(x_j)|| \Leftrightarrow ||\phi(x) - \phi(x_i)||^2 \leq ||\phi(x) - \phi(x_j)||^2 \Leftrightarrow$$
$$2 - 2K(x,x_i) \leq 2 - 2K(x,x_j) \Leftrightarrow K(x,x_i) \geq K(x,x_j) \Leftrightarrow e^{-\frac{||x-x_i||^2}{2\sigma^2}} \geq e^{-\frac{||x-x_j||^2}{2\sigma^2}} \Leftrightarrow$$
$$-\frac{||x-x_i||^2}{2\sigma^2} \geq -\frac{||x-x_j||^2}{2\sigma^2} \Leftrightarrow ||x-x_i||^2 \leq ||x-x_j||^2 \Leftrightarrow ||x-x_i|| \leq ||x-x_j||.$$

[483] Vedeţi problema 74 de la capitolul de *Fundamente*.

Cu alte cuvinte, dacă o instanţă de test x are drept cel mai apropiat vecin punctul x_i în spaţiul iniţial, acest lucru rămâne valabil şi în spaţiul de trăsături. Aşadar, decizia algoritmului 1-NN este identică în ambele spaţii. Aceeaşi concluzie este valabilă şi pentru k-NN.

***Observaţie*: Este însă posibil ca folosind ponderarea sau alte măsuri de distanţă (decât cea euclidiană) kernel-izarea să funcţioneze cu succes pentru k-NN.**

3.2 Învăţare bazată pe memorare — Probleme propuse

16. **(Algoritmul k-NN: acurateţe; comparaţie cu un simplu clasificator aleator)**

prelucrare de Liviu Ciortuz, după CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, HW1, pr. 5.bc

a. Enunţaţi [succint] *regula de decizie* a algoritmului k-NN pentru o instanţă de test x_q.

Care este *bias*-ul inductiv al algoritmului k-NN?

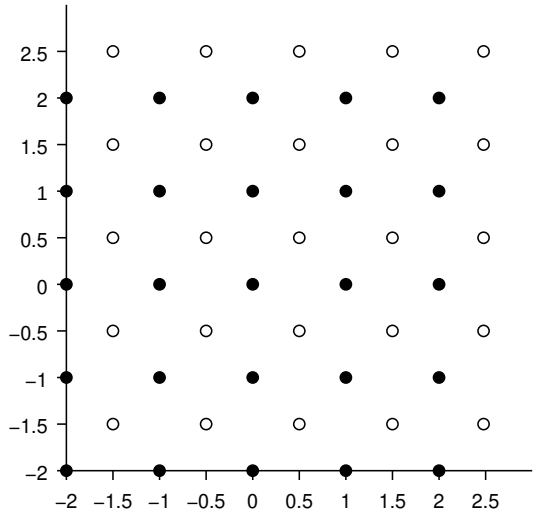
În continuare veţi lucra pe datele din figura de mai jos. Veţi aplica algoritmul k-NN, folosind distanţa euclidiană.

k-NN funcţionează bine atunci când instanţele dintr-o aceeaşi clasă sunt plasate într-una sau mai multe zone din spaţiu relativ bine delimitate, fără întrepătrunderi puternice.

***Obiectivul* nostru acum este să analizăm ce se întâmplă atunci când datele sunt puternic mixate. Rezultatele pe care le veţi obţine la calculul erorilor vor fi exprimate sub formă de numere [fracţionare] din intervalul $[0,\ 1]$.**

***Observaţie importantă*:**

În cazul în care există două sau mai multe instanţe situate exact pe „marginea“ [adică, pe conturul circular al] k-NN-vecinătăţii asociate instanţei de clasificat, se va considera că toate aceste instanţe aparţin respectivei vecinătăţi, iar fiecare dintre ele dispune de un vot întreg.



b. Pentru $k = 1$, calculaţi eroarea la antrenare şi eroarea la cross-validare cu metoda “leave-one-out” (CVLOO).

Ce puteţi spune comparând cele două rezultate? (Care este legătura între *bias*-ul inductiv al lui k-NN şi puterea de generalizare a lui 1-NN pe astfel de date?)

c. Pentru $k = 2$, calculaţi eroarea la cross-validare cu metoda “leave-one-out” (CVLOO).

d. Considerăm $k = 50$. (Remarcaţi faptul că în total în setul nostru de date sunt 50 de instanţe.) De această dată, vom impune ca algoritmul k-NN să ia decizia în mod *probabilist*. Aceasta înseamnă că dacă în vecinătatea k-NN a unei instanţe de test există n vecini pozitivi şi m vecini negativi, atunci algoritmul k-NN va returna (pentru instanţa respectivă) decizia + cu

probabilitatea $n/(n+m)$ şi decizia $-$ cu probabilitatea $m/(n+m)$. În consecinţă, pentru întreg setul de date vom putea calcula o *eroare medie*.

Calculaţi *eroarea medie* la antrenare pentru algoritmul 50-NN pe datele de mai sus. Cunoaşteţi o metodă de clasificare foarte simplă care obţine pe aceste date rezultate la fel de bune / proaste precum 50-NN?

17. **(Algoritmul k-NN: acurateţe; comparaţii pentru diferite valori ale lui k)**

• ○ ***CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, HW1, pr. 5.a***

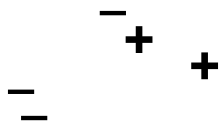
Considerăm două clase, notate cu C_1 şi C_2, în spaţiul euclidian bidimensional. Datele din clasa C_1 sunt distribuite în mod uniform într-un cerc de rază r. Datele din clasa C_2 sunt distribuite în mod uniform într-un alt cerc de rază r. (*Observaţie*: Numărul de date din cele două clase nu este neapărat acelaşi.) Centrele celor două cercuri sunt situate la o distanţă strict mai mare decât $4r$.

Arătaţi că este posibil ca [la antrenare] acurateţea algoritmului 1-NN aplicat pe aceste date să fie *strict* mai mare decât acurateţea algoritmului k-NN, pentru un anumit număr întreg $k \geq 3$, impar, ales în mod convenabil.

18. **(Algoritmul 1-NN: calculul erorii la CVLOO)**

∗ ***CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, midterm exam, pr. 1.7***

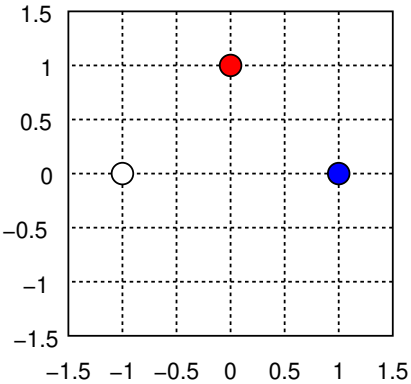
Care este eroarea clasificatorului 1-NN la cross-validare de tip "Leave-One-Out" pe setul de date alăturat?



19. **(Algoritmul 1-NN: graniţe / suprafeţe de decizie)**

• ***CMU, 2013 fall, W. Cohen, E. Xing, final exam, pr. 3.6***

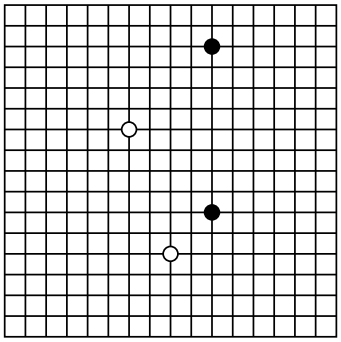
Trasaţi graniţele de decizie (engl., *decision boundaries*) produse de clasificatorul 1-NN la aplicarea pe datele din figura alăturată. Diversele culori ale punctelor reprezintă clase diferite.

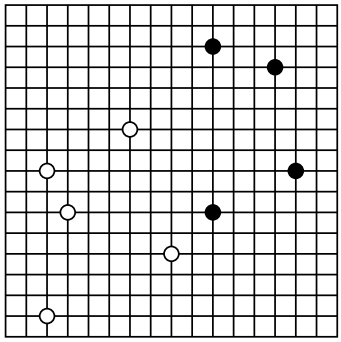


20. **(Algoritmul 1-NN: graniţe / suprafeţe de decizie)**

∗ ***CMU, 2008 fall, Eric Xing, HW1, pr. 3***

În fiecare din figurile următoare se dau câteva puncte / instanţe în spaţiul bidimensional, care sunt etichetate cu • (instanţe pozitive) sau ◦ (instanţe negative). Indicaţi în fiecare caz graniţele / suprafeţele de decizie pentru algoritmul 1-NN presupunând că se foloseşte distanţa euclidiană.

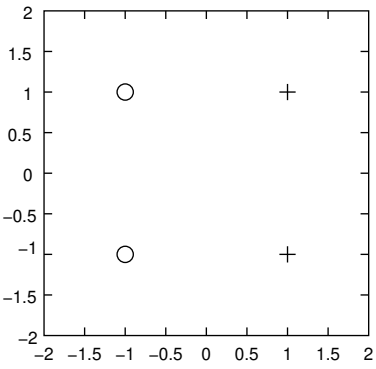


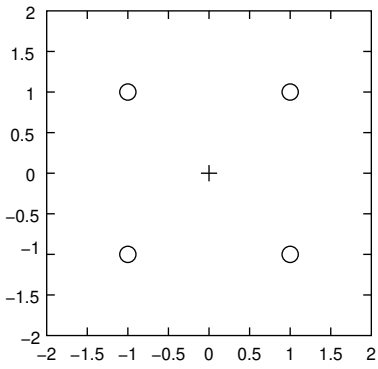


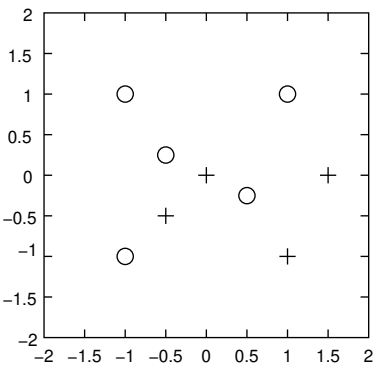
21. **(Algoritmul 1-NN: diagrame Voronoi)**

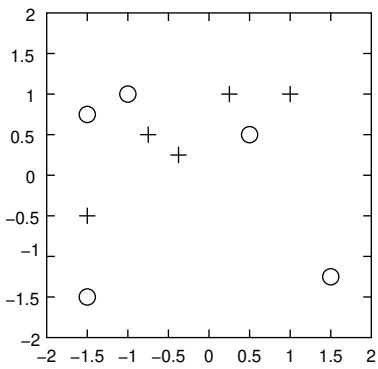
• ∗ ***CMU, 2010 fall, Ziv Bar-Joseph, HW1, pr. 3.1***

Desenaţi suprafeţele de decizie pentru clasificatorul 1-NN pentru fiecare dintre seturile de date din figurile de mai jos. Folosiţi distanţa euclidiană. Haşuraţi fin zonele corespunzătoare clasei +.





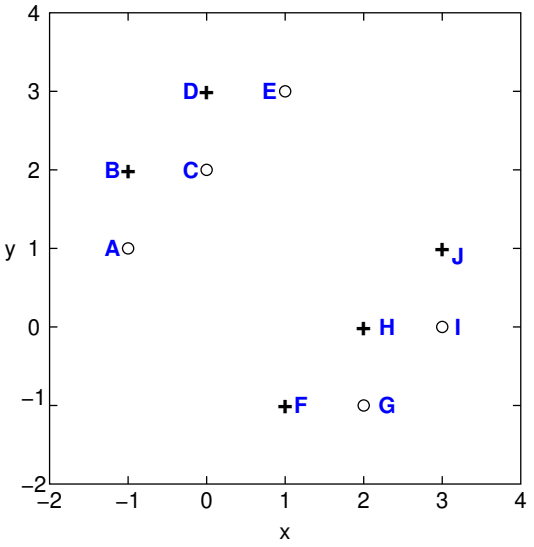




22. **(Algoritmul k-NN: diagrama Voronoi, eroarea la CVLOO; comparaţie pentru diferite valori ale lui k)**

• ***prelucrare de L. Ciortuz, după CMU, 2012 spring, Ziv Bar-Joseph, midterm exam, pr. 2***

La acest exerciţiu veţi aplica algoritmul k-NN folosind distanţa euclidiană pe setul de date din figura de mai jos. Fiecare punct aparţine la una din două clase, desemnate cu $+$ şi respectiv $\circ$.



a. Pentru $k = 1$, trasaţi diagrama Voronoi şi haşuraţi zona / zonele de decizie corespunzătoare etichetei $+$.

b. Care este eroarea la cross-validare cu metoda "Leave-One-Out" (CVLOO) dacă se foloseşte algoritmul 1-NN?

c. Care dintre următoarele valori ale lui k va conduce la o valoare minimă a erorii de tip CVLOO: 3, 5, 7 sau 9? Comentaţi succint rezultatul.

***Indicaţii*:**

1. k-NN-vecinătăţile vor fi construite în manieră *inclusivă*.[484] Vă *cerem(!)* să puneţi în evidenţă toate cazurile de acest tip. Vedeţi, spre exemplu, liniile 2 şi 3 din tabelul de mai jos, coloana 1-NN vecinătăţilor.

2. În caz de *paritate la voturi* (dar doar în acest caz!), se va considera că se aplică (în mod intuitiv) ponderarea distanţelor în sensul prezentat la curs.

3. Dacă veţi şti să exploataţi *simetriile*, veţi avea mult mai puţin de elaborat la nivel de detaliu!

4. Pentru convenienţă, puteţi completa tabelul de mai jos. Din cauza spaţiului restrâns, vă sugerăm ca la rubrica *Vecinătate*, în dreptul fiecărui punct (A, B, ..., J), pentru fiecare din valorile 3, 5 şi 9 ale lui k să menţionaţi doar punctele care constituie *extensia* de la vecinătatea *1-NN* la vecinătatea *3-NN* şi aşa mai departe.

[484] Adică, dacă notăm cu

– $x_1, x_2, \ldots, x_n$ instanţele de antrenament,

– x o instanţă oarecare căreia i se aplică la un moment dat procedura de cross-validare LOO cu algoritmul k-NN (unde k este fixat),

– $d(x, x_{i_1}) \leq d(x, x_{i_2}) \leq \ldots \leq d(x, x_{i_n})$ secvenţa ordonată a distanţelor de la x la fiecare din instanţele de antrenament,

şi există $l > 0$ astfel încât $x_{i_k} = x_{i_{k+1}} = \ldots = x_{i_{k+l}} < x_{i_{k+l+1}}$ sau $k + l = n$, atunci în *k-NN vecinătatea* lui x vor fi incluse toate instanţele $x_{i_{k+1}}, \ldots, x_{i_{k+l}}$.

Data	Et.	Vecinătate					Clasif. la CVLOO					Eroare? (da/nu)				
		1−	3−	5−	7−	9−NN	1−	3−	5−	7−	9−NN	1−	3−	5−	7−	9−NN
A	$\circ$	B														
B	$+$	$A \mid C$														
C	$\circ$	$B \mid D$														
...																

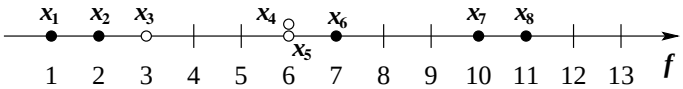
23. **(Diagrame Voronoi pentru dataset-uri din $\mathbb{R}$, folosind 1-NN şi 5-NN; calcularea erorii la CVLOO pentru algoritmul 1-NN; comparaţie cu algoritmul ID3 cu atribute numerice continue)**

prelucrare de L. Ciortuz, după

• ○ ∗ ***MIT, ML 6036 course, Review Material Problems, ex. 37***

***Comentariu*:** În acest exerciţiu, spre deosebire de toate celelalte cazuri când am construit diagrame Voronoi (întotdeauna pentru algoritmul 1-NN şi seturi de date din $\mathbb{R}^2$), vom construi astfel de diagrame pentru seturi de date din $\mathbb{R}$, mai întâi pentru algoritmul 1-NN şi apoi pentru algoritmul 5-NN. (Absolut similar se poate proceda şi pentru alte valori ale lui $k \neq 1$, atunci când datele de antrenament sunt din $\mathbb{R}$.)

În desenul de mai jos este reprezentat un set de antrenament care conţine 8 instanţe, fiecare dintre ele având doar o trăsătură (engl., feature), notată cu f.



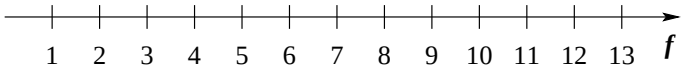
Remarcaţi faptul că sunt două instanţe pentru care valoarea trăsăturii f este aceeaşi, şi anume 6. Aceste două instanţe sunt reprezentate prin două simboluri ○, situate unul deasupra celuilalt, dar de fapt ele ar fi trebuit să fie reprezentate ca două simboluri ○ suprapuse (unul peste celălalt), întrucât aceste instanţe au exact aceeaşi valoare pentru trăsătura f.

Vă readucem aminte *convenţia* noastră de notare: simbolul • desemnează instanţe pozitive (+), iar simbolul ○ instanţe negative (−).

a. La acest punct al problemei veţi folosi algoritmul 1-NN.

***Convenţii / reguli*:**
1. În cazul în care există două sau mai multe instanţe situate exact pe „marginea" [adică, pe conturul circular al] k-NN-vecinătăţii asociate instanţei de clasificat, se va considera că toate aceste instanţe aparţin respectivei vecinătăţi, iar fiecare dintre ele dispune de un vot întreg.
2. În caz de *paritate de voturi*, veţi alege eticheta instanţei de antrenament care este situată la *distanţă minimă* către stânga faţă de instanţa de test respectivă. (Distanţa minimă este 0 atunci când instanţa de test coincide cu o instanţă de antrenament!)
3. Aceste reguli vor fi aplicate şi la punctele următoare.

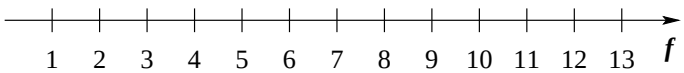
Vă cerem ca pe linia de mai jos să haşuraţi — şi, bineînţeles, să delimitaţi prin separatori decizionali — zonele în care algoritmul 1-NN va prezice semnul +, dat fiind setul de date de antrenament din figura de mai sus.



În prealabil, vă cerem să stabiliţi cu ce etichetă vor fi clasificate instanţele de test $f = 2.5$ şi $f = 6.5$.[485] Justificaţi.

b. Dacă faceţi cross-validare folosind metoda "leave-one-out" pe acest set de date, în conjuncţie cu algoritmul 1-NN, cât va fi eroarea produsă? Veţi arăta în mod clar — alcătuind un *tabel* care să conţină 1-NN-vecinătăţile respective — cum anume aţi ajuns la rezultatul pe care l-aţi indicat.

c. Similar cu punctul a, însă aici veţi folosi algoritmul 5-NN.[486]



***Indicaţie*: Justificaţi în mod riguros rezultatul, referindu-vă la diverse intervale de valori (sau valori particulare) ale lui f:**

Cazul 1: $f \in (-\infty, 4)$**:** ...

Cazul 2: ...

...

În prealabil, vă cerem să stabiliţi cu ce etichetă vor fi clasificate instanţele de test $f = 4$, $f = 6$ şi $f = 7$.[487] Justificaţi.

***Observaţie*: Se poate demonstra faptul că în cazul seturilor de date de antrenament $x_1, x_2, \ldots, x_n \subset \mathbb{R}$ (adică, atunci când se lucrează pe axa reală) k-NN-vecinătăţile nu sunt discontinue, adică: dat fiind un punct de test oarecare $x_q \in \mathbb{R}$, dacă instanţele de antrenament x_i şi x_j, cu $x_i \leq x_j$, se află în k-NN-vecinătatea lui x_q, atunci orice instanţă de antrenament x_l care satisface proprietatea $x_i \leq x_l \leq x_j$ aparţine şi ea respectivei k-NN-vecinătăţi a lui x_q.[488]**

d. Construiţi — în mod riguros — arborele de decizie corespunzător setului de date din enunţ, considerând f ca fiind atribut numeric continuu. Comparaţi rezultatul obţinut de data aceasta cu rezultatele care au fost produse de clasificatorii 1-NN şi 5-NN.

[485] Acestea sunt [toate] cazurile când 1-NN-vecinătatea unui punct de test de pe axa reală conţine instanţe având etichete diferite.

[486] Fără să aplicaţi 5-NN pentru instanţele de test $f = 2.5$ şi $f = 6.5$.

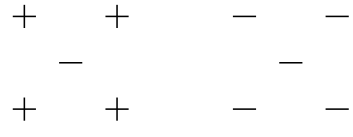
[487] Acestea sunt [toate] cazurile când 5-NN-vecinătatea unui punct de test de pe axa reală conţine mai mult de 5 instanţe de antrenament.

[488] O proprietate similară are loc în cazul clusterizării de instanţe pe axa reală. Vedeţi problema 47.c de la capitolul *Clusterizare*.

24. **(Algoritmul k-NN: CVLOO: comparaţie pentru diferite valori ale lui k; eroarea la antrenare: comparaţie cu alţi clasificatori)**

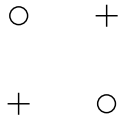
● ○ ***CMU, 2010 fall, Aarti Singh, midterm exam, pr. 2***

a. Care dintre clasificatorii de mai jos realizează o eroare de tip CVLOO (Leave-One-Out Cross-validation) mai mare pe setul de date alăturat?



☐ **1-NN** ☐ **3-NN**

b. Considerăm setul de date din figura alăturată. Care dintre clasificatorii de mai jos obţine / obţin eroare nulă la antrenare pe acest set de date?



☐ **arborii de decizie ID3 de adâncime 2** ☐ **regresia logistică**

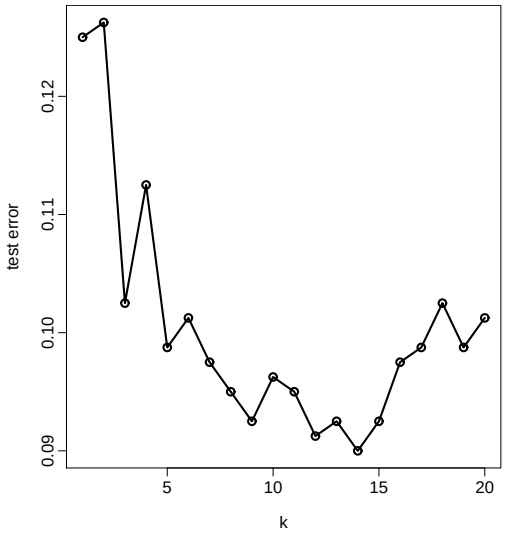
☐ **clasificatorul 3-NN** ☐ **SVM (cu nucleu pătratic)**

25. **(Algoritmul k-NN: alegerea valorii convenabile pentru k)**

prelucrare de Liviu Ciortuz, după CMU, (?) spring, ML course 10-701, HW1, pr. 5

Pe un anumit set de date format din date de antrenament, date de validare şi date de test, după ce a fost antrenat algoritmul k-NN pentru diferite valori ale lui k, rezultatele obţinute la validare au fost reprezentate în graficul care urmează.

Care este — în conformitate cu aceste rezultate — valoarea optimală care trebuie aleasă pentru k, în vederea folosirii ulterioare pe datele de test? Justificaţi alegerea făcută.



26. **(Algoritmul k-NN: întrebări de ordin calitativ)**

☐ ● ○ ***CMU, 2012 spring, Ziv Bar-Joseph, HW1, pr. 3***

Ştim că la aplicarea algoritmului k-NN, clasificarea unei instanţe date se face pe baza votului majoritar obţinut în „vecinătatea“ instanţei respective. Presupunem că se dau două clase de instanţe, fiecare clasă având $n/2$ puncte, întrepătrunse într-o anumită măsură, într-un spaţiu bidimensional.

a. Descrieţi ce se întâmplă cu *eroarea la antrenare* (folosind toate datele disponibile) când numărul k al vecinilor consideraţi variază de la n la 1.

b. Schiţaţi grafic cum anume ar evolua *eroarea la generalizare* (de exemplu, reţinând o parte din date pentru testare) atunci când k variază. Explicaţi modul în care aţi raţionat.

c. Propuneţi o metodă de determinare a unei valori adecvate pentru k.

d. La folosirea algoritmului k-NN, odată ce s-a stabilit valoarea lui k, toţi cei mai apropiaţi k vecini ai punctului de clasificat au ponderi egale (adică, aceeaşi importanţă) la stabilirea etichetei respectivului punct. Sugeraţi o modificare a algoritmului k-NN care elimină această limitare.

e. Daţi două motive pentru care este de preferat să nu folosim algoritmul k-NN atunci când dimensiunea spaţiului datelor de intrare este mare.

27. (Compararea clasificatorilor 1-NN şi Bayes Optimal: o margine superioară mai bună pentru *rata medie a erorii asimptotice* a lui 1-NN)

∗ ***Liviu Ciortuz, 2014, bazat pe un rezultat din***
■ ***"An Elementary Introduction to Statistical Learning Theory", S. Kulkarni, G. Harman, 2011, pag. 68-69***

La problema 10 am demonstrat că *rata medie a erorii* clasificatorului 1-NN este mărginită asimptotic[489] de dublul ratei medii a erorii clasificatorului Bayes Optimal.

Arătaţi că — în aceleaşi condiţii ca la problema 10 — se poate obţine o margine chiar mai bună:

$$E[\lim_{n\to\infty} Error_{1\text{-}NN}] \leq 2E[Error_{Bayes}](1 - E[Error_{Bayes}]).$$

28. (Adevărat sau Fals?)

○ ***CMU, 2010 fall, Ziv Bar-Joseph, midterm, pr. 1.bc***

Care dintre următoarele afirmaţii sunt adevărate pentru clasificatorii k-NN? (Justificaţi pe scurt răspunsul, în dreptul fiecărui punct.)

a. Acurateţea la antrenare creşte pe măsură ce creşte valoarea lui k.

b. Graniţa de decizie este mai netedă (engl., smoother) pe măsură ce valoarea lui k scade.

c. k-NN nu necesită o procedură explicită de antrenare.

d. Graniţa de decizie este liniară.

e. Este posibil ca un clasificator binar 1-NN să clasifice întotdeauna orice instanţă de test ca fiind pozitivă, chiar dacă în setul de date de antrenament există instanţe negative.

[489] Adică, atunci când $n \to \infty$, unde n este numărul de instanţe de antrenament.

29. **(Întrebări calitative despre design-ul unor experimente din Învăţarea Automată: OK ori ...problematic?)**

• ○ ***CMU, 2009, Geoff Gordon, midterm exam, pr. 3***

Fiecare din punctele de mai jos prezintă pe scurt design-ul unui experiment practic de învăţare automată. Analizaţi fiecare din aceste cazuri, indicând apoi dacă respectivul experiment este *ok* ori *problematic* (încercuiţi varianta pe care o alegeţi). Dacă este *problematic*, identificaţi TOATE defectele [de concepţie ale] design-ului respectiv.

a. O echipă de proiectare raportează o *eroare* mică *la antrenare* şi susţine că metoda folosită este bună.

Ok
Problematic

b. O echipă de proiectare susţine că este un mare succes faptul că a obţinut 98% *acurateţe la antrenare* pentru un task de clasificare binară care are următorul specific: unul din cele două cazuri se întâlneşte foarte rar comparativ cu celălalt caz. (O astfel de problemă o constituie, de exemplu, identificarea tranzacţiilor bancare frauduloase.) Datele lor au constat din 50 de exemple pozitive şi 4950 de exemple negative.

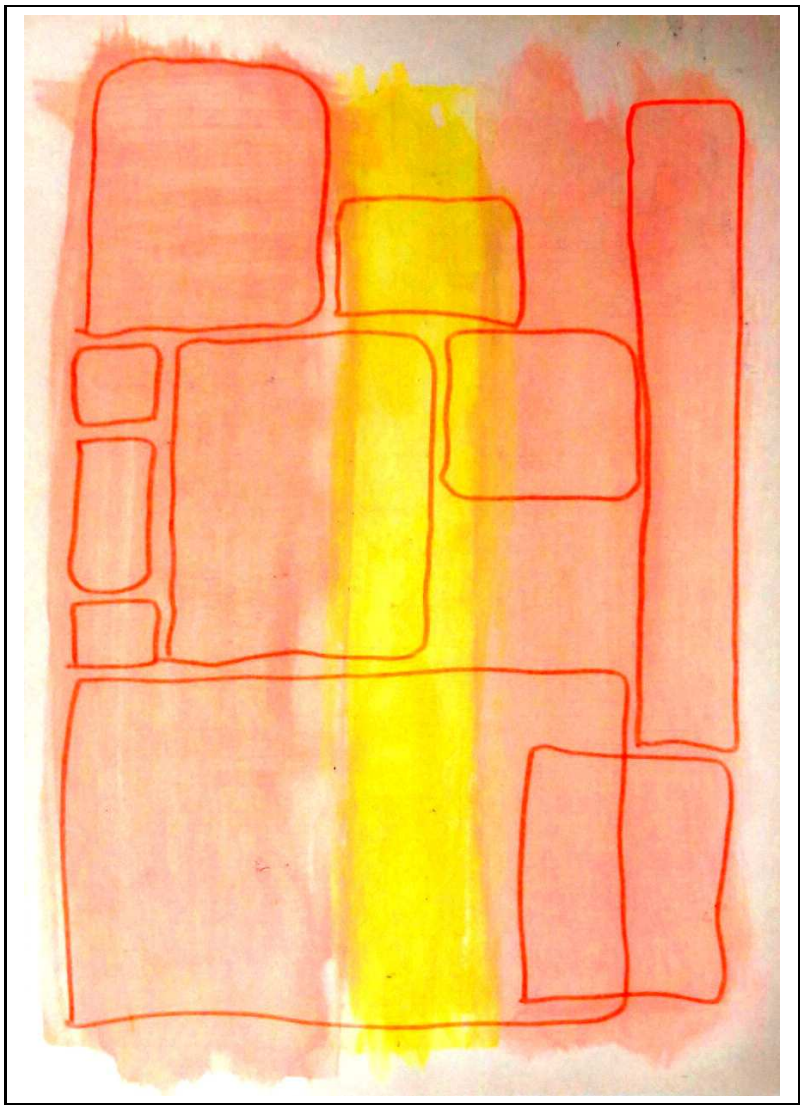
Ok
Problematic

c. O echipă de proiectare şi-a împărţit datele de care dispune în date de antrenament şi date de test. Folosind datele de antrenament, ei au construit un *model* de clasificare caracterizat de anumiţi *parametri*. Apoi, făcând *cross-validare*, au ales cea mai bună setare a parametrilor. La final, au raportat *eroarea* obţinută pe *datele de test*.

Ok
Problematic

d. O echipă de proiectare a efectuat o procedură de *selecţie a atributelor* (engl., features) pe toate datele şi apoi a redus setul mare de atribute la un set mai mic. După aceea, membrii echipei au împărţit datele în date de test şi date de antrenament. Au construit *modelul* de clasificare pe datele de antrenament folosind mai multe setări ale parametrilor modelului, şi au raportat cea mai bună *eroare la testare* pe care au obţinut-o.

Ok
Problematic



© *M. Romanică*

4 Arbori de decizie

Sumar

Noţiuni preliminare

– **partiţie a unei mulţimi: ex. 93 de la cap. *Fundamente*;**

– **proprietăţi elementare ale funcţiei logaritm; formule uzuale pentru calcule cu logaritmi;**

• **Elemente de *teoria informaţiei* (vedeţi secţiunea corespunzătoare din cap. *Fundamente*, în special ex. 55):**

  – **entropie, definiţie: T. Mitchell, *Machine Learning*, 1997 (desemnată în continuare simplu prin *cartea ML*), pag. 57; ex. 2.a, ex. 41.a, ex. 36.a;**
  – **entropie condiţională specifică: ex. 14.a;**
  – **entropie condiţională medie: ex. 2.cd, ex. 36.c;**
  – **câştig de informaţie (definiţie: cartea ML, pag. 58): ex. 2.cd, ex. 5.a, ex. 34, ex. 41.b, ex. 36.e;**

– ***arbori de decizie*, văzuţi *ca structură de date*: ex. 1, ex. 32**
şi, respectiv, ca program în logica propoziţiilor: ex. 2.e, ex. 40.bc, ex. 46.c;
• (P0) *expresivitatea arborilor de decizie* cu privire la *funcţii booleene*: ex. 33;

– ***spaţiu de versiuni* pentru un concept (de învăţat): ex. 1, ex. 3, ex. 32, ex. 39.**

Algoritmul ID3

– **pseudo-cod, versiune simplificată:**

> **create the root *node*;**
> **assign all training examples to the root;**
>
> **Main loop:**
> 1. **$A \leftarrow$ the "best" decision attribute for the current *node*;**
> 2. **for each value of the attribute A, create a new descendant of the *node*;**
> 3. **sort the training examples to the leaf nodes;**
> 4. **if the training examples are perfectly classified, then STOP;**
>   **else iterate over the new leaf nodes;**

pseudo-cod, versiune completă: cartea ML, pag. 56;

– ***bias-ul inductiv*: *ibidem*, pag. 63-64;**

– **exemple simple de aplicare: ex. 2, ex. 3, ex. 4.a, ex. 5, ex. 37, ex. 38, ex. 39.a, ex. 40, ex. 41, ex. 42;**

– **ID3 ca algoritm *per se*:**

  • **este un *algoritm de căutare*;**
  spaţiul de căutare — mulţimea tuturor arborilor de decizie care se pot construi cu atributele de intrare în *nodurile de test* şi cu valorile atributului de ieşire în *nodurile de decizie* — este de dimensiune exponenţială în raport cu numărul de atribute: ex. 1, ex. 3, ex. 32, ex. 39;
  ID3 are ca *obiectiv* căutarea unui arbore / *model* care $i.$ să explice cât mai

bine datele (în particular, atunci când datele sunt *consistente*, modelul trebuie să fie *consistent* cu acestea), *ii.* să fie cât mai *compact*, din motive de *eficienţă* la generalizare / testare şi *iii.* în final să aibă o [cât mai] bună putere de *generalizare*;[490]

- **de tip *divide-et-impera* (⇒ "*Iterative Dichotomizer*"), recursiv;**
  - ***1-step look-ahead*;**
  - **ID3 ar putea fi văzut şi ca algoritm de *optimizare*;**[491]
- ***greedy* ⇒ nu garantează obţinerea soluţiei optime d.p.v. al numărului de niveluri / noduri:**
  ex. 4, ex. 21.a, ex. 40 (vs. ex. 39.b, ex. 3.b), ex. 50;
- **complexitate de timp, cf. *Weka* book:**[492]
  la antrenare, în anumite condiţii: $\mathcal{O}(d\,m \log m)$; la testare $\mathcal{O}(d)$,
  unde d este numărul de atribute, iar m este numărul de exemple;

– **ID3 ca algoritm de învăţare automată:**

- ***bias*-ul inductiv al algoritmului ID3:**
  [dorim ca modelul să aibă structură ierarhică, să fie compatibil / consistent cu datele de antrenament dacă acestea sunt consistente (adică, necontradictorii), iar] *arborele* produs de *ID3* trebuie să aibă un număr cât mai mic de niveluri / noduri;
- **algoritm de învăţare de tip *"eager"*;**
- ***analiza erorilor*:**
  la antrenare: ex. 7.a, ex. 10.a, ex. 45;[493] **[acurateţe la antrenare: ex. 6;]**
  la validare: ex. 46.d;
  la n-fold cross-validare
  la cross-validare leave-one-out (CVLOO): ex. 10.b, ex. 49.bc;[494]
- ***robusteţe la „zgomote" şi overfitting*: ex. 10, ex. 21.bc, ex. 49, ex. 57.c;**[495]
- ***zone de decizie* şi *graniţe de separare* / *decizie* pentru arbori de decizie cu variabile continue: ex. 10, ex. 48, ex. 49, ex. 50.**

Extensii / variante ale algoritmului ID3

– ***atribute cu valori continue*: ex. 10-12, ex. 14.c, ex. 47-51; cap. *Învăţare bazată pe memorare*, ex. 11.b;**

– **atribute discrete cu multe valori: ex. 13, ex. 52;**

– **atribute cu valori nespecificate / absente pentru unele instanţe;**

– **atribute cu diferite costuri asociate: ex. 14;**

– **reducerea caracterului "eager" al învăţării: ex. 16;**

[490] LC: Alternativ, putem spune că algoritmul ID3 produce o *structură* de tip *ierarhie* (arbore) între diferite *partiţionări* ale setului de instanţe de antrenament, această ierarhie fiind generată pe baza *corespondenţei* dintre atributul de *ieşire* şi atributele de *intrare*, care sunt adăugate la model câte unul pe rând.

[491] LC: Am putea să-l interpretăm pe ID3 ca fiind un algoritm care caută între diferitele *distribuţii probabiliste* discrete care pot fi definite pe setul de date de antrenament una care să satisfacă cerinţa de *ierarhizare*, şi pentru care entropia să fie *minimală* (vedeţi proprietatea de structuralitate de la ex. 62 de la cap. *Fundamente*). Cerinţa ca arborele ID3 să fie *minimal* (ca număr de niveluri / noduri) este însă mai importantă, mai practică şi mai uşor de înţeles.

[492] "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations", Ian Witten, Eibe Frank (3rd ed.), Morgan Kaufmann Publishers, 2011.

[493] De asemenea, ex. 4.ab de la capitolul *Clasificare bayesiană*.

[494] De asemenea, ex. 4.ab de la capitolul *Clasificare bayesiană*.

[495] De asemenea, ex. 4.ab de la capitolul *Clasificare bayesiană*.

– **reducerea caracterului "greedy" al învăţării:**
IG cu "2-step look-ahead": ex. 17, ex. 18;

– **folosirea altor *măsuri de „impuritate"* în locul câştigului de informaţie:**
Gini Impurity, Misclassification Impurity: ex. 15;

– **reducerea *overfitting*-ului:**
reduced-error pruning (folosind un set de date de validare): cartea ML, pag. 69-71; A. Cornuéjols, L. Miclet, *Apprentissage artificiel*, 2010, pag. 418-421;
rule post-pruning: cartea ML, pag. 71-72; ex. 54
top-down vs. bottom-up pruning, folosind un criteriu bazat pe câştigul de informaţie: ex. 19, ex. 53;
pruning folosind testul statistic χ^2: ex. 20, ex. 55.

Proprietăţi ale arborilor ID3

- **(P1) arborele produs de algoritmul ID3 este *consistent* (adică, în concordanţă) cu datele de antrenament, dacă acestea sunt *consistente* (adică, necontradictorii). Altfel spus, *eroarea la antrenare* produsă de algoritmul ID3 pe orice set de *date consistente* este 0: ex. 2-4, ex. 37, ex. 39;**
- **(P2) arborele produs de algoritmul ID3 *nu* este în mod neapărat *unic*: ex. 3, ex. 39;**
- **(P3) arborele ID3 nu este neapărat *optimal* (ca nr. de noduri / niveluri): ex. 4, ex. 21.a, ex. 40, ex. 50;**
- **(P4) algoritmul ID3 este sensibil la prezenţa *atributelor superflue*: ex. 4;**
- **(P5) influenţa *atributelor identice/duplicate* şi, respectiv, a *instanţelor multiple* asupra arborelui ID3: ex. 8;**[496]
- **(P6) algoritmul ID3 este sensibil la prezenţa *atributelor cu foarte multe valori*: ex. 13, ex. 52;**
- **(P7) *o margine superioară* pentru *eroarea la antrenare* a algoritmului ID3, în funcţie de numărul de valori ale variabilei de ieşire: ex. 7.b;**
- **(P8) *graniţele de separare / decizie* pentru arborii ID3 cu *atribute de intrare continue* sunt întotdeauna paralele cu axele de coordonate: ex. 10, ex. 48, ex. 49, ex. 50, precum şi cap. *Învăţare bazată pe memorare*, ex. 11.b;**
- **(P9) *zonele de decizie* produse de algoritmul ID3 nu sunt în mod neapărat unice, fiindcă arborele de decizie creat de ID3 nu este determinat în mod unic (vedeţi proprietatea (P2));**
- **(P10) algoritmul ID3 este sensibil la prezenţa „zgomotelor" în date (relativ la etichetare), putând produce overfitting: ex. 10, ex. 21.bc, ex. 49, ex. 57.c.**

  ***Observaţie*: Următoarele trei proprietăţi se referă la arbori de decizie în general, nu doar la arbori ID3.**

- **(P11) *adâncimea maximă* a unui arbore de decizie, când atributele de intrare sunt categoriale: numărul de atribute: ex. 56.c;**

[496] Algoritmul ID3 nu este sensibil (în esenţă) la *duplicarea atributelor de intrare*, însă este sensibil la *duplicarea instanţelor de antrenament.*

- **(P12) o *margine superioară* pentru *adâncimea* unui arbore de decizie când atributele de intrare sunt continue, iar datele de antrenament sunt (ne)separabile liniar: ex. 11;**
- **(P13) o *margine superioară* pentru numărul de *noduri-frunză* dintr-un arbore de decizie, în funcţie de numărul de exemple şi de numărul de atribute de intrare, atunci când acestea (atributele de intrare) sunt binare: ex. 9.**

- **Alte metode de învăţare automată bazate pe arbori: arbori de regresie (CART).**

Învăţare automată de tip ansamblist folosind arbori de decizie: Algoritmul AdaBoost

- **Noţiuni preliminare:**
  distribuţie de probabilitate discretă, factor de normalizare pentru o distribuţie de probabilitate, ipoteze „slabe“ (engl., weak hypothesis), compas de decizie (engl., decision stump), prag de separare (engl., threshold split) pentru un compas de decizie, prag exterior de separare (engl., outside threshold split), eroare ponderată la antrenare (engl., weighted training error), vot majoritar ponderat (engl., weighted majority vote), overfitting, ansambluri de clasificatori (vedeţi ex. 71), funcţii de cost / pierdere (engl., loss function) (vedeţi ex. 23.b şi ex. 29);

– **pseudo-codul algoritmului AdaBoost: ex. 22;**[497]
  proprietăţi de bază: ex. 22;
  alte proprietăţi, precum şi două margini superioare pentru eroarea la antrenare: ex. 23.a-d;
  convergenţa algoritmului, precum şi o condiţie suficientă pentru învăţabilitate empirică γ-slabă: ex. 23.e, ex. 68;

– **exemple de aplicare: ex. 24, 60, 61, 62, 63, 64, 65 şi 66.**

- **AdaBoost ca algoritm *per se*:**
  algoritm *iterativ*,
  algoritm *de căutare* (spaţiul de căutare este mulţimea combinaţiilor liniare care se pot construi peste clasa de ipoteze „slabe“ considerate),
  algoritm *greedy* (dacă la fiecare iteraţie se alege cea mai bună ipoteză „slabă“),
  algoritm de *optimizare secvenţială*; minimizează o margine superioară pentru eroarea la antrenare;
  varianta de pseudo-cod de la pr. 26 (şi mai ales varianta de AdaBoost generalizat de la pr. 29) evidenţiază faptul că AdaBoost face *optimizare pe coordonate* (engl., coordinate descent), în raport cu componentele h_t şi respectiv α_t;

– ***învăţabilitate empirică γ-slabă*:**
  definiţie: ex. 23.e
  exemplificarea unor cazuri când nu există *garanţie* pentru învăţabilitate γ-slabă: ex. 25, 67;

– **AdaBoost ca algoritm de *optimizare secvenţială* în raport cu funcţia de cost / „pierdere“ negativ-exponenţială: ex. 26;**

[497] Vedeţi pseudo-codul algoritmului AdaBoost.M1 din cartea „Probabilistic machine learning: An introduction“ de Kevin Murphy, pag. 610, MIT Press, 2022. (Cf. https://probml.github.io/pml-book/book1.html, accesat la 24.02.2022.)

– **marginea de votare: ex. 27, 28 şi 69;**
proprietăţi ale marginilor de votare:

- $Margin_k(x_i) \in [-1, +1]$
- $Margin_k(x_i) = y_i \, \bar{f}_k(x_i)$ **unde** $\bar{f}_k(x_i) \stackrel{not.}{=} \sum_{t=1}^{k} \bar{\alpha}_t \, h_t(x_i)$
- x_i **este corect clasificat la iteraţia** $k \Leftrightarrow Margin_k(x_i) \geq 0$
- $Margin_k(x_i) > Margin_k(x_j) \Leftrightarrow D_{k+1}(i) < D_{k+1}(j)$**;**

– ***selectarea trăsăturilor*** **folosind AdaBoost; aplicare la clasificarea de documente: ex. 70;**

– **AdaBoost şi overfitting: ex. 73;**

– **o margine superioară pentru eroarea la generalizare produsă de algoritmul AdaBoost: ex. 30.**

– **AdaBoost poate folosi "confidence-rated classifiers" ca ipoteze "slabe" (în locul compaşilor de decizie): ex. 74;**

– **o variantă generalizată a algoritmului AdaBoost [în raport cu funcţia de cost]:**[498] **ex. 29 şi ex. 76;**

– **AdaBoost multi-class: ex. 77;**

– **recapitulare (întrebări cu răspuns *adevărat* / *fals*): ex. 31 şi 78.**

- **Proprietăţi ale algoritmului AdaBoost:**

(P0) AdaBoost poate produce rezultate diferite atunci când are posibilitatea să aleagă între două sau mai multe [cele mai bune] ipoteze „slabe": ex. 24, 60;

(P1) $\varepsilon_i > \varepsilon_j \Leftrightarrow \alpha_i < \alpha_j$ **(consecinţă imediată din relaţia** $\alpha_t = \ln \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}$**): ex. 22.***i***;**

(P2) Din relaţia de definiţie pentru distribuţia D_{t+1} **rezultă** $Z_t = e^{-\alpha_t} \cdot (1-\varepsilon_t) + e^{\alpha_t} \cdot \varepsilon_t = 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$ **(ex. 22.***ii* **şi 22.***iii***) şi** $\varepsilon_t \in (0, 1/2) \Rightarrow Z_t \in (0, 1)$ **(ex. 22.***iv***).**

(P3) O consecinţă din relaţia (245) **şi (P2):** $D_{t+1}(i) = \begin{cases} \frac{1}{2\varepsilon_t} D_t(i), & i \in M \\ \frac{1}{2(1-\varepsilon_t)} D_t(i), & i \in C \end{cases}$

(ex. 22.*v***);**

(P4) $err_{D_{t+1}}(h_t) = \frac{1}{2}$ **(ex. 22.***vi***);**
ca o *consecinţă*, rezultă că ipoteza h_t **nu poate fi reselectată şi la iteraţia** $t+1$**; ea poate fi reselectată la o iteraţie ulterioară;**

(P5) $D_{t+1}(i) = \frac{1}{m \prod_{t'=1}^{t} Z_{t'}} e^{-y_i f_t(x_i)}$**, unde** $f_t(x_i) \stackrel{def.}{=} \sum_{t'=1}^{t} \alpha_{t'} h_{t'}(x_i)$ **(ex. 23.a).**
Produsul $y_i f_t(x_i)$ **se numeşte *margine algebrică*;**

(P6) $err_S(H_t) \leq \prod_{t'=1}^{t} Z_t$**, adică eroarea la antrenare comisă de ipoteza combinată produsă de AdaBoost este majorată de produsul factorilor de normalizare: ex. 23.b;**

(P7) AdaBoost nu optimizează în mod direct $err_S(H_t)$**, ci marginea sa superioară,** $\prod_{t'=1}^{t} Z_t$**; optimizarea se face în mod secvenţial (greedy): la iteraţia** t **se**

[498] Pentru AdaBoost văzut ca instanţă a unui algoritm general de învăţare ansamblistă bazat pe minimizarea secvenţială a unei funcţii de cost / pierdere, vedeţi ex. 26.d şi ex. 75.

minimizează valoarea lui Z_t ca funcţie de α_t, ceea ce conduce la $\alpha_t = \ln \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}$ (ex. 23.c);

(P8) $err_S(H_t)$ **nu neapărat descreşte de la o iteraţie la alta; în schimb, descresc marginile sale superioare:** $\prod_{t'=1}^{t} Z_{t'}$ **şi** $\exp(-\sum_{t'=1}^{t} \gamma_{t'}^2)$**, unde** $\gamma_t = 1/2 - \varepsilon_t$ **(ex. 23.be);**

(P9) O proprietate de monotonie a *costurilor minime* negativ-exponenţiale determinate de către AdaBoost la iteraţii succesive: $J_t^* \leq J_{t-1}^*, \forall t \geq 1$**, unde** $J_t \stackrel{not.}{=} \frac{1}{m} \sum_{i=1}^{m} \exp(-y_i \sum_{t'=1}^{t} \alpha_{t'} h_{t'}(x_i))$**, iar** $J_t^* \stackrel{not.}{=} \min_{h \in \mathcal{H}, \alpha' \in \mathbb{R}_+} J_t(h, \alpha'), \forall t \in \mathbb{N}^*$**: ex 26.e;**

(P10) O condiţie suficientă pentru învăţabilitate γ-slabă, bazată pe *marginea de votare*: la ficare iteraţie a algoritmului AdaBoost, media marginilor de votare ale instanţelor de antrenament în raport cu distribuţia D_t să fie de cel puţin 2γ: ex. 28, ex. 69.d;

(P11) Orice mulţime formată din m instanţe din $\mathbb{R}$ care sunt etichetate în mod consistent poate fi corect clasificată de către o combinaţie liniară formată din cel mult m compaşi de decizie: ex. 71.ac;

(P11′) Orice mulţime de instanţe distincte [şi etichetate] din $\mathbb{R}$ este γ-slab învăţabilă cu ajutorul compaşilor de decizie: ex. 72.d.

- **Alte metode de învăţare ansamblistă bazate pe arbori de decizie: Bagging, Random Forests.**

4.1 Arbori de decizie — Probleme rezolvate

4.1.1 Algoritmul ID3

1. **(Arbori de decizie; optimalitate, relativ la numărul de noduri)**

Reprezentaţi arborele / arborii de decizie care are / au numărul minim de noduri posibile şi corespunde / corespund funcţiei booleene $(\neg A \vee B) \wedge \neg(C \wedge A)$ definită peste atributele booleene A, B şi C.

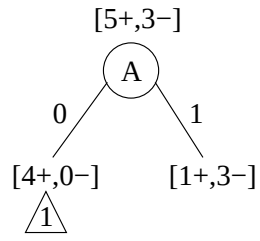
Răspuns:

A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Vom determina arborele de decizie optimal (ca număr de noduri) parcurgând în mod *exhaustiv spaţiul de versiuni*, adică mulţimea tuturor arborilor de decizie (construiţi cu variabilele A, B şi C) care sunt *consistenţi* cu funcţia dată. Aşadar, vom examina ce se întâmplă când în nodul rădăcină se pun pe rând atributele A, B şi respectiv C.

Notăm cu X funcţia $(\neg A \vee B) \wedge \neg(C \wedge A)$, ale cărei valori sunt date în tabelul alăturat.

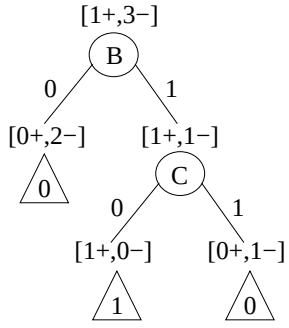
• ***Cazul 1*: Dacă în nodul rădăcină se plasează atributul A, mulţimea de exemple va fi re-partiţionată conform reprezentării alăturate.**

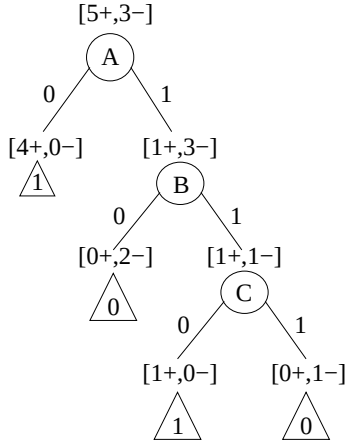


Subarborele drept al acestui arbore va trebui să reprezinte arborele de decizie pentru funcţia

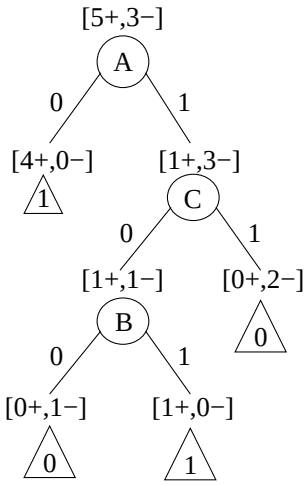
$$X_1 = X[A/1] = (\neg 1 \vee B) \wedge \neg(C \wedge 1) = B \wedge \neg C,$$

pentru care o reprezentare optimă este redată mai jos, în partea stângă:



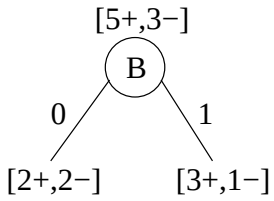


Prin urmare, un arbore optim (ca număr de noduri) care are variabila A în nodul rădăcină este cel reprezentat mai sus, în partea dreaptă.



***Observaţie*: Evident, există încă un arbore optim care are variabila A în nodul rădăcină (el corespunde unei alte reprezentări optimale a conjuncţiei $B \wedge \neg C$ faţă de cea de mai sus). Vedeţi desenul alăturat.**

- ***Cazul 2*: Dacă în nodul rădăcină se alege atributul B, se obţine partiţia**



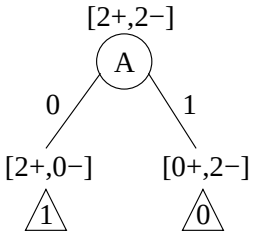
Subarborele stâng şi subarborele drept trebuie să reprezinte arborele de decizie pentru funcţiile

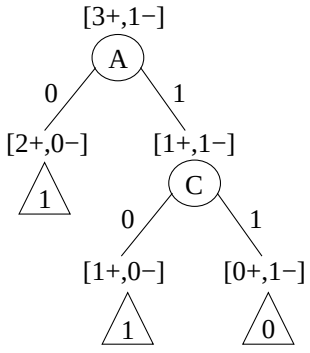
$$X_2 = X[B/0] = (\neg A \vee 0) \wedge \neg(C \wedge A) = \neg A \wedge (\neg C \vee \neg A) = (\neg A \wedge \neg C) \vee \neg A = \neg A,$$

şi respectiv

$$X_3 = X[B/1] = (\neg A \vee 1) \wedge \neg(C \wedge A) = 1 \wedge (\neg C \vee \neg A) = \neg C \vee \neg A,$$

care au ca reprezentări optime arborii de mai jos:

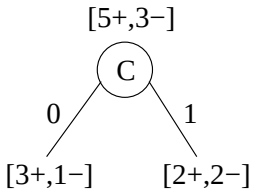




Pentru arborele din dreapta există un arbore de decizie echivalent, obţinut prin interschimbarea lui A cu C.

Prin urmare, orice arbore optim având variabila B în rădăcină are 3 niveluri şi 4 noduri, aşadar cu un nod (de test) mai mult decât cel determinat în primul caz.

• ***Cazul 3*: În sfârşit, când în nodul rădăcină se alege atributul C, obţinem partiţia**



Subarborele stâng şi subarborele drept trebuie să reprezinte arborele de decizie pentru funcţiile

$$X_4 = X[C/0] = (\neg A \vee B) \wedge \neg(0 \wedge A) = (\neg A \vee B) \wedge \neg 0 = \neg A \vee B$$

şi respectiv

$$X_5 = X[C/1] = (\neg A \vee B) \wedge \neg(1 \wedge A) = (\neg A \vee B) \wedge \neg A = \neg A.$$

Urmând un raţionament similar cu cel de la cazul anterior, putem spune că orice arbore optim cu atributul C în rădăcină are 3 niveluri şi 4 noduri, cu un nod (de test) mai mult decât cel determinat în primul caz.

Aşadar, putem concluziona că arborii de decizie optimi corespunzători funcţiei date sunt cei determinaţi în primul caz.

2. **(Algoritmul ID3: aplicare)**

■ • ***CMU, 2002 spring, A. Moore, midterm example questions, pr. 2***

Ai naufragiat pe o insulă pustie, unde nu găseşti niciun alt fel de hrană decât ciuperci. Despre unele dintre aceste ciuperci se ştie că sunt otrăvitoare, despre altele se ştie că sunt comestibile, iar despre restul nu se ştie ce fel sunt. Ai rămas singur pe insulă — foştii tăi camarazi, fiind epuizaţi de foame, au folosit metoda 'trial and error'... — şi ai la dispoziţie următoarele date:

Exemplu	*Uşoară*	*Mirositoare*	*ArePete*	*Netedă*	*Comestibilă*
A	1	0	0	0	1
B	1	0	1	0	1
C	0	1	0	1	1
D	0	0	0	1	0
E	1	1	1	0	0
F	1	0	1	1	0
G	1	0	0	1	0
H	0	1	0	0	0
U	0	1	1	1	?
V	1	1	0	1	?
W	1	1	0	0	?

Atunci când nu vei mai avea la dispoziţie pentru a supravieţui decât ciuperci U, V, sau W, ai putea estima care dintre ele sunt comestibile, folosind arbori de decizie.

În primele trei întrebări care urmează, ne vom referi la ciupercile $A - H$:

a. Care este entropia atributului *Comestibilă*?

b. Doar privind datele — adică fără a face explicit calculul câştigului de informaţie (engl., information gain) pentru cele patru atribute — poţi determina ce atribut vei alege ca rădăcină a arborelui de decizie?

c. Calculează câştigul de informaţie pentru atributul pe care l-ai ales la întrebarea precedentă.

d. Elaborează întregul arbore de decizie ID3 bazat pe datele din tabel şi apoi clasifică ciupercile U, V, W.

e. Exprimă cu ajutorul calculului propoziţional (logica predicatelor de ordinul 0) clasificarea produsă de arborele de decizie obţinut. (*Comestibilă* $\leftrightarrow \ldots$)

f. Există vreun risc dacă vei consuma ciuperci care au fost clasificate de arborele de decizie ca fiind comestibile? De ce da? sau, de ce nu?

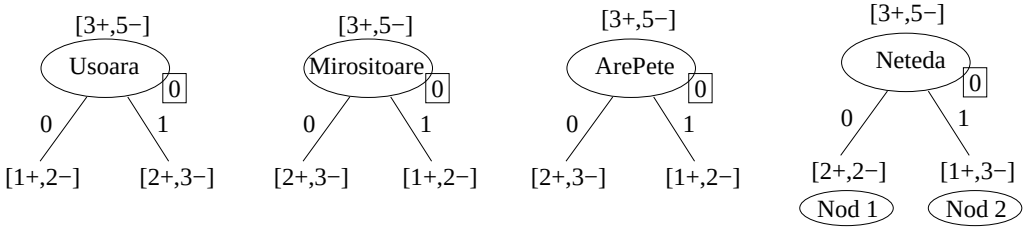
Răspuns:

a. Entropia atributului *Comestibilă* este:

$$\begin{aligned} H_{Comestibilă} &\overset{not.}{=} H[3+,5-] \overset{def.}{=} -\frac{3}{8}\log_2\frac{3}{8} - \frac{5}{8}\log_2\frac{5}{8} = \frac{3}{8}\log_2\frac{8}{3} + \frac{5}{8}\log_2\frac{8}{5} = \\ &= \frac{3}{8}3 - \frac{3}{8}\log_2 3 + \frac{5}{8}3 - \frac{5}{8}\log_2 5 = 3 - \frac{3}{8}\log_2 3 - \frac{5}{8}\log_2 5 \approx \\ &\approx 0.9544 \end{aligned}$$

***Observaţie* (1): Notaţia $[3+,5-]$ simbolizează o mulţime partiţionată în 3 exemple pozitive şi 5 exemple negative. Vom folosi acest gen de notaţie peste tot în continuare, cu mici variaţii determinate de valorile pe care le poate lua atributul de ieşire. De exemplu, dacă vorbim despre o mulţime cu 5 obiecte roşii, 3 albastre şi 4 verzi, am putea nota: $[5R, 3A, 4V]$.**

b. În rădăcina arborelui de decizie se alege atributul care aduce cel mai mare câştig de informaţie. Adică, atributul care, intuitiv vorbind, partiţionează cel mai bine datele de antrenament în raport cu atributul de ieşire. În cazul nostru, variantele pe care le avem la dispoziţie pentru rădăcina arborelui (nodul 0) sunt:



Este uşor de observat că atributele *Uşoară*, *Mirositoare* şi *ArePete* împart mulţimea exemplelor în mod similar: o submulţime cu 3 elemente, dintre care unul este pozitiv, iar două sunt negative, şi o submulţime cu 5 elemente, dintre care două sunt pozitive, iar trei sunt negative.

Dacă am considera un arbore de decizie cu un singur nod de test în care plasăm atributul *Netedă*, atunci numărul minim de erori la antrenare pe care îl putem obţine este 3, utilizând următoarea clasificare:

- *Netedă* $= \mathbf{0}$ **:** *Comestibilă* $= \mathbf{1} \Rightarrow$ **ciupercile** E **şi** H **sunt clasificate greşit**
- *Netedă* $= \mathbf{1}$ **:** *Comestibilă* $= \mathbf{0} \Rightarrow$ **ciuperca** C **este clasificată greşit**

Dacă, în schimb, vom pune în rădăcina arborelui de decizie unul dintre celelalte trei atribute, spre exemplu atributul *Uşoară*, şi dacă vom lua votul majoritar în fiecare nod descendent din nodul rădăcină, eroarea rezultată la antrenare va fi aceeaşi ca mai sus (3/8), însă toate instanţele vor fi clasificate la fel (şi anume, negativ). Dacă nu lucrăm cu vot majoritar pentru ambii descendenţi, ci doar pentru cel cu entropie mai mică (în vreme ce pentru celălalt nod descendent luăm decizia contrară), se observă că pentru atributul *Uşoară* vom obţine 4 erori pe setul de antrenament, iar pentru atributul *Netedă* vom obţine 3 erori.

Sumarizând, suntem înclinaţi să credem că ar fi o alegere sensibil mai bună să punem în rădăcină atributul *Netedă*. Pentru o justificare numerică riguroasă a acestei alegeri folosind criteriul maximizării câştigului de informaţie, vedeţi punctul d.

c. Pentru a obţine câştigul de informaţie pentru atributul *Netedă*, se fac calculele:

$$\begin{aligned} H_{0/Netedă} &\overset{def.}{=} \frac{4}{8}H[2+,2-] + \frac{4}{8}H[1+,3-] = \frac{1}{2}\cdot 1 + \frac{1}{2}\left(\frac{1}{4}\log_2\frac{4}{1} + \frac{3}{4}\log_2\frac{4}{3}\right) \\ &= \frac{1}{2} + \frac{1}{2}\left(\frac{1}{4}\cdot 2 + \frac{3}{4}\cdot 2 - \frac{3}{4}\log_2 3\right) = \frac{1}{2} + \frac{1}{2}\left(2 - \frac{3}{4}\log_2 3\right) \\ &= \frac{1}{2} + 1 - \frac{3}{8}\log_2 3 = \frac{3}{2} - \frac{3}{8}\log_2 3 \approx 0.9056 \end{aligned}$$

$$IG_{0/Netedă} \overset{def.}{=} H_{Comestibilă} - H_{0/Netedă} = 0.9544 - 0.9056 = 0.0488$$

***Observaţie* (2): În cele de mai sus am notat cu $H_{0/Netedă}$ entropia *partiţiei* [mulţimii de exemple de antrenament] determinate de alegerea atributului *Netedă* în nodul 0,[499] iar cu $IG_{0/Netedă}$ câştigul de informaţie corespunzător acestei alegeri. În general, prin notaţia $H_{n/A}$ vom înţelege entropia partiţiei determinate de alegerea atributului A în nodul n.**

d. Arborele de decizie ID3 se construieşte pornind din rădăcină şi alegând atributul pentru fiecare nod de test în modul următor:

***Nodul 0* (rădăcina):**
Să verificăm dacă alegerea făcută la punctul b este cea corectă:

$$\begin{aligned} H_{0/Uşoară} &\overset{def.}{=} \frac{3}{8}H[1+,2-] + \frac{5}{8}H[2+,3-] \\ &= \frac{3}{8}\left(\frac{1}{3}\log_2\frac{3}{1} + \frac{2}{3}\log_2\frac{3}{2}\right) + \frac{5}{8}\left(\frac{2}{5}\log_2\frac{5}{2} + \frac{3}{5}\log_2\frac{5}{3}\right) \\ &= \frac{3}{8}\left(\frac{1}{3}\log_2 3 + \frac{2}{3}\log_2 3 - \frac{2}{3}\cdot 1\right) + \frac{5}{8}\left(\frac{2}{5}\log_2 5 - \frac{2}{5}\cdot 1 + \frac{3}{5}\log_2 5 - \frac{3}{5}\log_2 3\right) \\ &= \frac{3}{8}\left(\log_2 3 - \frac{2}{3}\right) + \frac{5}{8}\left(\log_2 5 - \frac{3}{5}\log_2 3 - \frac{2}{5}\right) \\ &= \frac{3}{8}\log_2 3 - \frac{2}{8} + \frac{5}{8}\log_2 5 - \frac{3}{8}\log_2 3 - \frac{2}{8} = \frac{5}{8}\log_2 5 - \frac{4}{8} \approx 0.9512 \end{aligned}$$

[499] Mai riguros, folosind terminologia din *Teoria informaţiei*, vom spune că notaţia $H_{0/Netedă}$ se referă la *entropia condiţională medie* a atributului de ieşire *Comestibilă* în raport cu atributul de intrare *Netedă*.

Urmează că

$$IG_{0/\,Uşoară} \stackrel{def.}{=} H_{Comestibilă} - H_{0/\,Uşoară} = 0.9544 - 0.9512 = 0.0032,$$

deci

$$IG_{0/\,Uşoară} = IG_{0/\,Mirositoare} = IG_{0/\,ArePete} = 0.0032 < IG_{0/\,Netedă} = 0.0488$$

Am avut deci dreptate să alegem atributul *Netedă* **la punctul** b**.**

***Observaţie importantă*:**
În loc să fi calculat efectiv aceste câştiguri de informaţie, pentru a determina atributul cel mai „bun", ar fi fost *suficient* să elaborăm un *raţionament* de tip *relaţional*, bazat pe comparaţia dintre valorile entropiilor condiţionale medii $H_{0/\,Netedă}$ **şi** $H_{0/\,Uşoară}$**:**

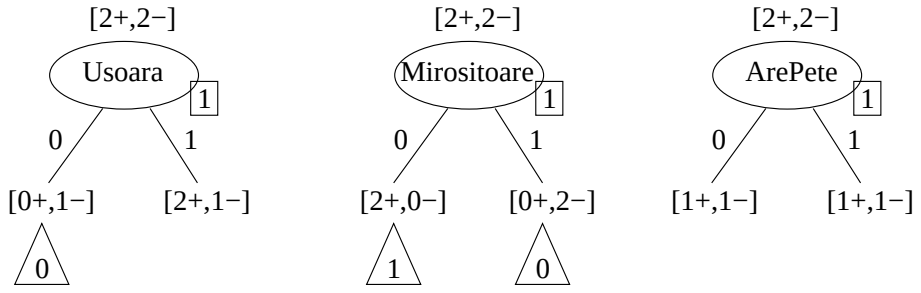
$$\begin{aligned} & IG_{0/\,Netedă} > IG_{0/\,Uşoară} \Leftrightarrow H_{0/\,Netedă} < H_{0/\,Uşoară} \\ \Leftrightarrow\ & \frac{3}{2} - \frac{3}{8}\log_2 3 < \frac{5}{8}\log_2 5 - \frac{1}{2} \Leftrightarrow 12 - 3\log_2 3 < 5\log_2 5 - 4 \\ \Leftrightarrow\ & 16 < 5\log_2 5 + 3\log_2 3 \Leftrightarrow 16 < 11.6096 + 4.7548 \ \textbf{(adev.)} \end{aligned}$$

În mod *alternativ*, ţinând cont de relaţia (286) **de la problema 36, putem proceda chiar *mai simplu* relativ la calcule (nu doar aici, ci ori de câte ori *nu* avem de-a face cu un *număr mare de instanţe*):**

$$H_{0/\,Netedă} < H_{0/\,Uşoară} \Leftrightarrow \frac{4^4}{\cancel{2^2}\cdot\cancel{2^2}}\cdot\frac{4^4}{3^3} < \frac{\cancel{3^3}}{\cancel{2^2}}\cdot\frac{5^5}{\cancel{2^2}\cdot\cancel{3^3}} \Leftrightarrow \frac{4^8}{3^3} < 5^5 \Leftrightarrow 4^8 < 3^3\cdot 5^5$$
$$\Leftrightarrow 2^{16} < 3^3\cdot 5^5 \Leftrightarrow 64\cdot 2^{10} < 27\cdot 25\cdot 125 \ \textbf{(adev.)}$$

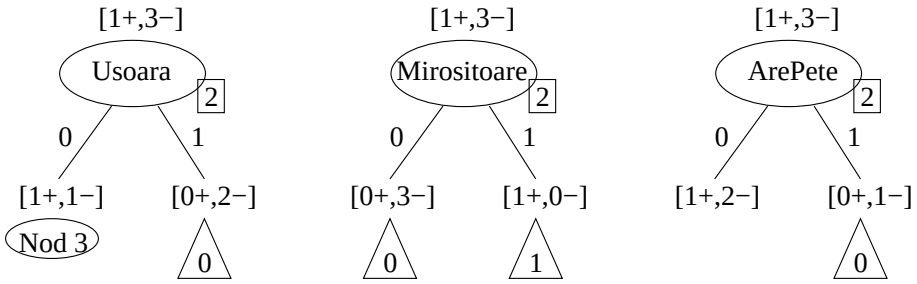
Vă sfătuim să *procedaţi aşa* la rezolvarea problemelor propuse din acest capitol, acolo unde este cazul.

***Nodul 1*: Trebuie să clasificăm acele exemple care au** *Netedă* $= 0$**; avem de ales între 3 atribute -** *Uşoară*, *Mirositoare* **şi** *ArePete*.



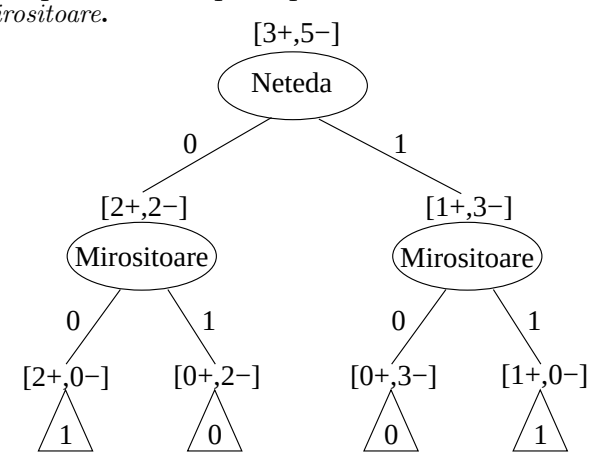
Avem $H_{1/\,Mirositoare} = \frac{2}{4}H[2+,0-] + \frac{2}{4}H[0+,2-] = 0$**. Oricare ar fi valorile pentru** $H_{1/\,ArePete}$ **şi** $H_{1/\,Uşoară}$**, întrucât ştim că entropia are întotdeauna valori nenegative, rezultă că atributul** *Mirositoare* **maximizează în nodul 1 câştigul de informaţie. În imaginea de mai sus valorile din triunghi reprezintă decizia luată de subarborele construit în nodul-frunză respectiv.**

***Nodul 2*: Avem de clasificat exemplele pentru care** *Netedă* $= 1$**. Atributele disponibile sunt:** *Uşoară*, *Mirositoare* **şi** *ArePete*.



Evident, $H_{2/Mirositoare} = \frac{3}{4}H[0+,3-] + \frac{1}{4}H[1+,0-] = \frac{3}{4}\cdot 0 + \frac{1}{4}\cdot 0 = 0$ **Aşadar, pentru nodul 2 putem alege atributul** *Mirositoare*.

Arborele complet arată astfel:



Parcurgând arborele construit, ciupercile U, V **şi** W **vor fi clasificate astfel:**

U	$\boldsymbol{Netedă = 1,\ Mirositoare = 1 \Rightarrow Comestibilă = 1}$
V	$\boldsymbol{Netedă = 1,\ Mirositoare = 1 \Rightarrow Comestibilă = 1}$
W	$\boldsymbol{Netedă = 0,\ Mirositoare = 1 \Rightarrow Comestibilă = 0}$

e. $Comestibilă \leftrightarrow (\neg Netedă \wedge \neg Mirositoare) \vee (Netedă \wedge Mirositoare)$

Acelaşi lucru poate fi exprimat şi sub forma unui pseudo-cod ***if*** $\cdots$ ***then Comestibilă else*** $\neg$ ***Comestibilă*****:**

IF (***Netedă*** $= 0$ **AND** ***Mirositoare*** $= 0$**) OR**
(***Netedă*** $= 1$ **AND** ***Mirositoare*** $= 1$**)**
THEN ***Comestibilă*****;**
ELSE $\neg$***Comestibilă*****;**

f. Arborele de decizie produs de către algoritmul ID3 elaborat mai sus este consistent cu datele de antrenament pe care le-am avut la dispoziţie (fiindcă aceste date sunt necontradictorii). Întrucât în realitate clasificarea poate depinde şi de alte trăsături / informaţii decât cele de care dispunem noi, nu avem garanţia că arborele ID3 face identificarea corectă a etichetei / clasei pentru toate instanţele din setul de test. Aşadar, nu putem fi siguri că nu ne vom îmbolnăvi dacă vom consuma ciupercile U **şi** V**, sau că ne vom îmbolnăvi dacă vom consuma ciuperca** W**. În multe aplicaţii practice, calitatea unui model de învăţare automată (în cazul de faţă, un arbore de decizie) se verifică pe un set de *date de validare*.**

3. **(Algoritmul ID3, aplicat pe expresii booleene; exploatarea simetriilor operaţiilor $\vee, \wedge$ în alegerea atributelor; analiza „optimalităţii“ arborelui ID3, ca număr de noduri de test)**

prelucrare de Liviu Ciortuz, după Tom Mitchell, "Machine Learning", 1997, ex. 3.1.b

Considerăm următoarea funcţie booleană: $A \vee (B \wedge C)$. Presupunem că această funcţie este deja definită — adică valoarea ei este cea cunoscută din logica propoziţiilor —, însă dorim să o reprezentăm ca arbore de decizie.

a. Aplicaţi algoritmul ID3 [tabelei de adevăr corespunzătoare] acestei funcţii.
***Observaţie*: Dacă exploataţi simetriile, veţi avea nevoie doar de puţine calcule, altfel vă veţi complica inutil.**

b. Arborele ID3 obţinut la punctul precedent este optimal?
Altfel spus, puteţi găsi un alt arbore de decizie, de adâncime mai mică sau cu număr mai mic de noduri (comparativ cu arborele obţinut la punctul a), care să reprezinte această funcţie? (Ţineţi cont că în fiecare nod al unui arbore de decizie se poate testa un sigur atribut.)

Răspuns:

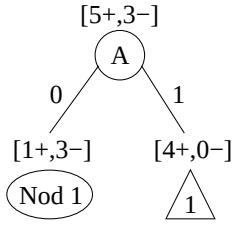
A	B	C	$Y = A \vee (B \wedge C)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

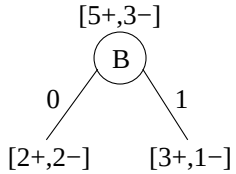
a. Observăm că funcţia dată este simetrică în B şi C, datorită comutativităţii operatorului logic $\wedge$. O consecinţă a acestui fapt este că dacă, pe parcursul algoritmului ID3, avem de ales (şi) între cele două atribute este nevoie să-l studiem doar pe unul dintre ele, celălalt comportându-se identic.

***Nodul 0* (rădăcină):**

$$H_{0/A} = \frac{4}{8}H[1+,3-] + \frac{4}{8}H[4+,0-] =$$
$$= \frac{1}{2}H[1+,3-] + \frac{1}{2}\cdot 0 = \frac{1}{2}H[1+,3-]$$

$$H_{0/B} = \frac{4}{8}H[2+,2-] + \frac{4}{8}H[3+,1-] =$$
$$= \frac{1}{2}\cdot 1 + \frac{1}{2}H[3+,1-] = \frac{1}{2} + \frac{1}{2}H[1+,3-]$$





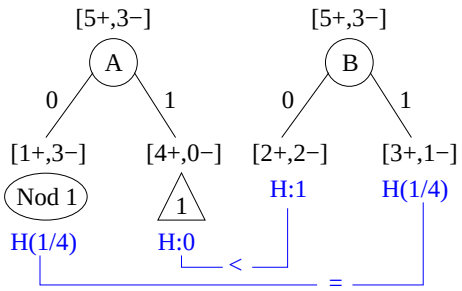
Este evident că $H_{0/A} < H_{0/B}$, deci vom alege atributul A în rădăcină.

***Observaţii importante*:**

1. La aceeaşi concluzie se putea ajunge *imediat* pe baza unui *raţionament* de tip *calitativ*, şi anume, comparând atent cei doi arbori („compaşi de decizie“) de mai sus. Mai precis, vom compara (două câte două) entropiile condiţionale

specifice din nodurile descendente, precum şi ponderile cu care se combină aceste entropii în scrierea entropiilor condiţionale medii corespunzătoare atributelor A şi B.

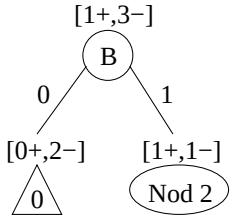
Putem pune în evidenţă acest fapt în figura alăturată, în care simbolul H, scris uneori însoţit de un argument (aşadar, ca $H(p)$), se referă la entropia unei variabile Bernoulli de parametru p.



Mai facem *precizarea* că semnele $<$ şi $=$ din figura de mai sus se referă de fapt nu [doar] la entropiile condiţionale specifice, ci [şi] la produsul acestora cu ponderile asociate în mod corespunzător: $\frac{4}{8}H[1+,3-] = \frac{4}{8}H[3+,1-]$ şi respectiv $\frac{4}{8}H[4+,0-] < \frac{4}{8}H[2+,2-]$.

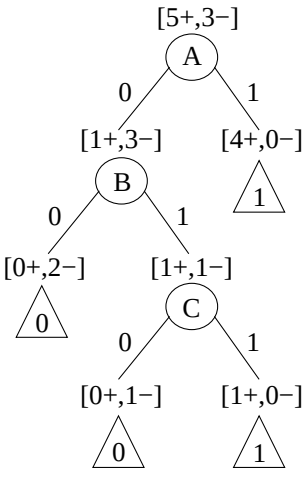
2. Pentru câteva formule de calcul convenabile pentru entropii şi câştiguri de informaţie relative la compaşi de decizie, atunci când se foloseşte calculatorul de buzunar, dar numărul de instanţe de antrenament asociate nu este prea mare, vedeţi problema 36.

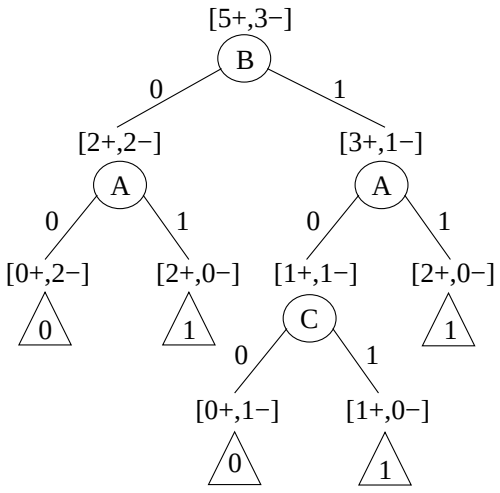
***Nodul 1*: Avem de clasificat instanţele care au $A = 0$ şi putem alege între atributele B şi C. Datorită simetriei, îl putem alege pe oricare dintre ele. Pentru fixare, îl alegem pe B.**



***Nodul 2*: La acest punct a mai rămas disponibil doar atributul C.**

Arborele construit de ID3 este cel reprezentat mai jos, în partea stângă:[500]





[500] Un alt arbore ID3 este cel obţinut din acesta interschimbând atributele B şi C. (Vedeţi *Observaţia* din enunţ.)

b. Pentru a vedea dacă arborele construit de algoritmul ID3 este cel optimal, trebuie să reconsiderăm toate deciziile pe care le-am luat în construirea acestuia:

– La nodul 1 al arborelui avem de clasificat exemplele pentru care $A = 0$, deci funcţia care trebuie reprezentată de subarborele în cauză este $f' = f[A/0] = 0 \vee (B \wedge C) = B \wedge C$, funcţie care este reprezentată în mod optimal de subarborele construit de ID3. (Notaţia $A/0$ semnifică faptul că variabila logică A este instanţiată la valoarea 0.) Prin urmare, nu există un arbore mai bun care să reprezinte funcţia dată şi să aibă în rădăcină atributul A.

– În rădăcină am ales atributul A în detrimentul celorlalte două atribute deoarece am demonstrat că aduce cel mai mare câştig de informaţie. Să vedem ce se întâmplă dacă alegem unul dintre atributele B sau C. După cum am discutat mai sus, datorită simetriei, pe oricare dintre cele două l-am alege, arborele rezultat ar avea aceeaşi formă. Pentru fixare, să-l alegem pe B.

Subarborele stâng şi drept vor trebui să reprezinte funcţiile:

$$f'' = f[B/0] = A \vee (0 \wedge C) = A \vee 0 = A$$

şi respectiv

$$f''' = f[B/1] = A \vee (1 \wedge C) = A \vee C.$$

Arborele minimal care poate fi construit în aceste circumstanţe este cel reprezentat mai sus în partea dreaptă. Acest arbore are 3 niveluri şi 4 noduri, cu un nod în plus faţă de cel construit de algoritmul ID3.

Putem deci conchide că arborele construit respectând specificaţiile algoritmului ID3 este cel optimal.

***Observaţie*:**

Această problemă pune în evidenţă două modalităţi de parcurgere a spaţiului de versiuni pentru un concept (în cazul de faţă un concept din logica propoziţiilor) care este reprezentat cu ajutorul arborilor de decizie. Pe de o parte avem explorarea (incompletă) făcută de algoritmul ID3 care este de tip "greedy", iar pe de altă parte avem explorarea exhaustivă. Prima strategie de explorare procedează la o căutare „orientată" a soluţiei (şi din această cauză este mai eficientă, dar se va vedea, ca revers, că nu asigură întotdeauna găsirea optimului), iar cea de-a doua strategie de explorare, deşi asigură găsirea optimului, nu este utilizabilă în cazurile (frecvente!) în care spaţiul de versiuni este foarte mare.

4. **(ID3, ca algoritm "greedy";
un exemplu când arborele ID3 nu este optimal
ca număr de noduri şi de niveluri)**

prelucrare de Liviu Ciortuz, după
■ ***CMU, 2003 fall, T. Mitchell, A. Moore, midterm exam, pr. 9.a***

Fie atributele binare de intrare A, B, C, atributul de ieşire Y şi următoarele exemple de antrenament:

A	B	C	Y
1	1	0	0
1	0	1	1
0	1	1	1
0	0	1	0

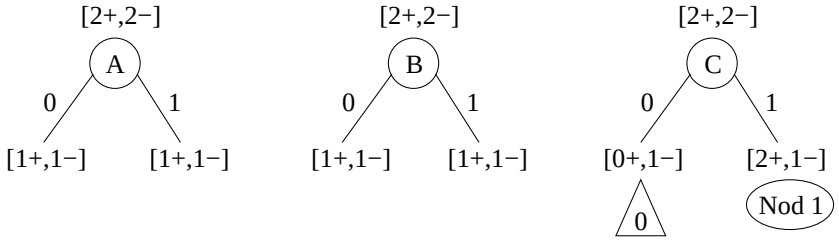
a. Determinaţi arborele de decizie calculat de algoritmul ID3. Este acest arbore de decizie *consistent* cu datele de antrenament?

b. Există un arbore de decizie de adâncime mai mică (decât cea a arborelui ID3) consistent cu datele de mai sus? Dacă da, ce concept (logic) reprezintă acest arbore?

Răspuns:

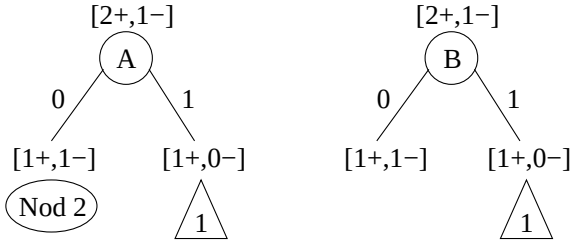
a. Se construieşte arborele de decizie cu algoritmul ID3 astfel:

***Nodul 0* (rădăcina):**



Observăm că $H_{0/A} = H_{0/B} = \frac{2}{4}H[1+,1-] + \frac{2}{4}H[1+,1-] = H[1+,1-] = 1$**, care este valoarea maximă a entropiei [condiţionale medii a] unei variabile booleene. Prin urmare,** $H_{0/C}$ **nu poate fi decât mai mică sau egală cu** $H_{0/A}$ **şi** $H_{0/B}$**. Deci vom alege în nodul rădăcină atributul** C**.**

***Nodul 1*: Avem de clasificat instanţele cu** $C = 1$**, deci alegerea se face între atributele** A **şi** B**.**

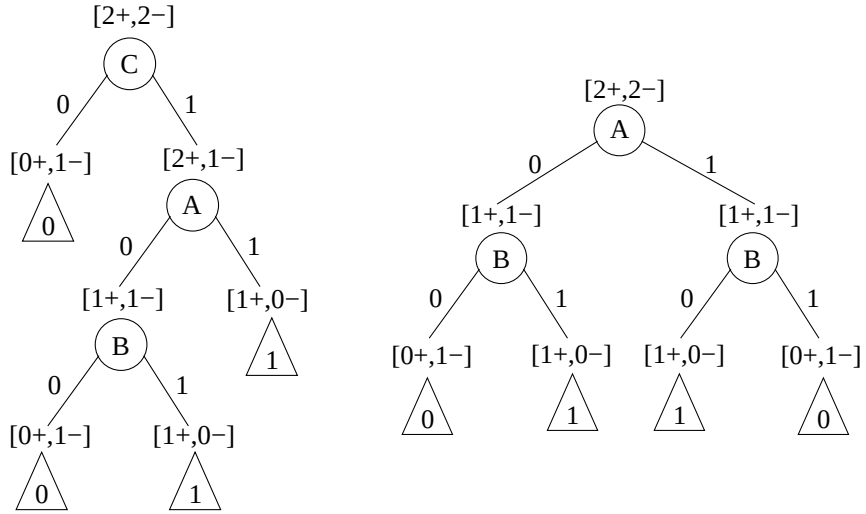


Cele două entropii condiţionale medii sunt egale:

$$H_{1/A} = H_{1/B} = \frac{2}{3}H[1+,1-] + \frac{1}{3}H[1+,0-]$$

Aşadar, putem alege oricare dintre cele două atribute. Pentru fixare, îl alegem pe A**.**

***Nodul 2*: La acest nod nu mai avem decât atributul B, deci îl vom pune pe acesta. Arborele complet este reprezentat în partea stângă:**



Prin construcţie, arborele ID3 este consistent cu datele de antrenament dacă acestea sunt consistente (i.e., necontradictorii). În cazul nostru, se verifică imediat că datele de antrenament sunt consistente.

b. Se observă că atributul de ieşire Y reprezintă de fapt funcţia logică A XOR B. Reprezentând această funcţie ca arbore de decizie, vom obţine arborele desenat mai sus în partea dreaptă. Acest arbore are cu un nivel mai puţin decât arborele construit cu algoritmul ID3. Prin urmare, arborele obţinut de algoritmul ID3 pe datele din enunţ *nu* este *optim* din punctul de vedere al numărului de niveluri. Aceasta este o *consecinţă* a caracterului "greedy" al algoritmului ID3, datorat faptului că la fiecare iteraţie alegem „cel mai bun" atribut în raport cu criteriul câştigului de informaţie. Se ştie că algoritmii de tip "greedy" nu grantează obţinerea optimului global.

5. **(Clasificare ternară: "decision stump" produs de ID3, pe date care conţin duplicări şi „zgomote")**

Presupunem că se dau şase date de antrenament (precizate în tabel) pentru o problemă de clasificare cu două atribute binare şi trei clase $Y \in \{1, 2, 3\}$. Se va crea un arbore ID3, bazat pe câştigul de informaţie.

X_1	X_2	Y
1	**1**	**1**
1	**1**	**1**
1	**1**	**2**
1	**0**	**3**
0	**0**	**2**
0	**0**	**3**

a. Calculaţi câştigul de informaţie atât pentru X_1 cât şi pentru X_2. Se va folosi aproximarea $\log_2 3 = 19/12$ şi se va scrie câştigul de informaţie sub formă de fracţii.

b. Pe baza rezultatelor anterioare, ce atribut va fi folosit pentru primul nod al arborelui ID3? Desenaţi arborele de decizie care rezultă folosind doar acest singur nod. Etichetaţi corespunzător nodul, ramurile şi eticheta prevăzută în fiecare frunză.

c. Cum va clasifica acest arbore instanţa determinată de $X_1 = 0$ şi $X_2 = 1$?

Răspuns:

a. Redăm formula pentru calculul câştigului de informaţie în varianta folosită de Tom Mitchell:

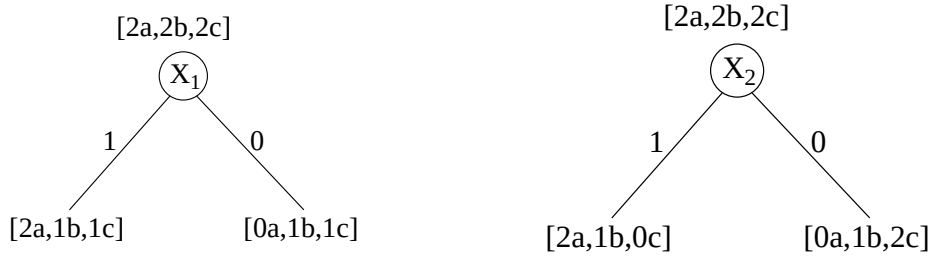
$$Gain(X_i) = Entropy(S) - \sum_{v \in Values(X_i)} \frac{|S_v|}{|S|} Entropy(S_v)$$

unde S este mulţimea celor 6 date de antrenament, iar

$$Entropy(S) = -\sum_{y \in Y} p_y \log_2 p_y$$

Notăm cu a clasa instanţelor având eticheta $Y = 1$, cu b clasa instanţelor cu $Y = 2$ şi cu c clasa instanţelor cu $Y = 3$. În mulţimea S există câte 2 elemente din fiecare clasă şi atunci putem scrie că $S = [2a, 2b, 2c]$.

Pentru nodul rădăcină, putem alege fie atributul X_1, fie atributul X_2, ceea ce determină următoarele împărţiri ale mulţimii S:



Putem calcula câştigurile de informaţie pentru cele două atribute:

$$Gain(X_1) = H[2a, 2b, 2c] - \left(\frac{4}{6} H[2a, 1b, 1c] + \frac{2}{6} H[0a, 1b, 1c]\right)$$

unde $H[2a, 2b, 2c]$ este o altă notaţie pentru entropia mulţimii compuse din două exemple de clasă a, două de clasă b şi două de clasă c.

Calculăm entropiile care intervin în formulă:

$$\begin{aligned}
H[2a, 2b, 2c] &= -\frac{2}{6}\log_2\frac{2}{6} - \frac{2}{6}\log_2\frac{2}{6} - \frac{2}{6}\log_2\frac{2}{6} \\
&= -3 \cdot \frac{2}{6}\log_2\frac{2}{6} = -\log_2\frac{1}{3} = \log_2 3 = \frac{19}{12} \\
H[2a, 1b, 1c] &= -\frac{2}{4}\log_2\frac{2}{4} - \frac{1}{4}\log_2\frac{1}{4} - \frac{1}{4}\log_2\frac{1}{4} = +\frac{1}{2}\cdot\log_2 2 + \frac{1}{4}\log_2 4 + \frac{1}{4}\log_2 4 \\
&= \frac{1}{2} + \frac{1}{4}\cdot 2 + \frac{1}{4}\cdot 2 = \frac{1}{2} + \frac{1}{2} + \frac{1}{2} = \frac{3}{2} \\
H[0a, 1b, 1c] &= 0 - \frac{1}{2}\cdot\log_2\frac{1}{2} - \frac{1}{2}\cdot\log_2\frac{1}{2} = \log_2 2 = 1
\end{aligned}$$

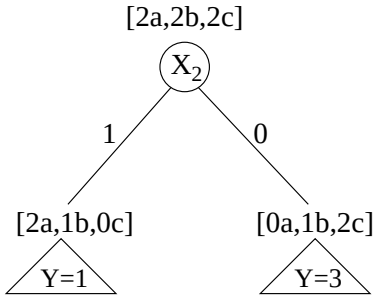
Înlocuind aceste valori numerice în formula câştigului de informaţie, obţinem:

$$Gain(X_1) = \frac{19}{12} - \left(\frac{2}{3}\cdot\frac{3}{2} + \frac{1}{3}\cdot 1\right) = \frac{19}{12} - \left(1 + \frac{1}{3}\right) = \frac{19}{12} - \frac{4}{3} = \frac{3}{12} = \frac{1}{4}$$

Se aplică aceleaşi formule şi pentru atributul X_2:

$$\begin{aligned} Gain(X_2) &= H[2a,2b,2c] - \left(\frac{3}{6}H[2a,1b,0c] + \frac{3}{6}H[0a,1b,2c]\right) \\ H[2a,1b,0c] &= H[0a,1b,2c] \\ &= -\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3} = -\frac{2}{3}(\log_2 2 - \log_2 3) - \frac{1}{3}(-1)\log_2 3 \\ &= -\frac{2}{3}\cdot 1 + \frac{2}{3}\cdot\frac{19}{12} + \frac{1}{3}\cdot\frac{19}{12} = \frac{19}{12} - \frac{2}{3} = \frac{11}{12} \\ \Rightarrow Gain(X_2) &= \frac{19}{12} - \left(\frac{1}{2}\cdot\frac{11}{12} + \frac{1}{2}\cdot\frac{11}{12}\right) = \frac{19}{12} - \frac{11}{12} = \frac{8}{12} = \frac{2}{3}. \end{aligned}$$

b. Deoarece $Gain(X_1) = \frac{3}{12}$, $Gain(X_2) = \frac{8}{12}$, şi deci $Gain(X_1) < Gain(X_2)$, se va alege atributul X_2 ca rădăcină a arborelui. Arborele de decizie construit din acest singur nod este cel din figura alăturată.



c. O instanţă care are $X_1 = 0$ şi $X_2 = 1$ va fi clasificată de acest arbore cu $Y = 1$ (cu probabilitate $2/3$).

6. **(Algoritmul ID3: aplicare pe date inconsistente, "decision stumps", calculul acurateţii la antrenare)**

• ○ ***CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, HW1, pr. 2.ab***

Tabelul de mai jos sumarizează situaţia celor 2201 de pasageri şi membri ai echipajului de la bordul vasului Titanic, în urma naufragiului din data de 15 Aprilie 1912. Pentru fiecare combinaţie de valori ale celor 3 variabile (Clasă, Sex, Vârstă) am indicat în tabel câţi oameni au supravieţuit şi câţi nu au supravieţuit. (*Observaţie*: Datele originale au patru valori pentru atributul Clasă; am comasat valorile II, III, şi Echipaj într-o singură valoare, denumită „Inferioară".)

			Supravieţuitori		
Clasa	**Sexul**	**Vârsta**	**Nu**	**Da**	**Total**
I	**Masculin**	**Copil**	**0**	**5**	**5**
I	**Masculin**	**Adult**	**118**	**57**	**175**
I	**Feminin**	**Copil**	**0**	**1**	**1**
I	**Feminin**	**Adult**	**4**	**140**	**144**
Inferioară	**Masculin**	**Copil**	**35**	**24**	**59**
Inferioară	**Masculin**	**Adult**	**1211**	**281**	**1492**
Inferioară	**Feminin**	**Copil**	**17**	**27**	**44**
Inferioară	**Feminin**	**Adult**	**105**	**176**	**281**
		Total	**1490**	**711**	**2201**

Pentru a vă uşura calculele pe care va trebui să le faceţi, am făcut noi totalurile pentru fiecare variabilă:

	Supravieţuitori		
Clasa	**Nu**	**Da**	**Total**
I	**122**	**203**	**325**
Inferioară	**1368**	**508**	**1876**

	Supravieţuitori		
Sexul	**Nu**	**Da**	**Total**
Masculin	**1364**	**367**	**1731**
Feminin	**126**	**344**	**470**

	Supravieţuitori		
Vârsta	**Nu**	**Da**	**Total**
Copil	**52**	**57**	**109**
Adult	**1438**	**654**	**2092**

a. Folosind un arbore de decizie, dorim să prezicem variabila de ieşire Y (Supravieţuitor), pornind de la atributele de intrare C (Clasa), S (Sexul), V (Vârsta). Utilizaţi criteriul câştigului de informaţie pentru a alege care dintre aceste trei atribute C, S sau V trebuie să fie folosit în nodul-rădăcină al arborelui de decizie.

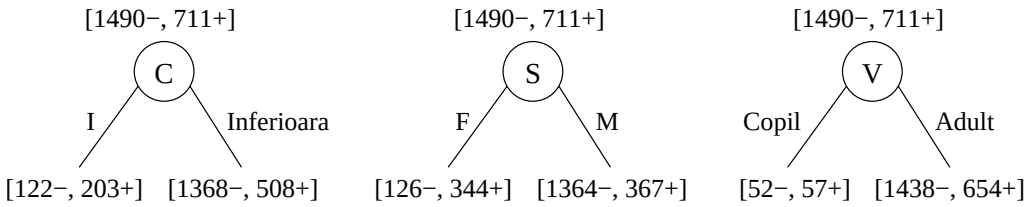
De fapt, ce vi se cere este să învăţaţi un arbore de decizie de adâncime 1 care foloseşte doar atributul din rădăcină pentru a clasifica datele. (Astfel de arbori de decizie de adâncime 1 sunt adesea numiţi în terminologia de limbă engleză "decision stumps".) Parcurgeţi toate etapele rezolvării, redând inclusiv calculele pentru câştigul de informaţie al fiecărui atribut.

b. Care este acurateţea [medie] obţinută pe datele de antrenament de către arborele de decizie cu adâncime 1 de la punctul precedent?

c. Dacă aţi crea un arbore de decizie care foloseşte toate cele trei variabile, care ar fi acurateţea lui [medie] pe datele de antrenament? (*Observaţie*: Nu trebuie neapărat să creaţi arborele de decizie pentru a afla răspunsul!)

Răspuns:

a. Totalurile care au fost furnizate în enunţ pentru fiecare dintre variabilele C, S şi V ne servesc foarte bine pentru a crea rapid cei trei "decision stumps":



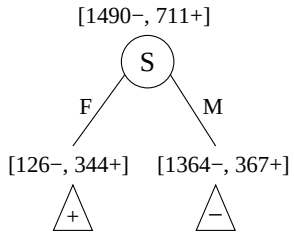
Analizând datele conform figurii de mai sus, se poate „intui" că atributul S va avea un câştig de informaţie (în raport cu atributul de ieşire Y – *Supravieţuitor*) mai bun decât al celorlalte două atribute de intrare (C şi V). Intuiţia se verifică făcând calculele:

$$\begin{aligned} IG(Y,C) &= H[1490-,711+] - \left(\frac{325}{2201}H[122-,203+] + \frac{1876}{2201}H[1368-,508+]\right) \\ &= 0.048501 \\ IG(Y,S) &= H[1490-,711+] - \left(\frac{470}{2201}H[126-,344+] + \frac{1731}{2201}H[1364-,367+]\right) \\ &= 0.142391 \end{aligned}$$

$$IG(Y,V) = H[1490-,711+] - \left(\frac{109}{2201}H[52-,57+] + \frac{2092}{2201}H[1438-,654+]\right)$$
$$= 0.006411.$$

Deci, într-adevăr, câştigul maxim de informaţie se obţine pentru atributul S.

b. Arborele de decizie de adâncime 1 care are în nodul rădăcină atributul S este cel din figura alăturată. Acurateţea [medie a] acestui arbore de decizie este:



$$\frac{470}{2201}\cdot\frac{344}{470} + \frac{1731}{2201}\cdot\frac{1364}{1731} = \frac{344+1364}{2201} = \frac{1708}{2201} = 0.776.$$

c. Se poate constata imediat că arborele ID3 produs pe datele din această problemă va avea 8 noduri-frunză, iar în fiecare dintre aceste noduri-frunză se va asigna câte una dintre mulţimile descrise (pe linie) în coloanele 4 şi 5 ale tabelului principal din enunţ: $[0-,5+]$, $[118-,57+],\ldots,[17-,27+]$, $[105-,176+]$. Decizia care va fi luată în fiecare nod-frunză este dictată de votul majoritar, şi anume: $+,-,\ldots,+$ şi respectiv $+$.

Putem calcula acurateţea [medie] astfel:

$$\frac{5+118+1+140+35+1211+27+176}{2201} = \frac{1713}{2201} = 0.778.$$

Se observă că se produce (din păcate) o creştere foarte mică în raport cu acurateţea celui mai bun "decision stump": doar 0.002.

7. **(Algoritmul ID3: cazul când există repetiţii şi inconsistenţe în datele de antrenament; o margine superioară pentru eroarea la antrenare în funcţie de numărul de valori ale variabilei de ieşire)**

CMU, 2002 fall, Andrew Moore, midterm exam, pr. 1.fg

A	B	C	Z
0	0	0	0
0	0	1	0
0	0	1	0
0	1	0	0
0	1	1	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	0	1
1	1	1	0
1	1	1	1

Presupunem că învăţăm un arbore de decizie care să prezică atributul de ieşire Z pornind de la atributele de intrare A, B, C. Se folosesc datele de antrenament din tabelul alăturat.

a. Care va fi eroarea la antrenare pe acest set de date? Exprimaţi răspunsul sub forma fracţiei de înregistrări care vor fi clasificate eronat ($n/12$).

b. Considerăm un arbore de decizie construit pe un set arbitrar de date. Dacă atributul de ieşire este cu valori discrete şi poate lua k valori distincte, care este maximul erorii la antrenare (exprimată ca fracţie)?

Răspuns:

a. Este uşor de observat că datele de antrenament conţin „inconsistenţe" (contradicţii, relativ la etichetare), şi anume la exemplele $(0,1,1)$ şi $(1,1,1)$.

Fiecare dintre aceste exemple sunt etichetate o dată cu 0 şi altă dată cu 1. Prin urmare, jumătate din aceste exemple vor fi clasificate eronat de arborele învăţat de către algoritmul ID3. Eroarea la antrenare va fi deci $\frac{2}{12}$.

b. Vom analiza pe rând mai multe cazuri, care sunt din ce în ce mai generale.

Cazul *i*: Mai întâi vom calcula eroarea la antrenare pentru cazul în care setul de date de antrenament este compus din k instanţe care sunt identice ca tupluri de valori pentru atributele ce le caracterizează, dar sunt clasificate pe rând cu fiecare din cele k valori posibile ale atributului de ieşire. Este evident că arborele de decizie învăţat va clasifica eronat $k-1$ instanţe. Aşadar, în acest caz, eroarea la antrenare va fi

$$E = \frac{k-1}{k}$$

Cazul *ii*: Aceeaşi eroare [la antrenare] ca mai sus se va înregistra dacă în locul fiecărei instanţe dintre cele considerate la cazul precedent vom avea l instanţe identice, inclusiv în ce priveşte eticheta. (În total sunt kl instanţe de antrenament.)

$$E = \frac{(k-1)\cdot l}{k\cdot l} = \frac{k-1}{k}$$

Cazul *iii*: Dacă relaxăm condiţia de mai sus considerând $l_1, l_2, \ldots, l_k$ instanţe identice, iar $l = \max_{i=1}^{k} l_i$, este imediat că eroarea maximă se va atinge în cazul $l_1 = l_2 = \ldots = l_k = l$, şi va avea aceeaşi valoare ca mai sus. Aşadar, în continuare vom putea renunţa la a considera factorul de multiplicare l, fără ca prin aceasta să restrângem generalitatea raţionamentului.

Cazul *iv*: Fie n exemple de antrenament (instanţe etichetate) dintre care d sunt distincte (ca tupluri de valori ale atributelor de intrare). Fie k_1, $k_2 \cdots k_d$ numărul de instanţe etichetate distincte pentru fiecare caz în parte din cele d. Atunci vom avea:

$$k_1 \le k,\, k_2 \le k, \cdots, k_d \le k \Rightarrow n = k_1 + k_2 + \cdots + k_d \le k\cdot d \text{ deci } n \le k\cdot d$$

Eroarea maximă la antrenare va fi dată de formula

$$E = \frac{(k_1-1)+(k_2-1)+\cdots+(k_d-1)}{n} = \frac{n-d}{n}$$

Avem:

$$E \le \frac{k-1}{k} \Leftrightarrow \frac{n-d}{n} \le \frac{k-1}{k} \Leftrightarrow k\cdot n - k\cdot d \le k\cdot n - n \Leftrightarrow k\cdot d \ge n \text{ (adev.)}$$

Prin urmare, eroarea maximă la antrenare care poate fi atinsă atunci când atributul de ieşire poate lua k valori distincte este $\frac{k-1}{k}$.

8. **(ID3, aspecte computaţionale: influenţa atributelor duplicate, respectiv a instanţelor de antrenament duplicate asupra arborelui ID3 rezultat)**

CMU, 2009 spring, Ziv Bar-Joseph, final exam, pr. 3.1-2

Se doreşte construirea unui arbore de decizie pentru n vectori / exemple de antrenament, cu m atribute.

a. Să presupunem că există i şi j astfel încât pentru TOŢI vectorii X din datele de antrenament, aceste atribute au valori egale (adică, $x_i = x_j$ pentru toţi vectorii, unde x_i este valoarea atributului i în vectorul X). Să presupunem de asemenea că în cazul în care ambele atribute duc la acelaşi câştig de informaţie vom folosi atributul i. Ştergerea atributului j din datele de antrenament poate schimba arborele de decizie obţinut? Explicaţi pe scurt.

b. Să presupunem că există în mulţimea de antrenament doi vectori egali X şi Z (adică, toate atributele lui X şi Z sunt exact la fel, inclusiv etichetele). Ştergerea vectorului Z din datele de antrenament poate schimba arborele de decizie obţinut? Explicaţi pe scurt.

Răspuns:

a. Nu, îndepărtarea atributului j nu schimbă arborele de decizie, deoarece atributele i şi j conduc la valori egale pentru câştigul de informaţie în fiecare nod al arborelui.

b. Da, în acest caz arborele de decizie se poate schimba, fiindcă entropia condiţională — care se calculează în fiecare nod pentru a determina atributul cu câştigul de informaţie cel mai mare — depinde de numărul de instanţe de antrenament luate în considerare.

9. **(Arbori de decizie: o margine superioară pentru numărul de noduri frunză, în funcţie de numărul atributelor şi numărul de exemple)**

CMU, 2005 fall, T. Mitchell, A. Moore, midterm exam, pr. 2.d

Presupunem că învăţăm un arbore de decizie folosind un set de R instanţe de antrenament descrise de M atribute de intrare având valori binare.

Care este numărul maxim posibil de noduri frunză din arborele de decizie, presupunând că fiecărui nod frunză îi este asociat măcar un exemplu de antrenament? Încercuiţi unul din răspunsurile de mai jos; justificaţi alegerea făcută.

R, $\log_2(R)$, R^2, 2^R, M, $\log_2(M)$, M^2, 2^M,
$\min(R, M)$, $\min(R, \log_2(M))$, $\min(R, M^2)$, $\min(R, 2^M)$,
$\min(\log_2(R), M)$, $\min(\log_2(R), \log_2(M))$, $\min(\log_2(R), M^2)$, $\min(\log_2(R), 2^M)$,
$\min(R^2, M)$, $\min(R^2, \log_2(M))$, $\min(R^2, M^2)$, $\min(R^2, 2^M)$,
$\min(2^R, M)$, $\min(2^R, \log_2(M))$, $\min(2^R, M^2)$, $\min(2^R, 2^M)$,
$\max(R, M)$, $\max(R, \log_2(M))$, $\max(R, M^2)$, $\max(R, 2^M)$,
$\max(\log_2(R), M)$, $\max(\log_2(R), \log_2(M))$, $\max(\log_2(R), M^2)$, $\max(\log_2(R), 2^M)$,
$\max(R^2, M)$, $\max(R^2, \log_2(M))$, $\max(R^2, M^2)$, $\max(R^2, 2^M)$,
$\max(2^R, M)$, $\max(2^R, \log_2(M))$, $\max(2^R, M^2)$, $\max(2^R, 2^M)$.

Răspuns:

Notăm cu $\max_{frunze}$ valoarea căutată. Trebuie luate în considerare două aspecte:

(a) fiecare nod frunză trebuie să clasifice măcar un exemplu de antrenament $\Rightarrow$ nu putem să avem mai multe frunze decât exemple de antrenament $\Rightarrow \max_{frunze} \leq R$

(b) fiecare atribut poate fi testat o singură dată pe un drum de la rădăcină la o frunză oarecare $\Rightarrow$ arborele obţinut va avea adâncimea cel mult M $\Rightarrow \max_{frunze} \leq 2^M$

$$\left.\begin{array}{l} \max_{frunze} \leq R \\ \max_{frunze} \leq 2^M \end{array}\right\} \Rightarrow \max_{frunze} \leq \min(R, 2^M)$$

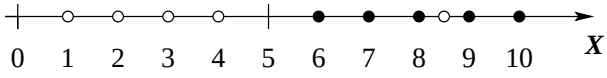
Această valoare poate fi atinsă: putem să luăm, de exemplu, cazul trivial când avem o singură instanţă de antrenament. Prin urmare, avem $\max_{frunze} = \min(R, 2^M)$.

10. **(Extensii ale algoritmului ID3: variabile de intrare continue; "decision stumps"; eroarea la antrenare, eroarea la CVLOO; overfitting)**

■ • ***CMU, 2002 fall, Andrew Moore, midterm exam, pr. 3***

Fie setul de date de mai jos. X este atribut de intrare şi ia valori reale, iar Y este variabilă de ieşire cu valori booleene. (*Observaţie*: Am reprezentat acest set de date sub formă grafică, marcând valoarea / eticheta $Y = 1$ prin bulină neagră, iar valoarea / eticheta $Y = 0$ prin cerculeţ alb.) Pe acest set de date se foloseşte algoritmul ID3 pentru învăţare de arbori de decizie.

X	Y
1	0
2	0
3	0
4	0
6	1
7	1
8	1
8.5	0
9	1
10	1



Algoritmul ID3 (extins) va trebui să decidă cum divide (engl., split) intervale de valori asociate variabilei reale X. Separarea în intervale diferite va fi stabilită cu ajutorul unor *praguri de separare* (engl., split thresholds), determinate în felul următor:

– **Mai întâi se ordonează crescător acele valori ale variabilei X care apar în datele de antrenament.**

– **Se stabilesc apoi perechi de valori consecutive pentru care există instanţe de antrenament care sunt etichetate în mod diferit pentru o valore, comparativ cu cealaltă valoare. Pentru fiecare pereche de valori de acest fel, va fi plasat un prag de separare la jumătatea distanţei dintre cele două valori.**

Iniţial, se alege pragul de separare care conduce la un câştig de informaţie maxim. Apoi, la fiecare nouă execuţie a buclei principale a algoritmului ID3 — vă readucem aminte că acest algoritm este recursiv — se va selecta câte un alt prag dintre cele rămase disponibile, aplicând acelaşi criteriu: maximizarea câştigului de informaţie.

De *exemplu*, pentru $X = 4$ avem o instanţă de antrenament negativă, iar pentru $X = 6$ avem o instanţă de antrenament pozitivă. Se poate arăta că algoritmul ID3 va decide să spliteze mai întâi la valoarea $X = 5$ (care reprezintă jumătatea distanţei dintre $X = 4$ şi $X = 6$) şi apoi la valoarea $X = 8.25$ (care reprezintă jumătatea distanţei dintre $X = 8$ şi $X = 8.5$).

Fie DT* arborele de decizie complet, obţinut de algoritmul ID3 fără a face pruning, iar DT2 arborele de decizie produs de ID3 urmat de pruning, care are doar două noduri frunză (deci DT2 face o singură divizare de interval).

a. Care este *eroarea la antrenare* produsă de DT2 respectiv DT* (exprimată ca număr de exemple clasificate eronat din totalul de 10 exemple)?

b. Care este *eroarea* produsă de DT2 respectiv DT* la *cross-validare* folosind metoda *Leave-One-Out* (CVLOO)?

Răspuns:

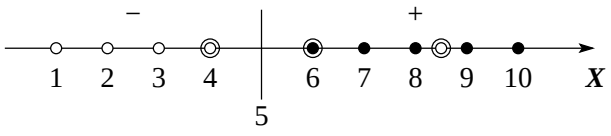
a. Deoarece DT2 reţine doar un singur split, şi anume cel de la pragul $X = 5$, regula de decizie pe care o reprezintă este:

IF $\mathbf{X \leq 5}$
THEN $\mathbf{Y = 0}$
ELSE $\mathbf{Y = 1}$

Este evident că această regulă produce o clasificare eronată doar pentru una dintre instanţele de antrenament, şi anume $X = 8.5$. Avem deci $E_{antren.}(DT2) = 1/10$.

Întrucât exemplele nu conţin inconsistenţe, arborele de decizie ID3 clasifică corect toate datele de antrenament. Avem deci $E_{antren.}(DT*) = 0/10 = 0$.

b. Figura de mai jos reprezintă împărţirea axei reale în *intervale / zone de decizie* conform arborelui (şi *pragului de decizie*) învăţat de către algoritmul DT2, folosind întregul set de exemple date. (Am încercut acele puncte care, după cum se va vedea mai jos, vor constitui cazuri aparte la calcularea erorii de tip CVLOO cu algoritmul DT2.)



În ce priveşte cross-validarea prin metoda "Leave-One-Out", se poate demonstra — calculele nu sunt arătate aici[501] — următorul fapt: pentru fiecare din cele 10 exemple ($X = 1$, $X = 2$, ..., $X = 10$) considerate pe rând, pragul de decizie identificat de algoritmul DT2 va fi $X = 5$, cu excepţia următoarelor două cazuri:

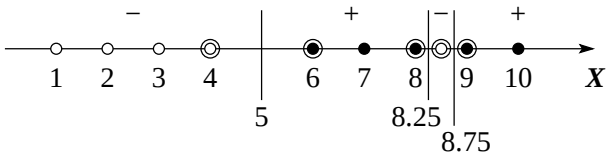
- **$X = 4$: în acest caz, splitarea se va face la mijlocul intervalului $[3, 6]$, deci la 4.5. Cum $4 \leq 4.5$, rezultă că exemplul $X = 4$ va fi clasificat corect;**
- **$X = 6$: splitarea se va face la mijlocul intervalului $[4, 7]$, deci la 5.5. Cum $6 > 5.5$, rezultă că exemplul $X = 6$ va fi clasificat corect.**

Atunci când punctul $X = 8.5$ este lăsat deoparte, pragul de decizie selectat fiind $X = 5$, va rezulta că punctul $X = 8.5$ este clasificat pozitiv (deci eronat), întrucât $8.5 > 5$.

Este imediat că pentru restul de 7 cazuri ($X = 1, 2, 3, 7, 8, 9, 10$), arborele determinat de algoritmul DT2 va clasifica corect punctul X.

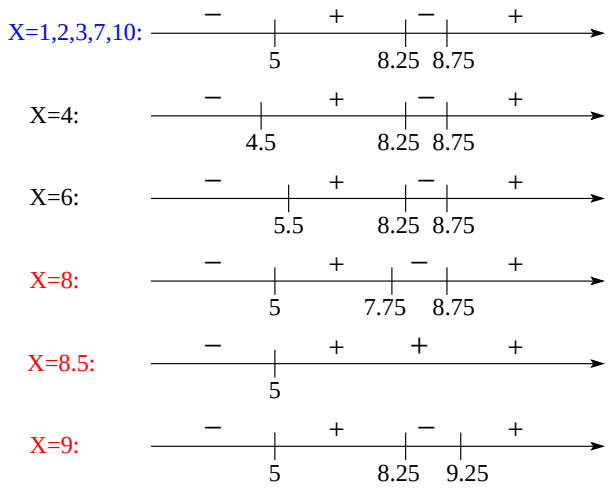
Aşadar, pentru DT2 rezultă $E_{CVLOO} = 1/10$.

Figura de mai jos reprezintă împărţirea axei reale în *intervale / zone de decizie* conform *pragurilor de decizie* (şi arborelui DT*) determinate de către algoritmul ID3 pe întregul set de date de antrenament.[502]



În această figură am încercuit exemplele care ar putea conduce la eroare la cross-validarea de tip "Leave-One-Out":

- **$X = 4$: corect clasificat, explicaţia este aceeaşi ca în cazul DT2;**
- **$X = 6$: idem;**
- **$X = 8$: split-ul trebuie făcut la mijlocul intervalului $[7, 8.5]$, adică 7.75. Cum $8 > 7.75$, rezultă că punctul $X = 8$ va fi clasificat negativ, ceea ce este eronat;**
- **$X = 8.5$: nu este nevoie decât de un singur split, arborele DT* învăţat în acest caz fiind identic cu cel construit de algoritmul DT2. Cum $8 > 5$, rezultă că punctul $X = 8.5$ va fi clasificat pozitiv, deci eronat;**



[501] Aceste calcule nu sunt dificile, mai ales dacă se utilizează raţionamente de tip „calitativ", aşa cum am arătat la problema 3.

[502] *Observaţie*: Atunci când ne raportăm doar la zonele de decizie în ansamblu, ordinea în care s-au stabilit testele în arborele ID3 este irelevantă! Acest fapt este valabil şi mai jos, unde discutăm despre eroarea la CVLOO cu algoritmul DT*.

• $X = 9$**: intervalul de split devine** $[8.5, 10]$**, split-ul făcându-se la** 9.25**. Cum** $9 \leq 9.25$**, rezultă că punctul** $X = 9$ **va fi clasificat negativ, ceea ce este eronat.**

Aşadar, pentru DT* avem $E_{CVLOO} = 3/10$**.**

În *concluzie*, se observă că $E_{antren.}(DT2) = 1/10 > 0 = E_{antren.}(DT*)$**, în vreme ce** $E_{CVLOO}(DT2) = 1/10 < 3/10 = E_{CVLOO}(DT*)$**. Acesta este un caz tipic de manifestare a fenomenului de *overfitting* (*supra-specializare*).**

11. **(Arbori de decizie cu variabile de intrare continue; o margine superioară pentru adâncimea arborilor în cazul (ne)separabilităţii liniare în** $\mathbb{R}^2$**)**

CMU, 2009 spring, Ziv Bar-Joseph, midterm exam, pr. 5.cd

Se consideră n **vectori bidimensionali** $(x = \{x_1, x_2\})$ **care pot fi clasificaţi folosind o funcţie [de regresie] liniară, adică există** $w \in \mathbb{R}^2$ **şi** $b \in \mathbb{R}$ **astfel încât:**

$$y = \begin{cases} +1 & \text{if } w \cdot x + b > 0 \\ -1 & \text{if } w \cdot x + b \leq 0 \end{cases}$$

a. Poate un arbore binar de decizie — atenţie!, nu neapărat arborele ID3 — să clasifice corect aceşti vectori? Dacă nu, justificaţi. Dacă da, determinaţi adâncimea maximă (adică numărul maxim de niveluri de test) ale unui astfel de arbore de decizie, care este optim (ca număr de niveluri de test).

b. Acum să presupunem că aceste n **date nu sunt separabile liniar (adică nu există** $w \in \mathbb{R}^2$ **şi** $b \in \mathbb{R}$ **cu proprietatea de mai sus). Poate un arbore de decizie (binar) să clasifice corect aceşti vectori? Dacă nu, justificaţi. Dacă da, care este adâncimea maximă a unui arbore de decizie corespunzător, optim ca număr de niveluri?**

Răspuns:

a. Da. O strategie posibilă pentru a construi un arbore de decizie binar care să clasifice corect aceste puncte este următoarea:

Mai întâi vom construi un arbore de decizie considerând doar atributul x_1**. În particular, făcând „înjumătăţiri" succesive ale mulţimii de valori ale acestui atribut, arborele rezultat va avea o adâncime maximă de** $\lceil \log_2 n \rceil$ **(adică partea întreagă superioară din** $\log_2 n$**) niveluri. În fiecare nod frunză al acestui arbore se va găsi o mulţime de puncte, însă nu neapărat cu aceeaşi clasificare. Totuşi, deoarece datele sunt liniar separabile, pentru fiecare dintre aceste mulţimi de puncte se poate găsi o valoare a atributului** x_2 **care să o împartă în două submulţimi clasificate corect.**

Prin urmare, arborele construit iniţial considerând doar valoarea x_1 **are nevoie să i se adauge doar cel mult câte un nod în fiecare frunză, nod care să clasifice corect datele luând în considerare valoarea** x_2**. Adâncimea totală a arborelui astfel construit este** $1 + \lceil \log_2 n \rceil$**, adică de ordinul** $O(\log n)$**.**

b. Da. Similar punctului anterior, se construieşte un arbore de decizie considerând doar atributul x_1**, ceea ce înseamnă o adâncime de** $\lceil \log_2 n \rceil$**. În fiecare**

nod frunză al acestui arbore se va găsi o mulţime de puncte, posibil cu clasificări diferite. Datele de antrenament nu mai sunt liniar separabile, deci nu pot fi clasificate corect printr-un singur nod.

Pentru fiecare astfel de nod frunză în care punctele nu au aceeaşi clasificare, se aplică algoritmul de determinare a arborelui de decizie, de această dată luând în considerare doar valoarea lui x_2. Aceasta presupune din nou o adâncime maximă a subarborelui de $\lceil \log_2 n \rceil$. În total, arborele obţinut are adâncimea maximă $\lceil \log_2 n \rceil + \lceil \log_2 n \rceil$, deci tot de ordinul $O(\log n)$.

12. (Algoritmul ID3 cu atribute discrete, respectiv atribute discrete şi un atribut continuu: aplicare; predicţie)

■ • ***CMU, 2012 fall, E. Xing, A. Singh, HW1, pr. 1.1***

Până în luna septembrie a anului 2012, 800 de planete extrasolare (numite în continuare exoplanete) au fost identificate în galaxia noastră. Nişte navete spaţiale super-secrete au fost trimise pentru a survola toate aceste exoplanete, cu scopul de a stabili dacă ele sunt locuibile sau nu de către oameni. Evident, a trimite câte o navetă spaţială la fiecare dintre aceste exoplanete este extrem de costisitor. De aceea, în această problemă vă propunem să elaboraţi un arbore de decizie pentru a prezice dacă o exoplanetă este locuibilă sau nu, folosind doar trăsături / caracteristici (engl., features) observabile cu ajutorul telescoapelor terestre.

a. În tabelul de mai jos vi se dau anumite date în legătură cu toate cele 800 de planete survolate până acum. Trăsăturile observate cu ajutorul telescoapelor sunt *Size* ("Big" sau "Small") şi *Orbit* ("Near" sau "Far").

Fiecare linie din tabel indică valori ale acestor două trăsături, caracterul habitabil ("Yes" sau "No"), precum şi de câte ori a fost identificată fiecare combinaţie de valori [pentru cele trei trăsături]. De exemplu, au fost identificate 20 de planete mari ("Big"), care sunt situate pe orbite apropiate ("Near") de soarele / steaua lor şi sunt locuibile.

Size	Orbit	Habitable	Count
Big	Near	Yes	20
Big	Far	Yes	170
Small	Near	Yes	139
Small	Far	Yes	45
Big	Near	No	130
Big	Far	No	30
Small	Near	No	11
Small	Far	No	255

Elaboraţi şi desenaţi arborele de decizie învăţat de către algoritmul ID3 pe aceste date. (Folosiţi criteriul câştigului de informaţie; nu aplicaţi pruningul.) La fiecare nod din arbore, scrieţi numărul de planete locuibile şi respectiv nelocuibile din datele de antrenament care sunt asociate cu nodul respectiv.

b. Pentru doar 9 dintre aceste exoplanete, a fost măsurată o a treia trăsătură, *Temperature* (exprimată în grade Kelvin), după cum se arată în tabelul de mai jos.

Refaceţi toţi paşii de la punctul a, de data aceasta folosind toate cele trei trăsături de intrare. Pentru trăsătura *Temperature* (văzută ca atribut numeric cu valori continue), la fiecare iteraţie va trebui să faceţi maximizarea în raport cu toate pragurile adecvate pentru separare binară (engl., binary thresholding splits). Iată un *exemplu* de test pentru o astfel de separare binară: $T \leq 250$ vs. $T > 250$.

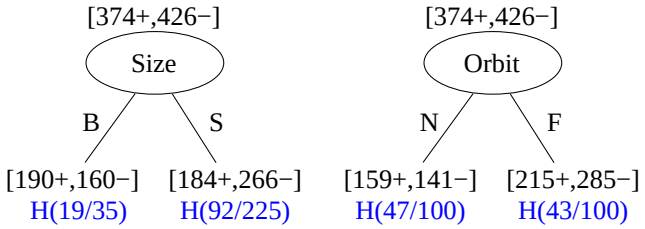
Size	Orbit	Temperature	Habitable
Big	Far	205	No
Big	Near	205	No
Big	Near	260	Yes
Big	Near	380	Yes
Small	Far	205	No
Small	Far	260	Yes
Small	Near	260	Yes
Small	Near	380	No
Small	Near	380	No

c. Conform arborelui de decizie pe care l-aţi obţinut la punctul b, cum va fi clasificată o planetă având trăsăturile (Big, Near, 280), locuibilă sau nelocuibilă?

***Indicaţie*: Este posibil să aveţi nevoie de următoarele valori pentru entropia ($H(p)$) unei variabile aleatoare Bernoulli de parametru p: $H(1/3) = 0.9182$, $H(2/5) = 0.9709$, $H(92/225) = 0.9759$, $H(43/100) = 0.9858$, $H(16/35) = 0.9946$, $H(47/100) = 0.9974$.**

Răspuns:

a. „Compaşii de decizie“ corespunzători nodului rădăcină sunt înfăţişaţi în desenul următor:



Sub fiecare nod descendent am notat entropia nodului respectiv, făcând referire la distribuţia Bernoulli şi dând (de fiecare dată) parametrului acestei distribuţii valoarea corespunzătoare. Pentru nodul descendent corespunzător lui Orbit = Near am ţinut cont şi de faptul că entropia distribuţiei Bernoulli, ca funcţie de parametru p, este simetrică faţă de valoarea $1/2$.[503]

Entropiile condiţionale medii corespunzătoare acestor doi „compaşi de decizie“ sunt:

$$\begin{aligned} H(Habitable|Size) &= \frac{35}{80} \cdot H\left(\frac{19}{35}\right) + \frac{45}{80} \cdot H\left(\frac{92}{225}\right) \\ &= \frac{35}{80} \cdot 0.9946 + \frac{45}{80} \cdot 0.9759 = 0.9841 \end{aligned}$$

[503] Este util să revedeţi explicaţiile date în *observaţia importantă* de la pagina 497.

$$H(Habitable|Orbit) = \frac{3}{8}\cdot H\left(\frac{47}{100}\right)+\frac{5}{8}\cdot H\left(\frac{43}{100}\right)$$
$$= \frac{3}{8}\cdot 0.9974+\frac{5}{8}\cdot 0.9858 = 0.9901$$

Suntem acum în măsură să desemnăm atributul care va fi plasat în nodul rădăcină al arborelui care va fi construit de algoritmul ID3 pe datele din enunţ: este atributul *Size*, întrucât $H(Habitable|Size) < H(Habitable|Orbit)$.

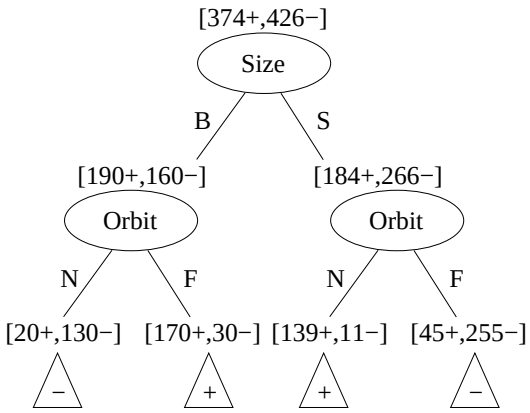
Doar cu titlu de informare, precizăm şi câştigurile de informaţie realizate de cele două atribute de intrare în raport cu atributul de ieşire:

$$IG(Habitable; Size) = H(Habitable) - H(Habitable|Size) = 0.9969 - 0.9841 = 0.0128$$

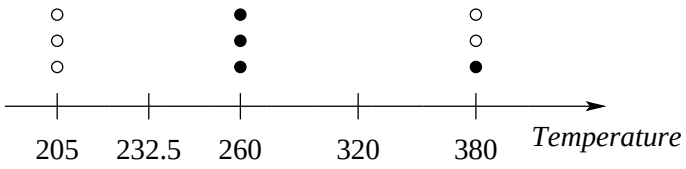
şi, similar,

$$IG(Habitable; Orbit) = 0.0067.$$

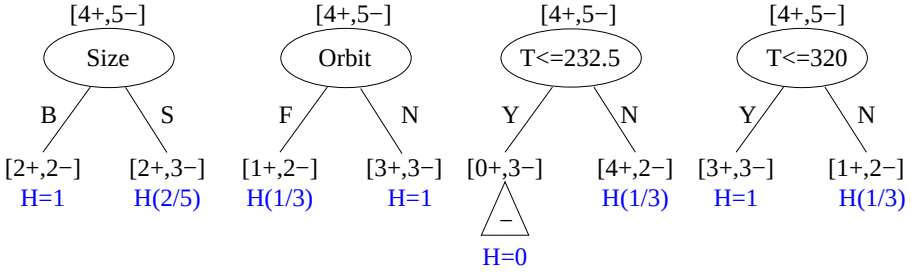
Întrucât nu avem decât două atribute de intrare, arborele de decizie va fi:



b. Pragurile de separare corespunzătoare atributului *Temperature* sunt determinate conform imaginii următoare:



***Nivelul 1* (rădăcina):**
„Compaşii de decizie“ corespunzători acestui nivel sunt:



Observând cu atenţie partiţiile formate,[504] vom constata că entropia condiţională medie pentru testul $Temperature \leq 232.5$ **(în raport cu atributul de ieşire** $Habitable$**) are o valoare mai mică decât fiecare dintre entropiile condiţionale medii** $H(Habitable|Orbit)$ **şi** $H(Habitable|Temperature \leq 320)$ **(care, de fapt, sunt egale între ele).[505]**

Relaţia dintre $H(Habitable|Temperature \leq 232.5)$ **şi** $H(Habitable|Size)$ **se determină cu ajutorul calculelor:**

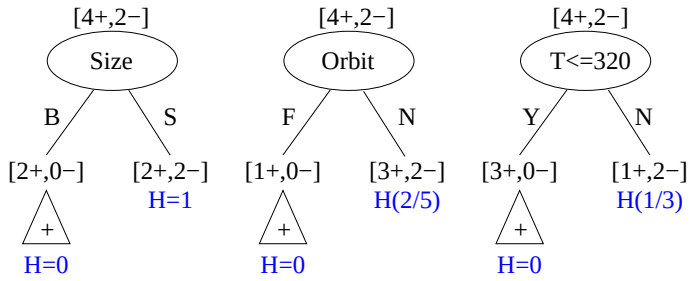
$$\begin{aligned} H(Habitable|Size) &= \frac{4}{9} + \frac{5}{9} \cdot H\left(\frac{2}{5}\right) = \frac{4}{9} + \frac{5}{9} \cdot 0.9709 = 0.9838 \\ H(Habitable|Temp \leq 232.5) &= \frac{2}{3} \cdot H\left(\frac{1}{3}\right) = \frac{2}{3} \cdot 0.9182 = 0.6121. \end{aligned}$$

Deşi nu mai este necesar, indicăm şi valorile câştigurilor de informaţie:

$$\begin{aligned} IG(Habitable; Size) &= 0.0072 \\ IG(Habitable; Orbit) &= 0.0183 \\ IG(Habitable; Temp \leq 232.5) &= 0.3788 \\ IG(Habitable; Temp \leq 320) &= 0.0183. \end{aligned}$$

Prin urmare, vom reţine pentru nivelul 1 testul $Temperature \leq 232.5$**.**

Nivelul 2 **(mai exact, aici ne limităm la datele cu** $Temperature > 232.5$**: „Compaşii de decizie“ corespunzători [completării] acestui nivel sunt:**



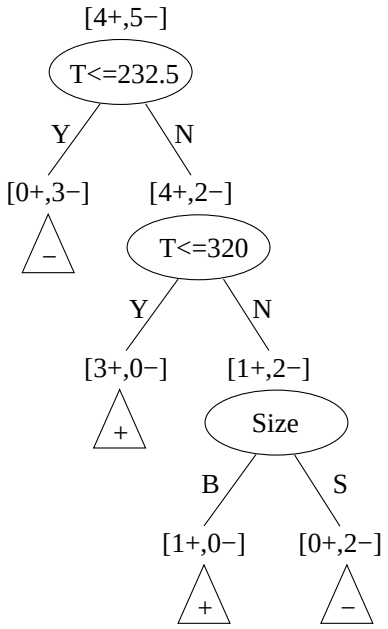
Este imediat că, pe aceste date, $H(Habitable|Temp \leq 320) < H(Habitable|Size)$ **şi, de asemenea,** $H(Habitable|Temp \leq 320) < H(Habitable|Orbit)$**.[506] Prin urmare, aici va fi ales testul** $Temp \leq 320$**.**

[504] Se compară două câte două entropiile condiţionale specifice, precum şi ponderile datelor respective în [raport cu] numărul total de instanţe asociate nodului părinte.

[505] Observaţi de exemplu că $H(Habitable|Orbit = F) > H(Habitable|Temperature \leq 232.5)$ şi $H(Habitable|Orbit = N > H(Habitable|Temperature > 232.5)$, iar ponderile corespunzătoare acestor entropii condiţionale specifice în calculul entropiilor condiţionale medii $H(Habitable|Orbit = N)$ şi $Temperature \leq 232.5$ sunt egale două câte două (şi anume, cu 3/9 şi respectiv 6/9).

[506] Mai exact, ar fi trebuit să scriem: $H(Habitable|Temp > 232.5, Temp \leq 320) < H(Habitable|Temp > 232.5, Size)$ şi respectiv $H(Habitable|Temp > 232.5, Temp \leq 320) < H(Habitable|Temp > 232.5, Orbit)$.

Nivelul 3 **(mai exact, aici ne limităm la datele cu** $Temperature > 320$**):**
Se poate observa că, pentru aceste date, atributul *Size* are putere discriminativă maximă. Aşadar, arborele de decizie final va fi cel din figura alăturată.



c. Conform arborelui de decizie obţinut la punctul anterior, o exoplanetă având trăsăturile (Big, Near, 280) va fi clasificată ca fiind locuibilă.

13. **(Extensii ale algoritmului ID3: cazul atributelor discrete cu număr mare de valori)**

CMU, 2008(?) spring, HW2, pr. 1

Se doreşte antrenarea unui arbore de decizie care să clasifice exemple cu două atribute de intrare X_1, X_2 **şi un atribut de ieşire** Y **care are valorile** 1 **şi** 2**. Primul atribut,** X_1**, este binar, pe când al doilea atribut,** X_2**, are 6 valori posibile** A, B, C, D, E, F**. Se dau următoarele 12 exemple de antrenament, câte 6 din fiecare clasă:**

$Y = 1$	$(1, A)$	$(0, E)$	$(1, B)$	$(1, B)$	$(1, F)$	$(0, D)$
$Y = 2$	$(0, A)$	$(0, C)$	$(1, E)$	$(0, F)$	$(0, B)$	$(1, D)$

Vă reamintim formula câştigului de informaţie la partajarea mulţimii de exemple S **în funcţie de valorile atributului** A**:**

$$Gain(S, A) = H(S) - H(S \mid A), \text{ cu } H(S \mid A) = \sum_{v \in valori(A)} \frac{|S_v|}{|S|} \cdot H(S_v),$$

unde $H(S)$ **este entropia mulţimii de exemple** S**, iar** $H(S \mid A)$ **este entropia condiţională medie a mulţimii** S **în raport cu atributul** A**, calculată aşa cum se vede mai sus, ca sumă ponderată a entropiilor submulţimilor lui** S **determinate de valorile atributului** A**.**

a. Determinaţi atributul ales în rădăcină folosind câştigul de informaţie. Folosiţi aproximarea $\log_2 3 = 1.585$**.**

b. Determinaţi atributul ales în nodul rădăcină folosind o măsură numită *gain ratio impurity*, adică alegeţi acel atribut care maximizează raportul

$$\frac{Gain(S, A)}{-\sum_v P(A = v) \cdot \log_2 P(A = v)} = \frac{H(S) - H(S \mid A)}{H(A)}.$$

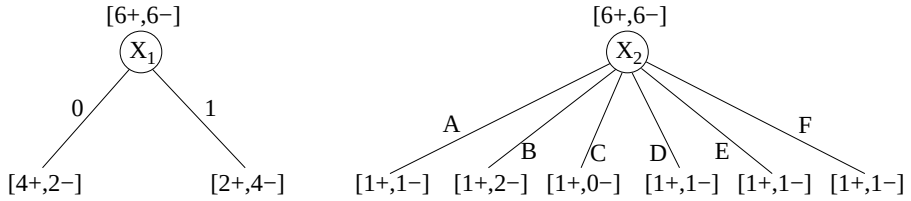
c. Având în vedere rezultatele de la punctele precedente, analizaţi utilitatea folosirii măsurii *gain ratio impurity* în cazurile în care atributele au numere diferite de valori posibile.

Răspuns:

a. Calculăm în primul rând entropia nodului rădăcină (sau a atributului de ieşire), care are 6 exemple pozitive şi 6 negative:

$$H(Y) = H[6+, 6-] = -\frac{6}{12}\log_2\frac{6}{12} - \frac{6}{12}\log_2\frac{6}{12} = 2\cdot\frac{1}{2}\log_2 2 = 1$$

În rădăcină poate fi ales fie atributul X_1, fie atributul X_2, obţinând următoarele împărţiri:



Dacă vom pune atributul X_1 în nodul rădăcină, câştigul de informaţie va fi:

$$\begin{aligned}
Gain(S, X_1) &= H(Y) - \frac{6}{12}H[4+, 2-] - \frac{6}{12}H[2+, 4-] = \\
&= 1 - \frac{6}{12}\left(-\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6}\right) - \frac{6}{12}\left(-\frac{2}{6}\log_2\frac{2}{6} - \frac{4}{6}\log_2\frac{4}{6}\right) \\
&= 1 - 2\cdot\frac{1}{2}\left(\frac{1}{3}\log_2 3 + \frac{2}{3}\log_2\frac{3}{2}\right) = 1 - \left(\log_2 3 - \frac{2}{3}\right) \\
&= \frac{5}{3} - \log_2 3 \approx 0.0817
\end{aligned}$$

Altminteri, punând atributul X_2 în nodul rădăcină, câştigul de informaţie este:

$$\begin{aligned}
Gain(S, X_2) &= H(Y) - 4\cdot\frac{2}{12}H[1+, 1-] - \frac{3}{12}H[1+, 2-] - \frac{1}{12}H[1+, 0-] \\
&= 1 - \frac{2}{3}\cdot 1 - \frac{3}{12}\left(-\frac{1}{3}\log_2\frac{1}{3} - \frac{2}{3}\log_2\frac{2}{3}\right) - \frac{1}{12}\cdot 0 \\
&= 1 - \frac{2}{3} - \frac{1}{4}\left(\frac{1}{3}\log_2 3 + \frac{2}{3}\log_2\frac{3}{2}\right) = 1 - \frac{2}{3} - \frac{1}{4}\left(\log_2 3 - \frac{2}{3}\right) \\
&= \frac{1}{2} - \frac{1}{4}\log_2 3 \approx 0.1037
\end{aligned}$$

Folosind drept criteriu de optimizat în fiecare nod câştigul de informaţie, în rădăcină vom pune atributul X_2, întrucât $Gain(S, X_1) < Gain(S, X_2)$.

b. Dacă vom pune atributul X_1 în nodul rădăcină, *gain ratio impurity* va fi:

$$\frac{Gain(S, X_1)}{-\sum_{i\in\{0,1\}} P(X_1 = i)\cdot\log_2 P(X_1 = i)} = \frac{Gain(S, X_1)}{-\frac{6}{12}\log_2\frac{6}{12} - \frac{6}{12}\log_2\frac{6}{12}} = \frac{Gain(S, X_1)}{1}$$

$$\approx 0.0817$$

În schimb, plasând atributul X_2 în nodul rădăcină, *gain ratio impurity* va fi:

$$\frac{Gain(S,X_2)}{-\sum_{j\in\{A,\ldots,F\}} P(X_2=j)\cdot\log_2 P(X_2=j)} = \frac{Gain(S,X_2)}{-4\cdot\frac{2}{12}\log_2\frac{2}{12}-\frac{3}{12}\log_2\frac{3}{12}-\frac{1}{12}\log_2\frac{1}{12}}$$

$$= \frac{Gain(S,X_2)}{\frac{2}{3}\log_2 6+\frac{1}{4}\log_2 4+\frac{1}{12}\log_2 12} = \frac{Gain(S,X_2)}{\frac{4}{3}+\frac{3}{4}\log_2 3} \approx 0.0411$$

Prin urmare, în nodul rădăcină vom pune atributul X_1, pentru care măsura *gain ratio impurity* are valoarea cea mai mare.

c. Câştigul de informaţie favorizează alegerea atributelor care au un număr mare de valori, indiferent dacă ele determină sau nu partiţionarea în mod semnificativ a datelor de antrenament. În schimb, măsura *gain ratio impurity* ia în considerare, prin cantitatea de la numitor (vedeţi definiţia), numărul de valori ale atributului respectiv, mai exact mărimea mulţimilor de instanţe asignate nodurilor-fii, care au fost generate ca urmare a alegerii respectivului atribut. Valoarea de la numitor va creşte odată cu numărul de noduri-fii, şi totodată cu numărul de noduri-fii care au asignate puţine exemple. Prin urmare, această măsură penalizează atributele cu multe valori, evitând favorizarea de care se face vinovat câştigul de informaţie într-o atare situaţie.

14. **(Extensii ale algoritmului ID3: cazul când se ia în considerare costul calculării atributelor; atribute continue; "decision stumps")**

CMU, 2008 spring, T. Mitchell, W. Cohen, HW1, pr. 1

Se dă următorul set de date. Acesta reprezintă fişele a 12 pacienţi ipotetici, ţinând cont de sex, vârstă (peste 60 ani sau nu), dacă suferă sau nu de diabet, dacă au pulsul mărit (sau nu) şi EKG-ul anormal (sau nu). Pacienţii sunt clasificaţi în final după cum prezintă (sau nu) aritmie.

Pacient	Sex	Peste60	Diabetic	Puls	EKG	AreAritmie
1	M	1	1	0	0	0
2	M	0	0	1	1	1
3	M	0	1	1	0	0
4	M	1	0	0	1	1
5	M	1	1	1	0	1
6	M	0	1	1	0	1
7	F	0	0	1	0	0
8	F	1	1	1	1	1
9	F	0	1	0	1	1
10	F	1	0	0	0	0
11	F	1	1	0	0	0
12	F	1	0	1	1	1

a. Calculaţi entropia condiţională specifică $H(\text{AreAritmie} \mid \text{Sex} = F)$.

b. Dacă pentru selectarea atributelor se foloseşte măsura $\frac{Gain^2(S,A)}{Cost(A)}$ în schimbul câştigului informational, care va fi atributul pus în nodul rădăcină? Se

consideră că *Cost*(Sex) = *Cost*(Peste60) = **1,** *Cost*(Diabetic) = **3,** *Cost*(Puls) = **2 şi** *Cost*(EKG) = **5.**

***Observaţie*: În calcule se vor utiliza următoarele aproximări:** $\log_2 3 = 1.585$, $\log_2 5 = 2.322$ **şi** $\log_2 7 = 2.807$.

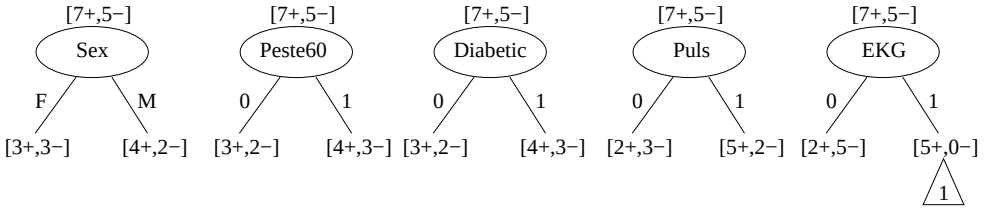
c. Să presupunem că, pentru un alt set de pacienţi, se cunoaşte vârsta lor exactă. Pentru exemplele pozitive, vârstele sunt: {40, 60, 62, 64, 70, 74, 75, 82}, iar pentru exemplele negative sunt: {33, 35, 42, 45, 49, 52, 58, 59, 80}. Să presupunem că toate celelalte atribute sunt predictori „slabi", prin urmare dorim ca arborele să aibă un singur nod, rădăcina, care să împartă exemplele cu valori continue ale atributului vârstă în două: *vârstă* $< k$ şi *vârstă* $\geq k$. Care va fi valoarea aleasă pentru k, bazat pe câştigul de informaţie?

Răspuns:

a. Entropia condiţională cerută este:

$$H(\text{AreAritmie} \mid \text{Sex} = F) = H[3+, 3-] = 1$$

b. În rădăcina arborelui de decizie se alege atributul A pentru care raportul $\frac{Gain^2(S,A)}{Cost(A)}$ este maxim. În cazul nostru, variantele pe care le avem sunt:



Se observă direct că *Gain*(S,EKG) **este mai mare decât** *Gain(S,A)* **pentru orice atribut** $A \neq$ EKG**. Însă şi** *Cost*(EKG) **este mai mare decât** *Cost(A)* **pentru orice** $A \neq$ EKG**. Aşadar, trebuie să facem calculele în detaliu.**

Entropia atributului de ieşire, AreAritmie**, este:**

$$\begin{aligned} H(\text{AreAritmie}) = H[7+, 5-] &= \frac{7}{12} \cdot \log_2 \frac{12}{7} + \frac{5}{12} \cdot \log_2 \frac{12}{5} \\ &= \frac{7}{12} \cdot \log_2 12 - \frac{7}{12} \cdot \log_2 7 + \frac{5}{12} \cdot \log_2 12 - \frac{5}{12} \cdot \log_2 5 \\ &= \log_2(3 \cdot 4) - \frac{7}{12} \cdot \log_2 7 - \frac{5}{12} \cdot \log_2 5 \\ &= \log_2 3 + \underbrace{\log_2 4}_{=2} - \frac{7}{12} \cdot \log_2 7 - \frac{5}{12} \cdot \log_2 5 \approx 0.98 \end{aligned}$$

Vom calcula câştigul de informaţie pentru fiecare din cele 5 atribute — se observă că pentru atributele Peste60 **şi** Diabetic**, câştigurile de informaţie sunt egale —, şi apoi vom face raportul $\frac{Gain^2}{Cost}$:**

Pentru atributul Sex**:**

$$\begin{aligned} Gain(S, \text{Sex}) &= H(\text{AreAritmie}) - \frac{6}{12} H[3+, 3-] - \frac{6}{12} H[4+, 2-] \\ &= H(\text{AreAritmie}) - \frac{1}{2} \cdot 1 - \frac{1}{2} \left(-\frac{4}{6} \log_2 \frac{4}{6} - \frac{2}{6} \log_2 \frac{2}{6} \right) \end{aligned}$$

$$= 0.98 - \frac{1}{2} - \frac{1}{2}\left(\log_2 3 - \frac{2}{3}\right) = 0.98 - \frac{1}{2} - \frac{1}{2}\log_2 3 + \frac{1}{3} \approx 0.02$$

Prin urmare, $\frac{Gain^2(S, \text{Sex})}{Cost(\text{Sex})} \approx \frac{0.02^2}{1} = 0.0004$**.**

Pentru atributele Peste60 **şi** Diabetic**:**

$$\begin{aligned} Gain(S, \text{Peste60}) &= Gain(S, \text{Diabetic}) \\ &= H(\text{AreAritmie}) - \frac{5}{12}H[3+, 2-] - \frac{7}{12}H[4+, 3-] \\ &= 0.98 - \frac{5}{12}\left(\frac{3}{5}\log_2\frac{5}{3} + \frac{2}{5}\log_2\frac{5}{2}\right) - \frac{7}{12}\left(\frac{4}{7}\log_2\frac{7}{4} + \frac{3}{7}\log_2\frac{7}{3}\right) \\ &= 0.98 - \frac{5}{12}\left(\log_2 5 - \frac{3}{5}\log_2 3 - \frac{2}{5}\cdot 1\right) - \frac{7}{12}\left(\log_2 7 - \frac{4}{7}\cdot 2 - \frac{3}{7}\log_2 3\right) \\ &= 0.98 - \frac{5}{12}\log_2 5 - \frac{7}{12}\log_2 7 + \frac{1}{2}\log_2 3 + \frac{5}{6} \approx 0.0009 \end{aligned}$$

Deci $\frac{Gain^2(S, \text{Peste60})}{Cost(\text{Peste60})} \approx \frac{0.0009^2}{1} = 81 \cdot 10^{-8}$ **şi** $\frac{Gain^2(S, \text{Diabetic})}{Cost(\text{Diabetic})} \approx \frac{0.0009^2}{3} = 27 \cdot 10^{-8}$**, ambele fiind nişte valori foarte mici.**

Pentru atributul Puls**:**

$$\begin{aligned} Gain(S, \text{Puls}) &= H(\text{AreAritmie}) - \frac{5}{12}H[2+, 3-] - \frac{7}{12}H[5+, 2-] \\ &= 0.98 - \frac{5}{12}\left(\frac{2}{5}\log_2\frac{5}{2} + \frac{3}{5}\log_2\frac{5}{3}\right) - \frac{7}{12}\left(\frac{5}{7}\log_2\frac{7}{5} + \frac{2}{7}\log_2\frac{7}{2}\right) \\ &= 0.98 - \frac{7}{12}\log_2 7 + \frac{1}{4}\log_2 3 + \frac{1}{3} \approx 0.072 \end{aligned}$$

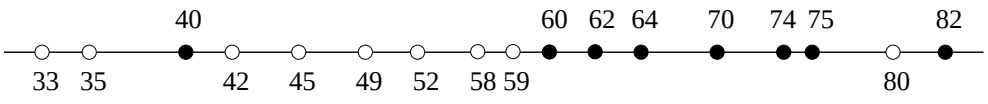
Prin urmare, $\frac{Gain^2(S, \text{Puls})}{Cost(\text{Puls})} \approx \frac{0.072^2}{2} = 0.002592$**.**

Pentru atributul EKG**:**

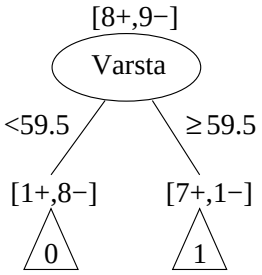
$$\begin{aligned} Gain(S, \text{EKG}) &= H(\text{AreAritmie}) - \frac{7}{12}H[2+, 5-] - \frac{5}{12}H[5+, 0-] \\ &= H(\text{AreAritmie}) - \frac{7}{12}\left(\frac{2}{7}\log_2\frac{7}{2} + \frac{5}{7}\log_2\frac{7}{5}\right) - \frac{5}{12}\cdot 0 \\ &= 0.98 + \frac{5}{12}\log_2 5 - \frac{7}{12}\log_2 7 + \frac{1}{6} \approx 0.476 \end{aligned}$$

Deci $\frac{Gain^2(S, \text{EKG})}{Cost(\text{EKG})} \approx \frac{0.476^2}{5} = 0.0453152$**. Este evident că aceasta este cea mai mare valoare, de aceea în nodul rădăcină va fi ales atributul *EKG*, deşi costul acestuia este cel mai mare.**

c. Putem reprezenta exemplele astfel:



Se constată uşor că alegerea cea mai bună este $k = 59.5$, arborele de decizie fiind cel din figura alăturată.



15. **(Alte criterii posibile pentru selecţia atributelor în ID3: Gini impurity şi Misclassification impurity)**

prelucrare de Liviu Ciortuz, după
■ • ***CMU, 2003 fall, T. Mitchell, A. Moore, HW1, pr. 4***

Entropia **este o mărime care cuantifică gradul de „impuritate“ (engl., impurity) al unui set de instanţe în raport cu etichetele asignate.[507] Algoritmul ID3 foloseşte entropia drept criteriu de partiţionare (engl., splitting criterion), calculând *câştigul de informaţie* pentru a decide care este atributul care trebuie testat în nodul curent. Există, însă, şi alte măsuri de impuritate care pot fi folosite, de asemenea, drept criterii de partiţionare. În această problemă, vom investiga două astfel de măsuri.**

Presupunem că nodul curent (n) din arborele de decizie aflat în curs de elaborare are asignate instanţe din k clase: $c_1, c_2, \ldots, c_k$. Definim[508]

Gini Impurity: $i(n) = 1 - \sum_{i=1}^{k} P^2(c_i)$

şi

Misclassification Impurity: $i(n) = 1 - \max_{i=1}^{k} P(c_i)$,

unde am notat cu $P(c_i)$ probabilitatea [sau: frecvenţa de apariţie a instanţelor aparţinând] clasei c_i în ansamblul instanţelor asignate la nodul curent.

a. Presupunem $k = 2$. Aşadar, în acest caz nodul n are două clase: c_1 şi c_2. Desenaţi un grafic în care cele trei măsuri de impuritate — *Entropia*, *Gini Impurity* şi *Misclassification Impurity* — sunt reprezentate în funcţie de $P(c_1)$.

b. Acum putem da definiţia unui nou criteriu de partiţionare, bazat pe măsurile de impuritate *Gini* şi *Misclassification*. În literatura de specialitate, acest nou criteriu este denumit uneori *Drop-of-Impurity* (pentru care propunem ca traducere în limba română termenul de *diminuarea impurităţii*). El reprezintă diferenţa dintre impuritatea nodului curent pe de o parte şi suma ponderată a impurităţilor fiilor pe de altă parte. În cazul partiţionării atributelor binare, definim *Drop-of-Impurity* ca fiind:

$$\Delta i(n) = i(n) - P(n_l)\, i(n_l) - P(n_r)\, i(n_r),$$

[507] În cazul clasificării binare, unii autori numesc partiţiile de forma $[n+, 0-]$ sau $[0+, m-]$ *partiţii pure*. Prin opoziţie, *partiţiile impure* sunt de forma $[n+, m-]$, cu $n \neq 0$ şi $m \neq 0$. Însă, în contextul termenului de *impuritate* de aici, condiţia $n, m \neq 0$ este ignorată.

[508] Indexul / coeficientul / raportul Gini este denumit după numele creatorului său, Corrado Gini, sociolog şi statistician italian (1884-1965).

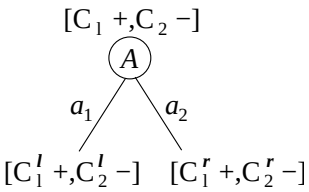
unde n_l şi n_r reprezintă fiul-stânga şi, respectiv, fiul-dreapta, care au fost derivaţi din nodul n după partiţionare, iar $P(n_l)$ şi respectiv $P(n_r)$ reprezintă probabilităţile „de selecţie“ pentru nodul n_l şi respectiv nodul n_r.

Folosind mai întâi *Gini Impurity* şi apoi *Misclassification Impurity*, calculaţi *Drop-of-Impurity* pentru următorul set de instanţe asignate nodului pentru care se testează atributul A luând valorile a_1 şi a_2. Am notat cu C variabila de ieşire (desemnând clasa) şi cu c_1 şi c_2 cele două valori ale ei.

A	a_1	a_1	a_1	a_2	a_2	a_2
C	c_1	c_1	c_2	c_2	c_2	c_2

c. Se poate crea un set de date de antrenament (sau: se poate modifica setul de date de mai sus) astfel încât, pe noul set, *Drop-of-Impurity* bazat pe *Misclassification* să fie 0 însă bazat pe *Entropy* şi, respectiv, *Gini* să fie diferit de 0?

***Sugestie*: Puteţi folosi următoarea *proprietate*, care este uşor de demonstrat:**
Dacă într-un set de date avem C_1 instanţe din clasa (sau: cu eticheta) c_1 şi C_2 instanţe din clasa c_2, cu $C_1 < C_2$, iar după partiţionarea în funcţie de valorile atributului A această relaţie se păstrează, adică $C_1^l < C_2^l$ şi $C_1^r < C_2^r$ (evident, cu $C_1 = C_1^l + C_1^r$ şi $C_2 = C_2^l + C_2^r$), unde l şi r desemnează nodul-fiu stâng şi respectiv nodul-fiu drept, atunci *Drop-of-Impurity* pentru *Misclassification* va fi 0. Însă, în aceleaşi condiţii, *Drop-of-Impurity* bazat pe *Gini* sau pe *Entropy* va avea, în general, valori nenule.



(Evident, *proprietatea* de mai sus se menţine dacă în locul relaţiei $<$ vom considera peste tot relaţia $>$.)

Răspuns:

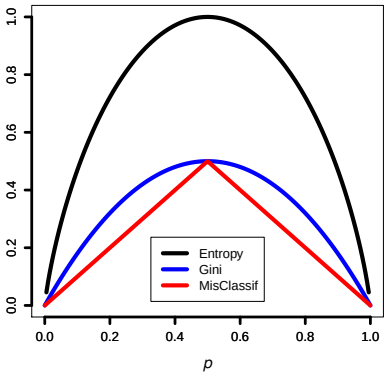
a. Vom scrie mai întâi expresiile celor trei funcţii, iar apoi vom trasa graficele lor:

$$\begin{aligned} \mathit{Entropy}(p) &= -p\log_2 p - (1-p)\log_2(1-p) \\ \mathit{Gini}(p) &= 1 - p^2 - (1-p)^2 = 2p(1-p) \end{aligned}$$

$$\mathit{MisClassif}(p) =$$

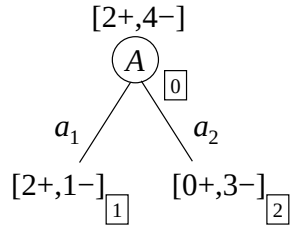
$$= \begin{cases} 1-(1-p), & \textbf{pentru } p \in [0;1/2) \\ 1-p, & \textbf{pentru } p \in [1/2;1] \end{cases}$$

$$= \begin{cases} p, & \textbf{pentru } p \in [0;1/2) \\ 1-p, & \textbf{pentru } p \in [1/2;1] \end{cases}$$



Se observă că toate cele trei funcţii iau valori în intervalul $[0;1]$, sunt simetrice în raport cu punctul $p = 1/2$, sunt strict crescătoare pe intervalul $[0;1/2]$ şi strict descrescătoare pe intervalul $[1/2;1]$, maximul fiecăreia dintre ele fiind obţinut pentru $p = 1/2$.

b. Partiţionarea datelor de antrenament în funcţie de valorile atributului A se face aşa cum se arată în figura alăturată. (În această figură şi, de asemenea, în calculele de mai jos, pentru convenienţă / simplitate, am asociat semnul + etichetei c_1 şi semnul − etichetei c_2.)



Aplicând formula $\Delta i(n) = i(n) - P(n_l)\, i(n_l) - P(n_r)\, i(n_r)$, vom obţine pentru *Drop-of-Impurity* următoarele valori:

Gini: $p = 2/6 = 1/3 \Rightarrow$

$$\left.\begin{array}{rcl} i(0) & = & 2 \cdot \frac{1}{3}(1 - \frac{1}{3}) = \frac{2}{3} \cdot \frac{2}{3} = \frac{4}{9} \\ i(1) & = & 2 \cdot \frac{2}{3}(1 - \frac{2}{3}) = \frac{4}{3} \cdot \frac{1}{3} = \frac{4}{9} \\ i(2) & = & 0 \end{array}\right\} \Rightarrow \Delta\, i(0) = \frac{4}{9} - \frac{3}{6} \cdot \frac{4}{9} = \frac{4}{9} - \frac{2}{9} = \frac{2}{9}.$$

Misclassification: $p = 1/3 < 1/2 \Rightarrow$

$$\left.\begin{array}{rcl} i(0) & = & p = \frac{1}{3} \\ i(1) & = & 1 - \frac{2}{3} = \frac{1}{3} \\ i(2) & = & 0 \end{array}\right\} \Rightarrow \Delta\, i(0) = \frac{1}{3} - \frac{1}{2} \cdot \frac{1}{3} = \frac{1}{6}.$$

c. Într-adevăr, putem justifica uşor *proprietatea* din enunţ:

$$\begin{aligned} \Delta i(n) &= \frac{C_1}{C_1 + C_2} - \left(\frac{\cancel{C_1^l + C_2^l}}{C_1 + C_2} \cdot \frac{C_1^l}{\cancel{C_1^l + C_2^l}} + \frac{\cancel{C_1^r + C_2^r}}{C_1 + C_2} \cdot \frac{C_1^r}{\cancel{C_1^r + C_2^r}} \right) \\ &= \frac{C_1}{C_1 + C_2} - \frac{C_1^l + C_1^r}{C_1 + C_2} = \frac{C_1}{C_1 + C_2} - \frac{C_1}{C_1 + C_2} = 0. \end{aligned}$$

Dacă în setul de date din enunţ una dintre instanţele (a_1, c_1) se modifică în (a_1, c_2) şi se adaugă o instanţă (a_2, c_1), atunci *Drop-of-Impurity* va avea următoarele valori:

Entropy: $\Delta\, i(0) = H[2+, 5-] - \left(\frac{3}{7} H[1+, 2-] + \frac{4}{7} H[1+, 3-] \right) = 0.006 \neq 0$;

Gini: $2 \left\{ \frac{2}{7} \left(1 - \frac{2}{7} \right) - \left[\frac{3}{7} \cdot \frac{1}{3} \left(1 - \frac{1}{3} \right) + \frac{4}{7} \cdot \frac{1}{4} \left(1 - \frac{1}{4} \right) \right] \right\} = 2 \left\{ \frac{10}{49} - \left[\frac{2}{21} + \frac{3}{28} \right] \right\}$

$= 2 \left(\frac{10}{49} - \frac{17}{84} \right) \neq 0$;

Misclassification: $\Delta\, i(0) = \frac{2}{7} - \left(\frac{3}{7} \cdot \frac{1}{3} + \frac{4}{7} \cdot \frac{1}{4} \right) = 0.$

16. (Algoritmul ID3 ca metodă de învăţare de tip "eager": posibilitatea suplimentării datelor de antrenament)

CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, HW1, pr. 3.4

Presupunem că, pornind de la un set de date de antrenament D, obţinem un arbore de decizie ID3, notat cu T. Ulterior, cineva ne mai dă un set suplimentar de date de antrenament, D'. Putem proceda într-unul din următoarele două moduri:

– Putem rula din nou ID3, de această dată pe datele de antrenament $D \cup D'$, obţinând arborele T_1. (Dezavantaj: dacă $|D|$ este foarte mare, această procedură poate fi costisitoare ca timp.)

– Putem extinde T, arborele ID3 obţinut pe mulţimea de antrenament D, ţinând cont de datele D'. Arborele nou, T_2, ar putea să nu fie la fel de bun ca T_1 (vedeţi cazul de mai sus), dar el este consistent cu datele din $D \cup D'$ în cazul în care aceste mulţimi de antrenament nu conţin zgomote. În special dacă $|D'|$ este mic, această metodă este acceptabilă din punct de vedere practic.

Propuneţi o procedură pentru obţinerea efectivă a arborelui T_2.

Răspuns:

Instanţele din D' se ataşează nodurilor frunză ale arborelui T, conform procedurii de clasificare de la arbori de decizie. Pentru fiecare dintre aceste noduri frunză, dacă instanţele ataşate nodului respectiv nu sunt toate etichetate identic, se aplică algoritmul ID3 (folosind mulţimea de atribute neutilizate pe drumul care uneşte nodul rădăcină al arborelui cu acest nod frunză). În acest mod se obţine un alt arbore de decizie T_2 care este consistent cu datele din $D \cup D'$.

17. **(Reducerea caracterului "greedy" al algoritmului ID3 prin calcularea câştigului de informaţie în maniera "2-step look-ahead")**

CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW4, pr. 1.2-6

Învăţarea automată a arborilor de decizie depinde mult de utilizarea unui mecanism "greedy" de selecţie a atributelor.

a. Dacă un set de date are A atribute booleene, calculaţi (ca expresie în funcţie de A) numărul total de apeluri la funcţia care calculează câştigul de informaţie pentru elaborarea întregului arbore de decizie.

Pentru *simplitate*, se va presupune că toate atributele sunt necesare pentru clasificarea unei instanţe, iar setul de date de antrenament conţine toate instanţele posibile (adică toate combinaţiile posibile de perechi atribut-valoare, inclusiv pentru atributul de ieşire).

b. Este posibil să îmbunătăţim algoritmul ID3, făcându-l să se comporte mai puţin "greedy", prin explorarea / prospectarea în avans (engl., look-ahead) a spaţiului de căutare. La o explorare cu 2 paşi înainte (engl., 2-step look-ahead), calculul câştigului de informaţie pentru un atribut a_i pe mulţimea de instanţe D va fi făcut cu ajutorul formulei

$$IG_{2-\text{step}}(D, a_i) = \max_{a_l, a_r} \left\{ \frac{n_l}{n_l + n_r} IG(D_l, a_l) + \frac{n_r}{n_l + n_r} IG(D_r, a_r) \right\},$$

unde

– a_l şi a_r sunt atributele din nodurile descendente din nodul marcat cu atributul a_i,

– D_l **şi** D_r **sunt seturile de instanţe asignate nodurilor descendente din** a_i**, iar** n_l **şi** n_r **reprezintă cardinalul mulţimii** D_l **şi respectiv** D_r**;**

– $IG(D_l, a_l)$ **este câştigul de informaţie calculat (în sens clasic) pentru atributul** a_l **pe setul** D_l**; similar,** $IG(D_r, a_r)$ **este câştigul de informaţie pentru atributul** a_r **pe setul** D_r**.**

b1. Explicaţi pe scurt de ce sunt necesari factorii $\frac{n_l}{n_l+n_r}$ **şi** $\frac{n_r}{n_l+n_r}$ **în formula de mai sus.**

b2. Dacă se folosesc A **atribute booleene, câte apeluri la funcţia** $IG(\ldots,\ldots)$ **sunt necesare pentru a stabili atributul din nodul rădăcină?**

b3. Vom evalua acum cât de costisitoare este această explorare în avans a spaţiului de căutare.
Dacă $A = 10$ **şi se face aceeaşi presupoziţie ca la punctul** a**, calculaţi câte niveluri complete din arborele ID3 standard se pot calcula cu acelaşi efort de calcul — exprimat ca număr de apeluri la funcţia** IG **— ca la stabilirea atributului rădăcină în varianta *2-step look-ahead*.**

b4. Metoda de învăţare a arborilor de decizie folosind *2-step look-ahead* crează o clasă de modele / ipoteze mai largă decât algoritmul ID3 simplu? Altfel spus, putem să calculăm în acest mod funcţii de clasificare pe care nu le putem reprezenta cu arborii de decizie standard?

Răspuns:

a. Datorită presupunerii făcute pentru *simplitate*, arborele de decizie final va fi un arbore binar complet cu $A+1$ **niveluri (incluzând şi nodurile de decizie), notate în mod convenţional de la** 0 **la** A**. Pe fiecare nivel** $i = \overline{0, A-1}$ **al arborelui binar se găsesc** 2^i **noduri. În fiecare dintre aceste noduri poate fi ales un atribut din cele** $A - i$ **rămase disponibile. Deci pentru determinarea nivelului** i **se fac** $2^i \cdot (A - i)$ **apeluri ale funcţiei** IG**.**

Desigur, pe penultimul nivel, $A-1$**, nu mai există decât un singur atribut rămas disponibil, prin urmare nu este necesar să se calculeze câştigul de informaţie. Aşadar, numărul total de apeluri ale funcţiei** IG **pentru elaborarea întregului arbore de decizie este:**

$$N_{IG} = A + 2(A-1) + 4(A-2) + \ldots + 2^{A-2}(A-(A-2)) = \sum_{i=0}^{A-2} 2^i(A-i)$$

b1. n_l **şi** n_r **reprezintă numărul de instanţe asignate nodurilor descendente din** a_i**, deci factorii** $\frac{n_l}{n_l+n_r}$ **şi** $\frac{n_r}{n_l+n_r}$ **reprezintă ponderile acestor sub-mulţimi în raport cu reuniunea lor. Cei doi factori vor pondera corespunzător câştigurile de informaţie de pe cele două ramuri din arborele de decizie. Dacă nu facem astfel de ponderări, este posibil să avem un câştig de informaţie mare pe o mulţime mică sau invers. În consecinţă, simpla însumare a celor două câştiguri de informaţie nu ar emula în mod veridic câştigul de informaţie pe întreg ansamblul lui** D**.**

b2. Pentru stabilirea atributului din nodul rădăcină făcând o explorare cu 2 paşi înainte, se calculează pentru fiecare dintre cele A **atribute câştigurile de informaţie corespunzătoare celor 2 descendenţi, adică:**

$$N_{IG_{2-\text{step}}}(\text{rădăcină}) = A \cdot 2(A-1) = 2A^2 - 2A$$

b3. Dacă $A = 10$, atunci pentru stabilirea atributului rădăcină în varianta *2-step look-ahead* se apelează funcţia IG de $N_{IG_{2-\text{step}}}(\text{rădăcină}) = 200 - 20 = 180$ de ori. Trebuie determinat numărul x de niveluri complete din arborele ID3 standard care se pot calcula folosind maxim 180 de apeluri ale funcţiei IG, adică argmax_x astfel încât $N_{IG}(x) = \sum_{i=0}^{x-1} 2^i(10-i) \leq 180$.

$$
\begin{array}{lcrrl}
x=1 & \Rightarrow & N_{IG}(1) = & 2^0(10-0) = & 10 \\
x=2 & \Rightarrow & N_{IG}(2) = & 10 + 2^1(10-1) = & 28 \\
x=3 & \Rightarrow & N_{IG}(3) = & 28 + 2^2(10-2) = & 60 \\
x=4 & \Rightarrow & N_{IG}(4) = & 60 + 2^3(10-3) = & 116 \\
x=5 & \Rightarrow & N_{IG}(5) = & 116 + 2^4(10-4) = & 212 \quad > 180
\end{array}
$$

Aşadar, se pot calcula 4 niveluri complete din arborele ID3 standard cu acelaşi efort de calcul ca la stabilirea atributului din rădăcină în varianta (mai puţin "greedy") a algoritmului ID3 cu *2-step look-ahead*.

b4. Nu. Clasa de modele / ipoteze pe care lucrează algoritmul ID3 cu *2-step look-ahead* este aceeaşi ca la algoritmul ID3 standard. Însă este foarte posibil ca arborele de decizie construit să fie mai bun dacă se face cătarea în maniera *2-step look-ahead*.

18. **(Îmbunătăţirea algoritmului ID3, folosind IG cu "2-step look-ahead")**

Liviu Ciortuz, folosind date de la CMU, 2008 fall, Eric Xing, HW2, pr. 3

Se consideră variabilele booleene X_1, X_2 şi X_3, precum şi clasificarea $Y = \{0, 1\}$. Fie setul de date de antrenament din tabelul de mai jos.

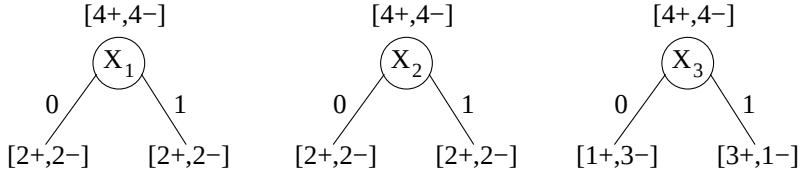
	X_1	X_2	X_3	Y
1	0	0	0	0
2	0	0	1	0
3	0	1	0	1
4	0	1	1	1
5	1	0	1	1
6	1	0	1	1
7	1	1	0	0
8	1	1	0	0

a. Elaboraţi arborele de decizie cu algoritmul ID3 standard.

b. Elaboraţi arborele de decizie cu algoritmul ID3 folosind câştigul de informaţie cu *2-step look-ahead*, aşa cum este acesta definit în problema 17.

Răspuns:

a. Pentru nodul rădăcină vom alege unul dintre atributele X_1, X_2 şi X_3. Corespunzător, vom obţine următoarele partajări ale setului de date de antrenament:

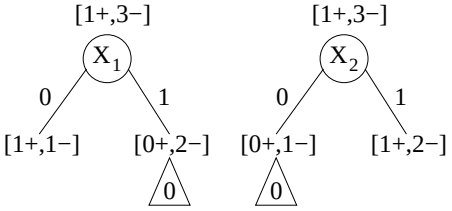


În cele ce urmează vom scrie câştigul de informaţie (IG) omiţând primul argument, întrucât se consideră cunoscut din context. Vom calcula câştigul de informaţie corespunzător fiecărui atribut:

$$IG(X_1) = IG(X_2) = H[4+,4-] - \left(\frac{4}{8}H[2+,2-] + \frac{4}{8}H[2+,2-]\right) = 1 - \left(\frac{1}{2} + \frac{1}{2}\right) = 0$$
$$IG(X_3) = H[4+,4-] - \left(\frac{4}{8}H[1+,3-] + \frac{4}{8}H[3+,1-]\right)$$

Însă $H[1+,3-] = H[3+,1-]$, iar valoarea corespunzătoare este sub-unitară, deci $IG(X_3) > 0$. Aşadar, în nodul rădăcină se alege atributul X_3.

Pentru nodul corespunzător ramurii $X_3 = 0$ se poate alege dintre atributele care au mai rămas, adică X_1 sau X_2.
Se calculează câştigurile de informaţie corespunzătoare:

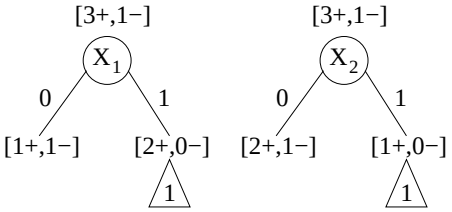


$$IG(X_1 \mid X_3 = 0) = H[1+,3-] - \left(\frac{2}{4}H[1+,1-] + \frac{2}{4}H[0+,2-]\right)$$
$$= 2 - \frac{3}{4}\log_2 3 - \left(\frac{1}{2} + 0\right) = \frac{3}{2} - \frac{3}{4}\log_2 3 = 0.311$$
$$IG(X_2 \mid X_3 = 0) = H[1+,3-] - \left(\frac{1}{4}H[0+,1-] + \frac{3}{4}H[1+,2-]\right)$$
$$= 2 - \frac{3}{4}\log_2 3 - \left(0 + \frac{3}{4}H[1+,2-]\right)$$
$$= 2 - \frac{3}{4}\log_2 3 - \frac{3}{4}\left(\frac{1}{3}\log_2 3 + \frac{2}{3}\log_2 \frac{3}{2}\right)$$
$$= 2 - \frac{3}{4}\log_2 3 - \frac{3}{4}\left(\log_2 3 - \frac{2}{3}\right) = \frac{5}{2} - \frac{3}{2}\log_2 3 = 0.122$$

Cum $IG(X_1 \mid X_3 = 0) > IG(X_2 \mid X_3 = 0)$, se alege atributul X_1.

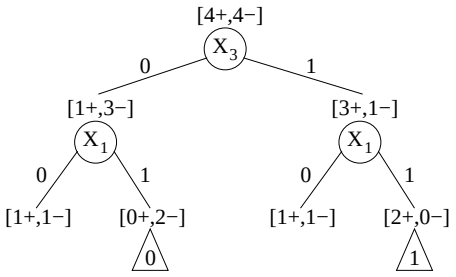
În cele de mai sus, s-au folosit notaţiile în maniera condiţională $IG(X_1 \mid X_3 = 0)$ şi $IG(X_2 \mid X_3 = 0)$ doar pentru a identifica în mod neambiguu care este nodul din arbore pentru care se calculează câştigul de informaţie respectiv.

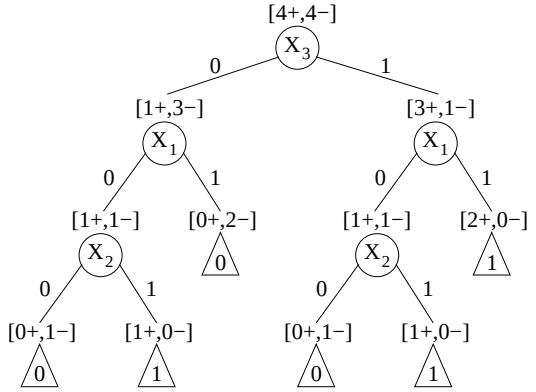
Pentru nodul corespunzător ramurii $X_3 = 1$ ce porneşte din nodul rădăcină se poate alege din nou unul dintre atributele X_1 şi X_2.
Pentru aceasta, se compară câştigurile de informaţie corespunzătoare. Se observă că acestea sunt egale cu cele din cazul precedent. Aşadar,



$$IG(X_1 \mid X_3 = 1) = IG(X_1 \mid X_3 = 0) = \frac{3}{2} - \frac{3}{4}\log_2 3 = 0.311$$
$$IG(X_2 \mid X_3 = 1) = IG(X_2 \mid X_3 = 0) = \frac{5}{2} - \frac{3}{2}\log_2 3 = 0.122$$

Prin urmare, se alege tot atributul X_1. Arborele de decizie construit până în acest moment este cel reprezentat mai jos în partea stângă:





În cele două noduri care au rămas, nu poate fi ales decât atributul X_2. Deci arborele de decizie complet construit de algoritmul ID3 este cel reprezentat mai sus în partea dreaptă.

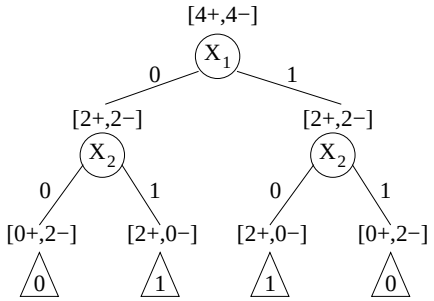
b. Dacă se foloseşte câştigul de informaţie cu *2-step look-ahead* pentru algoritmul ID3, atunci la alegerea fiecărui nod se iau în considerare [şi] toate combinaţiile posibile de noduri de pe nivelul următor, utilizându-se formula:

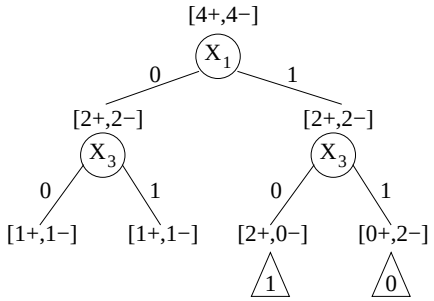
$$IG_{2-\text{step}}(D, a_i) = \max_{a_l, a_r} \left\{ \frac{n_l}{n_l + n_r} IG(D_l, a_l) + \frac{n_r}{n_l + n_r} IG(D_r, a_r) \right\}$$

Pentru nodul rădăcină se poate alege, la fel ca la punctul precedent, unul dintre atributele X_1, X_2 şi X_3. Aşadar, câştigul de informaţie corespunzător atributului X_1 va fi calculat astfel:

$$IG_{2-\text{step}}(X_1) = \max \begin{cases} \frac{4}{8} IG(X_2 \mid X_1 = 0) + \frac{4}{8} IG(X_2 \mid X_1 = 1) \\ \frac{4}{8} IG(X_2 \mid X_1 = 0) + \frac{4}{8} IG(X_3 \mid X_1 = 1) \\ \frac{4}{8} IG(X_3 \mid X_1 = 0) + \frac{4}{8} IG(X_2 \mid X_1 = 1) \\ \frac{4}{8} IG(X_3 \mid X_1 = 0) + \frac{4}{8} IG(X_3 \mid X_1 = 1) \end{cases}$$

Pentru a determina această valoare, va trebui să calculăm cele 4 câştiguri de informaţie (în sens clasic) implicate în formulă, şi este util să reprezentăm două din cele 4 situaţii posibile:





$$\begin{aligned} IG(X_2 \mid X_1 = 0) &= IG(X_2 \mid X_1 = 1) = IG(X_3 \mid X_1 = 1) \\ &= H[2+, 2-] - \left(\frac{1}{2}H[0+, 2-] + \frac{1}{2}H[2+, 0-]\right) = 1 - 0 = 1 \\ IG(X_3 \mid X_1 = 0) &= H[2+, 2-] - \left(\frac{1}{2}H[1+, 1-] + \frac{1}{2}H[1+, 1-]\right) = 1 - \left(\frac{1}{2} + \frac{1}{2}\right) = 0 \end{aligned}$$

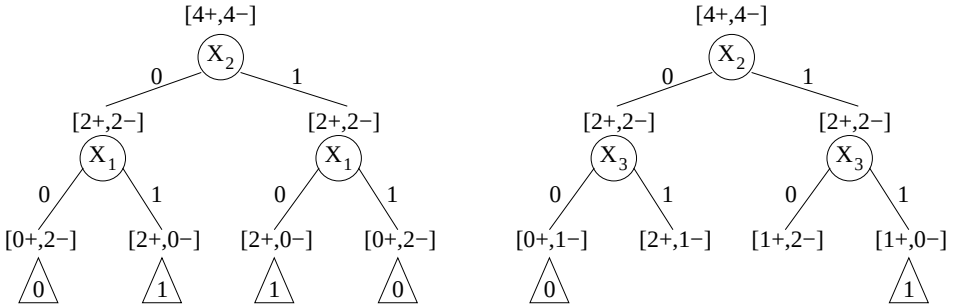
Prin urmare,

$$IG_{2-\text{step}}(X_1) = \max \left\{ \begin{array}{l} \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 1 \\ \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 1 \\ \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 1 \\ \frac{1}{2} \cdot 0 + \frac{1}{2} \cdot 1 \end{array} \right. = \max\left\{1, 1, \frac{1}{2}, \frac{1}{2}\right\} = 1.$$

Câştigul de informaţie corespunzător atributului X_2 plasat în nodul rădăcină este:

$$IG_{2-\text{step}}(X_2) = \max \left\{ \begin{array}{l} \frac{4}{8}IG(X_1 \mid X_2 = 0) + \frac{4}{8}IG(X_1 \mid X_2 = 1) \\ \frac{4}{8}IG(X_1 \mid X_2 = 0) + \frac{4}{8}IG(X_3 \mid X_2 = 1) \\ \frac{4}{8}IG(X_3 \mid X_2 = 0) + \frac{4}{8}IG(X_1 \mid X_2 = 1) \\ \frac{4}{8}IG(X_3 \mid X_2 = 0) + \frac{4}{8}IG(X_3 \mid X_2 = 1) \end{array} \right.$$

Vom reprezenta din nou două dintre situaţiile posibile:



$$\begin{aligned} IG(X_1 \mid X_2 = 0) &= IG(X_1 \mid X_2 = 1) = H[2+, 2-] - 0 = 1 \\ IG(X_3 \mid X_2 = 0) &= IG(X_3 \mid X_2 = 1) = H[2+, 2-] - \left(\frac{1}{4}H[0+, 1-] + \frac{3}{4}H[2+, 1-]\right) = \\ &= 1 - \frac{3}{4}\left(\log_2 3 - \frac{2}{3}\right) = \frac{3}{2} - \frac{3}{4}\log_2 3 = 0.311 \end{aligned}$$

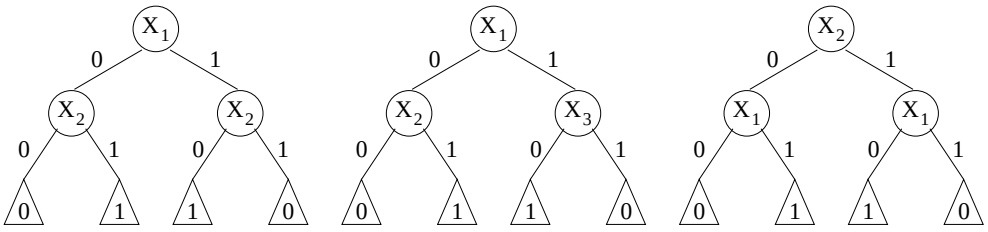
Prin urmare,

$$IG_{2-\text{step}}(X_2) = \max \left\{ \begin{array}{l} \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 1 \\ \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot 0.311 \\ \frac{1}{2} \cdot 0.311 + \frac{1}{2} \cdot 1 \\ \frac{1}{2} \cdot 0.311 + \frac{1}{2} \cdot 0.311 \end{array} \right. = \max\{1, 0.655, 0.655, 0.311\} = 1$$

Pentru calculul câştigului de informaţie corespunzător atributului X_3 plasat în nodul rădăcină, au fost calculate la punctul a cele 4 câştiguri de informaţie în sens clasic implicate în formulă, deci rezultă:

$$
\begin{aligned}
IG_{2-\text{step}}(X_3) &= \max \begin{cases} \frac{4}{8}IG(X_1 \mid X_3=0)+\frac{4}{8}IG(X_1 \mid X_3=1) \\ \frac{4}{8}IG(X_1 \mid X_3=0)+\frac{4}{8}IG(X_2 \mid X_3=1) \\ \frac{4}{8}IG(X_2 \mid X_3=0)+\frac{4}{8}IG(X_1 \mid X_3=1) \\ \frac{4}{8}IG(X_2 \mid X_3=0)+\frac{4}{8}IG(X_2 \mid X_3=1) \end{cases} \\
&= \max \begin{cases} \frac{1}{2}\left(\frac{3}{2}-\frac{3}{4}\log_2 3\right)+\frac{1}{2}\left(\frac{3}{2}-\frac{3}{4}\log_2 3\right) \\ \frac{1}{2}\left(\frac{3}{2}-\frac{3}{4}\log_2 3\right)+\frac{1}{2}\left(\frac{5}{2}-\frac{3}{2}\log_2 3\right) \\ \frac{1}{2}\left(\frac{5}{2}-\frac{3}{2}\log_2 3\right)+\frac{1}{2}\left(\frac{3}{2}-\frac{3}{4}\log_2 3\right) \\ \frac{1}{2}\left(\frac{5}{2}-\frac{3}{2}\log_2 3\right)+\frac{1}{2}\left(\frac{5}{2}-\frac{3}{2}\log_2 3\right) \end{cases} = \max \begin{cases} \frac{3}{2}-\frac{3}{4}\log_2 3 \\ 2-\frac{9}{8}\log_2 3 \\ 2-\frac{9}{8}\log_2 3 \\ \frac{5}{2}-\frac{3}{2}\log_2 3 \end{cases} \\
&= \max\{0.311, 0.216, 0.216, 0.122\} = 0.311
\end{aligned}
$$

Comparând $IG_{2-\text{step}}(X_1)$, $IG_{2-\text{step}}(X_2)$ şi $IG_{2-\text{step}}(X_3)$, obţinem valoarea maximă 1 fie pentru X_1, fie pentru X_2 în nodul rădăcină. Având în vedere calculele realizate pentru a obţine aceste valori, nu mai sunt necesare operaţii suplimentare pentru determinarea arborelui de decizie construit de ID3 cu metoda *2-step look-ahead*. Există de fapt 3 arbori de decizie optimi (ca număr de niveluri şi / sau noduri) pentru aceste date de antrenament, şi anume:



Este important de remarcat faptul că algoritmul ID3 cu *2-step look-ahead* identifică toate aceste trei soluţii optimale, în vreme ce algoritmul ID3 standard nu identifică niciuna dintre ele.

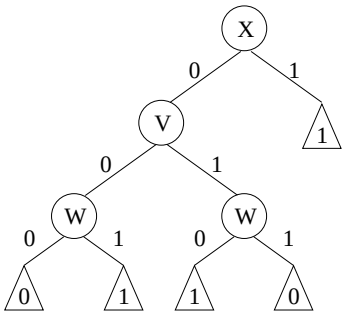
19. **(O strategie de pruning pentru arborele ID3: eliminarea nodurilor cu $IG < \varepsilon$; explorare top-down vs. bottom-up)**

prelucrare de Liviu Ciortuz, după

■ ● ○ ***CMU, 2006 spring, Carlos Guestrin, midterm exam, pr. 4***

Puteţi constata uşor că, aplicând algoritmul ID3 pe datele din tabelul de mai jos, se va obţine arborele de decizie alăturat.

V	W	X	Y
0	0	0	0
0	1	0	1
1	0	0	1
1	1	0	0
1	1	1	1



a. Pentru un astfel de arbore de decizie, o strategie simplă de trunchiere (engl., pruning) în vederea contracarării fenomenului de "overfitting" constă în a parcurge arborele de sus în jos, începând deci cu nodul-rădăcină şi identificând fiecare nod de test pentru care câştigul de informaţie (sau un alt criteriu fixat în avans) are o valoare mai mică decât o valoare pozitivă, mică, fixată de la început, ε. Orice astfel de nod de test este imediat înlocuit — împreună cu subarborele corespunzător lui — cu un nod de decizie, conform etichetei majoritare a instanţelor asignate nodului de test. Această strategie se numeşte "top-down pruning".

Care este arborele de decizie obţinut aplicând această strategie pe arborele de mai sus, dacă se consideră $\varepsilon = 0.0001$? Care este eroarea la antrenare pentru noul arbore?

b. O altă posibilitate de a face pruning este să parcurgem arborele de decizie începând cu părinţii nodurilor-frunză şi să eliminăm în mod recursiv acele noduri de test pentru care câştigul de informaţie (sau un alt criteriu ales) este mai mic decât ε. Aceasta este strategia de "bottom-up pruning".

***Observaţie*: Spre deosebire de strategia top-down, în varianta de pruning de tip bottom-up nu vor fi eliminate noduri (cu $IG < \varepsilon$) pentru care există descendenţi al căror câştig de informaţie este mai mare sau egal cu ε.**

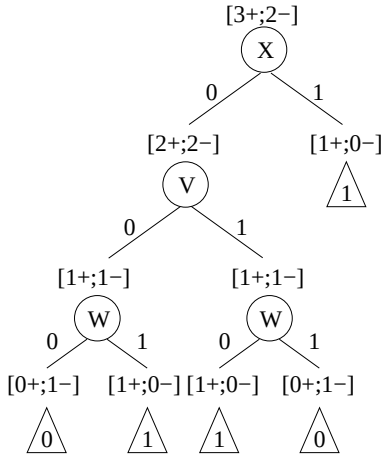
Ce arbore se obţine făcând "bottom-up pruning" pe arborele dat mai sus, dacă se consideră $\varepsilon = 0.0001$? Care este eroarea la antrenare pentru arborele rezultat?

c. Stabiliţi în ce situaţii ar fi indicat să alegem strategia "bottom-up pruning" în loc de "top-down pruning" şi viceversa. Comparaţi acurateţea la antrenare şi complexitatea computaţională a celor două strategii de pruning.

d. Cât este înălţimea — adică, numărul de niveluri de test — pentru arborele returnat de ID3 urmat de "bottom-up pruning"? Puteţi găsi un arbore de decizie având o înălţime mai mică, dar care clasifică perfect setul de antrenament? Ce concluzie putem trage despre calitatea [outputului] algoritmului ID3?

Răspuns:

Înainte de a rezolva efectiv punctele a şi b, vom augmenta arborele de decizie dat cu informaţiile referitoare la numărul de instanţe (pozitive şi, respectiv, negative) asignate fiecărui nod de test. Obţinem astfel figura alăturată.



a. Câştigul de informaţie al atributului X plasat în nodul-rădăcină este:

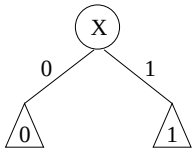
$$H[3+;2-] - 1/5 \cdot 0 - 4/5 \cdot 1 = 0.971 - 0.8 = 0.171 > \varepsilon.$$

Prin urmare, acest nod nu va fi eliminat din arbore.

Câştigul de informaţie al atributului V este:

$$H[2+;2-] - 1/2 \cdot 1 - 1/2 \cdot 1 = 1 - 1 = 0 < \varepsilon.$$

Aşadar, nodul reprezentat de atributul V va fi eliminat şi vom obţine arborele de decizie [trunchiat], reprezentat în figura alăturată. Menţionăm că am fi putut la fel de bine (din punctul de vedere al numărului de erori la antrenare) să alegem decizia $Y = 1$ în nodul-fiu stâng, însă în acel caz arborele s-ar fi redus de fapt la un singur nod de decizie (cu outputul $Y = 1$).
Eroarea la antrenare produsă de acest arbore este $2/5$.



b. Câştigul de informaţie al celor două noduri marcate cu atributul W în arborele dat în enunţ este acelaşi, şi anume 1 (se poate verifica imediat). Prin urmare, aplicând strategia de "bottom-up pruning", arborele de decizie rămâne identic cu cel iniţial. Evident, eroarea la antrenare pentru acest arbore este 0, întrucât datele de antrenament nu conţin inconsistenţe.

c. Din cauza faptului că la top-down pruning, odată cu un nod de test pentru care câştigul de informaţie (IG) este mai mic decât valoarea ε se elimină întregul subarbore care are ca rădăcină acel nod de test, această strategie este mai rapidă decât (sau, în cel mai rău caz, la fel de rapidă / lentă ca şi) pruning-ul de tip bottom-up.

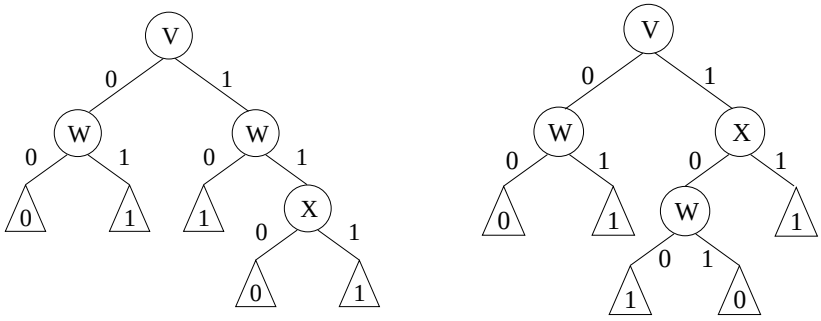
Aşa cum s-a menţionat în *observaţia* din enunţ şi s-a exemplificat apoi la punctele a şi b, dezavantajul pruning-ului de tip top-down este că odată ce este eliminat un nod cu IG mai mic decât ε, este posibil ca între descendenţii săi (eliminaţi) să fie şi noduri care au câştigul de informaţie mai mare sau egal cu ε. Pruning-ul bottom-up nu are acest dezavantaj; el este deci mai „conservativ".

O *problemă* care poate însă să apară şi în cazul pruning-ului de tip bottom-up este faptul că în nodurile apropiate de nodurile de decizie (acestea din urmă fiind nodurile-frunză), câştigul de informaţie se calculează uneori pe mulţimi mici de exemple, deci testul $IG \geq \varepsilon$ nu este neapărat semnificativ din punct de vedere statistic. Aşadar, este *recomandabil* ca în astfel de situaţii să se folosească un test statistic, de exemplu *testul* χ^2. (A se vedea problemele 20 şi 55.)

În ce priveşte comparaţia dintre acurateţile arborilor obţinuţi prin aplicarea celor două variante de pruning: se observă că arborele mai simplu (cel de la punctul a) are o eroare la antrenare mai mare decât arborele mai complex (cel de la punctul b), însă este mai probabil ca acesta din urmă să producă overfitting.

d. Din rezolvarea dată la punctul b, rezultă imediat că înălţimea arborelui obţinut prin aplicarea algoritmului ID3 urmat de pruning de tip bottom-up este 3. (Nu se iau în considerare nodurile frunză.) Vom demonstra — exact ca la problema 1 — că nu există un arbore de decizie consistent cu datele de antrenament, care să aibă adâncimea strict mai mică decât 3:

Se observă uşor din tabelul dat în enunţ că $Y = (V \text{ XOR } W) \vee (V \wedge V \wedge X)$, aşadar variabilele V şi W au rol simetric în definirea funcţiei reprezentate de variabila Y. Punând atributul X în nodul rădăcină, arborele de decizie minimal care se poate obţine este cel dat în enunţ (sau, echivalent, arborele care se obţine din acesta interschimbând V şi W). Dacă, în schimb, punem atributul V în nodul rădăcină, se pot obţine doi arbori de decizie minimali, aşa cum se arată grafic mai jos. (Şi similar, dacă în nodul rădăcină punem atributul W.)



20. **(ID3 cu post-pruning: folosirea testului statistic χ^2 pentru limitarea overfitting-ului)**

prelucrare de Liviu Ciortuz, după
■ • ***CMU, 2010 fall, Ziv Bar-Joseph, HW2, pr. 2.1***

În acest exerciţiu vom face pruning asupra unui arbore ID3 (după ce s-a făcut antrenarea pe toate datele disponibile), folosind o metodă statistică de testare / verificare a ipotezelor.

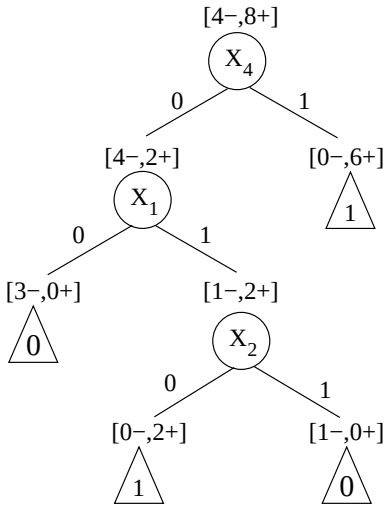
După ce a fost învăţat arborele de decizie, vizităm fiecare nod intern (inclusiv nodul rădăcină) şi testăm dacă atributul care a fost pus în nodul respectiv nu este cumva necorelat cu eticheta / clasa specificată de către atributul de ieşire.

Pentru aceasta, mai întâi presupunem că atributul din nodul respectiv este independent de atributul de ieşire (aceasta este aşa-numita „ipoteză nulă“),

iar apoi folosim testul χ^2 al lui Pearson pentru a genera o „statistică", care poate constitui temeiul pentru respingerea „ipotezei nule". Dacă ipoteza nulă nu poate fi respinsă, eliminăm sub-arborele din nodul respectiv (de fapt, îl înlocuim cu un nod de decizie).

Pentru a ilustra aceste chestiuni, considerăm arborele de decizie de mai jos (partea dreaptă); el a fost construit pornind de la datele din tabelul alăturat lui, folosind algoritmul ID3.

X_1	X_2	X_3	X_4	*Class*
1	1	0	0	0
1	0	1	0	1
0	1	0	0	0
1	0	1	1	1
0	1	1	1	1
0	0	1	0	0
1	0	0	0	1
0	1	0	1	1
1	0	0	1	1
1	1	0	1	1
1	1	1	1	1
0	0	0	0	0



a. Pentru fiecare nod intern din arborele de decizie vom crea o *tabelă de contingenţă* (engl., contingency table) pentru exemplele de antrenare care sunt asignate nodului respectiv. Tabela aceasta va avea coloanele etichetate cu cele c clase / valori ale variabilei de ieşire. Similar, valorile atributului testat în nodul respectiv (având în total r valori) vor fi asignate liniilor tabelei. Dacă acceptăm o uşoară simplificare în forma de exprimare, vom putea spune că un element oarecare $O_{i,j}$ din tabela de contingenţă reprezintă numărul de „observaţii" (adică, instanţe de antrenament asignate nodului respectiv) pentru care valoarea atributului testat este i, iar eticheta / clasa este j.

Calculaţi tabelele de contingenţă pentru cele trei noduri de test ale arborelui de decizie dat mai sus. Apoi, pornind de la datele conţinute în fiecare dintre aceste matrice de contingenţă, estimaţi (în sensul verosimilităţii maxime) probabilităţile pentru valorile variabilei de ieşire (*Class*), precum şi pentru valorile atributului din nodul corespunzător.

b. Pentru a aplica testul statistic χ^2, avem nevoie să calculăm pentru fiecare nod intern al arborelui de decizie încă o tabelă (pe care o vom nota cu E), în care să consemnăm *numărul aşteptat* de apariţii (engl., expected counts) ale instanţelor de antrenament la nodul respectiv, pentru fiecare pereche de indici i, j având semnificaţia de mai sus. Acest număr aşteptat este numărul (mediu) de instanţe de antrenament pe care le-am „observa" în nodul respectiv dacă atributul selectat şi clasa (variabila de ieşire) ar fi independente.

Derivaţi o formulă pentru calculul fiecărui element (notat $E_{i,j}$) din această a doua tabelă.

***Întrebări ajutătoare*: Care este probabilitatea ca exemplele de antrenament asignate nodului respectiv să aibă o anumită etichetă (j)? Ţinând cont de**

această probabilitate, precum şi de presupoziţia de independenţă formulată prin ipoteza „nulă“, care este numărul de exemple cu o anumită valoare (i) pentru atributul selectat în nodul respectiv, care ar trebui (i.e., „ne aşteptăm“) să aibă acea etichetă / clasă (j)?

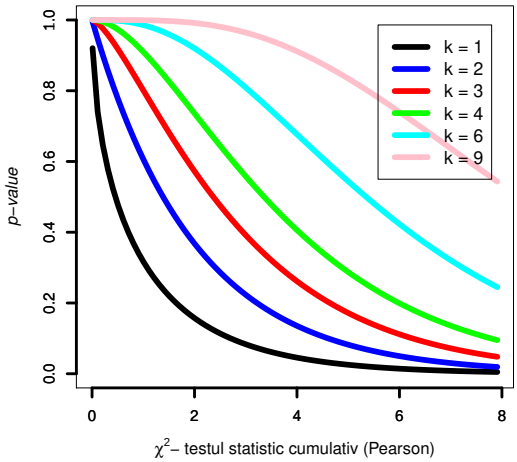
Folosind formula pe care tocmai aţi derivat-o, calculaţi matricea E pentru fiecare dintre cele trei noduri de test ale arborelui de decizie dat mai sus.

c. Date fiind cele două tabele pentru nodul considerat, puteţi calcula acum testul statistic χ-pătrat:

$$\chi^2 = \sum_{i=1}^{r}\sum_{j=1}^{c}\frac{(O_{i,j}-E_{i,j})^2}{E_{i,j}}$$

Puteţi introduce valoarea calculată χ^2 precum şi numărul de grade de libertate $(r-1)(c-1)$ într-un program[509] sau într-un calculator on-line[510] pentru a calcula o aşa-numită p-valoare (engl., p-value).[511]

În general, dacă $p < 0.05$, se va considera că nu avem suficientă *evidenţă* în favoarea ipotezei „nule“, care afirma că atributul selectat şi clasa (variabila de ieşire) sunt independente, şi o vom respinge. Într-o astfel de situaţie, spunem că testul [din nodul respectiv] este *semnificativ* din punct de vedere statistic.



Pentru fiecare dintre cele trei noduri interne din arborele de decizie dat mai sus, găsiţi p-valoarea corespunzătoare şi precizaţi dacă testul din nodul respectiv este sau nu semnificativ d.p.v. statistic. Cât de multe noduri interne vor fi eliminate din arbore dacă la pruning impunem condiţia $p \geq 0.05$ [pentru a elimina un nod de test din arbore şi a-l înlocui cu un nod de decizie]?

Răspuns:

a. Pentru fiecare nod din arborele ID3 vom alcătui matricea de contingenţă asociată, pornind de la partiţionările mulţimilor de exemple care au fost asignate (de către algoritmul ID3) descendenţilor nodului respectiv. Apoi, din

[509] Folosiţi 1-chi2cdf(x,df) în MATLAB sau CHIDIST(x,df) în Excel.

[510] `https://www.socscistatistics.com/pvalues/chidistribution.aspx` (accesat la 9.11.2022) este un astfel de calculator.

[511] *Statistica* χ^2 este aproximată de *distribuţia* χ^2 cu numărul corespunzător (k) de grade de libertate. (Distribuţia χ^2 reprezintă suma pătratelor a k variabile gaussiene standard independente.)

p-valoarea despre care este vorba mai sus reprezintă probabilitatea ca distribuţia χ^2 să ia valori mai mari sau egale cu valoarea considerată (i.e., valoarea calculată pentru statistica χ^2). Aşadar, p-valoarea pentru testul χ^2 se calculează făcând diferenţa dintre 1 şi valoarea funcţiei de distribuţie cumulative (c.d.f.) pentru distribuţia χ^2 cu k de grade de libertate. Vedeţi site-ul https://en.m.wikipedia.org/wiki/Chi-square_distribution#Table_of_.CF.872_value_vs_p-value (accesat la 5.09.2015).

fiecare matrice de contingenţă vom estima (în sensul verosimilităţii maxime) probabilităţile pentru valorile variabilei de ieşire (*Class*), precum şi probabilităţile pentru valorile atributului din acel nod. (Atenţie la condiţionarea probabilităţilor!)

O_{X_4}	$Class = 0$	$Class = 1$
$X_4 = 0$	4	2
$X_4 = 1$	0	6

$$\overset{N=12}{\Rightarrow} \begin{cases} P(X_4 = 0) = \frac{6}{12} = \frac{1}{2},\ P(X_4 = 1) = \frac{1}{2} \\ P(Class = 0) = \frac{4}{12} = \frac{1}{3},\ P(Class = 1) = \frac{2}{3} \end{cases}$$

$O_{X_1 \mid X_4=0}$	$Class = 0$	$Class = 1$
$X_1 = 0$	3	0
$X_1 = 1$	1	2

$$\overset{N=6}{\Rightarrow} \begin{cases} P(X_1 = 0 \mid X_4 = 0) = \frac{3}{6} = \frac{1}{2} \\ P(X_1 = 1 \mid X_4 = 0) = \frac{1}{2} \\ P(Class = 0 \mid X_4 = 0) = \frac{4}{6} = \frac{2}{3} \\ P(Class = 1 \mid X_4 = 0) = \frac{1}{3} \end{cases}$$

$O_{X_2 \mid X_4=0, X_1=1}$	$Class = 0$	$Class = 1$
$X_2 = 0$	0	2
$X_2 = 1$	1	0

$$\overset{N=3}{\Rightarrow} \begin{cases} P(X_2 = 0 \mid X_4 = 0, X_1 = 1) = \frac{2}{3} \\ P(X_2 = 1 \mid X_4 = 0, X_1 = 1) = \frac{1}{3} \\ P(Class = 0 \mid X_4 = 0, X_1 = 1) = \frac{1}{3} \\ P(Class = 1 \mid X_4 = 0, X_1 = 1) = \frac{2}{3} \end{cases}$$

b. Considerăm i o valoare arbitrar aleasă (dar fixată) pentru atributul de intrare A care este testat în nodul curent, iar j o valoare arbitrar aleasă (de asemenea, fixată) pentru atributul de ieşire *Class* (renotat cu C). Ţinând cont de presupoziţia de independenţă stipulată de ipoteza „nulă“, putem scrie:

$$P(A = i, C = j) = P(A = i) \cdot P(C = j)$$

Probabilităţile $P(A = i)$ şi $P(C = j)$ pot fi estimate — în sensul verosimilităţii maxime (MLE) —, cu ajutorul celor N instanţe de antrenament asignate nodului respectiv. Instanţele pentru care atributul A are valoarea i sunt tocmai cele din linia i a matricei. În mod similar, instanţele care au eticheta / clasa j sunt cele din coloana j a matricei de count-uri observate. Aşadar,

$$P(A = i) = \frac{\sum_{k=1}^{c} O_{i,k}}{N} \text{ şi } P(C = j) = \frac{\sum_{k=1}^{r} O_{k,j}}{N}$$

În consecinţă,

$$P(A = i, C = j) = \frac{(\sum_{k=1}^{c} O_{i,k})\,(\sum_{k=1}^{r} O_{k,j})}{N^2},$$

iar valoarea aşteptată — repetăm, în condiţiile presupoziţiei de independenţă — pentru numărul de instanţe având atributul $A = i$ şi clasa $C = j$ va fi dată de formula

$$E_{i,j} = N \cdot P(A = i, C = j) = \frac{(\sum_{k=1}^{c} O_{i,k})\,(\sum_{k=1}^{r} O_{k,j})}{N}$$

Folosind probabilităţile calculate la punctul precedent şi ţinând cont de presupoziţia de independenţă, calculăm numărul de observaţii aşteptate în fiecare nod, pentru a completa matricele E:

E_{X_4}	$Class = 0$	$Class = 1$
$X_4 = 0$	2	4
$X_4 = 1$	2	4

$E_{X_1\|X_4=0}$	$Class = 0$	$Class = 1$
$X_1 = 0$	2	1
$X_1 = 1$	2	1

$E_{X_2\|X_4=0,X_1=1}$	$Class = 0$	$Class = 1$
$X_2 = 0$	$\frac{2}{3}$	$\frac{4}{3}$
$X_2 = 1$	$\frac{1}{3}$	$\frac{2}{3}$

Ca să *exemplificăm* cum am procedat, detaliem mai jos calculul pentru primul element din matricea E_{X_4}:

$$N = 12,\ P(X_4 = 0) = \frac{1}{2} \textbf{ şi } P(Class = 0) = \frac{1}{3} \Rightarrow$$
$$N \cdot P(X_4 = 0, Class = 0) = N \cdot P(X_4 = 0) \cdot P(Class = 0) = 12 \cdot \frac{1}{2} \cdot \frac{1}{3} = 2$$

c. Aplicăm pentru fiecare nod din arborele de decizie formula de calcul a valorilor / statisticilor χ^2 care ne-a fost dată în enunţ:

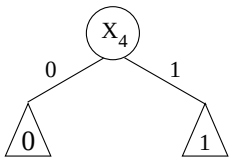
$$\chi^2{}_{X_4} = \frac{(4-2)^2}{2} + \frac{(2-4)^2}{4} + \frac{(0-2)^2}{2} + \frac{(6-4)^2}{4} = 2 + 1 + 2 + 1 = 6$$

$$\chi^2{}_{X_1|X_4=0} = \frac{(3-2)^2}{2} + \frac{(0-1)^2}{1} + \frac{(1-2)^2}{2} + \frac{(2-1)^2}{1} = 3$$

$$\chi^2{}_{X_2|X_4=0,X_1=1} = \frac{\left(0-\frac{2}{3}\right)^2}{\frac{2}{3}} + \frac{\left(2-\frac{4}{3}\right)^2}{\frac{4}{3}} + \frac{\left(1-\frac{1}{3}\right)^2}{\frac{1}{3}} + \frac{\left(0-\frac{2}{3}\right)^2}{\frac{2}{3}} = \frac{4}{9} \cdot \frac{3}{4} \cdot 9 = 3$$

Accesând pagina web indicată în enunţ, am obţinut p-valorile următoare: 0.0143, 0.0833 şi 0.0833.

În consecinţă, cu un grad de încredere de cel puţin 95%, nodurile situate pe nivelurile 1 şi 2 din arborele ID3 pot fi eliminate. Pentru nodul rădăcină, ipoteza „nulă“ nu se verifică, adică variabilele X_4 şi *Class* nu sunt independente. Arborele obţinut în urma pruning-ului este cel din figura alăturată.



***Observaţie*: Este de remarcat faptul că arborele ID3 furnizat în enunţ are (întâmplător) atât pentru atributul X_4 (din nodul rădăcină) cât şi pentru nodul X_1 (de pe primul nivel) acelaşi câştig de informaţie, 0.4591. În urma testului χ^2, se va elimina însă doar nodul care-l conţine pe X_1. Valoarea statisticii χ^2 asociate nodului X_1 este mult mai mică (3, faţă de 6, cât este pentru nodul care-l conţine pe X_4), deci vom avea suficientă „evidenţă“ pentru a trage concluzia, cu un grad de încredere de 95%, că variabila de ieşire, *Class*, şi $X_1|X_4 = 0$ sunt independente.**[512] **Dacă am fi aplicat o metodă de pruning**

[512] Observaţi şi faptul că nodul care conţine atributul X_1 are mai puţine instanţe asociate (şi anume, jumătate) faţă de cele asociate nodului rădăcină.

bazată pe câştigul de informaţie, aceasta n-ar fi putut să trateze în mod diferit cele două noduri, adică să-l păstreze pe unul şi să-l elimine pe celălalt.

21. **(Adevărat sau Fals?)**

a. ***Liviu Ciortuz***

Algoritmul ID3 garantează obţinerea arborelui de decizie optimal (ca număr de niveluri sau de noduri).

CMU, 2002 spring, A. Moore, midterm example questions, pr. 1.c

b. Întrucât arborii de decizie pot învăţa să clasifice instanţe într-un număr discret de clase (deci nu învaţă funcţii cu valori reale), este imposibil ca ei să manifeste fenomenul de overfitting.

CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, midterm, pr. 1.4

c. Fie A şi B doi algoritmi de clasificare automată. Algoritmul A este mai bun decât algoritmul B dacă eroarea la antrenare a algoritmului A este mai mică decât eroarea la antrenare a algoritmului B. Justificaţi.

CMU, 2005 spring, C. Guestrin, T. Mitchell, midterm, pr. 2.c

d. Presupunem că avem m instanţe şi că vom folosi jumătate dintre ele pentru antrenarea unui clasificator oarecare (nu neapărat ID3) şi jumătate pentru testare. Diferenţa dintre eroarea la antrenare şi eroarea la testare descreşte pe măsură ce numărul m creşte.

Răspuns:

a. Fals. ID3 nu garantează obţinerea arborelui optim (relativ la numărul de niveluri şi / sau noduri), ci încearcă să găsească o soluţie convenabilă, însă fără să caute în mod exhaustiv în tot spaţiul soluţiilor. Mai exact, căutarea soluţiei se face în manieră "greedy", maximizând un anumit criteriu (e.g., câştigul de informaţie) la fiecare iteraţie. O astfel de căutare nu garantează obţinerea optimului.

b. Fals. Arborii de decizie manifestă fenomenul de overfitting. Prima parte a afirmaţiei din enunţ este adevarată — arborii de decizie pot învăţa să clasifice instanţe într-un număr discret de clase —, însă a doua parte este falsă, deci avem o implicaţie de forma: $T \rightarrow F \equiv \neg T \vee F \equiv F$.

c. Fals, fiindcă la testare algoritmul B poate să aibă o eroare mai mică decât algoritmul A. Într-un astfel de caz, se spune că algoritmul A este "overfit" (rom., supra-antrenat).

d. Adevărat. Pe măsură ce dispunem de tot mai multe date de antrenament, dacă datele de antrenament sunt inconsistente, eroarea la antrenare va creşte, fiindcă va fi din ce în ce mai greu ca modelul învăţat să se adapteze la „zgomotele" din date. Similar, eroarea la testare va descreşte, fiindcă producem un clasificator care este din ce în ce mai puţin afectat de "overfitting" pe datele de antrenament. Cele două erori vor converge la aşa-numita *eroare adevărată* (engl., true error) fiindcă diferenţele statistice dintre datele de antrenament şi datele de testare vor dispărea.

4.1.2 Algoritmul AdaBoost

22. **(Algoritmul AdaBoost: formulare; demonstrarea unor relaţii de bază)**

prelucrare de Liviu Ciortuz, după
■ ● ○ ***CMU, 2015 fall, Z. Bar-Joseph, E. Xing, HW4, pr. 2.1***
CMU, 2009 fall, Carlos Guestrin, HW2, pr. 3.1

Fie o mulţime de m exemple de antrenament, $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, cu $x_i \in \mathcal{X}$ (unde $\mathcal{X}$ poate fi de exemplu $\mathbb{R}^d$, cu $d \in \mathbb{N}^*$) şi $y_i \in \{-1, +1\}$ pentru $i = 1, \ldots, m$.

Presupunem că dispunem de un algoritm A care este un *clasificator* automat *„slab“* (engl., weak classifier). A primeşte ca *input* o distribuţie probabilistă oarecare D definită peste mulţimea S de exemple de antrenament şi produce ca *output* o *ipoteză* $h : \mathcal{X} \to \{-1, +1\}$. Faptul că algoritmul A este un clasificator „slab“ înseamnă că orice astfel de ipoteză h produsă de el este [doar cu puţin] mai bună decât ghicirea / alegerea aleatorie (engl., random guessing).

Algoritmul AdaBoost[513] este un algoritm iterativ care livrează ca *output* [un clasificator bazat pe] o *combinaţie liniară de ipoteze* care sunt produse de către clasificatorul „slab“ A, câte una la fiecare iteraţie. Concret, AdaBoost lucrează astfel:

- **Iniţial (adică pentru $t = 1$), se foloseşte distribuţia uniformă $D_1(i) = \dfrac{1}{m}$, $i = 1, \ldots, m$.**
- **La fiecare iteraţie $t = 1, \ldots, T$,**
  - **folosind distribuţia probabilistă D_t, rulează algoritmul „slab“ de clasificare A pe setul de exemple de antrenament S, obţinând ipoteza h_t;**
  - **calculează *eroarea ponderată la antrenare* produsă de ipoteza h_t,**

$$\varepsilon_t \overset{not.}{=} err_{D_t}(h_t) \overset{def.}{=} \Pr_{D_t}(\{x | y \neq h_t(x)\}), \tag{243}$$

iar după aceea,[514] ponderea / „votul“

$$\alpha_t \overset{not.}{=} \frac{1}{2} \ln \frac{1 - \varepsilon_t}{\varepsilon_t} \tag{244}$$

asociat ipotezei h_t;[515]

  - **defineşte o nouă distribuţie probabilistă D_{t+1}, folosind „regula de actualizare“ (engl., update rule)**

$$D_{t+1}(i) = \frac{1}{Z_t} D_t(i)\, e^{-\alpha_t y_i h_t(x_i)}, \text{ pentru } i = 1, \ldots, m \text{ şi } \forall t \geq 1, \tag{245}$$

unde Z_t este factorul de normalizare.

[513] Yoav Freund, Robert Schapire, “Experiments with a new boosting algorithm“, 1996, “A Decision Theoretic Generalization of On-Line Learning and an Application to Boosting“, 1997 şi “A short introduction to boosting“, 1999.

[514] Dacă la o iteraţie $t \leq T$ clasificatorul slab A nu poate produce nicio ipoteză mai bună decât alegerea aleatorie (altfel spus, $\varepsilon_t = 1/2$) sau dacă, dimpotrivă, A nu poate produce decât o ipoteză perfectă (adică $\varepsilon_t = 0$), atunci algoritmul AdaBoost trebuie oprit.

[515] Am notat cu $\Pr_{D_t}(E)$ probabilitatea lui E (eveniment aleatoriu oarecare) în raport cu distribuţia probabilistă D_t. Remarcaţi faptul că din $\varepsilon_t \in (0, 1/2)$ rezultă $(1 - \varepsilon_t)/\varepsilon_t > 1$, deci $\alpha_t > 0$.

- **În final, algoritmul AdaBoost va livra — ca ipoteză învăţată — funcţia** $H_T \stackrel{def.}{=} sign\left(\sum_{t=1}^{T} \alpha_t h_t\right)$**; ea va acţiona asupra instanţelor de test** x **conform [principiului]** ***votului ponderat majoritar*** **(engl., weighted majority vote).**

***Observaţie importantă*:** Această formulare a algoritmului AdaBoost nu impune nicio restricţie asupra ipotezei h_t furnizate de către clasificatorul slab A la iteraţia t, cu excepţia condiţiei $\varepsilon_t < 1/2$. Totuşi, într-o formulare ulterioară a algoritmului AdaBoost, ca în problema 28, dar şi într-un cadru mai extins, ca în problema 29, se poate cere / recomanda în mod explicit[516] ca ipoteza h_t să fie aleasă *minimizând* (eventual aproximativ) criteriul erorii ponderate la antrenare pe o întreagă clasă de ipoteze (separatori decizionali). Astfel de ipoteze pot fi, de *exemplu*, arborii de decizie de adâncime 1 (compaşi de decizie; engl., decision stumps). În mod implicit la toate exerciţiile unde vom exemplifica aplicarea algoritmului AdaBoost, *recomandarea* aceasta va fi aplicată.

Demonstraţi următoarele relaţii:

i. $\varepsilon_i > \varepsilon_j \Leftrightarrow \alpha_i < \alpha_j$ **(consecinţă din relaţia** (244)**)**

ii. $Z_t = e^{-\alpha_t} \cdot (1 - \varepsilon_t) + e^{\alpha_t} \cdot \varepsilon_t$ **(consecinţă din relaţia** (245)**)**

iii. $Z_t = 2\sqrt{\varepsilon_t(1 - \varepsilon_t)}$ **(consecinţă din** *ii* **şi din relaţia** (244)**)**

iv. $0 < Z_t < 1$ **(consecinţă din** *iii*.**)**

v. $D_{t+1}(i) = \begin{cases} \dfrac{D_t(i)}{2\varepsilon_t}, & i \in M \stackrel{not.}{=} \{i | y_i \neq h_t(x_i)\} \\ \dfrac{D_t(i)}{2(1 - \varepsilon_t)}, & i \in C \stackrel{not.}{=} \{i | y_i = h_t(x_i)\} \end{cases}$
(consecinţă din relaţia (245) **şi** *iii*.**)**

vi. $err_{D_{t+1}}(h_t) = 1/2$**, unde** $err_{D_{t+1}}(h_t) \stackrel{not.}{=} \Pr_{D_{t+1}}(\{x_i | h_t(x_i) \neq y_i\})$
(consecinţă din relaţia (245) **şi** *iii*.**)**

***Observaţii*:**

1. Vom explica acum de ce anume se foloseşte regula de actualizare a distribuţiilor D_t sub forma dată (245). Dubla inegalitate $0 < \varepsilon_t < \dfrac{1}{2}$ implică $\alpha_t > 0$ şi, prin urmare, rezultă că $e^{\alpha_t} > 1$, iar $e^{-\alpha_t} < 1$. Pentru instanţe corect clasificate de către ipoteza h_t, vom avea $y_i h_t(x_i) = 1$, iar pentru instanţe incorect clasificate $y_i h_t(x_i) = -1$. Rezultă că la calculul distribuţiei D_{t+1}, algoritmul AdaBoost măreşte probabilitatea alocată instanţelor incorect clasificate şi diminuează probabilitatea alocată instanţelor corect clasificate.
2. Faptul că $err_{D_{t+1}}(h_t) = 1/2$ (egalitate care trebuie demonstrată la punctul *vi*.) înseamnă că algoritmul ia exact „masa“ de probabilitate $\gamma_t \stackrel{not.}{=} \dfrac{1}{2} - \varepsilon_t$ (care reprezintă cu cât este mai bună ipoteza h_t decât alegerea aleatorie) de la mulţimea de instanţe care au fost corect clasificate de către ipoteza h_t şi o distribuie în mod proporţional la mulţimea de instanţe incorect clasificate de către acelaşi h_t, pentru ca algoritmul slab A să se concentreze ulterior asupra „învăţării“ unui model (sau a unei ipoteze) care să clasifice corect şi aceste instanţe. Termenul de *boosting adaptiv* — din care a derivat numele algoritmului AdaBoost — corespunde exact acestei strategii, care este pusă la lucru prin *creşterea* probabilităţior asociate instanţelor incorect clasificate.
3. Prin ipoteză, clasificatorul slab A produce pentru distribuţia D_{t+1} o ipoteză (h_{t+1}) având eroarea ε_{t+1} strict mai mică decât 1/2. Prin urmare, rezultatul $err_{D_{t+1}}(h_t) = 1/2$

[516] Vedeţi de exemplu MIT, 2006 fall, Tommi Jaakkola, HW4, problema 3.

(demonstrat la punctul *vi.*) va avea drept *consecinţă* faptul că în mod cert ipoteza h_t nu va fi [produsă şi, deci, nici] aleasă la iteraţia $t+1$.

Răspuns:

i. **Plecând de la definiţia $\alpha_t = \ln\sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}$ care a fost dată în pseudo-codul algoritmului AdaBoost, putem scrie mai întâi**

$$\alpha_i < \alpha_j \Leftrightarrow \ln\sqrt{\frac{1-\varepsilon_i}{\varepsilon_i}} < \ln\sqrt{\frac{1-\varepsilon_j}{\varepsilon_j}}.$$

Apoi, întrucât atât funcţia $\ln$ **cât şi funcţia rădăcină pătrată sunt strict crescătoare, rezultă că**

$$\alpha_i < \alpha_j \Leftrightarrow \frac{1-\varepsilon_i}{\varepsilon_i} < \frac{1-\varepsilon_j}{\varepsilon_j} \overset{\varepsilon_i,\varepsilon_j>0}{\Leftrightarrow} \varepsilon_j(1-\varepsilon_i) < \varepsilon_i(1-\varepsilon_j)$$
$$\Leftrightarrow \varepsilon_j - \cancel{\varepsilon_i\varepsilon_j} < \varepsilon_i - \cancel{\varepsilon_i\varepsilon_j} \Leftrightarrow \varepsilon_i > \varepsilon_j$$

Aşadar, are loc echivalenţa $\alpha_i < \alpha_j \Leftrightarrow \varepsilon_i > \varepsilon_j$**.**

ii. **Întrucât Z_t este factorul de normalizare în scrierea probabilităţilor $D_{t+1}(i)$, conform relaţiei (245) putem scrie:**[517]

$$\begin{aligned} Z_t &= \sum_{i=1}^{m} D_t(i)\cdot e^{-\alpha_t y_i h_t(x_i)} = \sum_{i\in C} D_t(i)\cdot e^{-\alpha_t y_i h_t(x_i)} + \sum_{i\in M} D_t(i)\cdot e^{-\alpha_t y_i h_t(x_i)} \\ &= \sum_{i\in C} D_t(i)\cdot e^{-\alpha_t} + \sum_{i\in M} D_t(i)\cdot e^{\alpha_t} = e^{-\alpha_t}\cdot \underbrace{\sum_{i\in C} D_t(i)}_{1-\varepsilon_t} + e^{\alpha_t}\cdot \underbrace{\sum_{i\in M} D_t(i)}_{\varepsilon_t} \\ &= e^{-\alpha_t}\cdot(1-\varepsilon_t) + e^{\alpha_t}\cdot\varepsilon_t. \end{aligned} \tag{246}$$

***Observaţie*: La problema 23.c vom arăta că valoarea $\alpha_t = \frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t}$ este punct de minim pentru expresia $e^{-\alpha_t}\cdot(1-\varepsilon_t) + e^{\alpha_t}\cdot\varepsilon_t$, dacă aceasta este văzută ca funcţie de argumentul α_t.**

iii. **Ţinând cont de relaţia de definiţie dată pentru ponderea α_t în enunţ, vom avea:**

$$e^{\alpha_t} = e^{\frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t}} = e^{\ln\sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}} = \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}, \tag{247}$$

ceea ce implică

$$e^{-\alpha_t} = \frac{1}{e^{\alpha_t}} = \sqrt{\frac{\varepsilon_t}{1-\varepsilon_t}}. \tag{248}$$

[517] Vom uza de faptul (specificat în enunţ la proprietatea *iv.*) că s-a notat cu C mulţimea indicilor acelor exemple care sunt corect clasificate la iteraţia t (adică, $C = \{i : y_i h_t(x_i) \geq 0\}$) şi cu M mulţimea indicilor acelor exemple care sunt incorect clasificate la aceeaşi iteraţie t (adică, $M = \{i : y_i h_t(x_i) < 0\}$).

În consecinţă, folosind relaţia (246)**, factorul de normalizare** Z_t **poate fi exprimat în funcţie de** ε_t **(eroarea ponderată comisă la antrenare de către ipoteza** h_t **în raport cu distribuţia** D_t**) astfel:**

$$Z_t = \sqrt{\frac{\varepsilon_t}{1-\varepsilon_t}} \cdot (1-\varepsilon_t) + \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}} \cdot \varepsilon_t = 2\sqrt{\varepsilon_t(1-\varepsilon_t)}. \tag{249}$$

iv. **Pe baza cunoştinţelor de analiză matematică de liceu, deducem imediat că maximul funcţiei de gradul al doilea** $\varepsilon_t(1-\varepsilon_t)$ **se atinge pentru** $\varepsilon_t = 1/2$**, iar valoarea acestui maxim este** $1/4$**. Ţinând cont că** $\varepsilon_t \in (0, 1/2)$**, din relaţia** (249) **rezultă că** $Z_t > 0$ **şi** $Z_t < 2\sqrt{\frac{1}{4}} = 1$**.**

v. **Pe baza relaţiei** (245)**, putem scrie imediat:**

$$D_{t+1}(i) = \frac{1}{Z_t} \cdot D_t(i) \cdot \begin{cases} e^{\alpha_t}, & \text{pentru } i \in M \\ e^{-\alpha_t}, & \text{pentru } i \in C. \end{cases}$$

Prin urmare,

$$i \in M \Rightarrow D_{t+1}(i) = \frac{1}{Z_t} \cdot D_t(i) \cdot e^{\alpha_t} \overset{(249),(247)}{=} \frac{1}{2\sqrt{\varepsilon_t(1-\varepsilon_t)}} \cdot D_t(i) \cdot \frac{\sqrt{1-\varepsilon_t}}{\sqrt{\varepsilon_t}} = \frac{D_t(i)}{2\varepsilon_t}$$

$$i \in C \Rightarrow D_{t+1}(i) = \frac{1}{Z_t} \cdot D_t(i) \cdot e^{-\alpha_t} \overset{(249),(248)}{=} \frac{1}{2\sqrt{\varepsilon_t(1-\varepsilon_t)}} \cdot D_t(i) \cdot \frac{\sqrt{\varepsilon_t}}{\sqrt{1-\varepsilon_t}} = \frac{D_t(i)}{2(1-\varepsilon_t)}.$$

vi. **Putem exprima eroarea ponderată produsă la antrenare de către ipoteza** h_t**, în raport cu distribuţia probabilistă** D_{t+1}**, în felul următor:**

$$\begin{aligned} err_{D_{t+1}}(h_t) &\overset{def.}{=} \Pr_{D_{t+1}}(\{x_i | h_t(x_i) \neq y_i\}) = \sum_{i=1}^{m} D_{t+1}(i) \cdot 1_{\{y_i \neq h_t(x_i)\}} \\ &\overset{(245)}{=} \sum_{i \in M} \frac{1}{Z_t} \cdot D_t(i) \cdot e^{\alpha_t} = \frac{1}{Z_t} \cdot e^{\alpha_t} \cdot \underbrace{\sum_{i \in M} D_t(i)}_{\varepsilon_t} \\ &= \frac{1}{Z_t} \cdot e^{\alpha_t} \cdot \varepsilon_t. \end{aligned} \tag{250}$$

Înlocuind în formula (250) **valoarea lui** Z_t **dată de expresia** (249)**, obţinem:**

$$err_{D_{t+1}}(h_t) = \frac{1}{Z_t} \cdot e^{\alpha_t} \cdot \varepsilon_t = \frac{1}{2\sqrt{\varepsilon_t(1-\varepsilon_t)}} \cdot \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}} \cdot \varepsilon_t = \frac{1}{2}.$$

23. **(Algoritmul AdaBoost: analiza teoretică a convergenţei erorii la antrenare; învăţabilitate empirică** γ**-slabă)**

prelucrare de Liviu Ciortuz, după

■ ● ○ ***CMU, 2015 fall, Z. Bar-Joseph, E. Xing, HW4, pr. 2.2-5***
CMU, 2009 fall, Carlos Guestrin, HW2, pr. 3.1
CMU, 2005 spring, T. Mitchell, C. Guestrin, HW2, pr. 1.1.3

Acest exerciţiu vă va ghida pas cu pas ca să demonstraţi că $err_S(H_T)$**,** ***eroarea la antrenare*** **a algoritmului AdaBoost, prezentat la problema 22 — văzută ca**

numărul de instanţe greşit clasificate din totalul de m instanţe — în anumite condiţii converge la 0, întrucât o margine superioară a ei descreşte în mod strict [uneori repede, adică cu o rată foarte mare] în raport cu numărul de iteraţii efectuate (T).

***Observaţie*: În acest exerciţiu se lucrează fără a cere ca la fiecare iteraţie a algoritmului AdaBoost să se aleagă cea mai bună ipoteză „slabă".**[518]

a. Notăm cu f_T combinaţia liniară de ipoteze $\sum_{t=1}^{T} \alpha_t h_t$.[519] **Arătaţi că din relaţia (245) rezultă $D_{T+1}(i) = \frac{1}{m \cdot \prod_{t=1}^{T} Z_t} e^{-y_i f_T(x_i)}$. (Această expresie reprezintă de fapt o formulă de calcul nerecursivă pentru valorile distribuţiilor probabiliste D_t.)**

b. Folosind rezultatul de la punctul a, arătaţi că $err_S(H_T) \leq \prod_{t=1}^{T} Z_t$, unde $err_S(H_T)$ desemnează eroarea produsă la antrenare de către AdaBoost:[520]

$$\frac{1}{m} \sum_{i=1}^{m} 1_{\{H_T(x_i) \neq y_i\}}.$$

Notaţia $1_{\{H_T(x_i) \neq y_i\}}$ desemnează, ca de obicei, *funcţia-indicator* pentru testul $H_T(x_i) \neq y_i$; aşadar, $1_{\{H_T(x_i) \neq y_i\}} = 1$ pentru acei i pentru care $H_T(x_i) \neq y_i$ şi 0 în caz contrar.[521]

***Sugestie* (1): Puteţi folosi inegalitatea: $1_{\{x<0\}} \leq e^{-x}$.**[522]

c. La acest punct vom vedea că algoritmul AdaBoost — în loc să optimizeze în mod direct eroarea produsă la antrenare, $err_S(H_T)$ —, se mulţumeşte să *minimizeze* în mod *greedy* (deci, secvenţial) produsul $\prod_{t=1}^{T} Z_t$, care reprezintă o *margine superioară* (engl., upper bound) pentru *eroarea la antrenare*, după cum am arătat la punctul b.

Se observă în pseudo-codul algoritmului AdaBoost că factorii de normalizare $Z_1, \ldots, Z_{t-1}$ sunt determinaţi în cadrul primelor $t-1$ iteraţii şi nu pot fi modificaţi la iteraţia t. Aşadar, a minimiza $\prod_{t=1}^{T} Z_t$ la iteraţia t în mod *greedy* revine la a minimiza Z_t.

Dacă facem abstracţie de valoarea atribuită ponderii α_t la iteraţia t (şi anume, $\frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$), atunci factorul de normalizare Z_t pentru distribuţia D_{t+1} va putea fi scris ca o funcţie de parametru α_t, pornind de la relaţia (246). Arătaţi că valoarea pentru care se atinge minimul acestei funcţii (în raport cu toate valorile posibile pentru α_t) este exact valoarea $\frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$, pe care am întâlnit-o în formularea algoritmului AdaBoost.

d. Arătaţi că $\prod_{t=1}^{T} Z_t \leq e^{-2\sum_{t=1}^{T} \gamma_t^2}$, unde $\gamma_t \stackrel{not.}{=} \frac{1}{2} - \varepsilon_t$.[523]

[518] Vedeţi *Observaţia importantă* de la problema 22.

[519] Observaţi că expresia funcţiei H_T din finalul pseudo-codului algoritmului AdaBoost — vedeţi enunţul problemei 22 — se poate rescrie ca $H_T(x) = sign(f_T(x))$.

[520] Vă reamintim că S este setul de date de antrenament.

[521] *Observaţie*: $H_T(x_i) \neq y_i \stackrel{def.}{\Leftrightarrow} sign(\sum_{t=1}^{T} \alpha_t h_t(x_i)) \neq y_i \Leftrightarrow sign(f_T(x_i)) \neq y_i \Leftrightarrow y_i sign(f_T(x_i)) = -1 \Leftrightarrow y_i f_T(x_i) < 0$. Aşadar, funcţiile $1_{\{H_T(x_i) \neq y_i\}}$ şi $1_{\{y_i f_T(x_i) < 0\}}$ sunt identice.

[522] Putem interpreta această inegalitate sub forma următoare: funcţia de cost / „pierdere" [negativ] exponenţială reprezintă o margine superioară pentru funcţia de cost / pierdere $0-1$ (engl., the $0-1$ loss function). Pentru o definiţie [de lucru] pentru funcţiile de cost / pierdere, vedeţi *Explicaţia* de la problema 29.

[523] LC: Putem numi variabila γ_t „ecart", fiindcă ea reprezintă diferenţa dintre 1/2, eroarea alegerii aleatorii, şi ε_t, eroarea ponderată a ipotezei h_t la antrenare.

***Sugestie* (2): De data aceasta, puteţi folosi o altă margine inferioară pentru funcţia de pierdere [negativ] exponenţială:** $1 - x \leq e^{-x}$.

e. Combinând rezultatele de la punctele c **şi** d**, observăm că eroarea la antrenare produsă de algoritmul AdaBoost se micşorează (cu o rată exponenţială) pe măsură ce** T **creşte. Întrucât rata aceasta este cuprinsă în intervalul** $(0, 1)$**, *convergenţa* ei este asigurată.**[524] **La acest punct vom pune în evidenţă o *condiţie suficientă* pentru ca eroarea la antrenare produsă de AdaBoost să conveargă la** 0**.**

Presupunem că există $\gamma > 0$ **astfel încât** $\gamma \leq \gamma_t$ **pentru orice** $t = 1, 2, \ldots$ **.**[525] **(Această proprietate se numeşte „învăţabilitate empirică**[526] γ**-slabă“ (engl., empirical** γ**-weak learnability), iar** γ **se numeşte *garanţie de învăţabilitate empirică slabă*.) Considerând** $\varepsilon > 0$ **fixat, determinaţi (în funcţie de** γ **şi** ε**) o margine superioară pentru cât de multe iteraţii sunt necesare pentru ca eroarea la antrenare produsă de algoritmul AdaBoost să fie mai mică decât** ε**, adică** $err_S(H_T) < \varepsilon$**. Vă cerem să exprimaţi răspunsul sub forma** $T = \mathcal{O}(\cdot)$**.**

Răspuns:

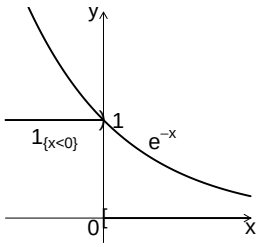
a. Pentru a demonstra egalitatea din enunţ, vom porni de la relaţia (245), exprimând probabilitatea $D_{T+1}(i)$ **recursiv în funcţie de** $D_T(i)$**, apoi de** $D_{T-1}(i)$**, până ajungem la** $D_1(i)$**:**

$$\begin{aligned}
D_{T+1}(i) &\overset{def.}{=} \frac{1}{Z_T} D_T(i)\, e^{-\alpha_T\, y_i\, h_T(x_i)} = D_T(i)\, \frac{1}{Z_T}\, e^{-\alpha_T\, y_i\, h_T(x_i)} \\
&\overset{def.}{=} D_{T-1}(i)\, \frac{1}{Z_{T-1}}\, e^{-\alpha_{T-1}\, y_i\, h_{T-1}(x_i)}\, \frac{1}{Z_T}\, e^{-\alpha_T\, y_i\, h_T(x_i)} \\
&= D_{T-1}(i)\, \frac{1}{Z_{T-1} Z_T}\, e^{-y_i(\alpha_{T-1}\, h_{T-1}(x_i) + \alpha_T\, h_T(x_i))} \\
&\;\;\vdots \\
&= D_j(i)\, \frac{1}{Z_j \ldots Z_{T-1} Z_T}\, e^{-y_i(\alpha_j\, h_j(x_i) + \ldots + \alpha_{T-1}\, h_{T-1}(x_i) + \alpha_T\, h_T(x_i))} \\
&\;\;\vdots \\
&= D_1(i)\, \frac{1}{\prod_{t=1}^{T} Z_t}\, e^{-\sum_{t=1}^{T} y_i\, \alpha_t\, h_t(x_i)} = \frac{1}{m \cdot \prod_{t=1}^{T} Z_t}\, e^{-y_i\, f_T(x_i)}.
\end{aligned}$$

Produsul de forma $y_i f_T(x_i)$ **(mai general,** $y_i f_t(x_i)$**) se numeşte *margine funcţională* a instanţei** x_i**.**

b. După cum s-a precizat în enunţ, putem exprima eroarea produsă de ipoteza H_T **(outputul algoritmului AdaBoost) pe setul de date de antrenament** S **cu ajutorul funcţiei de cost / pierdere** $0 - 1$**:**

$$err_S(H_T) = \frac{1}{m} \sum_{i=1}^{m} 1_{\{y_i f_T(x_i) < 0\}}.$$



[524] Se ştie că orice şir mărginit şi monoton este convergent.
[525] LC: De fapt, este suficient ca această condiţie să fie îndeplinită de la o iteraţie oarecare t_0 încolo.
[526] Adjectivul *empiric* se referă la faptul că eroarea analizată este *eroarea la antrenare*.

Folosind inegalitatea menţionată în *Sugestia* (1) din enunţ, putem scrie:

$$err_S(H_T) \leq \frac{1}{m}\sum_{i=1}^{m} e^{-y_i\, f_T(x_i)}.$$

Conform rezultatului de la punctul a, vom putea substitui $e^{-y_i\, f_T(x_i)}$ cu produsul $D_{T+1}(i)\cdot m\cdot\prod_{t=1}^{T} Z_t$ şi vom obţine:

$$\begin{aligned} err_S(H_T) &\leq \frac{1}{m}\sum_{i=1}^{m}\left(D_{T+1}(i)\cdot m\cdot\prod_{t=1}^{T} Z_t\right) = \sum_{i=1}^{m}\left(D_{T+1}(i)\prod_{t=1}^{T} Z_t\right) \\ &= \left(\prod_{t=1}^{T} Z_t\right)\cdot\underbrace{\left(\sum_{i=1}^{m} D_{T+1}(i)\right)}_{1} = \prod_{t=1}^{T} Z_t. \end{aligned}$$

Egalitatea $\sum_{i=1}^{m} D_{T+1}(i) = 1$ se justifică prin faptul că D_{T+1} reprezintă o distribuţie probabilistă.

c. Vom porni de la relaţia (246), care a fost demonstrată la problema 22.*ii*, şi anume $Z_t = \varepsilon_t\cdot e^{\alpha_t} + (1-\varepsilon_t)\cdot e^{-\alpha_t}$. Pentru a facilita înţelegerea raţionamentului care urmează, întrucât aici α_t este lăsat liber, îl vom renota cu α, deci expresia anterioară va fi văzută ca funcţie de acest α, adică $Z(\alpha) = \varepsilon_t\cdot e^{\alpha} + (1-\varepsilon_t)\cdot e^{-\alpha}$. (Observaţi că Z_t dinainte tocmai a fost şi el renotat mai simplu: Z.)

Ca de obicei, pentru a găsi minimul expresiei $\varepsilon_t\cdot e^{\alpha} + (1-\varepsilon_t)\cdot e^{-\alpha}$, în care ε_t (eroarea produsă de ipoteza care tocmai a fost produsă de clasificatorul „slab“ A) este considerată constantă, vom calcula derivata ei în raport cu α şi apoi vom egala cu 0 această derivată:

$$\frac{\partial}{\partial\alpha}\left(\varepsilon_t\cdot e^{\alpha} + (1-\varepsilon_t)\cdot e^{-\alpha}\right) = 0 \Leftrightarrow \varepsilon_t\cdot e^{\alpha} - (1-\varepsilon_t)\cdot e^{-\alpha} = 0 \Leftrightarrow$$

$$\varepsilon_t\cdot (e^{\alpha})^2 = 1-\varepsilon_t \Leftrightarrow e^{2\alpha} = \frac{1-\varepsilon_t}{\varepsilon_t} \Leftrightarrow 2\alpha = \ln\frac{1-\varepsilon_t}{\varepsilon_t} \Leftrightarrow \alpha = \frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t} = \ln\sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}.$$

Remarcaţi faptul că fracţia $\dfrac{1-\varepsilon_t}{\varepsilon_t}$ este pozitivă (deci i se poate aplica logaritmul), fiindcă $\varepsilon_t \in (0,\, 1/2)$. Chiar mai mult, $\alpha > 0$, fiindcă $\dfrac{1-\varepsilon_t}{\varepsilon_t} > 1$, datorită aceluiaşi motiv ca mai înainte. Se poate verifica imediat (analizând semnele derivatei de mai sus) că $\alpha = \dfrac{1}{2}\ln\dfrac{1-\varepsilon_t}{\varepsilon_t}$ este într-adevăr punctul în care se atinge *minimul* expresiei $\varepsilon_t\cdot e^{\alpha} + (1-\varepsilon_t)\cdot e^{-\alpha}$, deci şi al lui Z_t (văzut ca funcţie de α):

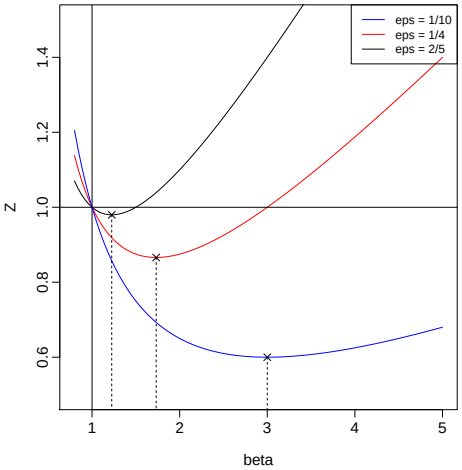
$$\varepsilon_t\cdot e^{\alpha} - (1-\varepsilon_t)\cdot e^{-\alpha} > 0 \Leftrightarrow e^{2\alpha} - \frac{1-\varepsilon_t}{\varepsilon_t} > 0 \Leftrightarrow \alpha > \frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t},$$

ceea ce înseamnă că funcţia $Z(\alpha) = \varepsilon_t\cdot e^{\alpha} + (1-\varepsilon_t)\cdot e^{-\alpha}$ este strict crescătoare pe intervalul $\left(\dfrac{1}{2}\ln\dfrac{1-\varepsilon_t}{\varepsilon_t}, +\infty\right)$ şi strict descrescătoare pe intervalul $\left(-\infty, \dfrac{1}{2}\ln\dfrac{1-\varepsilon_t}{\varepsilon_t}\right]$.

Pentru a ilustra grafic acest rezultat, vom renota e^{α} cu β, deci vom rescrie Z ca o funcţie de argumentul β:

$$Z(\beta) = \varepsilon_t \cdot \beta + (1 - \varepsilon_t) \cdot \frac{1}{\beta}.$$

Graficul funcţiei $Z(\beta)$ pentru trei valori diferite ale lui ε_t este prezentat în figura alăturată. Conform raţionamentului de mai sus, minimul funcţiei $Z(\beta)$ este atins pentru valoarea $\beta_{min} = \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}$. Vechea expresie a lui Z_t o vom regăsi astfel: $Z_t \overset{not.}{=} Z(\beta_{min}) \overset{calcul}{=} 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$.[527]

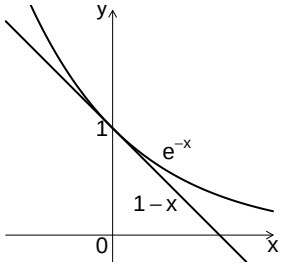


d. Folosind relaţia (249) care a fost dedusă la problema 22.*iii* şi apoi relaţia dată în enunţ pentru a exprima legătura dintre ε_t şi „ecartul“ γ_t, vom putea scrie:

$$\begin{aligned}\prod_{t=1}^{T} Z_t &= \prod_{t=1}^{T} 2 \cdot \sqrt{\varepsilon_t(1-\varepsilon_t)} = \prod_{t=1}^{T} 2 \cdot \sqrt{\left(\frac{1}{2} - \gamma_t\right)\left(1 - \left(\frac{1}{2} - \gamma_t\right)\right)} \\ &= \prod_{t=1}^{T} 2 \cdot \sqrt{\left(\frac{1}{2} - \gamma_t\right)\left(\frac{1}{2} + \gamma_t\right)} = \prod_{t=1}^{T} 2 \cdot \sqrt{\left(\frac{1}{4} - \gamma_t^2\right)} = \prod_{t=1}^{T} \sqrt{1 - 4\gamma_t^2}.\end{aligned}$$

Utilizând inegalitatea din *Sugestia* (2) din enunţ, rezultă $1 - 4\gamma_t^2 \leq e^{-4\gamma_t^2}$ şi, mai departe:

$$\begin{aligned}\prod_{t=1}^{T} Z_t &\leq \prod_{t=1}^{T} \sqrt{e^{-4\gamma_t^2}} = \prod_{t=1}^{T} \sqrt{(e^{-2\gamma_t^2})^2} = \prod_{t=1}^{T} e^{-2\gamma_t^2} \\ &= e^{-2\sum_{t=1}^{T} \gamma_t^2}.\end{aligned}$$



[527] Iată alte câteva *proprietăţi* ale funcţiei $Z(\beta)$, care pot fi demonstrate cu uşurinţă (şi care sunt reflectate în graficul de mai sus):

– $\lim_{\beta \searrow 0} Z(\beta) = +\infty$.

– $\lim_{\beta \to 1} Z(\beta) = Z(1) = \varepsilon_t + 1 - \varepsilon_t = 1$.

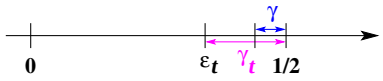
– $\lim_{\beta \to +\infty} Z(\beta) = +\infty$.

– Dreapta de ecuaţie $y = \varepsilon_t \beta$ este asimptotă oblică la $+\infty$ pentru funcţia $Z(\beta)$, fiindcă $\lim_{\beta \to +\infty} \frac{Z(\beta)}{\beta} = \varepsilon_t$, iar $\lim_{\beta \to +\infty}(Z(\beta) - \varepsilon_t \beta) = 0$.

– Atunci când $\varepsilon_t \nearrow \frac{1}{2}$, rezultă că $\sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}} \searrow 1$, deci $\ln \sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}} \searrow 0$, adică $\alpha_t \searrow 0$ (şi, deci, $\beta_{min} \searrow 1$). Totodată, $Z_t \overset{not.}{=} Z(\beta_{min}) = 2\sqrt{\varepsilon_t(1-\varepsilon_t)} \nearrow 1$.

– Atunci când $\varepsilon_t \searrow 0$, rezultă pe de o parte că $\sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}} \nearrow +\infty$, deci $\alpha_t \to +\infty$ (şi $\beta_{min} \nearrow +\infty$), iar pe de altă parte că $Z_t \overset{not.}{=} Z(\beta_{min}) = 2\sqrt{\varepsilon_t(1-\varepsilon_t)} \searrow 0$.

e. Combinând rezultatele pe care le-am obţinut la punctele c şi e, rezultă $err_S(H_T) \leq e^{-2\sum_{t=1}^{T}\gamma_t^2}$. Apoi, ţinând cont că $\gamma \leq \gamma_t$ pentru $t = 1, 2, \ldots$, vom avea:[528]



$$err_S(H_T) \leq e^{-2T\gamma^2}. \tag{251}$$

Prin urmare, pentru ca inegalitatea $err_S(H_T) < \varepsilon$ să aibă loc, este suficient să impunem restricţia următoare:

$$-2T\gamma^2 < \ln\varepsilon \Leftrightarrow 2T\gamma^2 > -\ln\varepsilon \Leftrightarrow 2T\gamma^2 > \ln\frac{1}{\varepsilon} \Leftrightarrow T > \frac{1}{2\gamma^2}\ln\frac{1}{\varepsilon}.$$

Aşadar, $T = \mathcal{O}\left(\frac{1}{\gamma^2}\ln\frac{1}{\varepsilon}\right)$.

În particular, atunci când considerăm $\varepsilon = \frac{1}{m}$, obţinem numărul T de iteraţii începând de la care [cu certitudine] vom avea $err_S(H_{T'}) = 0$ pentru orice $T' \geq T$: $T = \left\lceil \frac{1}{2\gamma^2}\ln m \right\rceil$.[529]

***Observaţie*: Din relaţia (251) rezultă că $err_S(H_T) \to 0$ atunci când $T \to \infty$.**

24. **(Algoritmul AdaBoost, folosind "decision stumps": aplicare pe date din $\mathbb{R}^2$)**

prelucrare de L. Ciortuz, după

■ ● ○ ***CMU, 2015 fall, Z. Bar-Joseph, E. Xing, HW4, pr. 2.6***

Considerăm setul de date de antrenament din tabelul de mai jos.

Rulaţi $T = 3$ iteraţii ale algoritmului AdaBoost, folosind drept clasificatori slabi *compaşi de decizie* (engl., decision stumps), care determină separatori liniari paraleli cu axele de coordonate (engl., axis-aligned separators). Reprezentaţi aceste date în planul euclidian; pe figura obţinută veţi trasa dreptele corespunzătoare ipotezelor „slabe" h_t,[530] **iar la final veţi completa tabelul de mai jos. Pentru pseudo-codul algoritmului AdaBoost, vedeţi problema 22. Vă rugăm să citiţi şi să reţineţi *Observaţia importantă* pe care am redactat-o imediat după acel pseudo-cod.**

x_i	X_1	X_2	y_i
x_1	1	2	+1
x_2	2	3	+1
x_3	3	4	−1
x_4	3	2	−1
x_5	3	1	−1
x_6	4	4	−1
x_7	5	4	−1
x_8	5	2	+1
x_9	5	1	+1

[528] Întrucât $\gamma_t \geq \gamma > 0$, au loc următoarele echivalenţe: $\gamma_t \geq \gamma \Leftrightarrow \gamma_t^2 \geq \gamma^2 \Leftrightarrow -2\gamma_t^2 \leq -2\gamma^2 \Leftrightarrow e^{-2\gamma_t^2} \leq e^{-2\gamma^2}$. Considerând această ultimă inegalitate pentru fiecare valoare a lui $t \in \{1, 2, \ldots\ T\}$ şi înmulţind membru cu membru, rezultă: $\prod_{t=1}^{T} e^{-2\gamma_t^2} \leq \prod_{t=1}^{T} e^{-2\gamma^2} \Leftrightarrow e^{-2\sum_{t=1}^{T}\gamma_t^2} \leq (e^{-2\gamma^2})^T \Leftrightarrow e^{-2\sum_{t=1}^{T}\gamma_t^2} \leq e^{-2T\gamma^2}$.

[529] *Observaţie*: Dacă toate instanţele de antrenament x_i sunt corect clasificate (de către ipoteza combinată $H_S(t)$) la iteraţia t, asta nu înseamnă că toate instanţele vor fi corect clasificate şi la iteraţia următoare (de către $H_S(t+1)$), fiindcă eroarea la antrenare $err_S(H_t)$ nu este în mod neapărat descrescătoare (în raport cu t). (Vedeţi de exemplu problema de tip implementare de la Stanford, 2016 fall, A. Ng, J. Duchi, HW2, pr. 6.d.) Însă, dacă pragul inferior γ pentru γ_t se menţine pentru orice $t = t_0, t_0+1, \ldots, t_0+T$, unde $t_0 \in \mathbb{N}$, iar T se calculează ca mai sus, atunci avem certitudinea că eroarea la antrenare $err_S(H_t)$ va rămâne 0 pentru orice $t \geq t_0 + T$.

[530] Însoţiţi fiecare dintre dreptele respective cu o pereche de semne + şi −, corespunzătoare zonelor de decizie determinate.

t	ε_t	α_t	$D_t(1)$	$D_t(2)$	$D_t(3)$	$D_t(4)$	$D_t(5)$	$D_t(6)$	$D_t(7)$	$D_t(8)$	$D_t(9)$	$err_S(H_T)$
1												
2												
3												

***Recomandare*:** Rolul acestui exerciţiu este de a vă ajuta să înţelegeţi pas cu pas cum anume lucrează în practică algoritmul AdaBoost. Vă sugerăm ca, după ce veţi fi înţeles rezolvarea acestui exerciţiu, să implementaţi mai întâi un program / o funcţie, care să calculeze *eroarea ponderată la antrenare* (engl., weighted training error) produsă de un anumit compas de decizie, în raport cu o distribuţie de probabilitate (D) definită pe setul de date de antrenament. Ulterior, puteţi extinde acest program la o implementare completă a algoritmului AdaBoost, conform pseudo-codului din problema 22.

Răspuns:

Faţă de reprezentarea grafică (de tip arbore de adâncime 1) cu care ne-am obişnuit în trecut pentru *compaşii de decizie*, aici vom lucra cu *reprezentarea analitică* următoare: pentru un atribut de tip continuu X care ia valori $x \in \mathbb{R}$ şi pentru un prag de separare (engl., split threshold) oarecare $s \in \mathbb{R}$ fixat, putem defini doi compaşi de decizie ca funcţii (mai exact, ca *funcţii-treaptă*) de variabila x:

$$sign(x-s) = \begin{cases} 1 & \text{dacă } x \geq s \\ -1 & \text{dacă } x < s \end{cases} \quad \text{şi} \quad sign(s-x) = \begin{cases} -1 & \text{dacă } x \geq s \\ 1 & \text{dacă } x < s. \end{cases}$$

Pentru convenienţă, în cele ce urmează — atunci când nu riscăm să creăm ambiguităţi — vom renota primul compas de decizie cu $X \geq s$ şi al doilea cu $X < s$.

Faptul că în acest exerciţiu se cere aplicarea algoritmului AdaBoost cu compaşi de decizie înseamnă că la fiecare iteraţie (t) clasificatorul „slab“ (notat cu A) din pseudo-codul algoritmului AdaBoost selectează un compas de decizie, şi anume — conform *Observaţiei importante* din enunţul problemei 22 — unul care are eroarea ponderată la antrenare minimă în raport cu distribuţia probabilistă curentă (D_t).[531]

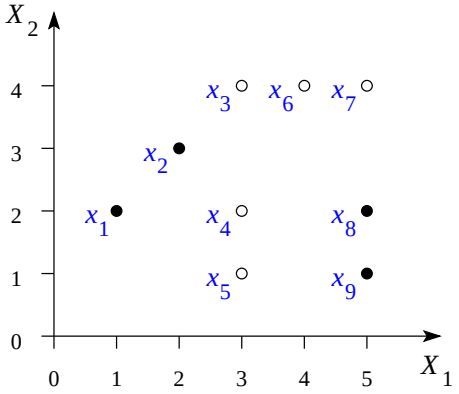
Exact ca atunci când am lucrat cu algoritmul ID3 cu atribute continue, vom considera câte un prag intermediar pentru fiecare pereche de valori succesive [luate de un atribut continuu X] corespunzătoare schimbărilor de etichete. Formal, pentru orice astfel de perechi de valori x_i, x_{i+1}, cu $y_i y_{i+1} < 0$ şi $x_i < x_{i+1}$, dar fără să mai existe vreun $x_j \in Val(X)$ cu proprietatea $x_i < x_j < x_{i+1}$, vom considera pragul $(x_i + x_{i+1})/2$.[532] **De asemenea, vom considera şi un *prag exterior* intervalului de valori ale atributului X în mulţimea de instanţe de antrenament.**[533]

[531] Deci algoritmul slab A nu este algoritmul ID3 care produce un compas de decizie cu câştig de informaţie minim!

[532] În cazul algoritmului ID3, există un *rezultat teoretic* (vedeţi problema 47) care demonstrează că nu este necesar să considerăm alte praguri pentru un atribut continuu X decât cele situate între perechi de valori succesive având etichete [de semne] contrare, fiindcă valoarea IG-ului celorlalte praguri este situată în mod cert sub valoarea IG-ului maximal pentru acel atribut. LC: În cazul algoritmului AdaBoost, se poate demonstra un rezultat similar care, în consecinţă, ne va permite să simplificăm aplicarea clasificatorului „slab“ (A).

[533] Compaşii de decizie corespunzători acestui prag „exterior“ pot fi puşi în corespondenţă cu arborii de decizie de adâncime 0 pe care i-am întâlnit în precedent.

Aşadar, la *prima iteraţie* a algoritmului AdaBoost (adică, pentru $t = 1$), *pragurile* de separare pentru valorile celor două variabile continue (X_1 şi X_2) care corespund celor două coordonate ale instanţelor de antrenament ($x_1, \ldots, x_9$) sunt $\frac{1}{2}$, $\frac{5}{2}$ şi $\frac{9}{2}$ pentru X_1, şi respectiv $\frac{1}{2}$, $\frac{3}{2}$, $\frac{5}{2}$ şi $\frac{7}{2}$ pentru X_2.



Observăm însă că se poate renunţa la pragul „exterior“ $\frac{1}{2}$ pentru X_2, deoarece compaşii de decizie corespunzător lui se comportă identic cu compaşii de decizie corespunzător pragului „exterior“ $\frac{1}{2}$ pentru X_1.[534]

Pentru compaşii de decizie corespunzători acestei prime iteraţii, *erorile ponderate la antrenare* sunt cele prezentate centralizat în tabelele de mai jos. Pentru calcule, am folosit egalităţile $err_{D_t}(X_1 \geq s) = 1 - err_{D_t}(X_1 < s)$ şi, similar, $err_{D_t}(X_2 \geq s) = 1 - err_{D_t}(X_2 < s)$, pentru orice prag s şi orice iteraţie $t = 1, 2, \ldots$. Aceste egalităţi sunt foarte uşor de demonstrat.

s	$\frac{1}{2}$	$\frac{5}{2}$	$\frac{9}{2}$
$err_{D_1}(X_1 < s)$	$\frac{4}{9}$	$\frac{2}{9}$	$\frac{4}{9} + \frac{2}{9} = \frac{2}{3}$
$err_{D_1}(X_1 \geq s)$	$\frac{5}{9}$	$\frac{7}{9}$	$\frac{1}{3}$

s	$\frac{1}{2}$	$\frac{3}{2}$	$\frac{5}{2}$	$\frac{7}{2}$
$err_{D_1}(X_2 < s)$	$\frac{4}{9}$	$\frac{1}{9} + \frac{3}{9} = \frac{4}{9}$	$\frac{2}{9} + \frac{1}{9} = \frac{1}{3}$	$\frac{2}{9}$
$err_{D_1}(X_2 \geq s)$	$\frac{5}{9}$	$\frac{5}{9}$	$\frac{2}{3}$	$\frac{7}{9}$

Se observă că eroarea ponderată minimă la antrenare ($\varepsilon_1 = 2/9$) este obţinută pentru compaşii de decizie $X_1 < 5/2$ şi $X_2 < 7/2$.[535] **Alegem drept *cea mai bună ipoteză* la această iteraţie pe $h_1 = sign\left(\frac{7}{2} - X_2\right)$, separatorul corespunzător fiind dreapta de ecuaţie $X_2 = \frac{7}{2}$. Ipoteza h_1 clasifică greşit instanţele x_4 şi x_5. Vom avea:**

$$\gamma_1 = \frac{1}{2} - \frac{2}{9} = \frac{5}{18}$$

$$\alpha_1 = \frac{1}{2} \ln \frac{1 - \varepsilon_1}{\varepsilon_1} = \ln \sqrt{\left(1 - \frac{2}{9}\right) : \frac{2}{9}} = \ln \sqrt{\frac{7}{2}} \approx 0.626$$

[534] De aceea în tabelele de mai jos, pentru aceste cazuri se foloseşte culoarea verde.
[535] În tabele, pentru erorile minime am folosit culoarea albastră.

Acum algoritmul trebuie să „pregătească“ o nouă distribuţie (D_2), pentru iteraţia următoare. Distribuţia D_2 va fi obţinută prin modificarea vechii distribuţii (D_1), în aşa fel încât algoritmul să se poată concentra cu preponderenţă asupra instanţelor care au fost greşit clasificate (engl., misclassified). Conform relaţiei (245), vom scrie:

$$D_2(i) = \frac{1}{Z_1} D_1(i) \underbrace{(e^{-\alpha_1})}_{\sqrt{2/7}}{}^{y_i\, h_1(x_i)} = \begin{cases} \frac{1}{Z_1} \cdot \frac{1}{9} \cdot \sqrt{\frac{2}{7}} & \text{pentru } i \in \{1,2,3,6,7,8,9\}; \\ \frac{1}{Z_1} \cdot \frac{1}{9} \cdot \sqrt{\frac{7}{2}} & \text{pentru } i \in \{4,5\}. \end{cases}$$

Vă reamintim că Z_1 este aşa-numitul *factor de normalizare* pentru distribuţia probabilistă D_2. Aşadar, din relaţia $\sum_i D_2(i) = 1$ rezultă:

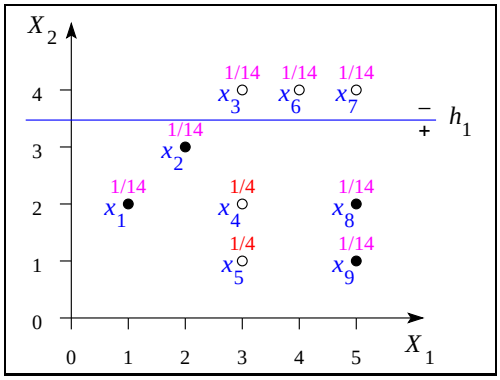
$$Z_1 = \frac{1}{9}\left(7 \cdot \sqrt{\frac{2}{7}} + 2 \cdot \sqrt{\frac{7}{2}}\right) = \frac{2\sqrt{14}}{9} = 0.8315$$

Prin urmare,

$$D_2(i) = \begin{cases} \frac{9}{2\sqrt{14}} \cdot \frac{1}{9} \cdot \sqrt{\frac{2}{7}} = \frac{1}{14} & \text{pentru } i \notin \{4,5\}; \\ \frac{9}{2\sqrt{14}} \cdot \frac{1}{9} \cdot \sqrt{\frac{7}{2}} = \frac{1}{4} & \text{pentru } i \in \{4,5\}. \end{cases}$$

Figura următoare prezintă grafic exemplele de antrenament (x_i, y_i), pentru $i = 1, \ldots, 9$, împreună cu probabilităţile care tocmai le-au fost asociate ($D_2(x_i)$), precum şi separatorul „învăţat“ la această primă iteraţie (h_1).

Probabilităţile corespunzătoare instanţelor clasificate eronat de către h_1 au fost scrise cu culoarea roşie. Se observă că aceste probabilităţi sunt mult mai mari decât probabilităţile celorlalte instanţe şi că împreună se sumează la valoarea $1/2$. Am notat cu $+$ şi respectiv $-$ cele două zone de decizie determinate de h_1.



***Observaţie* (1): Dacă, drept ipoteză h_1, în locul lui $sign\left(\frac{7}{2} - X_2\right)$ am fi ales $sign\left(\frac{5}{2} - X_1\right)$, atunci cursul rezolvării ulterioare ar fi fost altul (deşi ambele ipoteze au aceeaşi eroare ponderată – minimală – la antrenare): x_8 şi x_9 ar fi primit ponderile $1/4$, iar x_4 şi x_5 ar fi primit ponderile $1/14$. În consecinţă, outputul algoritmului AdaBoost nu este în mod neapărat unic determinat!**

Iteraţia $t = 2$:

Vom proceda similar cu iteraţia precedentă, doar că, de data aceasta, vom folosi distribuţia D_2.

s	$\frac{1}{2}$	$\frac{5}{2}$	$\frac{9}{2}$
$err_{D_2}(X_1 < s)$	$\frac{4}{14}$	$\frac{2}{14}$	$\frac{2}{14}+\frac{2}{4}+\frac{2}{14}=\frac{11}{14}$
$err_{D_2}(X_1 \geq s)$	$\frac{10}{14}$	$\frac{12}{14}$	$\frac{3}{14}$

s	$\frac{1}{2}$	$\frac{3}{2}$	$\frac{5}{2}$	$\frac{7}{2}$
$err_{D_2}(X_2 < s)$	$\frac{4}{14}$	$\frac{1}{4}+\frac{3}{14}=\frac{13}{28}$	$\frac{2}{4}+\frac{1}{14}=\frac{8}{14}$	$\frac{2}{4}=\frac{1}{2}$
$err_{D_2}(X_2 \geq s)$	$\frac{10}{14}$	$\frac{15}{28}$	$\frac{6}{14}$	$\frac{1}{2}$

***Observaţie* (2): Conform rezultatului teoretic de la punctul *vi* al problemei 22, luarea în calcul a compasului de decizie corespunzător testului $X_2 \geq 7/2$ este aici superfluă, întrucât acest compas de decizie a fost ales ca ipoteză optimală la iteraţia precedentă. L-am pus totuşi în tabel (folosind culoarea roşie), de dragul realizării unei prezentări exhaustive.**

Cea mai bună ipoteză este acum $h_2 = sign\left(\frac{5}{2} - X_1\right)$; separatorul corespunzător acestei ipoteze este dreapta de ecuaţie $X_1 = \frac{5}{2}$. Urmează să explicităm cum am calculat eroarea ε_2 şi, aferent, să calculăm „ecartul" γ_2, precum şi ponderea α_2, după care vom trece la determinarea noii distribuţii (D_3).

$$\varepsilon_2 = P_{D_2}(\{x_8, x_9\}) = \frac{2}{14} = \frac{1}{7} = 0.143, \quad \text{deci } \gamma_2 = \frac{1}{2} - \frac{1}{7} = \frac{5}{14}$$

$$\alpha_2 = \ln\sqrt{\frac{1-\varepsilon_2}{\varepsilon_2}} = \ln\sqrt{\left(1-\frac{1}{7}\right) : \frac{1}{7}} = \ln\sqrt{6} = 0.896$$

$$D_3(i) = \frac{1}{Z_2} \cdot D_2(i) \cdot \underbrace{(e^{-\alpha_2})}_{1/\sqrt{6}}{}^{y_i\, h_2(x_i)} = \begin{cases} \frac{1}{Z_2} \cdot D_2(i) \cdot \frac{1}{\sqrt{6}} & \text{dacă } h_2(x_i) = y_i; \\ \frac{1}{Z_2} \cdot D_2(i) \cdot \sqrt{6} & \text{în caz contrar} \end{cases}$$

$$= \begin{cases} \frac{1}{Z_2} \cdot \frac{1}{14} \cdot \frac{1}{\sqrt{6}} & \text{pentru } i \in \{1,2,3,6,7\}; \\ \frac{1}{Z_2} \cdot \frac{1}{4} \cdot \frac{1}{\sqrt{6}} & \text{pentru } i \in \{4,5\}; \\ \frac{1}{Z_2} \cdot \frac{1}{14} \cdot \sqrt{6} & \text{pentru } i \in \{8,9\}. \end{cases}$$

Din relaţia $\sum_i D_3(i) = 1$ rezultă:

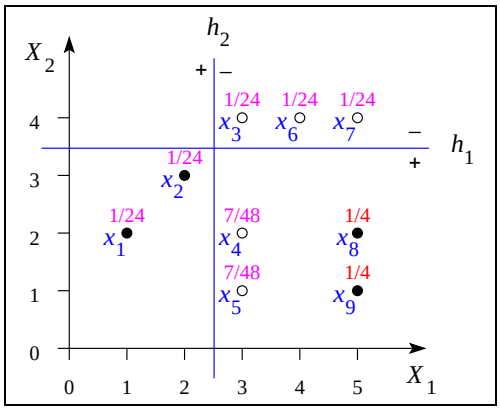
$$Z_2 = 5 \cdot \frac{1}{14} \cdot \frac{1}{\sqrt{6}} + 2 \cdot \frac{1}{4} \cdot \frac{1}{\sqrt{6}} + 2 \cdot \frac{1}{14} \cdot \sqrt{6} = \frac{5}{14\sqrt{6}} + \frac{1}{2\sqrt{6}} + \frac{\sqrt{6}}{7} = \frac{12+12}{14\sqrt{6}}$$

$$= \frac{24}{14\sqrt{6}} = \frac{2\sqrt{6}}{7} \approx 0.7$$

şi, prin urmare

$$D_3(i) = \begin{cases} \frac{7}{2\sqrt{6}} \cdot \frac{1}{14} \cdot \frac{1}{\sqrt{6}} = \frac{1}{24} & \text{pentru } i \in \{1,2,3,6,7\}; \\ \frac{7}{2\sqrt{6}} \cdot \frac{1}{4} \cdot \frac{1}{\sqrt{6}} = \frac{7}{48} & \text{pentru } i \in \{4,5\}; \\ \frac{7}{2\sqrt{6}} \cdot \frac{1}{14} \cdot \sqrt{6} = \frac{1}{4} & \text{pentru } i \in \{8,9\}. \end{cases}$$

Facem din nou reprezentarea grafică a datelor de antrenament, împreună cu noile probabilităţi asociate (D_3) şi cei doi separatori care au fost „învăţaţi“ până acum (h_1 şi h_2); vedeţi figura alăturată. Este instructiv să comparăm această figură cu cea dinainte (adică, pentru $t = 1$), pentru a observa evoluţia probabilităţilor de la o iteraţie la alta. Instanţele x_1, x_2, x_3, x_6 şi x_7 au acum ponderile foarte mici, pentru că au fost clasificate corect atât de către h_1 cât şi de către h_2.



Iteraţia $t = 3$:

Erorile ponderate la antrenare pentru compaşii de decizie, în raport cu distribuţia D_3 sunt:

s	$\frac{1}{2}$	$\frac{5}{2}$	$\frac{9}{2}$
$err_{D_3}(X_1 < s)$	$\frac{2}{24} + \frac{2}{4} = \frac{7}{12}$	$\frac{2}{4}$	$\frac{2}{24} + 2 \cdot \frac{7}{48} + 2 \cdot \frac{1}{4} = \frac{21}{24}$
$err_{D_3}(X_1 \geq s)$	$\frac{5}{12}$	$\frac{2}{4}$	$\frac{3}{24} = \frac{1}{8}$

s	$\frac{1}{2}$	$\frac{3}{2}$	$\frac{5}{2}$	$\frac{7}{2}$
$err_{D_3}(X_2 < s)$	$\frac{7}{12}$	$\frac{7}{48} + \frac{2}{24} + \frac{1}{4} = \frac{23}{48}$	$2 \cdot \frac{7}{48} + \frac{1}{24} = \frac{1}{3}$	$2 \cdot \frac{7}{48} = \frac{7}{24}$
$err_{D_3}(X_2 \geq s)$	$\frac{5}{12}$	$\frac{25}{48}$	$\frac{2}{3}$	$\frac{17}{24}$

Similar cu *Observaţia* precedentă (vedeţi iteraţia $t = 2$), facem menţiunea că am fi putut renunţa la elaborarea compasului de decizie corespunzător testului $X_1 < 5/2$.

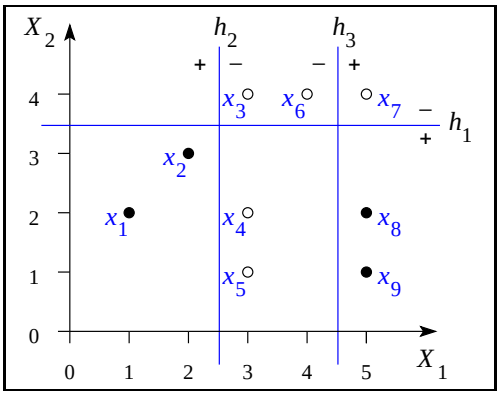
Acum cea mai bună ipoteză este $h_3 = sign\left(X_1 - \frac{9}{2}\right)$, iar separatorul corespunzător este dreapta de ecuaţie $X_1 = \frac{9}{2}$. Vom explicita cum am calculat eroarea ε_3 şi apoi vom calcula „ecartul“ γ_3, precum şi ponderea α_3:

$$\varepsilon_3 = P_{D_3}(\{x_1, x_2, x_7\}) = 2 \cdot \frac{1}{24} + \frac{1}{24} = \frac{3}{24} = \frac{1}{8}$$

$$\gamma_3 = \frac{1}{2} - \frac{1}{8} = \frac{3}{8}$$

$$\alpha_3 = \ln \sqrt{\frac{1-\varepsilon_3}{\varepsilon_3}} = \ln \sqrt{\left(1 - \frac{1}{8}\right) : \frac{1}{8}} = \ln \sqrt{7} = 0.973$$

Reprezentarea grafică a datelor de antrenament împreună cu cei trei separatori învăţaţi este furnizată în figura alăturată.



În final, vom pune rezultatele pe care le-am obţinut până acum în tabelul care a fost dat în enunţ.

t	ε_t	α_t	$D_t(1)$	$D_t(2)$	$D_t(3)$	$D_t(4)$	$D_t(5)$	$D_t(6)$	$D_t(7)$	$D_t(8)$	$D_t(9)$	$err_S(H_t)$
1	2/9	$\ln\sqrt{7/2}$	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	1/9	2/9
2	2/14	$\ln\sqrt{6}$	1/14	1/14	1/14	1/4	1/4	1/14	1/14	1/14	1/14	2/9
3	1/8	$\ln\sqrt{7}$	1/24	1/24	1/24	7/48	7/48	1/24	1/24	1/4	1/4	0

***Observaţie* (3): Tabelul de mai jos vă serveşte ca să înţelegeţi cum anume se calculează $err_S(H_t)$. În acest tabel, am marcat cu culoarea roşie clasificările eronate făcute de ipotezele h_t, pentru $t = 1, 2, 3$.**

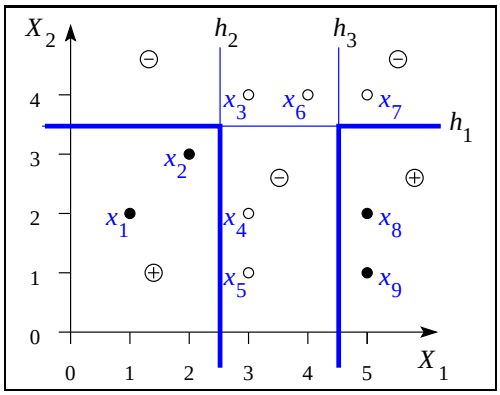
t	α_t	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
1	0.626	+1	+1	−1	+1	+1	−1	−1	+1	+1
2	0.896	+1	+1	−1	−1	−1	−1	−1	−1	−1
3	0.973	−1	−1	−1	−1	−1	−1	+1	+1	+1
	$H_T(x_i)$	+1	+1	−1	−1	−1	−1	−1	+1	+1

Se observă că H_1 (deci, la finalul primei iteraţii a algoritmului AdaBoost) a clasificat greşit instanţele x_4 şi x_5, iar H_2 (deci, la finalul celei de-a doua iteraţii) instanţele x_8 şi x_9. La finalul celei de-a treia iteraţii, eroarea produsă pe datele de antrenament (de către ipoteza H_3) este 0. Vă readucem aminte că $H_T(x_i) \stackrel{def.}{=} sign\left(\sum_{t=1}^{T} \alpha_t\, h_t(x_i)\right)$, deci notând $x_i = (x_{i,1}, x_{i,2})$, putem scrie

$$H_3(x_i) = sign\left(\ln\sqrt{\frac{7}{2}} \cdot sign(\frac{7}{2} - x_{i,2}) + \ln\sqrt{6} \cdot sign(\frac{5}{2} - x_{i,1}) + \ln\sqrt{7} \cdot sign(x_{i,1} - \frac{9}{2})\right).$$

***Observaţie* (4): Se poate constata imediat că o instanţă nouă aleasă în mod arbitrar în partea din stânga sus a figurii care reprezintă datele de antrenament (de exemplu, instanţa $(1, 4)$) va fi clasificată de către ipoteza H_3 — care a fost „învăţată“ de către algoritmul AdaBoost după cele trei iteraţii — ca fiind negativă, fiindcă $-\alpha_1 + \alpha_2 - \alpha_3 = -0.626 + 0.896 - 0.973 < 0$.**

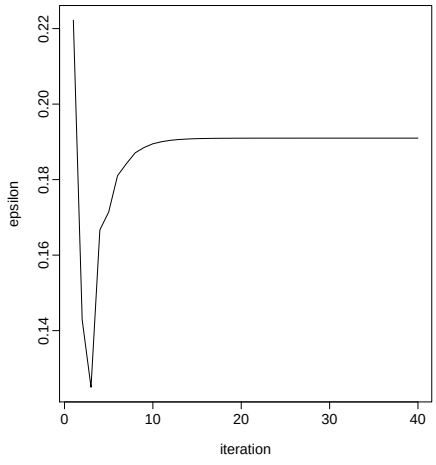
Făcând şi alte raţionamente similare, putem conchide că *zonele de decizie* şi *graniţele de decizie* produse de către AdaBoost pentru acest set de date de antrenament vor fi cele indicate în figura alăturată.



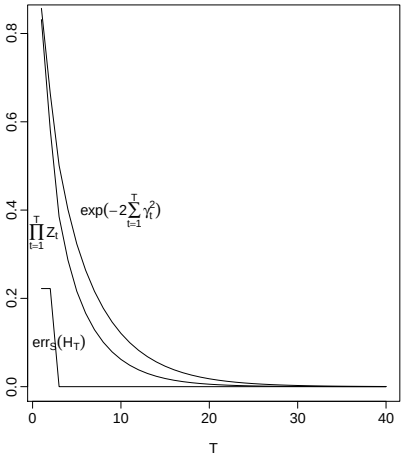
***Observaţie* (5): Execuţia algoritmului AdaBoost ar putea continua (dacă am fi fixat iniţial un $T > 3$), chiar dacă am obţinut $err_S(H_t) = 0$ la iteraţia $t = 3$. Dacă veţi elabora detaliile, veţi vedea că pentru $t = 4$ am obţine ca ipoteză optimă $X_2 < \frac{7}{2}$ (care a fost selectată şi la iteraţia $t = 1$). Această ipoteză ar produce acum la antrenare eroarea ponderată $\varepsilon_4 = \frac{1}{6}$, deci ar primi în noul output H_4 factorul $\alpha_4 = \ln\sqrt{5}$ (care s-ar alătura factorului $\alpha_1 = \ln\sqrt{7/2}$). Astfel se va întări încrederea în ipoteza $X_2 < \frac{7}{2}$. Să reţinem, deci, că algoritmul AdaBoost poate selecta de mai multe ori o aceeaşi ipoteză „slabă“ (însă niciodată la iteraţii consecutive, conform pr. 22.*vi*).**

În final prezentăm două grafice utile, care au fost realizate de către studentul Sebastian Ciobanu:

Evoluţia erorii ε_t în raport cu t:



Cele două margini superioare (eng., upper bounds) pentru eroarea empirică a ipotezei combinate H_T (cf. ex. 23.bd):

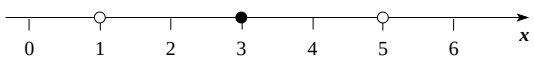


25. **(AdaBoost şi *non*-învăţabilitate γ-slabă: exemplificare pe un set de date din $\mathbb{R}$)**

prelucrare de Liviu Ciortuz, după

■ • ○ ***CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, final, pr. 8.a-e***

În această problemă vom studia modul în care se comportă algoritmul AdaBoost pe un set de date foarte simplu din $\mathbb{R}$, $x_1 = 1, x_2 = 3, x_3 = 5$, ilustrat în figura alăturată.



Vă readucem aminte *convenţia* noastră de notare: simbolul • desemnează instanţe pozitive, iar simbolul ○ instanţe negative.

Vom folosi ca ipoteze „slabe" (h_t) compaşi de decizie. Vă readucem aminte că un compas de decizie alege o valoare reală constantă s şi clasifică toate punctele $x > s$ ca fiind într-o clasă, iar toate celelalte puncte ($x \leq s$) în cealaltă clasă.

a. *i*. **Care este valoarea ponderilor / probabilităţilor asignate iniţial fiecărei instanţe?** *ii*. **Marcaţi pe desenul de mai sus separatorul decizional (graniţa) corespunzătoare primului compas de decizie. Indicaţi zona pozitivă şi zona negativă, de o parte şi de alta a separatorului decizional.** *iii*. **Încercuiţi instanţa a cărei pondere / probabilitate creşte la execuţia primei iteraţii de boosting.** *iv*. **Calculaţi ponderile / probabilităţile asignate fiecărei instanţe după prima iteraţie.**

b. Poate oare algoritmul AdaBoost să clasifice în mod perfect toate aceste exemple de antrenament? Dacă răspunsul este *nu*, justificaţi în detaliu. Dacă răspunsul este *da*, care este numărul minim de iteraţii în care se atinge eroarea la antrenare 0?

Răspuns:

***Observaţie*: La stabilirea distribuţiilor probabiliste D_{t+1} (pentru $t = 1, 2, \ldots$), vom ţine cont**

– în primul rând de rezultatul de la exerciţiul 22.*vi*, şi anume că $D_{t+1}(h_t) = 1/2$,

– iar apoi de faptul — specificat în relaţia de definiţie (245) pentru distribuţia D_{t+1} — că probabilităţile instanţelor de antrenament care au fost corect (respectiv incorect) catalogate de către ipoteza slabă h_t vor fi modificate (pentru iteraţia $t+1$) proporţional, şi anume cu factorul $e^{-\alpha_t}$ (respectiv e^{α_t}).[536]

a. La iteraţia $t = 1$, se lucrează cu distribuţia probabilistă $D_1(i) = 1/3$ pentru $i \in \{1, 2, 3\}$ şi, indiferent de cum am fixa pragul compasului de decizie h_1,[537] **— în legătură cu care nu precizăm deocamdată decât că trebuie să aibă eroarea ponderată la antrenare minimală, ca de obicei — o [singură] instanţă este**

[536] Mai precis, folosind notaţiile de la ex. 22, $D_{t+1}(i) = \frac{1}{Z_t} D_t(i)\, e^{-\alpha_t}$, pentru $i \in C$, şi $D_{t+1}(i) = \frac{1}{Z_t} D_t(i)\, e^{\alpha_t}$, pentru $i \in M$.

[537] Atât pentru varianta când se foloseşte prag exterior (engl., outside threshold) cât şi pentru varianta când *nu* se foloseşte prag exterior. Vedeţi precizările de mai jos.

incorect clasificată. Dacă folosim praguri exterioare, pentru ipoteza slabă h_1 putem alege pragul 0 (vedeţi primul desen de mai jos). În acest caz, instanţa incorect clasificată este $x_2 = 3$.

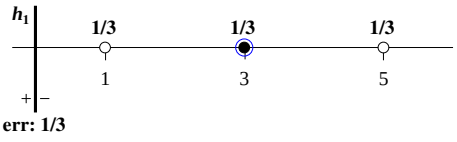
Pentru iteraţia $t + 1 = 2$, instanţa x_2 va primi (cf. rezultatului teoretic de la ex. 22.*vi*) ponderea / probabilitatea $1/2$, în vreme ce probabilitatea care rămâne, şi anume $1 - (1/2) = 1/2$ va fi împărţită în mod egal între instanţele $x_1 = 1$ şi $x_3 = 5$; aşadar acestea vor primi acum ponderile $1/4$.

b. La iteraţia $t = 2$, instanţa incorect clasificată de către ipoteza slabă h_2 (vedeţi detaliile din al doilea desen de mai jos) este $x_3 = 5$. În continuare, la ieraţia $t + 1 = 3$, instanţa x_3 va primi probabilitatea $1/2$, iar restul masei de probabilitate $(1/2)$ va fi împărţit în mod proporţional între instanţele $x_1 = 1$ şi $x_2 = 3$, deci acestea vor avea acum probabilităţile $\frac{1}{6} = \frac{1}{2} \cdot \frac{1}{3}$ şi respectiv $\frac{1}{3} = \frac{1}{2} \cdot \frac{2}{3}$.

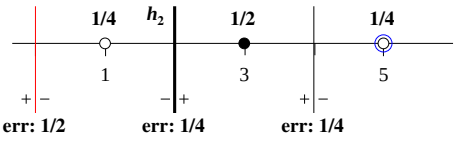
Erorile ponderate ε_t se calculează uşor, la fel şi ponderile α_t, aşa că vom reda acum în mod direct rezultatele obţinute la fiecare iteraţie (presupunând, ca şi mai sus, că folosim praguri exterioare).

Se observă că după $T = 3$ iteraţii, algoritmul AdaBoost obţine eroarea la antrenare $err_S(H_3) = 0$.

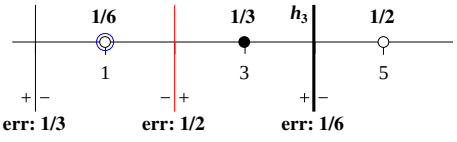
• $t = 1$:



• $t = 2$:



• $t = 3$:



• $t = 1$:
$\varepsilon_1 = 1/3 \Rightarrow \alpha_1 = \ln\sqrt{2} = 0.3465$,
$err_S(H_1) = 1/3$

• $t = 2$:
$\varepsilon_2 = 1/4 \Rightarrow \alpha_2 = \ln\sqrt{3} = 0.5493$

	x_1	x_2	x_3
α_1	$-$	$-$	$-$
α_2	$-$	$+$	$+$
$H_2(x_i)$	$-$	$+$	$+$

$\Rightarrow err_S(H_2) = 1/3$

• $t = 3$:
$\varepsilon_3 = 1/6 \Rightarrow \alpha_3 = \ln\sqrt{5} = 0.8047$

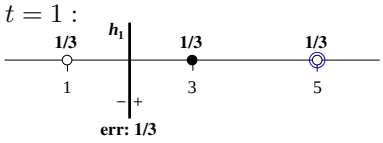
	x_1	x_2	x_3
α_1	$-$	$-$	$-$
α_2	$-$	$+$	$+$
α_3	$+$	$+$	$-$
$H_3(x_i)$	$-$	$+$	$-$

$\Rightarrow err_S(H_3) = 0$

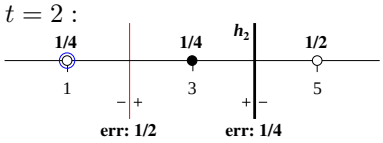
***Atenţie*! Întrucât se va dovedi a fi un fapt foarte instructiv, vom elabora acum o *a doua soluţie*, pentru care însă ne vom auto-impune din start să nu folosim praguri exterioare.[538] Iată soluţia obţinută de data aceasta (mai exact, rezultatul primelor 4 iteraţii):**

[538] Aceasta a fost, în esenţă, soluţia dată pentru acest exerciţiu la CMU, însă ea este greşită, iar când spunem aceasta ne bazăm pe rezultatul teoretic de la problema 72.d, cu care această soluţie (de la CMU) vine în contradiţie.

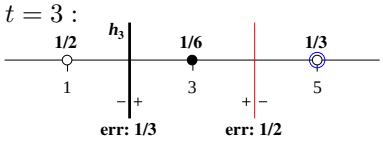
• $t = 1$:



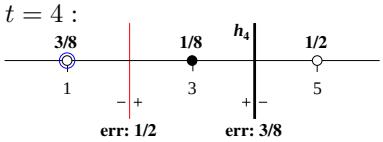
• $t = 2$:



• $t = 3$:



• $t = 4$:



• $t = 1$:
$\varepsilon_1 = 1/3 \Rightarrow \alpha_1 = \ln\sqrt{2} = 0.3465,\ err_S(H_1) = 1/3$

• $t = 2 : \varepsilon_2 = 1/4 \Rightarrow \alpha_2 = \ln\sqrt{3} = 0.5493$

	x_1	x_2	x_3
α_1	$-$	$+$	$+$
α_2	$+$	$+$	$-$
$H_2(x_i)$	$+$	$+$	$-$

$\Rightarrow err_S(H_2) = 1/3$

• $t = 3 : \varepsilon_3 = 1/3 = \varepsilon_1 \Rightarrow \alpha_3 = \ln\sqrt{2} = 0.3465 = \alpha_1$

	x_1	x_2	x_3
α_1	$-$	$+$	$+$
α_2	$+$	$+$	$-$
α_3	$-$	$+$	$+$
$H_3(x_i)$	$-$	$+$	$+$

$\Rightarrow err_S(H_3) = 1/3$

• $t = 4 : \varepsilon_4 = 3/8 \Rightarrow \alpha_4 = \ln\sqrt{5/3} = 0.2554$

	x_1	x_2	x_3
α_1	$-$	$+$	$+$
α_2	$+$	$+$	$-$
α_3	$-$	$+$	$+$
α_4	$+$	$+$	$-$
$H_4(x_i)$	$+$	$+$	$-$

$\Rightarrow err_S(H_4) = 1/3$

Se observă că

– la oricare două iteraţii consecutive, pragurile alese ($s = 2$ şi $s = 4$) vor alterna, conform unei consecinţe imediate de la relaţia demonstrată la ex. 22.*vi* (vedeţi *Observaţia* 2 de la pagina 542);

– instanţa $x_2 = 3$ este clasificată corect de către oricare ipoteză selectată, h_t, de aceea, conform relaţiei (245) ponderea ei va scădea mereu (observaţi valorile $1/3, 1/4, 1/6, 1/8$). Mai mult, se poate demonstra că această probabilitate tinde descrescător la 0.[539]

– conform aceleiaşi relaţii de la ex. 22.*vi*, va rezulta că la fiecare iteraţie una dintre instanţele $x_1 = 1$ şi $x_3 = 5$ va avea probabilitatea $1/2$, iar cealaltă instanţă va avea probabilitatea $1/2$ minus probabilitatea instanţei (corect calsificată) $x_2 = 3$. Aşadar, această ultimă probabilitate va tinde crescător la $1/2$. În consecinţă, în acest caz — adică, atunci când nu se lucrează cu praguri exterioare, pe acest set de date de antrenament — nu există *garanţie de învăţabilitate empirică slabă* ($\gamma > 0$).

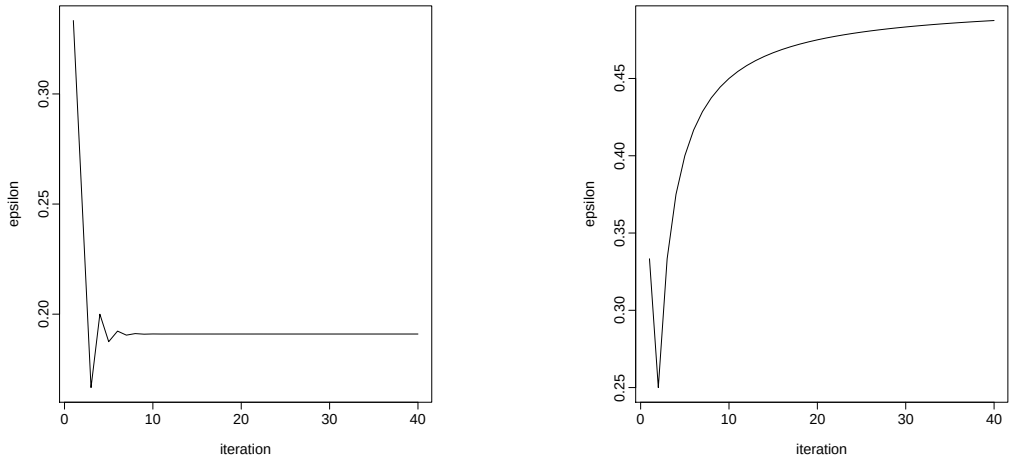
***Observaţie*: Se poate constata — din tabelele pentru $err_S(H_T)$ şi în special din ultimul tabel — că $H_T(x_1)$ va avea întotdeauna semnul contrar semnului lui $H_T(x_3)$, deoarece la fiecare iteraţie (t) instanţele x_1 şi x_3 vor fi clasificate cu semne contrare de către ipoteza slabă selectată, h_t. Prin urmare, în acest caz — repetăm, atunci când nu se lucrează cu praguri exterioare, pe acest set de date de antrenament — nu este posibil ca AdaBoost să obţină eroare la antrenare 0.**

[539] Vedeţi graficul de mai jos. Analitic, la această concluzie se poate ajunge prin reducere la absurd, pornind de la rezultatul teoretic obţinut la ex. 23.e, coroborat cu *Observaţia* de mai jos.

În *concluzie*, este absolut necesar ca în toate cazurile, la aplicarea algoritmului AdaBoost să fie folosite praguri exterioare (mai precis, este suficient un prag exterior). Altfel, este posibil ca AdaBoost să nu poată să clasifice perfect datele de antrenament.[540]

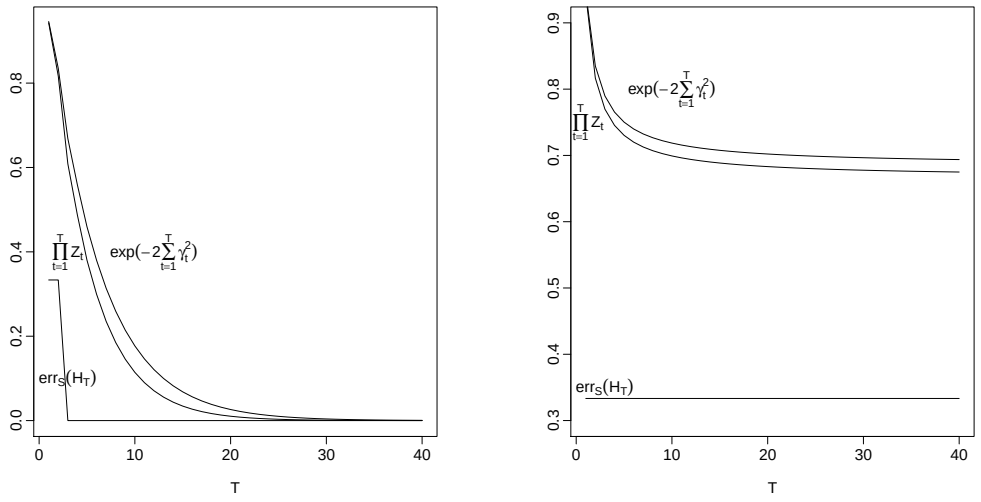
Graficele următoare, făcute de Sebastian Ciobanu, arată evoluţia valorilor *erorilor ponderate la antrenare* care au fost produse atunci când am rulat 40 de iteraţii cu algoritmul AdaBoost pe aceste date, în cele două variante: cu şi respectiv fără prag exterior. Se observă clar existenţa unui *prag* $\gamma > 0$ *de învăţabilitate empirică slabă* în primul caz, şi lipsa unui astfel de prag în cel de-al doilea caz.

cu prag exterior: **fără prag exterior:**



În sfârşit, următoarele grafice, făcute tot de Sebastian Ciobanu, pun în evidenţă legătura dintre rezultatele obţinute mai sus şi cele două *margini superioare* pentru *eroarea la antrenare* produsă de *ipoteza combinată* H_T livrată la ieşire de algoritmul AdaBoost pe aceste date (vedeţi ex. 23.bd).

cu prag exterior: **fără prag exterior:**



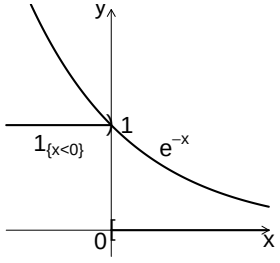
[540] Exerciţiul 67 pune în evidenţă o situaţie (adică, un set de date) pentru care AdaBoost nu obţine învăţabilitate empirică γ-slabă, chiar dacă se folosesc [şi] compaşi de decizie exteriori.

26. **(Algoritmul AdaBoost ca algoritm de *optimizare secvenţială*: deducerea regulii de calcul a ponderilor α_t folosind *metoda minimizării secvenţiale* a costului [negativ] exponenţial; monotonia costurilor [negativ] exponenţiale minime: $J_t^* \leq J_{t-1}^*$)**

prelucrare de Liviu Ciortuz, după
■ • ○ CMU, 2008 fall, Eric Xing, HW3, pr. 4.1.1
CMU, 2012 fall, E. Xing, A. Singh, HW4, pr. 3.ab

La problema 23.c am văzut că în algoritmul AdaBoost se urmăreşte (în mod indirect) să se minimizeze eroarea la antrenare $err_S(H_T)$ prin minimizarea *secvenţială* a marginii sale superioare $\prod_{t=1}^{T} Z_t$. Aceasta înseamnă că la fiecare iteraţie t (unde $1 \leq t \leq T$) alegem valoarea ponderii α_t astfel încât să minimizăm Z_t (văzut ca funcţie de α_t, conform relaţiei (246) de la problema problema 22.*ii*).

Aici veţi vedea că o altă cale de a explica modul în care funcţionează algoritmul AdaBoost este determinată de *obiectivul* de a minimiza în mod *greedy* şi secvenţial *costul / pierderea exponenţială* (engl., the exponential loss):[541]



$$J_T \overset{def.}{=} \frac{1}{m}\sum_{i=1}^{m} \exp(-y_i f_T(x_i)) \overset{not.}{=} \frac{1}{m}\sum_{i=1}^{m} \exp(-y_i \sum_{t=1}^{T} \alpha_t h_t(x_i)). \tag{252}$$

Aceasta revine la a spune că la fiecare iteraţie $t \in \{1, \ldots, T\}$ urmărim să alegem — pe lângă cea mai bună ipoteză „slabă“ h_t — o valoare pentru ponderea α_t astfel încât costul / „pierderea“ J_t să fie minimizată.

În acest execiţiu se demonstrează că această nouă strategie va conduce la aceeaşi regulă de actualizare pentru ponderea α_t folosită în algoritmul AdaBoost, adică $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$.

a. Arătaţi că

$$J_t = c_t \cdot \underbrace{\sum_{i=1}^{m} D_t(i) \cdot \exp(-y_i \alpha_t h_t(x_i))}_{not.: \tilde{J}_t}$$

unde c_t este un factor constant, care nu depinde de α_t şi nici de h_t, dar pe care îl veţi preciza, iar

$$\tilde{J}_t = (1-\varepsilon_t) \cdot e^{-\alpha_t} + \varepsilon_t \cdot e^{\alpha_t},$$

unde ε_t este eroarea ponderată la antrenare pentru ipoteza „slabă“ h_t.

[541] Pentru o definiţie [de lucru] pentru funcţiile de cost / pierdere, vedeţi *Explicaţia* de la problema 29.
Veţi observa că pentru orice h_t şi α_t fixaţi, $J_t \geq err_S(H_t) \overset{def.}{=} \frac{1}{m}\sum_{i=1}^{m} 1_{\{y_i \neq sign(f_t(x_i))\}}$. Pentru aceasta, analizaţi separat cazurile $y_i f_t(x_i) < 0$ şi $y_i f_t(x_i) \geq 0$. Pentru a minimiza $err_S(H_t)$, aici vom urmări (în principal) să minimizăm J_t în funcţie de α_t.

***Sugestie*: Veţi putea folosi rezultatul obţinut la problema 23.a, şi anume că $D_t(i) \propto \exp(-y_i f_{t-1}(x_i))$, adică probabilitatea $D_t(i)$ este proporţională cu $\exp(-y_i f_{t-1}(x_i))$.**[542]

b. Arătaţi că dacă se consideră $\alpha_t > 0$ fixat, atunci pentru a minimiza expresia / valoarea lui J_t trebuie să alegem dintre toate ipotezele „slabe" pe aceea (notată cu h_t) pentru care eroarea ponderată la antrenare (notată cu ε_t) este minimă.

c. Arătaţi că dacă se consideră h_t fixat (deci şi ε_t fixat), atunci minimizarea lui J_t revine la a atribui lui α_t valoarea $\ln\sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}$.

d. Dacă notăm cu J_t^* valoarea optimă a lui J_t (văzut ca funcţie de ipoteza „slabă" h_t şi votul $\alpha_t > 0$), veţi constata că are loc o anumită relaţie de inegalitate între J_t^* şi J_{t-1}^* pentru orice $t > 1$. Care este această relaţie? Justificaţi.

e. Care este relaţia dintre J_t^* pe de o parte şi Z_1, Z_2, ..., Z_t, pe de altă parte?

f. Ca o *consecinţă* a punctelor b şi c, rescrieţi pseudo-codul algoritmului AdaBoost ca un algoritm de *optimizare secvenţială* a costului [negativ] exponenţial al clasificării instanţelor de antrenament.

Răspuns:

a. La iteraţia t vom avea:

$$\begin{aligned}
J_t &= \frac{1}{m}\sum_{i=1}^{m}\exp(-y_i f_t(x_i)) = \frac{1}{m}\sum_{i=1}^{m}\exp\left(-y_i\left(\sum_{t'=1}^{t-1}\alpha_{t'}h_{t'}(x_i)\right) - y_i\alpha_t h_t(x_i)\right)\\
&= \frac{1}{m}\sum_{i=1}^{m}\exp(-y_i f_{t-1}(x_i))\cdot\exp(-y_i\alpha_t h_t(x_i))\\
&= \frac{1}{m}\sum_{i=1}^{m}\Big(m\prod_{t'=1}^{t-1}Z_{t'}\Big)\cdot D_t(i)\cdot\exp(-y_i\alpha_t h_t(x_i))\\
&= \Big(\prod_{t'=1}^{t-1}Z_{t'}\Big)\cdot\sum_{i=1}^{m}D_t(i)\cdot\exp(-y_i\alpha_t h_t(x_i)) \qquad (253)\\
&\propto \sum_{i=1}^{m}D_t(i)\cdot\exp(-y_i\alpha_t h_t(x_i)) \stackrel{not.}{=} \tilde{J}_t. \qquad (254)
\end{aligned}$$

Mai departe, ţinând cont de faptul că $y_i \in \{-1,+1\}$ şi $h_t(x_i) \in \{-1,+1\}$ pentru $i = 1,\ldots,m$ şi $t = 1,\ldots,T$, putem scrie expresia $\tilde{J}_t$ astfel:

$$\begin{aligned}
\tilde{J}_t &= \sum_{i=1}^{m}D_t(i)\cdot\exp(-y_i\alpha_t h_t(x_i))\\
&= \sum_{i\in C}D_t(i)\exp(-\alpha_t) + \sum_{i\in M}D_t(i)\exp(\alpha_t)\\
&= \exp(-\alpha_t)\underbrace{\sum_{i\in C}D_t(i)}_{1-\varepsilon_t} + \exp(\alpha_t)\underbrace{\sum_{i\in M}D_t(i)}_{\varepsilon_t}
\end{aligned}$$

[542] Mai exact, $D_t(i) = \frac{1}{mZ_1\ldots Z_{t-1}}\exp(-y_i f_{t-1}(x_i))$.

$$= \quad (1-\varepsilon_t)\cdot e^{-\alpha_t} + \varepsilon_t \cdot e^{\alpha_t}, \tag{255}$$

unde, ca şi la problema 22, C este mulţimea [indicilor] exemplelor de antrenament care sunt corect clasificate de către ipoteza h_t, iar M este mulţimea [indicilor] exemplelor de antrenament care sunt incorect clasificate de către h_t.

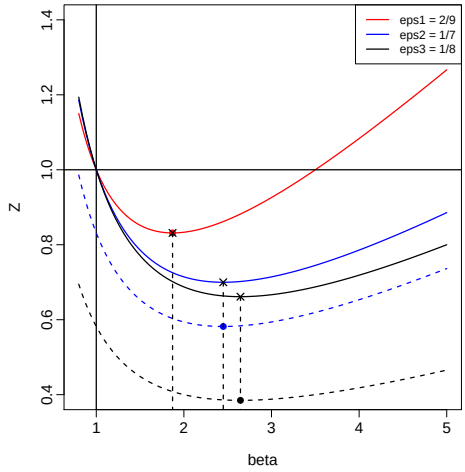
b. Dacă rescriem expresia (255) sub forma

$$e^{-\alpha_t} + \varepsilon_t \underbrace{(e^{\alpha_t} - e^{-\alpha_t})}_{>0},$$

se observă că dacă fixăm α_t atunci a minimiza expresia J_t în raport cu h_t revine la a minimiza ε_t (care nu depinde de α_t).

c. Expresia (255) este identică cu expresia (246) pentru factorul de normalizare Z_t de la problema 22.*ii*.[543] La problema 23.c am arătat că expresia (246) îşi atinge minimul pentru $\alpha_t = \frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t}$. Prin urmare, ţinând cont de faptul că ε_t se consideră fixat, „costul" J_t va fi minimizat pentru acelaşi $\alpha_t = \frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t}$.

***Exemplificare*: În graficul alăturat, realizat de către Sebastian Ciobanu, sunt reprezentate funcţiile $\tilde{J}_t$ şi J_t corespunzătoare celor trei iteraţii realizate de către algoritmul AdaBoost, care au fost descrise la problema 24. Pentru convenienţă, am folosit notaţia $\beta = e^{\alpha}$ şi am reprezentat funcţiile în raport cu acest argument. În acest grafic, liniile continue corespund funcţiilor $\tilde{J}_t$, iar liniile discontinue corespund funcţiilor J_t asociate lor: $J_t(\beta) = \left(\prod_{t'=1}^{t-1} Z_{t'}\right)\cdot \tilde{J}_t(\beta)$. Remarcaţi că din relaţia (253) rezultă că $J_1(\beta) = \tilde{J}_1(\beta), \forall \beta$.**



d. Valorile minime ale funcţiilor J_t nu cresc niciodată de la o iteraţie la alta, fiindcă $\min_{h\in\mathcal{H},\alpha\in\mathbb{R}_+} J_t(h,\alpha) \leq \underbrace{\frac{1}{m}\sum_{i=1}^{m}\phi(y_i, f_{t-1}(x_i))}_{J_t(_,0)}$. Introducând notaţia $J_t^* = \min_{h\in\mathcal{H},\alpha\in\mathbb{R}_+} J_t(h,\alpha)$, putem scrie aceasta inegalitate sub forma $J_t^* \leq J_{t-1}^*$, $\forall t \geq 1$.

e. Din relaţia (253) putem deduce imediat că

$$J_t^* = \prod_{t'=1}^{t} Z_{t'},$$

fiindcă expresia (255) are valoarea Z_t atunci când îi atribuim lui α_t valoarea

[543] [S. Ciobanu:] De fapt, se putea vedea chiar din relaţia (254) că $\tilde{J}_t = Z_t$ (văzut ca funcţie de α_t), ţinând cont de faptul că Z_t este factor de normalizare în definiţia distribuţiei D_{t+1} (vedeţi relaţia (245)).

$\frac{1}{2}\ln\frac{1-\varepsilon_t}{\varepsilon_t}$.[544] **Pe graficul precedent se poate vedea că într-adevăr $J_t^* \leq J_{t-1}^*$, pentru $t \in \{2,3\}$, întrucât $Z_1 = 0.8315$, $Z_2 = 0.7$ şi $Z_3 = 0.6614$, cf. problemei 24.**

***Observaţie*: În consecinţă, *graficul* valorilor J_t^* (pentru $j = 1, 2, \ldots$) coincide cu graficul marginii superioare $\prod_{t'=1}^{t} Z_{t'}$ pentru eroarea la antrenare $err_S(H_t)$ produsă de ipoteza combinată H_t. (Vedeţi exerciţiile 23, 24 şi 25.)**

f. Ca urmare a punctelor b şi c, algoritmul AdaBoost poate fi *reformulat* ca un algoritm de *optimizare secvenţială* a costurilor / „pierderilor" J_t. La fiecare iteraţie a algoritmului se adaugă în mod „greedy" câte o ipoteză „slabă" $h \in \mathcal{H}$ la *ipoteza combinată* curentă pentru a minimiza costul determinat de funcţia de cost / pierdere ϕ, şi cu *obiectivul* final de a minimiza eroarea la antrenare $err_S(H_T)$. (De aceea această metodă de optimizare se numeşte *optimizare secvenţială*.) Iată care este noua formulare a algoritmului AdaBoost:

Intrare: $\{(x_i, y_i)\}_{i=1,\ldots,m}$ – setul de date de antrenament, T – numărul de iteraţii de executat, $\mathcal{H}$ – setul de ipoteze „slabe", $\phi(y, y') \stackrel{def.}{=} \exp(-yy')$ – funcţia de cost / pierdere.

Procedura:

Initializare: $f_0(x) = 0$ şi $D_1(i) = 1/m$ pentru $i = 1, \ldots, m$
pentru t de la 1 la T execută:
begin
1. Calculează

$$(h_t, \alpha_t) = \arg\min_{h\in\mathcal{H},\alpha\in\mathbb{R}_+} \underbrace{\sum_{i=1}^{m} \phi(y_i, f_{t-1}(x_i) + \alpha h(x_i))}_{mJ_t(h,\alpha)}$$

2. Actualizează clasificatorul $f_t(x) = f_{t-1}(x) + \alpha_t h_t(x)$
şi calculează noua distribuţie, D_{t+1}
end
returnează clasificatorul *sign* $(f_T(x))$

Precizăm faptul că minimizarea efectuată la pasul 1 din corpul iterativ al acestui algoritm este făcută *pe coordonate*. Aceasta înseamnă că mai întâi minimizăm funcţia J_t în raport cu argumentul h (care nu depinde de α) şi apoi — cu h fixat la valoarea pe care tocmai am găsit-o, h_t — minimizăm funcţia J_t în raport cu argumentul α (care depinde de h_t).

***Observaţie importantă*:**
La problema 29 vom arăta că putem adapta algoritmul nou introdus astfel încât să lucrăm cu orice funcţie de cost dorim în locul funcţiei de cost [negativ] exponenţiale ϕ. Ca pregătire pentru aceasta, la punctul următor vom demonstra aici un *rezultat similar* cu cel de la punctul b:

A minimiza [în funcţie de ε_t] expresia $\frac{\partial}{\partial\alpha} J_t(\alpha)|_{\alpha=0}$,[545] revine la a identifica dintre erorile ponderate ε_t (corespunzătoare ipotezelor h_t) pe cea mai mică.[546]

[544] LC: Mulţumesc lui Sebastian Ciobanu pentru dialogul în urma căruia am identificat această proprietate. Din relaţia $J_t^* = \prod_{t'=1}^{t} Z_{t'}$, ştiind că $Z_t \in (0,1)$ (vedeţi problema 22.*iv*), se obţine imediat o a doua demonstraţie pentru faptul că $J_t^* \leq J_{t-1}^*$, $\forall t \geq 1$.

[545] Valorea acestei expresii se obţine astfel: se calculează mai întâi derivata parţială a funcţiei J_t în raport cu α, după care se determină valoarea acestei derivate parţiale în punctul $\alpha = 0$.

[546] Pentru a înţelege de ce în această *proprietate* a fost ales drept *criteriu de minimizat* expresia $\frac{\partial}{\partial\alpha} J_t(\alpha)|_{\alpha=0}$, vedeţi graficul precedent, precum şi explicaţiile de la problema 29.B.

Într-adevăr,

$$\Big(\frac{\partial}{\partial\alpha}J_t(\alpha)\Big)|_{\alpha=0} \propto \Big(\frac{\partial}{\partial\alpha}\tilde{J}_t(\alpha)\Big)|_{\alpha=0} = \Big(\frac{\partial}{\partial\alpha}((1-\varepsilon)e^{-\alpha}+\varepsilon e^{\alpha})\Big)|_{\alpha=0}$$
$$= \Big(-(1-\varepsilon)e^{-\alpha}+\varepsilon e^{\alpha}\Big)|_{\alpha=0} = -(1-\varepsilon)+\varepsilon = 2\varepsilon-1. \qquad (256)$$

Aşadar, minimul expresiei $\frac{\partial}{\partial\alpha}J_t(\alpha)|_{\alpha=0}$ se atinge pentru eroarea ponderată ε minimă.

27. **(Noţiunea de *margine de votare* în conexiune cu algoritmul AdaBoost; o referire la chestiunea overfitting-ului)**

prelucrare de Liviu Ciortuz, după
■ □ • · ***CMU, 2016 spring, W. Cohen, N. Balcan, HW4, pr. 3.3***

Deşi la aplicarea algoritmului AdaBoost *complexitatea modelului* produs *creşte* la fiecare iteraţie, *în general nu se produce overfitting.* Motivul este că modelul capătă din ce în ce mai multă „încredere“ pe măsură ce numărul de iteraţii executate creşte. Această „încredere“ (engl., confidence) poate fi exprimată din punct de vedere matematic cu ajutorul noţiunii de *margine de votare* (engl., voting margin).[547]

Vă readucem aminte că,[548] **după efectuarea celor T iteraţii, algoritmul AdaBoost livrează la ieşire clasificatorul H_T, definit astfel:**

$$H_T(x) = sign\Big(\sum_{t=1}^{T}\alpha_t\, h_t(x)\Big).$$

În mod similar, putem defini *clasificatorul ponderat intermediar* (engl., intermediate weighted classifier) după k iteraţii: $H_k(x) = sign\Big(\sum_{t=1}^{k}\alpha_t\, h_t(x)\Big)$. Întrucât outputul acestei funcţii este $+1$ sau -1, aceasta nu ne comunică nimic despre încrederea [pe care o putem avea] în deciziile acestui clasificator. De aceea, fără a afecta regula de decizie în sine, o putem redefini sub forma

$$H_k(x) = sign\Big(\sum_{t=1}^{k}\bar{\alpha}_t\, h_t(x)\Big),$$

unde $\bar{\alpha}_t = \frac{\alpha_t}{\sum_{t'=1}^{k}\alpha_{t'}}$, deci ponderile (engl., weights) ipotezelor „slabe“ sunt acum normalizate, în sensul că $\sum_{t=1}^{k}\bar{\alpha}_t = 1$ şi, desigur, $\bar{\alpha}_t > 0$.[549]

***Marginea de votare* a instanţei de antrenament x_i după iteraţia k se defineşte ca fiind diferenţa dintre suma ponderilor / voturilor normalizate ale acelor h_t**

[547] În mod concret, „încrederea“ în clasificarea lui x_i la iteraţia t va fi definită ca $|Margin_k(x_i)|$, unde $Margin_k(x_i)$ urmează să fie definită mai jos.

[548] Vedeţi enunţul problemei 22.

[549] Aceasta normalizare, referitoare la ponderile α_t (asociate ipotezelor „slabe“ h_t), nu trebuie confundată cu normalizarea sau, de fapt, normalizările reprezentate de factorii Z_t din pseudo-codul algoritmului AdaBoost — vedeţi enunţul problemei 22 —, care se referă la distribuţiile probabiliste D_t (asociate, la fiecare iteraţie, setului de instanţe de antrenament).

— evident, $t \in \{1, \ldots, k\}$ — care clasifică corect instanţa x_i şi suma ponderilor / voturilor normalizate ale acelor h_t care îl clasifică incorect pe x_i.[550] Aşadar,

$$Margin_k(x_i) = \sum_{t:h_t(x_i)=y_i} \bar{\alpha}_t \; - \sum_{t:h_t(x_i)\neq y_i} \bar{\alpha}_t.$$

***Observaţie*: Este imediat că $Margin_k(x_i) \in [-1, 1]$, pentru orice instanţă x_i, cu $i = 1, \ldots, m$ şi pentru orice iteraţie k a algoritmului AdaBoost.**

a. Fie $f_k(x) \stackrel{not.}{=} \sum_{t=1}^{k} \bar{\alpha}_t \, h_t(x)$. Arătaţi că $Margin_k(x_i) = y_i \, f_k(x_i)$ pentru orice instanţă de antrenament x_i, cu $i = 1, \ldots, m$.

Aşadar, avem în acest fel o legătură foarte semnificativă între noţiunea de *margine de votare* nou-introdusă şi noţiunea de *margine funcţională* introdusă la rezolvarea problemei 23.a: atunci când toate voturile α_t sunt normalizate, noţiunea de *margine de votare* coincide cu noţiunea de *margine funcţională*. (De aceea, în cele ce urmează, atunci când nu este niciun pericol de confuzie ne vom referi la aceste două noţiuni pur şi simplu prin termenul *margine*.)

***Consecinţă imediată*: x_i, o instanţă de antrenament oarecare, este corect clasificată de către ipoteza combinată H_k produsă de către AdaBoost la iteraţia k dacă şi numai dacă $Margin_k(x_i) \geq 0$.**

b. Dacă $Margin_k(x_i) > Margin_k(x_j)$, care dintre probabilităţile asociate celor două instanţe va fi mai mare la iteraţia $k+1$ (adică, $D_{k+1}(i)$ sau $D_{k+1}(j)$)?

***Sugestie*: Din relaţia $D_{k+1}(i) = \dfrac{1}{m \cdot \prod_{t=1}^{k} Z_t} \cdot \exp(-y_i \cdot \sum_{t=1}^{k} \alpha_t \cdot h_t(x_i))$,[551] care a fost demonstrată la problema 23.a, putem deduce $D_{k+1}(i) \propto \exp(-y_i f_k(x_i))$.[552]**

Răspuns:

a. Vom demonstra egalitatea cerută pornind de la termenul din partea dreaptă:

$$\begin{aligned} y_i \, f_k(x_i) &= y_i \sum_{t=1}^{k} \bar{\alpha}_t \, h_t(x_i) = \sum_{t=1}^{k} \bar{\alpha}_t \, y_i \, h_t(x_i) = \sum_{t:h_t(x_i)=y_i} \bar{\alpha}_t - \sum_{t:h_t(x_i)\neq y_i} \bar{\alpha}_t \\ &= Margin_k(x_i). \end{aligned} \tag{257}$$

b. Conform relaţiei pe care tocmai am demonstrat-o la punctul precedent, inegalitatea $Margin_k(x_i) > Margin_k(x_j)$ este echivalentă cu $y_i \, f_k(x_i) > y_j \, f_k(x_j)$. La rândul ei, aceasta din urmă este echivalentă cu $-y_i \, f_k(x_i) < -y_j \, f_k(x_j)$, deci şi cu $\exp(-y_i \, f_k(x_i)) < \exp(-y_j \, f_k(x_j))$. Ţinând cont de *Sugestia* din enunţ, rezultă imediat că $D_{k+1}(i) < D_{k+1}(j)$.

Aşadar, se verifică *intuiţia* conform căreia este natural ca instanţele care sunt [cel] mai bine clasificate, deci care au o margine [mai] mare la o anumită iteraţie, să primească la iteraţia următoare o probabilitate mai mică, algoritmul AdaBoost concentrându-se asupra instanţelor care sunt incorect (sau, mai puţin bine) clasificate.

[550] Noţiunea de *margine geometrică* (definită altfel decât cele două tipuri de *margine* de aici) este specifică clasificatorului SVM. Vedeţi capitolul *Maşini cu vectori-suport*.

[551] *Atenţie*! Spre deosebire de definiţia funcţiei f_k care a fost dată la punctul *a*, definiţia funcţiei f_k de la problema 23 nu include normalizarea „voturilor" α_t.

[552] Observaţi că o *sugestie* similară a fost făcută şi la problema 26.a.

***Observaţie importantă*: Se poate constata practic[553] că în cursul execuţiei sale, algoritmul AdaBoost tinde să mărească per ansamblu, de la o iteraţie la alta, marginile corespunzătoare exemplelor de antrenament,[554] iar o margine mai mare [pentru aceste exemple] implică în mod obişnuit o eroare la testare / generalizare mai mică. Aceasta este o explicaţie pentru faptul că, deşi numărul de „parametri" ai modelului rezultat creşte cu 2 la fiecare iteraţie a algoritmului AdaBoost (şi, deci, complexitatea creşte),[555] *în general* acest algoritm *nu produce overfitting*.[556]**

28. (AdaBoost: o [altă] condiţie suficientă pentru învăţabilitate γ-slabă: media [probabilistă a] marginilor de votare să fie de cel puţin 2γ, la fiecare iteraţie a algoritmului)

prelucrare de Liviu Ciortuz, după
■ □ • ○ ***CMU, 2016 spring, W. Cohen, N. Balcan, HW4, pr. 3.1.4***

***Introducere*:** La problema 23.e am făcut cunoştinţă cu noţiunea de *învăţabilitate empirică γ-slabă* (engl., emipirical γ-weak learnability). Concret, am demonstrat acolo că eroarea la antrenare produsă de algoritmul AdaBoost descreşte [rapid] la 0 atunci când există $\gamma > 0$ astfel încât $\gamma_t \geq \gamma$ pentru orice t, unde $\gamma_t \overset{def.}{=} \frac{1}{2} - \varepsilon_t$, iar ε_t este eroarea ponderată la antrenare a ipotezei „slabe" h_t.[557] În acest exerciţiu vom demonstra o nouă *condiţie suficientă* pentru ca să aibă loc învăţabilitatea empirică γ-slabă, făcând apel la noţiunea de *margine de votare* (engl., voting margin), care a fost prezentată la problema 27. Concret, vom arăta că atunci când există $\theta > 0$ astfel încât media [în sens probabilist a] marginilor de votare ale instanţelor de antrenament este mărginită inferior de θ la fiecare iteraţie a algoritmului AdaBoost,[558] proprietatea de învăţabilitate empirică γ-slabă este „garantată", cu $\gamma = \theta/2$.[559]

Presupunem că dispunem de setul de date de antrenament $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ şi că există ipotezele „slabe" $h_1, \ldots, h_k$ din spaţiul de ipoteze $\mathcal{H}$, precum şi coeficienţii pozitivi $\alpha_1, \ldots, \alpha_k$ cu proprietatea $\sum_{j=1}^{k} \alpha_j = 1$. Presupunem, de asemenea, că există $\theta > 0$ astfel încât, pentru o distribuţie probabilistă D (oarecare) peste setul de date de antrenament S are loc inegalitatea

$$E_{i\sim D}[Margin_k(x_i)] \geq \theta. \tag{258}$$

Vom demonstra că de fiecare dată când condiţia (258) este satisfăcută există [cel puţin] o ipoteză $h_l \in \{h_1, \ldots, h_k\}$ pentru care eroarea ponderată la antre-

[553] Vedeţi de exemplu problemele de tip implementare MIT, 2001 fall, Tommi Jaakkola, HW3, pr. 2.4 şi MIT, 2009 fall, Tommi Jaakkola, HW3, pr. 1.4.

[554] Acest fapt se poate deduce din relaţia (252) de la problema 26, care dă expresia *costului* de minimizat de către algoritmul AdaBoost. Concret, a minimiza $E_T = \sum_{i=1}^{m} \exp(-y_i f_T(x_i))$ implică, în idee, a maximiza, pe cât posibil, fiecare *margine* $y_i f_T(x_i)$.

[555] Cei doi parametri adăugaţi la fiecare iteraţie a algoritmului AdaBoost sunt unul pentru identificarea ipotezei h_t (de exemplu, în cazul compaşilor de decizie, pragul corespunzător, s_t), iar celălalt ponderea α_t.

[556] Există totuşi situaţii în care algoritmul AdaBoost produce overfitting. Vedeţi de exemplu pr. 73.

[557] Această condiţie nu este satisfăcută întotdeauna. Vedeţi, de exemplu, soluţia a doua de la problema 25 (adică soluţia care a fost obţinută fără a folosi compaşi de decizie cu „prag" exterior în raport cu instanţele de antrenament).

[558] De fapt, este suficient ca această condiţie să fie satisfăcută de la o anumită iteraţie t_0 încolo.

[559] În enunţul original al acestui exerciţiu se formulează o altă condiţie suficientă, care este mai „tare" decât cea adoptată de noi: există $\theta > 0$ astfel încât marginile de votare ale instanţelor de antrenament sunt mărginite inferior de θ la fiecare iteraţie a algoritmului AdaBoost.

nare în raport cu distribuţia D este de cel mult $\frac{1}{2}-\frac{\theta}{2}$, adică $\varepsilon_l \leq \frac{1}{2}-\frac{\theta}{2}$.[560] **Demonstraţia va fi făcută în doi paşi, care corespund punctelor a şi b de mai jos.**

***Observaţie*:**
Relaţia (258) implică faptul că există $i \in \{1,\ldots,m\}$ cu proprietatea $Margin_k(x_i) \geq \theta$. Ştim din enunţul problemei 27 că $Margin_k(x_i) \in [-1,\,1]$, deci inegalitatea $Margin_k(x_i) \geq \theta > 0$ implică $\theta \in (0,\,1]$ şi, în consecinţă, $\frac{\theta}{2} \in (0,\,1/2]$.

a. Fie D este o distribuţie probabilistă oarecare definită pe S. Ca şi la problema 27, vom nota $f_k(x_i) = \sum_{l=1}^{k} \alpha_l h_l(x_i)$. Arătaţi că atunci când

$$E_{i\sim D}[Margin_k(x_i)] \overset{(257)}{=} E_{i\sim D}[y_i f_k(x_i)] \geq \theta,$$

adică media marginilor pentru instanţele $x_1,\ldots,x_m$ este mărginită inferior de θ, există cel puţin o ipoteză „slabă“ h_l din mulţimea $\{h_1,\ldots,h_k\}$ astfel încât $E_{i\sim D}[y_i h_l(x_i)] \geq \theta$.[561]

b. Arătaţi că inegalitatea $E_{i\sim D}[y_i h_l(x_i)] \geq \theta$ este echivalentă cu inegalitatea $\mathrm{err}_D(h_l) \leq \frac{1}{2}-\frac{\theta}{2}$, unde, ca şi la problema 22, $\mathrm{err}_D(h_l) \overset{not.}{=} \Pr_{i\sim D}[y_i \neq h_l(x_i)]$ desemnează eroarea ponderată la antrenare a ipotezei „slabe“ h_l în raport cu distribuţia probabilistă D.

Răspuns:

a. Ţinând cont de relaţia (257), inegalitatea (258) se rescrie (conform definiţiei mediei) sub forma

$$\sum_{i=1}^{m} y_i f_k(x_i) \cdot D(i) \geq \theta. \tag{259}$$

Vom folosi *metoda reducerii la absurd* pentru a demonstra că există cel puţin un indice $l \in \{1,\ldots,k\}$ astfel încât $E_{i\sim D}[y_i h_l(x_i)] \geq \theta$. Presupunem că pentru orice $l = 1,\ldots,k$ are loc inegalitatea $E_{i\sim D}[y_i h_l(x_i)] < \theta$, adică $\sum_{i=1}^{m} y_i h_l(x_i) \cdot D(i) < \theta$. Prin înmulţirea ambilor termeni ai acestei inegalităţi cu $\alpha_l > 0$ rezultă

$$\sum_{i=1}^{m} y_i h_l(x_i) \cdot D(i) \cdot \alpha_l < \theta \cdot \alpha_l \text{ pentru } l = 1,\ldots,k.$$

Însumând membru cu membru aceste inegalităţi pentru $l = 1,\ldots,k$, rezultă

$$\sum_{l=1}^{k}\sum_{i=1}^{m} y_i h_l(x_i) \cdot D(i) \cdot \alpha_l < \sum_{l=1}^{k} \theta \cdot \alpha_l \Leftrightarrow \sum_{i=1}^{m} y_i\, D(i) \left(\sum_{l=1}^{k} h_l(x_i)\alpha_l\right) < \theta \sum_{l=1}^{k} \alpha_l \Leftrightarrow$$
$$\sum_{i=1}^{m} y_i f_k(x_i) \cdot D(i) < \theta, \tag{260}$$

[560] În particular, D poate fi considerată ca fiind distribuţia calculată de AdaBoost pentru iteraţia k.
[561] Putem vedea produsele $y_i h_l(x_i)$ (care iau valori în mulţimea $\{-1,+1\}$) ca fiind generate de către o variabilă aleatoare, cu distribuţia D. Observaţi că $E_{i\sim D}[y_i h_l(x_i)]$ este media acestei variabile aleatoare.

fiindcă $\sum_{l=1}^{k} \alpha_l = 1$ **şi** $f_k(x_i) \overset{not.}{=} \sum_{l=1}^{k} \alpha_l h_l(x_i)$**.**

Inegalitatea (260) **se rescrie sub forma** $E_{i\sim D}[y_i f_k(x_i)] < \theta$**. Evident, aceasta contrazice relaţia** (259)**. Prin urmare, presupunerea făcută anterior este falsă. Rezultă că există** $l \in \{1, \ldots, k\}$ **astfel încât** $E_{i\sim D}[y_i h_l(x_i)] \geq \theta$**.**

b. După cum am menţionat deja la rezolvarea punctului a**, inegalitatea** $E_{i\sim D}[y_i h_l(x_i)] \geq \theta$ **se rescrie în mod echivalent sub forma** $\sum_{i=1}^{m} y_i\, h_l(x_i) \cdot D(i) \geq \theta$**. Ţinând cont de faptul că** $y_i \in \{-1, +1\}$ **şi** $h_l(x_i) \in \{-1, +1\}$ **pentru** $i = 1, \ldots, m$ **şi** $l \in \{1, \ldots, k\}$**, putem scrie următorul şir de echivalenţe:**[562]

$$\sum_{i=1}^{m} y_i\, h_l(x_i) \cdot D(i) \geq \theta \;\Leftrightarrow \sum_{i:y_i = h_l(x_i)} D(x_i) - \sum_{i:y_i \neq h_l(x_i)} D(x_i) \geq \theta \Leftrightarrow$$

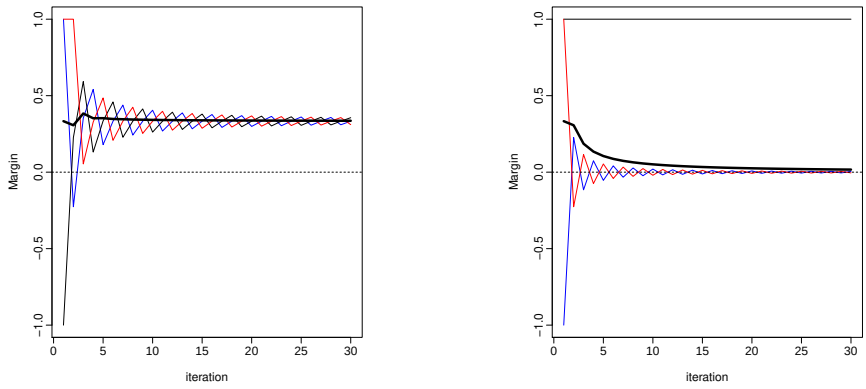
$$(1 - \varepsilon_l) - \varepsilon_l \geq \theta \Leftrightarrow 1 - 2\varepsilon_l \geq \theta \Leftrightarrow 2\varepsilon_l \leq 1 - \theta \Leftrightarrow \varepsilon_l \leq \frac{1}{2} - \frac{\theta}{2} \Leftrightarrow$$

$$\overset{def.}{\Leftrightarrow} \mathrm{err}_D(h_l) \leq \frac{1}{2} - \frac{\theta}{2}.$$

În *consecinţă*, o *condiţie suficientă* pentru a asigura γ**-învăţabilitate slabă pe un dataset de antrenament** S **este ca media [probabilistă a] marginilor de votare să fie de cel puţin** $2\gamma \overset{not.}{=} \theta$**, pentru orice exemplu de antrenament, la fiecare iteraţie a algoritmului AdaBoost (de fapt, este destul ca această condiţie să fie îndeplinită de la o anumită iteraţie** t_0 **încolo).**[563]

***Exemplificare*: Următoarele grafice, făcute de Sebastian Ciobanu, pun în evidenţă evoluţia marginilor de votare, precum şi a mediei lor (vedeţi linia mai îngroşată), pentru datele de la problema 25, pentru fiecare dintre cele două variante de rezolvare care au fost prezentate acolo:**

cu prag exterior **fără prag exterior**



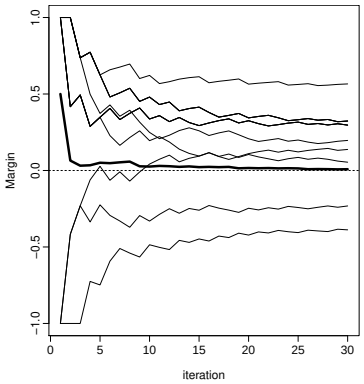
Se constată uşor că în prima variantă (şi anume, când se foloseşte prag exterior) există un prag inferior θ **pentru media marginilor de votare (şi chiar pentru toate marginile), începând chiar cu una din primele iteraţii. În schimb, în a doua variantă (şi anume, când nu se foloseşte prag exterior), este evident că nu va exista un prag inferior** θ **pentru media marginilor de votare, întrucât**

[562] Ca şi la problema 22, vom nota cu ε_l eroarea ponderată produsă la antrenare de către ipoteza h_l.

[563] Observaţi că această *condiţie suficientă* este mai „tare“ / restrictivă decât condiţia pentru învăţabilitate γ-slabă de la problema 23.e, fiindcă prezenta condiţie o implică pe cea de la problema 23.e.

la fiecare iteraţie una dintre instanţele $x_1 = 1$ şi $x_3 = 5$ este clasificată eronat în mod flagrant. (Se observă însă că instanţa $x_2 = 3$ este clasificată corect la fiecare iteraţie, ba chiar cu „maximum“ de voturi, de aceea marginea ei de votare este întotdeauna 1. Probabilitatea asociată acestei instanţe este însă din ce în ce mai mică.)

Graficul alăturat, făcut tot de Sebastian Ciobanu, prezintă evoluţia marginilor de votare, precum şi a mediei lor (vedeţi linia mai îngroşată), pentru datele de la problema 67, pentru care se poate arăta că nu admite [nici măcar direct, adică în maniera de la problema 23.e] garanţie de învăţabilitate γ-slabă.



***Observaţii*:**

1. Aşa cum am procedat la problema 23.e, putem formula şi la această problemă o *condiţie suficientă* pentru ca eroarea la antrenare $H_{T'}$, ipoteza combinată produsă de AdaBoost la iteraţia T', să rămână 0, după ce s-a constatat că există $\theta > 0$ astfel încât timp de un anumit număr de iteraţii (T) este satisfăcută inegalitatea $E_{i\sim D_k}[Margin_k(x_i)] \geq \theta$. Concret, se poate arăta uşor că $T = \left\lceil \frac{2}{\theta^2} \ln m \right\rceil$.

2. În afară de proprietatea de mai sus (1.), se poate constata că proprietatea de bază pe care am demonstrat-o în acest exerciţiu

$$E_{i\sim D}[Margin_k(x_i)] \geq \theta \Rightarrow \exists h_t : err_D(h_t) \leq \frac{1}{2} - \frac{\theta}{2} \tag{261}$$

poate avea utilitate practică doar în cazul în care în algoritmul AdaBoost nu impunem condiţia să fie aleasă la fiecare iteraţie cea mai bună ipoteză „slabă“ (în sensul minimizării erorii ponderate la antrenare). În această situaţie, ipoteza „slabă“ h_t, a cărei existenţă este stipulată în relaţia (261), poate fi înlocuită cu ipoteza h'_t dacă, bineînţeles, $err_{D_t}(h'_t) < err_{D_t}(h_t)$.

3. Chiar dacă această *condiţie suficientă* — există $\theta > 0$ astfel încât media $E_{i\sim D_t}[\boldsymbol{Margin}_t(x_i)]$ este mai mare sau egală cu θ pentru orice t de la un t_0 încolo timp de T de iteraţii — nu este satisfăcută, totuşi studiul evoluţiei marginilor de votare are o mare valoare informativă.[564]

4. La capitolul *Maşini cu vectori-suport* (SVM) se arată ca pentru o altă categorie de margini — şi anume, *margini geometrice* —, se poate formula în mod riguros o *problemă de optimizare* care se pretează (inclusiv) la tratarea *excepţiilor* de la clasificare. În urma rezolvării acelei probleme de optimizare se obţine un clasificator / separator care este foarte convenabil, din multe puncte de vedere.

[564] Vedeţi de exemplu problemele de tip implementare MIT, 2001 fall, Tommi Jaakkola, HW3, pr. 1.4 şi MIT, 2009 fall, Tommi Jaakkola, HW3, pr. 2.4.

29. **(O generalizare a algoritmului AdaBoost, pentru diverse funcţii de cost / pierdere)**

prelucrare de Liviu Ciortuz, după
■ ● ○ ***MIT, 2003 fall, Tommi Jaakkola, HW4, pr. 2.1-3***

A. [*Comentariu* — fără „cerinţe de rezolvat“]

În această problemă vom deriva un *algoritm de boosting*, *mai general* decât algoritmul AdaBoost din problema 22.[565] Noul algoritm va putea fi aplicat la o întreagă clasă de *funcţii de cost* / *pierdere* (engl., loss functions), în particular pentru funcţia de cost [negativ] exponenţială.

***Scopul* nostru este să generăm *funcţii de discriminare* (engl., discriminant functions) de forma următoare:**

$$f_T(x) = \alpha_1 h(x;\theta_1) + \ldots + \alpha_T h(x;\theta_T).$$

În această expresie, $x \in \mathbb{R}^d$, iar θ_i este un parametru sau set / vector de parametri (tot dintr-un spaţiu de tip $\mathbb{R}^n$), depinzând de tipul ipotezelor „slabe“ h. În continuare veţi putea presupune că aceste ipoteze $h(x;\theta_i)$ sunt compaşi de decizie (engl., decision stumps), ale căror predicţii pot fi $+1$ sau -1. Orice alte categorii de ipoteze „slabe“ vor putea fi folosite în acest cadru. Vom adăuga în mod *secvenţial* componente la funcţia generală de discriminare, într-o manieră care va separa (atât cât este posibil) *estimarea* parametrilor θ ai ipotezelor „slabe“ de *setarea* voturilor / ponderilor α.

***Explicaţie*:** Vom începe prin a defini noţiunea de *funcţie de cost* / *pierdere*, pe care o vom folosi mai jos.[566] Singurele *restricţii* pe care o astfel de funcţie va trebui să le îndeplinească sunt următoarele: *i.* să fie nenegativă, *ii.* să fie *monoton descrescătoare* şi *iii.* să fie *derivabilă*.[567] În contextul nostru, argumentul unei funcţii de cost va fi *marginea algebrică* $y_i\, f_T(x_i)$, care reprezintă o măsură a acordului dintre eticheta y_i asociată instanţei de antrenament x_i pe de o parte şi [semnul dat de] valoarea funcţiei de discriminare f_T pe de altă parte. Vrem ca, pentru fiecare instanţă x_i, „costul“ să fie cu atât mai mic cu cât valoarea funcţiei de discriminare este mai mult în acord cu valoarea ± 1 a etichetei y_i (adică, cu cât marginea algebrică $y_i f_T(x_i)$ are o valoare mai mare). Funcţia de cost [negativ] *exponenţială* pe care am întâlnit-o deja la problema 26, adică

$$\text{Loss}(y_i f_T(x_i)) = \exp(-y_i f_T(x_i))$$

satisface cele două restricţii pe care tocmai le-am menţionat mai sus.[568]

***Criteriul de evaluare* / „*estimare*“ pentru combinaţia de ipoteze „slabe“ este pur şi simplu *costul* / *pierderea empirică* (engl., empirical loss), definită prin expresia**

$$J_T = \frac{1}{m} \sum_{i=1}^{m} \text{Loss}(y_i f_T(x_i)), \tag{262}$$

[565] Sămânţa pentru acest nou algoritm a fost sădită la *Observaţia importantă* de la finalul rezolvării problemei 26.

[566] Veţi vedea că definiţia pe care o dăm aici pentru funcţii de cost diferă uşor de cea pe care am folosit-o la problema 88 de la capitolul de *Fundamente*.

[567] Într-o accepţiune mai largă, condiţia *ii.* din definiţia funcţiei de cost devine: să fie *convexă*. (Această accepţiune va fi folosită la punctul *c* al acestei probleme.)

[568] La problema 88 de la capitolul *Fundamente* am ilustrat / reprezentat pe un acelaşi grafic trei funcţii de cost: funcţia de cost [negativ] exponenţială pentru AdaBoost, funcţia de cost logistică pentru regresia logistică şi funcţia de cost *hinge* pentru maşini cu vectori-suport (SVM).

unde sumarea se face parcurgând toate exemplele de antrenament disponibile.

Aşadar, vrem să *derivăm algoritmul de boosting* în aşa fel încât să putem lucra cu *orice funcţie de cost / pierdere* de tipul discutat mai sus. În acest scop, la *iteraţia curentă* (t) presupunem că am inclus deja în combinaţia liniară un număr de $t-1$ componente (ipoteze „slabe"):

$$f_{t-1}(x) = \alpha_1 h(x;\hat{\theta}_1) + \ldots + \alpha_{t-1} h(x;\hat{\theta}_{t-1}) \tag{263}$$

şi că dorim să mai adăugăm încă una, $h(x;\theta)$. *Criteriul de evaluare* pentru funcţia de discriminare — inclusiv ultima componentă adăugată, al cărei „vot" asociat este α —, este dat de expresia

$$J_t(\alpha,\theta) = \frac{1}{m}\sum_{i=1}^{m} \text{Loss}(y_i f_{t-1}(x_i) + y_i\,\alpha\,h(x_i;\theta)).$$

Remarcaţi faptul că urmărim să explicăm / explorăm doar modul în care acest „obiectiv" / *criteriu* depinde de alegerea ultimei componente ($h(x_i;\theta)$) şi de votul corespunzător (α), fiindcă parametrii precedentelor $t-1$ componente, împreună cu voturile asociate lor au fost deja setate şi nu vor mai fi modificate.

B. Pentru elaborarea ciclului repetitiv al noului algoritm, *mai întâi* vom urmări să identificăm *noua componentă*, mai precis spus valoarea parametrului θ în aşa fel încât să maximizăm „potenţialul" ei (adică, al noii componente) de a reduce costul empiric — „potenţial" în sensul că ulterior vom putea să ajustăm „votul" asociat acestei componente, ca să reducem [şi mai mult] costul empiric. Concret, la *Pasul 1* al iteraţiei t vom seta θ astfel încât să *minimizăm* — în raport cu diferitele valori ale parametrului θ — valoarea expresiei următoare:[569]

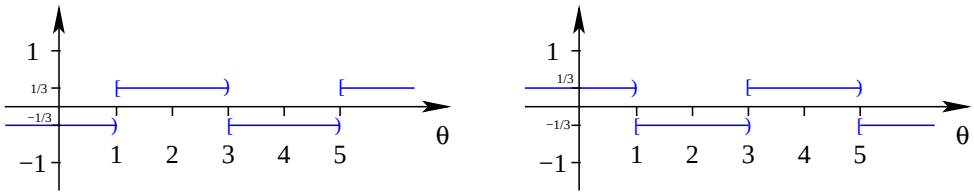
$$\begin{aligned}
\frac{\partial}{\partial\alpha} J_t(\alpha,\theta)_{|\alpha=0} &= \frac{1}{m}\sum_{i=1}^{m} \frac{\partial}{\partial\alpha}\text{Loss}(y_i f_{t-1}(x_i) + y_i\,\alpha\,h(x_i;\theta))_{|\alpha=0} \\
&= \frac{1}{m}\sum_{i=1}^{m} [dL(y_i f_{t-1}(x_i) + y_i\,\alpha\,h(x_i;\theta))\cdot y_i h(x_i;\theta)]_{|\alpha=0} \\
&= \frac{1}{m}\sum_{i=1}^{m} dL(y_i f_{t-1}(x_i))\,y_i\,h(x_i;\theta),
\end{aligned} \tag{264}$$

unde prin $dL(z)$ am notat derivata $\dfrac{\partial \text{Loss}(z)}{\partial z}$. (Am aplicat formula de derivare a funcţiilor compuse.)

***Exemplificare*:** Graficele următoare, corespunzătoare datelor de la problema 25, reprezintă cele două variante ale expresiei / funcţiei (de argumentul θ) $\dfrac{\partial}{\partial\alpha} J_t(\alpha,\theta)_{|\alpha=0}$ calculate pentru iteraţia $t=1$ a algoritmului AdaBoost folosind funcţia de cost [negativ] exponenţială. Aceste două variante corespund semnelor care pot fi alese pentru zonele de decizie determinate de compasul de decizie de prag (generic) θ: $+|-$ şi respectiv $-|+$.[570]

[569] Pentru o *justificare informală* [de ce se procedează aşa], puteţi face analogie cu graficul de la soluţia problemei 26.c. Faceţi corelaţia între pantele tangentelor la curbele din grafic pentru valoarea $\alpha = 0$ pe de o parte şi valorile minime ale funcţiilor de cost (J_t) pe de altă parte.

[570] Pentru calcularea valorilor expresiei $\dfrac{\partial}{\partial\alpha} J_t(\alpha,\theta)_{|\alpha=0}$ în acest caz, se poate folosi relaţia (256).



Remarcaţi faptul că expresia $\frac{\partial}{\partial\alpha}J_t(\alpha,\theta)_{|\alpha=0}$ exprimă în mod precis cât de „repede" (în funcţie de θ) vom putea începe să reducem costul empiric dacă vom creşte în mod gradual „votul" (α) pentru noua componentă, $h(x;\theta)$. Minimizarea expresiei (264) în raport cu θ pare a fi o modalitate rezonabilă de „estimare" a parametrului θ al noii componente, $h(x;\theta)$. Acest plan ne permite ca mai întâi să setăm θ la o anumită valoare ($\hat{\theta}_t$) şi apoi, la *Pasul 2* al iteraţiei t, să alegem valoarea lui α în aşa fel încât să minimizăm costul empiric. (Concret, votul α_t va fi determinat egalând cu 0 derivata parţială a lui $J_t(\alpha,\hat{\theta}_t)$ în raport cu α.)

Acum vom modifica uşor algoritmul schiţat mai sus în aşa fel încât să semene mai mult cu un *algoritm de boosting* (pregătind elaborarea *Pasului 3* al iteraţiei curente).[571] Vom defini următoarele *ponderi* (engl., weights) şi *ponderi normalizate* asociate exemplelor de antrenament:[572]

$$\begin{aligned} W_i^{(t)} &= -dL(y_i f_{t-1}(x_i)), \text{ pentru } i=1,\ldots,m \text{ şi} \\ \tilde{W}_i^{(t)} &= \frac{W_i^{(t)}}{\sum_{j=1}^m W_j^{(t)}}, \text{ pentru } i=1,\ldots,m. \end{aligned} \tag{265}$$

Aceste ponderi sunt, în mod evident, nenegative, întrucât funcţia de cost / pierdere este descrescătoare şi derivabilă (deci derivata ei trebuie să fie negativă sau zero). *Intuiţia* este că ponderea unei instanţe x_i este cu atât mai mică (respectiv mai mare) cu cât marginea algebrică $y_i f_{t-1}(x_i)$ este mai mare (respectiv mai mică), adică cu cât x_i este mai bine (respectiv mai prost) clasificat.

a. Arătaţi că a minimiza (în funcţie de θ) expresia $\frac{\partial}{\partial\alpha}J_t(\alpha,\theta)_{|\alpha=0}$ revine la a identifica ipoteza „slabă" care are cea mai bună eroare ponderată la antrenare, ε_t, unde

$$\varepsilon_t \overset{not.}{=} \sum_{i\,:\,y_i\neq h(x_i;\hat{\theta}_t)} \tilde{W}_i^{(t)}$$

***Observaţie*: Proprietatea afirmată aici este o generalizare a *Observaţiei importante* de la finalul rezolvării problemei 26. Acolo foloseam funcţia de cost negativ-exponenţială, iar aici considerăm** Loss **o funcţie de cost oarecare.**

[571] Termenul de *boosting* se referă la faptul că instanţele incorect clasificate la o iteraţie (de către ipoteza „slabă" aleasă la acea iteraţie) vor avea la iteraţia următoare probabilităţi mai mari, pentru ca algoritmul să se concentreze cu preponderenţă asupra lor.

[572] *Ponderilor normalizate* $\tilde{W}_i^{(t)}$ de aici le corespund în pseudo-codul algoritmului AdaBoost dat la problema 22 *probabilităţile* $D_t(i)$. Este îndreptăţit să folosim pentru $\tilde{W}_i^{(t)}$ şi denumirea de probabilităţi (*atenţie*!, pentru iteraţia t), fiindcă $\tilde{W}_i^{(t)} \in [0,\,1]$ şi $\sum_{i=1}^m \tilde{W}_i^{(t)} = 1$.

C. Acum putem începe să formulăm paşii corpului iterativ al noului algoritm de boosting într-un pseudo-cod similar cu algoritmul AdaBoost care a fost prezentat la problema 22:[573]

***Pasul 1*: Identifică o ipoteză / un clasificator $h(x;\hat{\theta}_t)$ care are o eroare ponderată la antrenare (engl., weighted training error) ε_t mai bună decât alegerea aleatorie, adică $\varepsilon_t < 1/2$, unde:**

$$\varepsilon_t \overset{not.}{=} \sum_{i\,:\,y_i \neq h(x_i;\hat{\theta}_t)} \tilde{W}_i^{(t)}. \tag{266}$$

***Pasul 2*: Setează „votul" α_t pentru noua componentă, minimizând costul empiric totală (engl., overall empirical loss):**

$$J_t(\alpha, \hat{\theta}_t) = \frac{1}{m}\sum_{i=1}^{m} \text{Loss}(y_i\, f_{t-1}(x_i) + y_i\, \alpha\, h(x_i;\hat{\theta}_t)),$$

adică

$$\alpha_t = \arg\min_{\alpha>0} J_t(\alpha, \hat{\theta}_t).$$

Se consideră că $f_0(x_i) = 0$ pentru $i = 1, \ldots, m$.

***Pasul 3*: Recalculează ponderile normalizate pentru următoarea iteraţie, astfel:**

$$\tilde{W}_i^{(t+1)} = -c_t \cdot dL(\underbrace{y_i\, f_{t-1}(x_i) + y_i\, \alpha_t\, h(x_i;\hat{\theta}_t)}_{y_i\, f_t(x_i)}) \text{ pentru } i = 1, \ldots, m,$$

unde constanta c_t este aleasă astfel încât $\sum_{i=1}^{m} \tilde{W}_i^{(t+1)} = 1$.

b. Arătaţi că, atunci când funcţia de cost este cea [negativ] *exponenţială*, adică $\text{Loss}(z) = \exp(-z)$, cei trei paşi din noul algoritm sunt în corespondenţă directă cu paşii algoritmului AdaBoost (de la problema 22). Mai concret, arătaţi că în acest caz setarea lui α_t (la Pasul 2) bazat pe noua ipoteză „slabă" $h(x;\hat{\theta}_t)$, precum şi actualizarea ponderii $\tilde{W}_i^{(t+1)}$ (la Pasul 3) conduc la un rezultat identic cu AdaBoost. (După cum am precizat deja, în problema 22 corespondentul lui $\tilde{W}_i^{(t+1)}$ este $D_{t+1}(i)$.)

MIT, 2003 fall, Tommi Jaakkola, HW4, pr. 2.3

c. Arătaţi că, pentru *orice funcţie de cost validă* de tipul discutat mai sus, componenta $h(x;\hat{\theta}_t)$, care a fost adăugată la iteraţia t, are eroarea ponderată la antrenare în raport cu ponderile actualizate $\tilde{W}_i^{(t+1)}$ exact $1/2$.[574]

CMU, 2008 fall, Eric Xing, HW3, pr. 4.1.2
CMU, 2008 fall, Eric Xing, midterm, pr. 5.1

d. Să presupunem acum că schimbăm funcţia obiectiv J_t în *media pătratelor erorilor*, adică

[573] Pentru partea de iniţializare, ca şi în cazul algoritmului AdaBoost, vom folosi $\tilde{W}_i^{(1)} = \frac{1}{m}$, pentru $i = 1, \ldots, m$. După aceea, corpul iterativ al algoritmului, format din paşii 1, 2 şi 3, se execută pentru $t = 1, \ldots, T$. Ca şi în cazul algoritmului AdaBoost de la problema 22, algoritmul generalizat va fi oprit dacă la o iteraţie oarecare t nu există nicio ipoteză $h(x;\theta)$ având eroarea ponderată la antrenare $\varepsilon_t \in (0,\, 1/2)$.

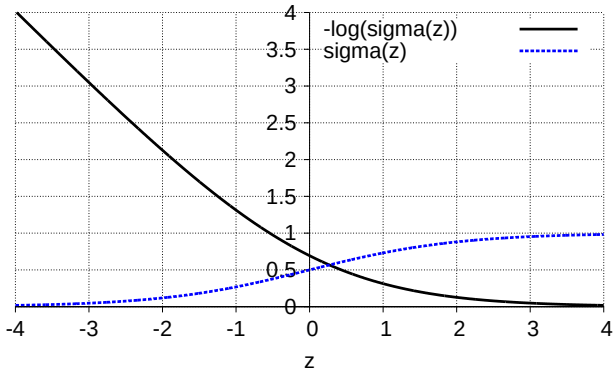
[574] Acest nou rezultat generalizează proprietatea care a fost demonstrată la problema 22.*vi*.

$$J_t = \frac{1}{m}\sum_{i=1}^{m}(y_i - f_t(x_i))^2 \textbf{ sau, alternativ, } \frac{1}{m}\sum_{i=1}^{m}[(y_i - f_t(x_i))^2 \cdot 1_{\{y_i f_t(x_i)<1\}}],$$

ca să fie satisfăcută condiţia de monotonie a funcţiei de cost Loss **pe întreaga axă reală.**[575] **Urmărim să optimizăm costul J_t, ca şi mai înainte, în mod secvenţial. Care este noua regulă de actualizare (engl., update rule) pentru α_t?**

D. *Funcţia* de cost *logistică* se defineşte astfel:

$$\text{Loss}(y_i f_T(x_i)) = \ln(1+\exp(-y_i f_T(x_i))).$$



MIT, 2006 fall, Tommi Jaakkola, HW4, pr. 3.a
MIT, 2009 fall, Tommi Jaakkola, HW3, pr. 2.1

e. Arătaţi că, atunci când folosim *funcţia de cost logistică* [în locul celei [negativ] exponenţiale], ponderile nenormalizate $W_i^{(t+1)}$ au valori mărginite superior de 1 (adică, $W_i^{(t+1)} < 1$). (Sugestie: Veţi exprima aceste ponderi în funcţie de $y_i f_t(x_i)$.)

MIT, 2003 fall, Tommi Jaakkola, HW4, pr. 2.2
MIT, 2011 fall, L. P. Kaelbling, HW5, pr. 1.1

f. În cazul funcţiei de cost *logistice*, care este valoarea / expresia ponderilor normalizate, $\tilde{W}_i^{(t+1)}$?
Ce puteţi spune despre ponderile normalizate pentru exemplele care sunt clasificate eronat în mod flagrant, comparativ cu cele care sunt clasificate uşor eronat de către combinaţia curentă („ansamblul" curent), $f_T(x) \stackrel{not}{=} \sum_{t=1}^{T}\alpha_t h(x;\hat{\theta}_t)$? Dacă datele de antrenament conţin exemple care au fost etichetate în mod greşit (engl., mislabeled), de ce credeţi că se preferă funcţia de cost logistică în locul celei [negativ] exponenţiale, $\text{Loss}(z) = \exp(-z)$**?**

Liviu Ciortuz, 2020

g. Presupunem că lucrăm cu *funcţia de cost logistică*. Care este în acest caz regula de actualizare pentru α_t?

***Indicaţie*: În loc să minimizaţi J_t în raport cu argumentul α (considerând ipoteza h_t deja aleasă / fixată), veţi urmări să minimizaţi în raport cu α**

[575] Observaţi că $(y_i - f(x_i))^2 = [y_i(1 - y_i f_t(x_i))]^2 = (1 - y_i f_t(x_i))^2 = (1 - z_i)^2$, unde am notat $z_i = y_i f_t(x_i)$. Funcţia $(1-z)^2$ este derivabilă şi convexă; este descrescătoare pe intervalul $(-\infty, 1]$ şi crescătoare pe intervalul $[1, +\infty)$. Varianta alternativă, adică funcţia $(1-z)^2 \cdot 1_{\{z<1\}}$, este descrescătoare pe toată axă reală.
Derivata funcţiei $(1-z)^2$ este $-2(1-z)$, iar derivata funcţiei $(1-z)^2 \cdot 1_{\{z<1\}}$ este $-2(1-z)\cdot 1_{\{z<1\}}$.

o *margine superioară* pentru *diferenţa* dintre J_t şi J^*_{t-1}, unde, aşa cum am definit la problema 26, $J^*_{t-1} \stackrel{not.}{=} \min_{h\in\mathcal{H},\alpha'\in\mathbb{R}_+} J_{t-1}(h,\alpha')$.[576]

MIT, 2006 fall, Tommi Jaakkola, HW4, pr. 3.b
MIT, 2009 fall, Tommi Jaakkola, HW3, pr. 2.2

h. Presupunem că lucrăm tot cu funcţia de *cost logistică*. Considerăm un set de date de antrenament liniar-separabil. Am dori să utilizăm o maşină cu vectori-suport liniară cu margine "hard" — adică, fără variabile / penalizări ecart (engl., slack penalties) — pe post de clasificator „slab".[577] **Presupunând că la *Pasul 1* al algoritmului AdaBoost generalizat se minimizează eroarea ponderată la antrenare ε_t, care va fi valoarea ponderii α_1 la prima iteraţie de boosting?**

Răspuns:

a. Pornind de la relaţia (264), o vom putea rescrie — folosind notaţia (265) — sub forma următoare:

$$\begin{aligned}\frac{\partial}{\partial\alpha}J_t(\alpha,\theta)_{|\alpha=0} &= -\frac{1}{m}\sum_{i=1}^m W_i^{(t)}\, y_i\, h(x_i;\theta)\\ &= -\frac{1}{m}\Big(\sum_j W_j^{(t)}\Big)\cdot\sum_{i=1}^m \frac{W_i^{(t)}}{\sum_j W_j^{(t)}}\, y_i\, h(x_i;\theta)\\ &= -\frac{1}{m}\Big(\sum_j W_j^{(t)}\Big)\cdot\sum_{i=1}^m \tilde{W}_i^{(t)}\, y_i\, h(x_i;\theta). \qquad (267)\end{aligned}$$

Întrucât factorul $\frac{1}{m}\sum_j W_j^{(t)}$ este pozitiv şi constant în raport cu θ, rezultă că a minimiza valoarea expresiei $\frac{\partial}{\partial\alpha}J_t(\alpha,\theta)_{|\alpha=0}$ este echivalent cu a minimiza expresia

$$-\sum_{i=1}^m \tilde{W}_i^{(t)} y_i h(x_i;\theta),$$

Notând — ca la problema 22 — mulţimile de indici $M = \{i = 1,\ldots,m \,|\, y_i \neq h(x_i;\hat{\theta}_t)\}$ şi $C = \{i = 1,\ldots,m \;\; | \, y_i = h(x_i;\hat{\theta}_t)\}$, este imediat că

$$\begin{aligned}-\sum_{i=1}^m \tilde{W}_i^{(t)} y_i h(x_i;\hat{\theta}_t) &= -\sum_{i\in M}\tilde{W}_i^{(t)} y_i h(x_i;\hat{\theta}_t) - \sum_{i\in C}\tilde{W}_i^{(t)} y_i h(x_i;\hat{\theta}_t)\\ &= \underbrace{\sum_{i\in M}\tilde{W}_i^{(t)}}_{\varepsilon_t} - \underbrace{\sum_{i\in C}\tilde{W}_i^{(t)}}_{1-\varepsilon_t} = 2\varepsilon_t - 1. \qquad (268)\end{aligned}$$

Aşadar,

$$\varepsilon_t = \frac{1}{2}\Big(1 - \sum_{i=1}^m \tilde{W}_i^{(t)} y_i h(x_i;\hat{\theta}_t)\Big). \qquad (269)$$

[576] Spre deosebire de J_t care depinde de α, J^*_{t-1} este constant, deci nu depinde de α. Prin urmare, a minimiza J_t în raport cu α este echivalent cu a minimiza $J_t - J^*_{t-1}$ în raport cu α. La finalul problemei 26.e am arătat că $J^*_t \geq J^*_{t-1}$ şi $J_t(h,0) = J^*_{t-1}$.

[577] LC: În locul acestui tip de SVM putem considera orice alt separator liniar consistent cu astfel de date de antrenament (adică, liniar separabile).

***Observaţii*:**
1. Întrucât $\tilde{W}_i^{(t)} \geq 0$ **şi** $\sum_{i=1}^m \tilde{W}_i^{(t)} = 1$**, iar** $y_i, h(x_i;\theta) \in \{-1,+1\}$**, rezultă că** $\sum_{i=1}^m \tilde{W}_i^{(t)} y_i h(x_i;\theta) \in [-1,+1]$**, deci** $1 - \sum_{i=1}^m \tilde{W}_i^{(t)} y_i h(x_i;\hat{\theta}_t) \in [0,+2]$**. Prin urmare, se verifică faptul că** $\varepsilon_t = \frac{1}{2}\left(1 - \sum_{i=1}^m \tilde{W}_i^{(t)} y_i h(x_i;\hat{\theta}_t)\right) \in [0,+1]$**.**
2. Mult mai important decât atât, din relaţiile (267) **şi** (268) **rezultă imediat că**

$$\frac{\partial}{\partial \alpha} J_t(\alpha,\theta)_{|\alpha=0} < 0 \Leftrightarrow 2\varepsilon_t - 1 < 0 \Leftrightarrow \varepsilon_t < 1/2.$$

b. Pentru prima parte, vom arăta că atunci când se foloseşte $\text{Loss}(z) = e^{-z}$**, valoarea obţinută pentru coeficientul** α **la minimizarea de la *Pasul 2* din algoritmul general (vedeţi partea stângă a egalităţii următoare) este aceeaşi cu valoarea lui** α **rezultată prin minimizarea efectuată de către algoritmul AdaBoost (vedeţi partea dreaptă a egalităţii următoare), adică**

$$\arg\min_{\alpha>0} \sum_{i=1}^m \text{Loss}(y_i f_{t-1}(x_i) + \alpha y_i h(x_i;\hat{\theta}_t)) = \arg\min_{\alpha>0} \sum_{i=1}^m \tilde{W}_i^{(t)} \exp(-\alpha y_i h(x_i;\hat{\theta}_t)),$$

unde, conform asignării de la execuţia *Pasului 3* al iteraţiei $t-1$**,**

$$\tilde{W}_i^{(t)} = c_{t-1} \cdot \exp(-y_i f_{t-1}(x_i)),$$

c_{t-1} **fiind constanta de normalizare (astfel încât ponderile însumate să dea valoarea 1). Calculând expresia funcţiei obiectiv din partea stângă a egalităţii de mai sus, obţinem:**[578]

$$\begin{aligned}
&\sum_{i=1}^m \text{Loss}(y_i f_{t-1}(x_i) + \alpha y_i h(x_i;\hat{\theta}_t)) = \sum_{i=1}^m \exp(-y_i f_{t-1}(x_i) - \alpha y_i h(x_i;\hat{\theta}_t)) \\
&= \sum_{i=1}^m \exp(-y_i f_{t-1}(x_i))\ \exp(-\alpha y_i h(x_i;\hat{\theta}_t)) \\
&= \frac{1}{c_{t-1}} \sum_{i=1}^m \tilde{W}_i^{(t)} \exp(-\alpha y_i h(x_i;\hat{\theta}_t)) \\
&= \frac{1}{c_{t-1}} \Big[\sum_{i \in \{i | y_i h(x_i;\hat{\theta}_t) = 1\}} \left(\tilde{W}_i^{(t)}\right) e^{-\alpha} + \sum_{i \in \{i | y_i h(x_i;\hat{\theta}_t) = -1\}} \left(\tilde{W}_i^{(t)}\right) e^{\alpha} \Big],
\end{aligned}$$

cantitate care este proporţională cu funcţia obiectiv minimizată de către algoritmul AdaBoost (vedeţi de asemenea problema 26.a). Prin urmare, minimizarea în raport cu α **conduce la acelaşi rezultat în ambii algoritmi.**

Pentru partea a doua, observaţi faptul că la *Pasul 3* (de la iteraţia t**) din algoritmul general asignarea ponderilor se face astfel:**

$$\tilde{W}_i^{(t+1)} = -c_t \cdot dL(y_i f_t(x_i)) = c_t \cdot \exp(-y_i f_t(x_i)),$$

ceea ce coincide cu modul cum procedează algoritmul AdaBoost (vedeţi problema 23.a).

c. La iteraţia t**, coeficientul** α_t **este ales minimizând** $J_t(\alpha,\hat{\theta}_t)$**, ceea ce implică** $\frac{\partial J_t(\alpha,\hat{\theta}_t)}{\partial \alpha} = 0$**. Întrucât**

[578] LC: A se vedea similaritatea cu demonstraţia de la problema 26.a.

$$\frac{\partial}{\partial\alpha}J_t(\alpha,\hat\theta_t) = \frac{1}{m}\sum_{i=1}^m\frac{\partial}{\partial\alpha}\text{Loss}(y_if_{t-1}(x_i)+y_i\alpha h(x_i;\hat\theta_t))$$
$$= \frac{1}{m}\sum_{i=1}^m\underbrace{dL(y_if_{t-1}(x_i)+y_i\alpha h(x_i;\hat\theta_t))}_{-\frac{\tilde W_i^{(t+1)}}{c_t}}y_ih(x_i;\hat\theta_t)\propto\sum_{i=1}^m\tilde W_i^{(t+1)}y_ih(x_i;\hat\theta_t),$$

rezultă că trebuie să avem

$$\sum_{i=1}^m\tilde W_i^{(t+1)}y_ih(x_i;\hat\theta_t)=0.$$

După aceasta, eroarea ponderată la antrenare produsă de $h(x;\hat\theta_t)$ (corespunzător ponderilor actualizate $\tilde W_i^{(t+1)}$ determinate de α_t) poate fi calculată într-un mod similar cu relaţia (269)**:**

$$\frac{1}{2}\Big(1-\underbrace{\sum_{i=1}^m\tilde W_i^{(t+1)}y_ih(x_i;\hat\theta_t)}_{0}\Big)=\frac{1}{2}(1-0)=\frac{1}{2}.$$

d. Vom lucra [mai întâi] cu $\text{Loss}(y_if_t(x_i))=(1-y_if_t(x_i))^2=(y_i-f_t(x_i))^2$, deci vom considera $J_t=\frac{1}{m}\sum_{i=1}^m(y_i-f_t(x_i))^2$.[579] **Pentru a găsi valoarea lui α_t, vom calcula derivata lui J_t în raport cu α_t şi apoi o vom egala cu zero.**

$$\frac{\partial J_t}{\partial\alpha_t} = \frac{1}{m}\frac{\partial\sum_{i=1}^m(y_i-f_t(x_i))^2}{\partial\alpha_t}=\frac{1}{m}\sum_{i=1}^m2(y_i-f_t(x_i))\frac{\partial(y_i-f_t(x_i))}{\partial\alpha_t}.$$

Ştim că[580]

$$f_t(x_i)=f_{t-1}(x_i)+\alpha_th_t(x_i).$$

În această expresie, f_{t-1} nu depinde de α_t. Înlocuind în expresia derivatei, obţinem

$$\frac{\partial J_t}{\partial\alpha_t} = \frac{2}{m}\sum_{i=1}^m(y_i-f_t(x_i))\frac{\partial(y_i-f_{t-1}(x_i)-\alpha_th_t(x_i))}{\partial\alpha_t}=\frac{2}{m}\sum_{i=1}^m(y_i-f_t(x_i))(-h_t(x_i)).$$

Egalând această derivată cu zero, vom avea:

$$\frac{\partial J_t}{\partial\alpha_t}=0\Leftrightarrow\sum_{i=1}^m(y_i-f_t(x_i))h_t(x_i)=0\Leftrightarrow\sum_{i=1}^m(y_i-f_{t-1}(x_i)-\alpha_th_t(x_i))h_t(x_i)=0$$
$$\Leftrightarrow\sum_{i=1}^m(y_i-f_{t-1}(x_i))h_t(x_i)=\alpha_t\sum_{i=1}^mh_t^2(x_i)\Leftrightarrow\alpha_t=\frac{\sum_{i=1}^m(y_i-f_{t-1}(x_i))h_t(x_i)}{\sum_{i=1}^m\underbrace{h_t^2(x_i)}_{1}}$$
$$\Leftrightarrow\alpha_t=\frac{1}{m}\sum_{i=1}^m(y_i-f_{t-1}(x_i))h_t(x_i),$$

[579] *Atenţie*! În acest caz, ponderile $W_i^{(t)}$ trebuie definite astfel: $W_i^{(t)}=|dL(y_if_{t-1}(x_i)|$, întrucât derivata funcţiei de cost $(1-z)^2$ nu este negativă pe toată axa reală.

[580] Atât aici cât şi în continuare, vom scrie $h_t(x_i)$ în loc de $h_t(x_i;\hat\theta_t)$ dacă se subînţelege din context că h_t a fost deja fixat.

cu condiţia ca această valoare să fie strict pozitivă.[581]

***Observaţie*: Egalitatea $h_t^2(x_i) = 1$ (pe care am folosit-o mai sus) decurge din presupunerea că ipotezele „slabe“ sunt compaşi de decizie.**

Dacă schimbăm funcţia obiectiv în $J_t = \frac{1}{m}\sum_{i=1}^{m}[(y_i - f_t(x_i))^2 \cdot 1_{\{y_i f_t(x_i)<1\}}]$, se verifică uşor că se obţine următoarea expresie pentru votul α_t:

$$\alpha_t = \frac{1}{\sum_{i=1}^{m} 1_{\{y_i f_{t-1}(x_i)<1\}}} \sum_{i=1}^{m} \left[(y_i - f_{t-1}(x_i))h_t(x_i) \cdot 1_{\{y_i f_{t-1}(x_i)<1\}}\right],$$

cu condiţia ca această valoare să fie strict pozitivă.

e. Ponderea $W_i^{(t+1)}$ a fost definită ca fiind $-\mathrm{dL}(y_i f_t(x_i))$, unde $\mathrm{dL}(z) \overset{not.}{=} \frac{\partial}{\partial z}(\ln(1+e^{-z}))$. Aşadar,

$$W_i^{(t+1)} = \frac{e^{-z_i}}{1+e^{-z_i}} = \frac{1}{1+e^{z_i}} < 1, \text{ unde } z_i = y_i f_t(x_i).$$

f. Ponderile normalizate au fost definite astfel:

$$\tilde{W}_i^{(t+1)} = c_t \cdot \frac{\exp(-y_i f_t(x_i))}{1+\exp(-y_i f_t(x_i))},$$

unde constanta de normalizare este

$$c_t = \left(\sum_{i=1}^{m} \frac{\exp(-y_i f_t(x_i))}{1+\exp(-y_i f_t(x_i))}\right)^{-1}.$$

Pentru exemplele de antrenament care sunt clasificate eronat în mod flagrant, $y_i\, f_t(x_i)$ are o valoare negativă mare în valoare absolută, aşadar $W_i^{(t+1)}$ va fi în acest caz aproape de [dar mai mic decât] 1. Pentru exemplele de antrenament care sunt clasificate uşor eronat, $W_i^{(t+1)}$ va fi aproape de [şi mai mare decât] $1/2$. Aşadar, ponderile normalizate pentru aceste două cazuri se vor afla într-un raport de cel mult $2:1$. Aceasta înseamnă că un singur outlier care este clasificat eronat în mod flagrant va avea întotdeuna o pondere care este de cel mult de două ori mai mare decât ponderea unei instanţe care este clasificată uşor eronat. Din acest motiv, algoritmul AdaBoost cu funcţie de cost logistică este robust la outliere.

În ce priveşte situaţia exemplelor de antrenament care au fost greşit etichetate, putem raţiona în felul următor. În cazul funcţiei de cost logistice, dacă avem un x_i care ar avea de drept etichetă $y_i = +1$, dar se consideră (în mod eronat) $y_i = -1$, costul este de aproximativ $f_t(x_i)$ dacă $f_t(x_i) > 0$,[582] **în vreme ce în cazul funcţiei de cost [negativ] exponenţiale costul este $\exp(f_t(x_i))$ care este în general mult mai mare decât $f_t(x_i)$. Cazurile simetrice ($f_t(x_i) \leq 0$ şi apoi $y_i = -1 \to +1$) se tratează în mod similar.**

[581] Verificarea condiţiei de minim este imediată. Dacă valoarea calculată aici pentru α_t este negativă, atunci $\min_{\alpha \geq 0} J_t$ se obţine pentru $\alpha = 0$. În acest caz, se poate renunţa la h_t şi se poate alege în schimb o altă ipoteză „slabă“, cu următoarea cea mai bună eroare ponderată la antrenare.

[582] Vedeţi graficul funcţiei de *cost logistic* din enunţul problemei de faţă.

g.[583] **Calculăm mai întâi valoarea expresiei $J_t - J^*_{t-1}$:**

$$\begin{aligned}
J_t - J^*_{t-1} &= \frac{1}{m}\sum_{i=1}^{m}\ln(1+\exp(-y_i f_t(x_i))) - \frac{1}{m}\sum_{i=1}^{m}\ln(1+\exp(-y_i f_{t-1}(x_i))) \\
&= \frac{1}{m}\sum_{i=1}^{m}\ln\frac{1+\exp(-y_i f_t(x_i))}{1+\exp(-y_i f_{t-1}(x_i))} \\
&= \frac{1}{m}\sum_{i=1}^{m}\ln\frac{\cancel{1+\exp(-y_i f_{t-1}(x_i))} - \cancel{\exp(-y_i f_{t-1}(x_i))} + \exp(-y_i f_t(x_i))}{1+\exp(-y_i f_{t-1}(x_i))} \\
&= \frac{1}{m}\sum_{i=1}^{m}\ln\Big(1+\frac{\exp(-y_i f_t(x_i)) - \exp(-y_i f_{t-1}(x_i))}{1+\exp(-y_i f_{t-1}(x_i))}\Big) \\
&= \frac{1}{m}\sum_{i=1}^{m}\ln\Big(1+\frac{\exp(-y_i f_{t-1}(x_i) - y_i\alpha h(x_i)) - \exp(-y_i f_{t-1}(x_i))}{1+\exp(-y_i f_{t-1}(x_i))}\Big) \\
&= \frac{1}{m}\sum_{i=1}^{m}\ln\Big(1+\frac{\exp(-y_i f_{t-1}(x_i))\exp(-y_i\alpha h(x_i)) - \exp(-y_i f_{t-1}(x_i))}{1+\exp(-y_i f_{t-1}(x_i))}\Big) \\
&= \frac{1}{m}\sum_{i=1}^{m}\ln\Big(1+\frac{\exp(-y_i f_{t-1}(x_i))[\exp(-y_i\alpha h(x_i)) - 1]}{1+\exp(-y_i f_{t-1}(x_i))}\Big) \\
&= \frac{1}{m}\sum_{i=1}^{m}\ln\left(1+\frac{\exp(-y_i\alpha h(x_i)) - 1}{\frac{1}{\exp(-y_i f_{t-1}(x_i))}+1}\right) \\
&= \frac{1}{m}\sum_{i=1}^{m}\ln\Big(1+\frac{\exp(-y_i\alpha h(x_i)) - 1}{\exp(y_i f_{t-1}(x_i))+1}\Big) \qquad (270)
\end{aligned}$$

Remarcaţi faptul că logaritmul din expresia (270) **există, întrucât**

$$\begin{aligned}
\frac{\exp(-y_i\alpha h(x_i)) - 1}{\exp(y_i f_{t-1}(x_i))+1} > -1 \quad &\Leftrightarrow \quad \exp(-y_i\alpha h(x_i)) - 1 > -\exp(y_i f_{t-1}(x_i)) - 1 \\
&\Leftrightarrow \quad \underbrace{\exp(-y_i\alpha h(x_i))}_{>0} > \underbrace{-\exp(y_i f_{t-1}(x_i))}_{<0},
\end{aligned}$$

inegalitate adevărată pentru $\forall\alpha$.

Mai departe, întrucât $\ln(1+z) \le z$ **pentru orice** $z > -1$,[584] **rezultă că**

$$J_t - J^*_{t-1} \le \frac{1}{m}\sum_{i=1}^{m}\frac{\exp(-y_i\alpha h(x_i)) - 1}{\exp(y_i f_{t-1}(x_i))+1} \qquad (271)$$

Expresia din partea dreaptă a inegalităţii (271) **este *marginea superioară* (engl., upper bound) pentru $J_t - J^*_{t-1}$ pe care o vom minimiza în raport cu α.**

Ştim din enunţ că $W_i^{(t)} = -dL(y_i f_{t-1}(x_i))$. Un calcul simplu ne arată că

$$\frac{\partial}{\partial z}\ln(1+e^{-z}) = \frac{-e^{-z}}{1+e^{-z}} = -\frac{1}{\frac{1}{e^{-z}}+1} = -\frac{1}{e^z+1}. \qquad (272)$$

[583] Soluţia care urmează a fost redactată iniţial de către studentul Ştefan Matcovici, cf. https://mitpress.mit.edu/sites/default/files/titles/content/boosting_foundations_algorithms/chapter007.html.

[584] Această inegalitate este foarte uşor de verificat analitic.

Rezultă că

$$W_i^{(t)} = -\Big(- \frac{1}{\exp(y_i f_{t-1}(x_i)) + 1} \Big) = \frac{1}{\exp(y_i f_{t-1}(x_i)) + 1} \text{ şi}$$
$$\tilde{W}_i^{(t)} = c_{t-1} \cdot \frac{1}{\exp(y_i f_{t-1}(x_i)) + 1},$$

unde c_{t-1} este constanta de normalizare a ponderilor $W_i^{(t)}$.
Prin urmare,

$$\begin{aligned} J_t - J_{t-1}^* &\leq \frac{1}{m} \sum_{i=1}^{m} \frac{1}{\exp(y_i f_{t-1}(x_i)) + 1} (\exp(-y_i \alpha h(x_i)) - 1) \\ &= \frac{1}{m\, c_{t-1}} \sum_{i=1}^{m} \tilde{W}_i^{(t)} \cdot (\exp(-y_i \alpha h(x_i)) - 1) \\ &\propto \sum_{i=1}^{m} \tilde{W}_i^{(t)} \cdot \exp(-y_i \alpha h(x_i)) - \underbrace{\sum_{i=1}^{m} \tilde{W}_i^{(t)}}_{1}. \end{aligned}$$

Aşadar, a minimiza marginea superioară pentru $J_t - J_{t-1}^*$ revine la a minimiza (în raport cu α) expresia $\sum_{i=1}^m \tilde{W}_i^{(t)} \cdot \exp(-y_i \alpha h(x_i))$, pe care am întâlnit-o şi la optimizarea costului [negativ] exponenţial (vedeţi problema 26.a). Putem conchide că şi în cazul funcţiei de cost logistice putem alege $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$.

***Observaţie*: Ţinând cont de faptul că aici nu se minimizează J_t ci o margine superioară pentru $J_t - J_{t-1}^*$, rezultă că atunci când se lucrează astfel (adică se foloseşte $\alpha_t = \frac{1}{2} \ln \frac{1-\varepsilon_t}{\varepsilon_t}$ dacă funcţia de cost este cea logistică), nu neapărat se va respecta proprietatea care a fost enunţată şi demonstrată la punctul c.**[585]

h. Conform enunţului, setul de date de antrenament este liniar-separabil. O maşină cu vectori-suport cu margine "hard" va produce pe un astfel de set de date un separator (liniar) care are eroarea ponderată la antrenare 0. Ca şi în cazul lui AdaBoost, algoritmul trebuie oprit.[586]

[585] Mulţumesc studentului Bogdan Palanici pentru discuţia care a dus la formularea acestei observaţii.

[586] Dacă am gândi în mod miop, am judeca astfel:
La *Pasul 1*, alegem $\hat{\theta}_1$, astfel încât să minimizăm $\frac{\partial J_t(\alpha, \theta)}{\partial \alpha}|_{\alpha=0}$. Echivalent, putem gândi că acest $\hat{\theta}_1$ este ales astfel încât să minimizăm suma ponderată $2\varepsilon_1 - 1 = -\sum_{i=1}^m \tilde{W}_i^{(1)} y_i h(x_i; \theta)$, unde $\tilde{W}_i^{(1)} = \frac{1}{m}$ pentru $i = 1, 2, \ldots, m$. Setul de date de antrenament fiind liniar-separabil, maşina cu vectori-suport cu margine "hard" va produce un separator $h(\cdot; \hat{\theta}_1)$ care satisface inegalitatea $y_i h(x_i; \hat{\theta}_1) \geq 1$ pentru $i = 1, 2, \ldots, m$.
La *Pasul 2*, trebuie să alegem α_1 astfel încât să minimizăm $J_t(\alpha_1, \hat{\theta}_1) = \frac{1}{m} \sum_{i=1}^m L(y_i h_0(x_i) + \alpha_1 y_i h(x_i; \hat{\theta}_1)) = \frac{1}{m} \sum_{i=1}^m L(\alpha_1 y_i h(x_i; \hat{\theta}_1))$. Remarcaţi faptul că $J_t(\alpha_1, \hat{\theta}_1)$ este o sumă de termeni care sunt strict decrescători în raport cu α_1 (pentru că $y_i h(x_i; \hat{\theta}_1) \geq 1$). Aşadar, şi suma $J_t(\alpha_1, \hat{\theta}_1)$ este strict decrescătoare în raport cu α_1. Aceasta ar implica faptul că algoritmul AdaBoost generalizat folosind funcţie de cost logistică va lua $\alpha_1 = \infty$. În general, dacă am avea un clasificator „de bază" care separă în mod perfect datele de antrenament, îi vom asocia un vot oricât putem de mare, ca să minimizăm costul. Lecţia este simplă: atunci când facem boosting, trebuie să folosim clasificatori „de bază" care nu sunt atât de puternici încât să separe în mod perfect datele de antrenament.

30. **(O margine superioară pentru eroarea la generalizare produsă de AdaBoost)**

□ • · ***Liviu Ciortuz, Andi Munteanu, după Princeton Univ., 2006 spring, COS 511, Rob Schapire, Lecture #11***

În această problemă vom calcula o margine superioară pentru eroarea pe care o obţine algoritmul AdaBoost la generalizare / testare, în funcţie de eroarea produsă la antrenare pe un set de date oarecare S. Vom folosi următoarele *notaţii*:

$\mathcal{H} \overset{not.}{=}$ **spaţiul ipotezelor „slabe“; vom presupune că $\mathcal{H}$ este finit** ($|\mathcal{H}| < +\infty$)

$co\,(\mathcal{H}) \overset{def.}{=} \{f(x) = \sum_j \alpha_j h_j(x) | \alpha_j \geq 0, \sum_j \alpha_j = 1, h_j \in \mathcal{H}\}$

(această mulţime se numeşte *înfăşurătoarea convexă* a lui $\mathcal{H}$)

$\mathcal{C}_N \overset{def.}{=} \{g(x) = \frac{1}{N} \sum_{j=1}^{N} h_j(x) | h_j \in \mathcal{H}\}, N \in \mathbb{N}^*$

(evident, $\mathcal{C}_N$ constituie o submulţime al lui $co\,(\mathcal{H}), \forall N \in \mathbb{N}^*$)

$D \overset{not.}{=}$ **o distribuţie de probabilitate peste $X \times \{-1, +1\}$**

$S \overset{not.}{=}$ **setul de exemple de antrenament**

$m \overset{not.}{=}$ **numărul de exemple de antrenament**

$P_D(\,\cdot\,) \overset{not.}{=}$ **probabilitatea unui set de exemple $(x, y) \sim D$**

$P_S(\,\cdot\,) \overset{not.}{=}$ **probabilitatea unui set de exemple (x, y) selectate în mod uniform din setul S.**

În cele ce urmează, ne vom concentra atenţia asupra unei combinaţii liniare de ipoteze „slabe“ $f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$, care a fost obţinută de către algoritmul AdaBoost după ce s-au executat T iteraţii pe setul de exemple S. Vom considera că $\sum_{t=1}^{T} \alpha_t = 1$; aşadar, voturile α_t sunt presupuse a fi normalizate.

***Observaţie*: În demonstraţia care urmează un rol esenţial îl va avea următoarea *proprietate interesantă*: funcţia $f \in co\,(\mathcal{H})$ poate fi aproximată cu ajutorul unor funcţii din $\mathcal{C}_N$. Într-adevăr, putem considera funcţiile $g \in \mathcal{C}_N$ definite prin expresia $g(x) = \frac{1}{N} \sum_{j=1}^{N} g_j(x)$, unde fiecare componentă g_j este selectată în mod aleatoriu din mulţimea de ipoteze „slabe“ $\{h_1, \ldots, h_N\}$, şi anume: g_j este desemnată ca fiind ipoteza „slabă“ h_t cu probabilitatea α_t.**[587] [588] **În aceste condiţii, este imediat că $E_g[g_j(x)] \overset{def.}{=} \sum_{t=1}^{T} \alpha_t h_t(x) = f(x)$.**[589]

În vederea determinării unei *margini superioare* în raport cu *eroarea la generalizare* produsă de algoritmul AdaBoost — calculul propriu-zis va fi elaborat la punctul e —, vom avea nevoie să demonstrăm patru rezultate intermediare

[587] Atenţie! Este posibil ca pentru doi indici diferiţi $j \neq j'$ să existe un acelaşi $t \in \{1, \ldots, T\}$ astfel încât să avem $g_j = g_{j'} = h_t$.

[588] În mod alternativ, g poate fi văzut ca un sondaj (engl., survey) efectuat asupra ipotezelor „slabe“ h_t. (Aceste ipoteze „slabe“ ar putea fi văzute ca „votanţi“.)

[589] Andi Munteanu a implementat un program pentru a „verifica“ această *proprietate interesantă* pe datele de la problema 25. Pentru cazul în care se lucrează cu compaşi de decizie exteriori, au fost obţinute următoarele aproximări ale voturilor normalizate α_t (şi anume, $\alpha_1 = 0.20376359$, $\alpha_2 = 0.32302264$, $\alpha_3 = 0.47321376$):

/ „leme“, care vor constitui, câte una pe rând, obiectul punctelor *a*-*d* care urmează.

a. (*Lema 1.*) Considerând instanţa de antrenament x arbitrar aleasă (dar fixată) în mulţimea X, precum şi un număr oarecare $\theta > 0$, demonstraţi că are loc următoarea inegalitate:

$$P_g\left(|f(x) - g(x)| > \frac{\theta}{2}\right) \leq 2\exp\left(-N\frac{\theta^2}{8}\right) \stackrel{not.}{=} \beta_\theta. \tag{273}$$

***Sugestie*: Folosiţi inegalitatea lui Hoeffding, enunţată la pr. 22.b, de la capitolul de *Fundamente*.**

b. (*Lema 2.*) Fie D' o distribuţie de probabilitate oarecare definită pe $X \times \{-1, +1\}$. Arătaţi că are loc următoarea inegalitate, care este asemănătoare cu cea de la punctul precedent, însă diferă de aceasta prin faptul că instanţa etichetată (x, y) este lăsată acum să urmeze distribuţia (nespecificată) D':

$$P_{D',g}\left(|yf(x) - yg(x)| > \frac{\theta}{2}\right) \leq \beta_\theta. \tag{274}$$

***Sugestie*: Puteţi folosi egalitatea $P_{D',g}(\,\cdot\,) = E_{D'}[P_g(\,\cdot\,)]$, unde P_g, este probabilitatea care a constituit obiectul inegalităţii de la punctul precedent. (Observaţi că P_g este probabilitate marginală în raport cu $P_{D',g}$.)**

c. (*Lema 3.*) Fie g definit ca mai sus, precum şi $\theta > 0$, ambii fixaţi. Considerăm $p_{g,\theta} \stackrel{not.}{=} P_D\left(yg(x) \leq \frac{\theta}{2}\right)$ şi $\hat{p}_{g,\theta} \stackrel{not.}{=} P_S\left(yg(x) \leq \frac{\theta}{2}\right)$, unde D şi S au fost introduse în prima parte a acestei probleme.[590] Demonstraţi că în raport cu alegerea setului de *date* de antrenament (S), are loc inegalitatea următoare:

$$P_{date}\left(p_{g,\theta} - \hat{p}_{g,\theta} > \varepsilon\right) \leq \exp(-2m\varepsilon^2). \tag{275}$$

***Sugestie*: Folosiţi din nou inegalitatea lui Hoeffding.**

d. (*Lema 4.*) Fie $\delta > 0$, oarecare. Arătaţi că are loc inegalitatea

$$P_{date}\left(\exists g \in \mathcal{C}_N \text{ şi } \theta > 0 : p_{g,\theta} > \hat{p}_{g,\theta} + \varepsilon\right) \leq \delta \tag{276}$$

dacă se ia

$$\varepsilon = \sqrt{\frac{\ln\left(\left(\frac{N}{2} + 1\right)|\mathcal{H}|^N/\delta\right)}{2m}}. \tag{277}$$

***Sugestie*: Folosiţi Lema 3 şi *inegalitatea lui Boole* (“union bounds”):**

$$P\left(\bigcup_{i=1}^{n} A_i\right) \leq \sum_{i=1}^{n} P(A_i),$$

iteraţia	α_1	α_2	α_3
1	0.0	0.0	1.0
10	0.0	0.8	0.2
10^2	0.15	0.41	0.44
10^3	0.207	0.327	0.466
10^4	0.2065	0.3209	0.4726
10^5	0.20326	0.32399	0.47275
10^6	0.202648	0.323807	0.473545
10^7	0.2037366	0.3230384	0.473225

[590] Funcţia g fiind construită în mod aleatoriu, rezultă că $p_{g,\theta}$ şi $\hat{p}_{g,\theta}$ pot fi văzute ca fiind variabile aleatoare.

unde $A_1, \ldots, A_n$ sunt evenimente aleatoare oarecare dintr-un spaţiu pe care a fost definită o distribuţie de probabilitate P.

***Consecinţă* dedusă din relaţia (276): Prin complementaritate, urmează că pentru orice $\delta \in (0,1]$, orice $g \in C_N$ şi orice $\theta > 0$ (de fapt, este suficient pentru $\theta \in (0,2]$), are loc inegalitatea**

$$p_{g,\theta} \leq \hat{p}_{g,\theta} + \varepsilon,$$

cu probabilitate de cel puţin $1-\delta$.

e. Acum vom „asambla" rezultatele care au fost demonstrate la punctele a-d.
***i*. Folosind Lema 2, arătaţi că**

$$P_D\left(yf(x) \leq 0\right) \leq E_g[P_D\left(yg(x) \leq \theta/2 \mid g\right)] + \beta_\theta,$$

iar după aceea, aplicând membrului drept al acestei inegalităţi Lema 4, va rezulta că pentru orice $\delta \in (0,1]$, inegalitatea următoare este satisfăcută cu [o] probabilitate de cel puţin $1-\delta$:

$$P_D\left(yf(x) \leq 0\right) \leq E_g[P_S\left(yg(x) \leq \theta/2 \mid g\right)] + \varepsilon + \beta_\theta,$$

unde ε a fost deja definit în relaţia (277).
***ii*. Folosind din nou Lema 2, arătaţi că**

$$E_g[P_S\left(yg(x) \leq \theta/2 \mid g\right)] + \varepsilon + \beta_\theta \leq P_S\left(yf(x) \leq \theta\right) + 4e^{-N\theta^2/8} + \sqrt{\frac{\ln\left[\left(\frac{N}{2}+1\right)|\mathcal{H}|^N/\delta\right]}{2m}}.$$

***iii*. Arătaţi că pentru $N = \left\lceil \frac{4}{\theta^2} \ln \frac{m}{\ln|\mathcal{H}|} \right\rceil$, inegalitatea rezultată de la punctele precedente (i şi ii), şi anume**

$$P_D\left(yf(x) \leq 0\right) \leq P_S\left(yf(x) \leq \theta\right) + 4e^{-N\theta^2/8} + \sqrt{\frac{\ln\left[\left(\frac{N}{2}+1\right)|\mathcal{H}|^N/\delta\right]}{2m}}$$

poate fi exprimată sub forma

$$P_D\left(yf(x) \leq 0\right) \leq P_S\left(yf(x) \leq \theta\right) + O\left(\frac{1}{\sqrt{m}}\sqrt{\frac{\ln m \cdot \ln|\mathcal{H}|}{\theta^2} - \ln\delta}\right).$$

Răspuns:

a. Fie $Z_j \overset{not.}{=} g_j(x)$. Rezultă că $g(x) = \frac{1}{N}\sum_{j=1}^N g_j(x) = \frac{1}{N}\sum_{j=1}^N Z_j \overset{not.}{=} \bar{Z}$. Întrucât variabilele aleatoare Z_j sunt mărginite de intervalul $[-1,1]$, aplicând inegalitatea lui Hoeffding, vom obţine:

$$\begin{aligned} P_g\left(|f(x) - g(x)| > \frac{\theta}{2}\right) = P_g\left(\left|E_g[Z] - \bar{Z}\right| > \frac{\theta}{2}\right) &\overset{Hoeffding}{\leq} 2\exp\left(-\frac{2N^2\frac{\theta^2}{4}}{4N}\right) \\ &= 2\exp\left(-\frac{N\theta^2}{8}\right). \end{aligned}$$

La scrierea primei egalităţi am folosit *Observaţia* din enunţ, conform căreia $f(x) = E_g[g_j(x)] = E_g[Z]$.

b. Folosind *sugestia* din enunţ şi apoi inegalitatea care a fost dedusă la punctul a (Lema 1), putem scrie:

$$\begin{aligned} & P_{D',g}\left(|yf(x) - yg(x)| > \frac{\theta}{2}\right) \\ \overset{not.}{=} \quad & P_{D',g}\left(\{(x,y)\Big| \, |yf(x) - yg(x)| > \frac{\theta}{2}\right) \\ = \quad & E_{D'}\left[P_g\left(|yf(x) - yg(x)| > \frac{\theta}{2}\Big| D'\right)\right] \\ \overset{(273)}{\leq} \quad & E_{D'}[\beta_\theta] = \beta_\theta. \end{aligned}$$

c. Definim variabilele aleatoare Q_i pentru $i = 1, \ldots, m$, unde $m = |S|$, astfel: $Q_i = \begin{cases} 1, \text{ dacă } y_i g(x_i) \leq \theta/2; \\ 0, \text{ în cazul contar}. \end{cases}$
Rezultă că

$$p_{g,\theta} \overset{not.}{=} P_D\left(yg(x) \leq \frac{\theta}{2}\right) = E_D[Q_i] \text{ şi } \hat{p}_{g,\theta} \overset{not.}{=} P_S\left(yg(x) \leq \frac{\theta}{2}\right) = \frac{1}{m}\sum_{i=1}^{m} Q_i \overset{not.}{=} \bar{Q}.$$

Ţinând cont că variabilele Q_i iau valori în mulţimea $\{0, +1\}$ şi aplicând a doua inegalitate a lui Hoeffding (vedeţi pr. 22.b), obţinem:

$$P_{date}\left(p_{g,\theta} - \hat{p}_{g,\theta} > \varepsilon\right) = P_{date}\left(E_D[Q_i] - \bar{Q} > \varepsilon\right) \leq \exp\left(-\frac{2m^2\varepsilon^2}{m \cdot 1^2}\right) = \exp\left(-2m\varepsilon^2\right).$$

d. Pentru a evalua probabilitatea $P_{date}\left(\exists g \in \mathcal{C}_N \text{ şi } \theta > 0 : p_{g,\theta} > \hat{p}_{g,\theta} + \varepsilon\right)$, vom face două *remarci*. Prima dintre ele se referă la numărul de valori pe care este suficient să le analizăm pentru C_N şi respectiv pentru θ. Din definiţia mulţimii C_N, rezultă $|\mathcal{C}_N| = |\mathcal{H}|^N$. Acum vom arăta că din infinitatea de valori pe care θ le poate lua, există doar un număr finit de valori care sunt de interes. Într-adevăr,

$$yg(x) \leq \theta/2 \Leftrightarrow \frac{y}{N}\sum_j g_j(x) \leq \frac{\theta}{2} \Leftrightarrow y\sum_j g_j(x) \leq \frac{N}{2}\,\theta \Leftrightarrow y\sum_{j=1}^{N} \underbrace{g_j(x)}_{\in\{-1,\,+1\}} \leq \left\lfloor \frac{N}{2}\,\theta \right\rfloor.$$

Din enunţul punctului e se observă că este suficient să considerăm $\theta \leq 1$. Prin urmare, valorile nenegative pe care le poate lua expresia $y\sum_{j=1}^{N} g_j(x)$ sunt $0, 1, \ldots, \left\lfloor \frac{N}{2} \right\rfloor$. Acestea sunt în număr de $\left\lfloor \frac{N}{2} \right\rfloor + 1$.

A doua *remarcă* este următoarea. Fie $\tilde{\theta} = \frac{2}{N}v$, cu $v \in \{0, 1, \ldots, \left\lfloor \frac{N}{2} \right\rfloor\}$. Este imediat că $p_{g,\theta} = p_{g,\tilde{\theta}}$ pentru orice $\theta \in \left[\tilde{\theta}, \tilde{\theta} + \frac{2}{N}\right)$:

$$p_{g,\tilde{\theta}} \overset{not.}{=} P_D\left(yg(x) \leq \frac{1}{2}\tilde{\theta}\right) = P_D\left(yg(x) \leq \frac{1}{2}\theta\right) \overset{not.}{=} p_{g,\theta}.$$

Similar, are loc egalitatea $\hat{p}_{g,\theta} = \hat{p}_{g,\tilde{\theta}}$.
Folosind aceste două remarci, putem scrie acum:

$$\begin{aligned}
& P_{date}\left(\exists g \in \mathcal{C}_N \text{ \textbf{şi} } \theta > 0 : p_{g,\theta} > \hat{p}_{g,\theta} + \varepsilon\right) \\
= \quad & P_{date}\left(\exists g \in \mathcal{C}_N \text{ \textbf{şi} } \theta > 0 : p_{g,\tilde{\theta}} > \hat{p}_{g,\tilde{\theta}} + \varepsilon\right) \\
\overset{\text{ineg. Boole}}{\leq} \quad & |\mathcal{H}|^N \left(\frac{N}{2} + 1\right) P_{date}\left(p_{g,\tilde{\theta}} > \hat{p}_{g,\tilde{\theta}} + \varepsilon\right) \\
\overset{(275)}{\leq} \quad & |\mathcal{H}|^N \left(\frac{N}{2} + 1\right) e^{-2\varepsilon^2 m}.
\end{aligned}$$

Egalând cu δ ultima expresie pe care am obţinut-o, vom avea:

$$\begin{aligned}
|\mathcal{H}|^N \left(\frac{N}{2} + 1\right) e^{-2\varepsilon^2 m} = \delta \Leftrightarrow \\
|\mathcal{H}|^N \left(\frac{N}{2} + 1\right) = \delta\, e^{2\varepsilon^2 m} \Leftrightarrow \\
2\varepsilon^2 m = \ln\left(|\mathcal{H}|^N \left(\frac{N}{2} + 1\right)/\delta\right) \Leftrightarrow \\
\varepsilon^2 = \frac{1}{2m} \ln\left(|\mathcal{H}|^N \left(\frac{N}{2} + 1\right)/\delta\right) \Leftrightarrow \\
\varepsilon = \sqrt{\frac{\ln\left[\left(\frac{N}{2} + 1\right)|\mathcal{H}|^N/\delta\right]}{2m}}.
\end{aligned}$$

e. Folosind lemele pe care le-am demonstrat anterior, vom obţine:

$$\begin{aligned}
& P_D\left(yf(x) \leq 0\right) \\
= \quad & P_{D,g}\left(yf(x) \leq 0\right) && (278) \\
= \quad & P_{D,g}\left(yf(x) \leq 0 \ \wedge yg(x) \leq \theta/2\right) + \\
& P_{D,g}\left(yf(x) \leq 0 \ \wedge yg(x) > \theta/2\right) && (279) \\
\leq \quad & P_{D,g}\left(yg(x) \leq \theta/2\right) + \underbrace{P_{D,g}\left(|yf(x) - yg(x)| > \theta/2\right)}_{\leq \beta_\theta} && (280) \\
\overset{\text{Lema 2}}{\leq} \quad & E_g[P_D\left(yg(x) \leq \theta/2 \mid g\right)] + \beta_\theta && (281) \\
\overset{\text{Lema 4}}{\leq} \quad & E_g[P_S\left(yg(x) \leq \theta/2 \mid g\right) + \varepsilon] + \beta_\theta \text{ \textbf{cu probabilitate} } > 1 - \delta, \\
& \text{\textbf{unde} } \varepsilon \text{ \textbf{a fost definit de relaţia} } (277); \\
= \quad & E_g[P_S\left(yg(x) \leq \theta/2 \mid g\right)] + \varepsilon + \beta_\theta && (282) \\
= \quad & P_{S,g}\left(yg(x) \leq \theta/2\right) + \varepsilon + \beta_\theta && (283) \\
= \quad & P_{S,g}\left(yg(x) \leq \theta/2 \wedge yf(x) \leq \theta\right) + \\
& P_{S,g}\left(yg(x) \leq \theta/2 \wedge yf(x) > \theta\right) + \varepsilon + \beta_\theta && (284) \\
\leq \quad & P_{S,g}\left(yf(x) \leq \theta\right) + \underbrace{P_{S,g}\left(|yf(x) - yg(x)| > \theta/2\right)}_{\leq \beta_\theta} + \varepsilon + \beta_\theta && (285) \\
\overset{\text{Lema 2}}{\leq} \quad & P_S\left(yf(x) \leq \theta\right) + 2\beta_\theta + \varepsilon \\
\overset{(273),(277)}{=} \quad & P_S\left(yf(x) \leq \theta\right) + 4e^{-N\theta^2/8} + \sqrt{\frac{\ln\left[\left(\frac{N}{2} + 1\right)|\mathcal{H}|^N/\delta\right]}{2m}}.
\end{aligned}$$

Iată câteva explicaţii succinte pentru deducerea relaţiilor de mai sus:

(278): P_D **este distribuţie / probabilitate marginală în raport cu** $P_{D,g}$**, iar** $yf(x)$ **nu depinde de** g**;**

(279): **se aplică proprietatea de aditivitate numărabilă din definiţia funcţiei de probabilitate;**

(280): **se aplică proprietatea** $A \subseteq B \Rightarrow P(A) \leq P(B)$**, respectiv se ţine cont de faptul că în condiţiile date avem** $|yf(x) - yg(x)| = -(yf(x) - yg(x)) > \theta/2$**;**

(281): P_D **este distribuţie / probabilitate marginală în raport cu** $P_{D,g}$**;**

(282): ε **este constant în raport cu alegerea lui** g**;**

(283): P_S **este distribuţie / probabilitate marginală în raport cu** $P_{S,g}$**;**

(284): **se aplică proprietatea de aditivitate numărabilă;**

(285): **se aplică proprietatea** $A \subseteq B \Rightarrow P(A) \leq P(B)$**, respectiv se ţine cont de faptul că în condiţiile date avem** $yf(x) - yg(x) < -\theta/2 < 0$**, deci** $|yf(x) - yg(x)| = -(yf(x) - yg(x)) > \theta/2$**.**

În fine, se poate arăta relativ uşor că atunci când se ia $N = \left\lceil \frac{4}{\theta^2} \ln \frac{m}{\ln |\mathcal{H}|} \right\rceil$**, ultima expresie de mai sus poate fi pusă sub forma**

$$P_S\left(yf(x) \leq \theta\right) + O\left(\frac{1}{\sqrt{m}} \sqrt{\frac{\ln m \cdot \ln |\mathcal{H}|}{\theta^2} - \ln \delta}\right).$$

Într-adevăr,

$$N = \left\lceil \frac{4}{\theta^2} \ln \frac{m}{\ln |\mathcal{H}|} \right\rceil \Rightarrow \frac{4}{\theta^2} \ln \frac{m}{\ln |\mathcal{H}|} \leq N < \frac{4}{\theta^2} \ln \frac{m}{\ln |\mathcal{H}|} + 1$$

$$\Rightarrow -\frac{1}{2} \ln \frac{m}{\ln |\mathcal{H}|} \geq -N \frac{\theta^2}{8} > -\frac{1}{2} \ln \frac{m}{\ln |\mathcal{H}|} - \frac{\theta^2}{8}$$

$$\Rightarrow \exp\left(-\frac{1}{2} \ln \frac{m}{\ln |\mathcal{H}|}\right) \geq \exp\left(-N \frac{\theta^2}{8}\right) > \exp\left(\frac{1}{2} \ln \frac{m}{\ln |\mathcal{H}|} - \frac{\theta^2}{8}\right)$$

$$\Rightarrow \sqrt{\frac{\ln |\mathcal{H}|}{m}} \geq e^{-N \frac{\theta^2}{8}} > \sqrt{\frac{\ln |\mathcal{H}|}{m}} \cdot e^{-\frac{\theta^2}{8}}.$$

Evident, $e^{-\theta^2/8} \leq 1$**. Aşadar, ordinul expresiei** $e^{-N \frac{\theta^2}{8}}$ **este** $O\left(\frac{\sqrt{\ln |\mathcal{H}|}}{\sqrt{m}}\right)$**.**

Apoi, întrucât

$$\varepsilon = \sqrt{\frac{\ln\left[\left(\frac{N}{2} + 1\right) |\mathcal{H}|^N / \delta\right]}{2m}} = \frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{m}} \cdot \sqrt{\ln\left[\left(\frac{N}{2} + 1\right) |\mathcal{H}|^N / \delta\right]} =$$

$$\frac{1}{\sqrt{2}} \cdot \frac{1}{\sqrt{m}} \cdot \sqrt{\ln\left(\frac{N}{2} + 1\right) + N \ln |\mathcal{H}| - \ln \delta},$$

ordinul expresiei $\frac{N}{2} + 1$ **este cel mult** $O\left(\frac{\ln m}{\theta^2}\right)$ **şi ordinul expresiei** $N \ln |\mathcal{H}|$ **este cel mult** $O\left(\frac{\ln m}{\theta^2} \cdot \ln |\mathcal{H}|\right)$**, rezultă că ordinul expresiei** $2\beta_\theta + \varepsilon$ **este cel mult** $O\left(\frac{1}{\sqrt{m}} \sqrt{\frac{\ln m \cdot \ln |\mathcal{H}|}{\theta^2} - \ln \delta}\right)$**.**

31. **(Algoritmul AdaBoost: Adevărat sau Fals?)**

□ • ○ ***MIT, 2003 fall, Tommi Jaakkola, final, pr. 3.1-2***
MIT, 2001 fall, Tommi Jaakkola, midterm, pr. 4.3
MIT, 2002 fall, Tommi Jaakkola, midterm, pr. 5.4
CMU, 2011 spring, Eric Xing, HW5, pr. 3.1.b

a. Adeseori, mai ales în practică, ε_t, eroarea ponderată produsă la antrenare de către ipoteza h_t / clasificatorul „slab“ A (măsurată relativ la ponderile de la începutul iteraţiei t) tinde să crească în raport cu t.

b. În decursul iteraţiilor executate de algoritmul AdaBoost, erorile ponderate ε_t (produse la antrenare, pe rând, de către ipotezele „slabe“ h_t, în ocurenţă, compaşii de decizie) pe de o parte, şi erorile produse la antrenare de către clasificatorii combinaţi H_t pe de altă parte, variază aproximativ la fel.

c. Ponderile / „voturile“ α_t asignate de către algoritmul AdaBoost clasificatorilor „slabi“ h_t asamblaţi sunt întotdeauna nenegative.

d. Probabilităţile / ponderile $D_t(i)$ alocate de către algoritmul AdaBoost exemplelor de antrenament care au fost clasificate eronat [de către ipoteza h_t] vor creşte cu un acelaşi factor multiplicativ.

e. Întotdeauna după ce algoritmul AdaBoost execută suficient de multe iteraţii, eroarea la antrenare produsă de ipoteza combinată H_t descreşte la o valoare care este oricât [dorim să fie] de apropiată de zero, indiferent de tipul de clasificatori „slabi“ folosiţi.

Răspuns:

a. Adevărat (în general). Modul de definire a probabilităţilor / ponderilor $D_t(i)$ asignate exemplelor de antrenament face ca algoritmul AdaBoost să se concentreze asupra exemplelor care sunt dificil de clasificat corect. După câteva iteraţii, cea mai mare parte a „masei“ de probabilitate va fi alocată acestor exemple „dificile“, iar eroarea ponderată la antrenare comisă de următoarea ipoteză „slabă“ va fi mai apropiată de $1/2$ (care reprezintă eroarea corespunzătoare alegerii aleatorii).

b. Fals. În vreme ce eroarea la antrenare produsă de către clasificatorul combinat H_t în mod tipic descreşte ca funcţie de t (numărul de iteraţii executate de AdaBoost), erorile ponderate la antrenare ε_t produse de ipotezele „slabe“ h_t în mod tipic devin din ce în ce mai mari (aşa cum am justificat deja la punctul precedent), fiindcă ponderile / probabilităţile $D_t(i)$ se alocă din ce în ce mai mult exemplelor care sunt dificil de clasificat.

c. Adevărat. După cum s-a specificat în pseudo-codul din problema 22, algoritmul AdaBoost alege la fiecare iteraţie (t) ipoteze „slabe“, care au o eroare ponderată la antrenare ε_t strict mai mică decât $1/2$. Prin urmare, $\ln((1-\varepsilon_t)/\varepsilon_t) > 0$, ceea ce înseamnă că „votul“ α_t este pozitiv.

d. Adevărat. Puteţi verifica acest fapt analizând formula (245) din problema 22, pentru actualizarea probabilităţilor $D_t(i)$. Întrucât pentru toate exemplele incorect clasificate avem $y_i \neq h_t(x_i)$, iar y_i şi $h_t(x_i)$ pot fi doar ± 1, rezultă că probablităţile / ponderile alocate lor vor fi multiplicate cu factorul $\exp(-\alpha_t y_i h_t(x_i)) = \exp(\alpha_t)$. După aceea se face normalizarea, cu un acelaşi factor, Z_t.

e. Fals. Dacă la o anumită iteraţie t a algoritmului AdaBoost clasificatorul „slab“ A nu poate produce nicio ipoteză care să aibă eroarea ponderată la antrenare $\varepsilon_t < 1/2$, atunci algoritmul AdaBoost se opreşte.[591] **La momentul respectiv, cea mai mică dintre erorile produse la antrenare de către clasificatorii [combinaţi] $H_{t'}$ cu $t' = 1, \ldots, t$, este fie 0 fie strict pozitivă.**

Însă, chiar şi în cazul în care clasificatorul „slab“ A produce la orice iteraţie ipoteze h_t cu eroare $\varepsilon_t < 0.5$ (deci $\gamma_t \overset{not.}{=} \frac{1}{2} - \varepsilon_t > 0$), deşi avem certitudinea că *marginea superioară* $\exp(-2\sum_{t'=1}^{t} \gamma_t^2)$ pentru $err_S(H_t)$ descreşte ca funcţie de t (vedeţi pr. 23.d), nu putem avea şi certitudinea că ea converge la 0, şi nici certitudinea că *eroarea* $err_S(H_t)$ însăşi descreşte mereu. Vedeţi, de exemplu, soluţia a doua de la problema 25 (cea care a fost obţinută fără a folosi prag exterior).

[591] Altminteri, adică atunci când $\varepsilon_t = 1/2$, ponderea ipotezei h_t respective ar fi $\alpha_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right) = \frac{1}{2} \ln 1 = 0$, deci h_t nu mai poate îmbunătăţi combinaţia liniară învăţată $f(x)$ şi, prin urmare, eroarea clasificatorului combinat H_t rămâne neschimbată.

4.2 Arbori de decizie — Probleme propuse

4.2.1 Algoritmul ID3

32. **(Arbori de decizie; optimalitate, ca număr minim de noduri)**

*

Reprezentaţi arborele / arborii de decizie care are / au numărul minim posibil de noduri (de test) şi corespunde / corespund funcţiei booleene $(A \text{ XOR } B) \wedge C$ definită peste atributele booleene A, B şi C.

33. **(Expresivitatea arborilor de decizie: un rezultat privind funcţiile booleene)**

Orice funcţie booleană (care primeşte n argumente din mulţimea $\{0,1\}$ şi întoarce un element din mulţimea $\{0,1\}$) poate fi reprezentată cu ajutorul unui arbore de decizie. Adevărat sau fals?

În cazul afirmativ, explicaţi succint cum anume poate fi construit arborele de decizie respectiv.

În cazul negativ, daţi un exemplu de funcţie booleană pentru care nu se poate construi un arbore de decizie consistent cu funcţia respectivă.

34. **(Calcularea câştigului de informaţie pe "decision stumps")**

■ □ • ***CMU, 2013 fall, W. Cohen, E. Xing, Sample Questions, pr. 4***

Studentul Timmy doreşte să ştie cum [ar trebui să procedeze cel mai bine ca] să promoveze examenul de învăţare automată. Pentru aceasta, a cules informaţii de la studenţii care au urmat acest curs în anii precedenţi şi apoi a decis să-şi construiască un *model* bazat pe arbori de decizie. A colectat în total nouă *instanţe* / exemple, descrise cu ajutorul a două *trăsături* (văzute în cele ce urmează ca două variabile aleatoare, S şi A): „este bine să stai şi să înveţi până noaptea târziu înainte de examen" (S) şi „este bine să mergi la toate cursurile şi seminariile" (A). Timmy dispune acum de următoarele „statistici" (care sunt de fapt *partiţionări* ale datelor sale):

$$Set(all) = [5+, 4-]$$
$$Set(S+) = [3+, 2-], Set(S-) = [2+, 2-]$$
$$Set(A+) = [5+, 1-], Set(A-) = [0+, 3-]$$

Presupunând că se foloseşte drept criteriu de selecţie a celei mai bune trăsături câştigul maxim de informaţie, ce trăsătură va alege Timmy? Care este valoarea câştigului de informaţie?

Puteţi folosi la calcule următoarele aproximaţii:

N	3	5	7
$\log_2 N$	1.5850	2.3219	2.8073

35. **(Implementare: compas de decizie, entropie, entropie condiţională specifică, entropie condiţională medie, câştig de informaţie)**

□ ***Liviu Ciortuz, 2016***

Folosind limbajul de programare pe care-l preferaţi, implementaţi un program care, pornind de la o structură de date de tip compas de decizie (engl., decision stump), calculează entropia, entropiile condiţionale specifice, entropia condiţională medie, precum şi câştigul de informaţie aferent.

În mod concret, programul va primi ca *input*

– m — numărul de valori posibile ale etichetei / atributului de ieşire (în mod implicit, se va considera $m = 2$);

– n — numărul de valori ale atributului (notat mai jos cu A) în raport cu care se face partiţionarea mulţimii de instanţe asociate nodului-rădăcină al compasului de decizie (valoarea implicită: $n = 2$);

– partiţiile (de fapt, count-urile) corespunzătoare nodurilor descendente. Pornind de la aceste partiţii, programul va calcula partiţia asociată nod-rădăcină al compasului de decizie.

De *exemplu*, pentru primul compas de decizie de la problema 34, inputul va avea forma $[3, 2]$, $[2, 2]$, în vreme ce pentru al doilea compas de decizie va fi $[5, 1]$, $[0, 3]$.

Programul va calcula şi apoi va afişa

– entropia atributului / variabilei de ieşire (notată aici cu Y);

– entropiile condiţionale specifice pentru fiecare descendent din nodul-rădăcină;

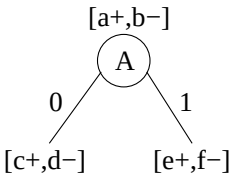
– entropia condiţională medie a atributului A;

– câştigul de informaţie al atributului [de ieşire] Y în raport cu atributul [de intrare] A.

36. **(Compaşi de decizie; entropie, entropia condiţională medie şi câştigul de informaţie: formule de calcul adecvate pentru folosirea calculatorului de buzunar)**

■ • ∗ ***UAIC, Iaşi, 2017, Sebastian Ciobanu, Liviu Ciortuz***

Fie compasul de decizie din figura alăturată. a, b, c, d, e şi f reprezintă count-uri corespunzătoare unui set de date de antrenament. După cum se observă, eticheta (sau, variabila de ieşire), notată cu Y, este binară, iar atributul (sau, variabila de intrare) A este de asemenea binar. Evident, $a = c + e$ şi $b = d + f$.



a. Arătaţi că entropia [variabilei de ieşire] corespunzătoare partiţiei asociate nodului de test este

$$H[a+,b-] \;=\; \frac{1}{a+b}\log_2\frac{(a+b)^{a+b}}{a^a b^b} \quad \textbf{dacă } a\neq 0 \textbf{ şi } b\neq 0.$$

b. Cum s-ar scrie formula corespunzătoare entropiei variabilei de ieşire în cazul când ea este ternară, iar partiţia din nodul de test [al compasului de decizie] este $[a+,b-,c*]$**?**

***Atenţie*! Nu există nicio legătură între acest ultim** c **şi count-ul c din compasul de decizie de mai sus.**

c. Presupunând că niciunul dintre c,d,e **şi** f **nu este nul, arătaţi că entropia condiţională medie corespunzătoare compasului de decizie din desenul de mai sus este**

$$H_{nod\,|\,atribut} \;=\; \frac{1}{a+b}\log_2\left(\frac{(c+d)^{c+d}}{c^c d^d}\cdot\frac{(e+f)^{e+f}}{e^e f^f}\right).$$

d. Să presupunem acum că unul dintre count-urile c,d,e **şi** f **este** 0**; pentru fixarea ideilor vom considera** $c=0$**. Elaboraţi formula entropiei condiţionale medii pentru compasul de decizie în acest caz.**

e. Demonstraţi următoarea formulă pentru câştigul de informaţie corespunzătoare compasului de decizie de mai sus, presupunând că a,b,c,d,e **şi** f **sunt strict pozitive:**

$$IG_{nod;atribut} \;=\; \frac{1}{a+b}\log_2\left(\frac{(a+b)^{a+b}}{a^a b^b}\cdot\frac{c^c d^d}{(c+d)^{c+d}}\cdot\frac{e^e f^f}{(e+f)^{e+f}}\right).$$

Observaţie* (1): Întrucât majoritatea calculatoarelor de buzunar *nu* au funcţia** $\log_2$ ***ci **funcţiile** $\ln$ **şi** $\lg$**, în formulele prezentate sau deduse la punctele** $a-e$ **ar fi de dorit să schimbăm baza logaritmului. Aceasta revine – pe lângă înlocuirea lui** $\log_2$ **cu** $\ln$ **sau** $\lg$ **– la înmulţirea membrului drept cu** $1/\ln 2$**, respectiv** $1/\lg 2$**.**

***Observaţie* (2): Întrucât, la aplicarea algoritmului ID3, pentru alegerea celui mai bun atribut de pus în nodul curent este suficient să calculăm entropiile condiţionale medii, va fi suficient să comparăm produsele de forma**

$$\frac{(c+d)^{c+d}}{c^c d^d}\cdot\frac{(e+f)^{e+f}}{e^e f^f} \tag{286}$$

pentru compaşii de decizie consideraţi la nodul respectiv şi să alegem minimul dintre aceste produse.

***Atenţie*! O problemă importantă care poate apărea la folosirea acestor formule în lucrul cu calculatorul de buzunar este *depăşirea capacităţii de reprezentare* a rezultatelor intermediare. Spre exemplu, la un calculator Sharp EL-531VH, putem lucra cu** 56^{56} **dar nu şi cu** 57^{57}**. Similar, pe calculatorul disponibil în [meniul *Accessories* din] sistemul de operare Linux Mint putem lucra cu** 179^{179} **dar nu şi cu** 180^{180}**. Din acest motiv, în cazul depăşirii capacităţii de reprezentare pe calculatoare de buzunar, vă sugerăm fie să folosiţi forma logaritmată a expresiei** (286)**, adică**

$$(c+d)\log(c+d)-c\log c-d\log d+(e+f)\log(e+f)-e\log e-f\log f,$$

fie să utilizaţi formulele de bază pentru entropii şi pentru câştigul de informaţie (vedeţi problema 55 de la capitolul de *Fundamente*).

***Comentariu*:**
Raţionamente relaţionale / „calitative“
în contextul calculării entropiilor condiţionale medii

Acest tip de raţionament — spre deosebire de *raţionamentul cantitativ*, reprezentat de calculul entropiei condiţionale (specifice şi apoi medii) folosind efectiv *definiţiile clasice*[592] — este justificat în felul următor.

Conform notaţiei folosite de Tom Mitchell în cartea *Machine Learning*, ştim că entropia condiţională medie a lui S (setul de date de antrenament) în raport cu atributul A este definită astfel:

$$H(S|A) \stackrel{def.}{=} \sum_{v \in Val(A)} \frac{|S_v|}{|S|} \cdot H(S|A=v) \stackrel{not.}{=} \sum_{v \in Val(A)} P(A=v) \cdot H(S|A=v), \qquad (287)$$

unde S_v este submulţimea lui S formată din acele instanţe pentru care $A = v$, iar $H(S|A = v)$ este entropia lui S_v (care este numită *entropie condiţională specifică*).

Considerăm acum A', un alt atribut din descrierea instanţelor de antrenament din setul S. Are loc următoarea proprietate:

***Proprietate*: Dacă există o corespondenţă injectivă $v \mapsto v'$,[593] unde $v \in Val(A)$ şi $v' \in Val(A')$, astfel încât**

$$\frac{|S_v|}{|S|} \cdot H(S|A=v) \leq \frac{|S_{v'}|}{|S|} \cdot H(S|A'=v') \text{ pentru orice } v \in Val(A), \qquad (288)$$

sau, scris echivalent

$$P(A=v) \cdot H(S|A=v) \leq P(A'=v') \cdot H(S|A'=v') \text{ pentru orice } v \in Val(A),$$

din relaţia (287) rezultă imediat că

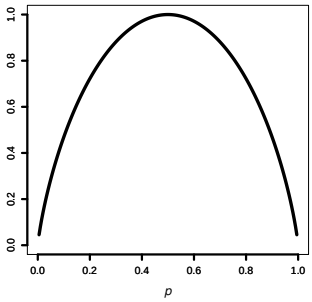
$$H(S|A) \leq H(S|A').$$

Evident, relaţia (288) este satisfăcută în următorul *caz particular*:

$$\frac{|S_v|}{|S|} \leq \frac{|S_{v'}|}{|S|} \text{ şi } H(S|A=v) \leq H(S|A'=v') \text{ pentru orice } v \in Val(A). \qquad (289)$$

Verificarea acestor relaţii (din varianta (289)) este mult mai uşor de realizat în mod vizual (decât varianta (288)), atunci când — aşa cum procedăm noi de obicei la aplicarea algoritmului ID3 — se lucrează cu compaşi de decizie pentru a determina „cel mai bun atribut“ de pus în nodul curent al arborelui de decizie aflat în curs de construire.

Atunci când se procedează aşa, dacă A şi A' sunt variabile Bernoulli (adică, fiecare dintre ele are câte două valori), pentru a verifica inegalităţile de tipul $H(S|A = v) \leq H(S|A' = v')$, se apelează de obicei la *monotonia* (şi *simetria*) funcţiei entropie pentru distribuţia Bernoulli. Graficul acestei funcţii este prezentat în figura alăturată.



[592] Vedeţi formulele date la problema 55 de la capitolul de *Fundamente*.

[593] Mai exact, ar fi trebuit să notăm corespondenţa injectivă respectivă cu $f : Val(A) \to Val(A')$ şi în loc de v' să scriem $f(v)$ peste tot în cele ce urmează. În text, vom păstra însă notaţia [mai] simplă: v şi corespondentul său v'. Evident, $S_{v'}$ este submulţimea lui S formată din acele instanţe pentru care $A' = v'$.

***Observaţii*:**
1. În ce priveşte corespondenţa injectivă a cărei existenţă am presupus-o / cerut-o mai sus, este suficient ca ea să fie definită pe acea submulţime a lui $Val(A)$ pentru care este îndeplinită condiţia

$$P(A = v) \neq 0 \text{ şi, mai ales(!), } H(S|A = v) \neq 0.$$

2. În diagramele din culegere (sau din slide-uri) unde utilizăm astfel de raţionamente relaţionale / „calitative", folosim următoarea *convenţie*: cu o linie continuă punem în evidenţă relaţii de tipul (289), iar printr-o linie punctată relaţii de tipul $H(S|A = v) \leq H(S|A' = v')$.
3. În exerciţii didactice — mai ales în condiţii de test sau examen, când timpul pus la dispoziţie este limitat! — este adeseori utilă combinarea raţionamentelor calitative cu cele cantitative,[594] întrucât adeseori inegalităţile din relaţiile (288) şi / sau (289) nu sunt satisfăcute pentru toate valorile $v \in Val(A)$.

37. **(Algoritmul ID3: aplicare)**

prelucrare de Liviu Ciortuz, după
· CMU, 2020 fall, Aarti Singh, Recitation

Fie următorul set de date, relativ la când anume un copil iese afară să se joace:

Day	Weather	Temperature	Humidity	Wind	Play
1	Sunny	Hot	High	Weak	No
2	Cloudy	Hot	High	Weak	Yes
3	Sunny	Mild	Normal	Strong	Yes
4	Cloudy	Mild	High	Strong	Yes
5	Rainy	Mild	High	Strong	No
6	Rainy	Cool	Normal	Strong	No
7	Rainy	Mild	High	Weak	Yes
8	Sunny	Hot	High	Strong	No
9	Cloudy	Hot	Normal	Weak	Yes
10	Rainy	Mild	High	Strong	No

În acest exerciţiu vă cerem să elaboraţi arborele învăţat de algoritmul ID3 pe acest set de date.

a. Stabiliţi în mod riguros ce atribut va pune algoritmul ID3 în nodul rădăcină.

Veţi folosi următorul tabel, în care se dau entropiile mai multor variabile Bernoulli, determinate de valoarea parametrului p:

p	1/4	1/3	2/5	3/7
$H(p)$	0.8112	0.9182	0.9709	0.9852

b. Satisfaceţi aceeaşi cerinţă, apelând acum la alte două metode: mai întâi raţionamentul relaţional („calitativ") care a fost prezentat la *Comentariul* de la pag. 593 şi apoi metoda expusă la problema 36 (vedeţi în special *Observaţia* (2) de acolo).

[594] Vedeţi şi metoda expusă la problema 2.d.

c. Stabiliţi ce atribute pot fi puse în nodurile de test de pe nivelul 2 (adică, în descendenţii nodului rădăcină), iar apoi completaţi arborele ID3.[595] **Este oare arborele [elaborat de algoritmul] ID3 unic? Care este eroarea pe setul de date de antrenare?**

38. **(Calcularea unor entropii; aplicarea algoritmului ID3)**

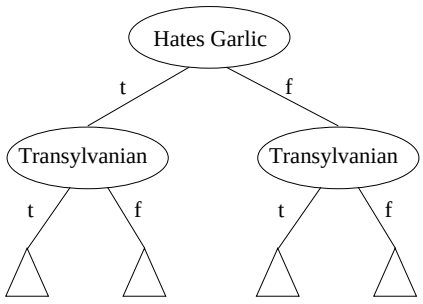
prelucrare de Liviu Ciortuz, după
• ***CMU, 2014 spring, Seyoung Kim, HW2, pr. 1.4.1-5***

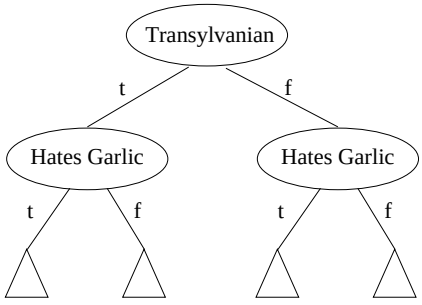
Centrul [de Medicină] pentru Controlul şi Prevenţia Maladiilor a fost sesizat în legătură cu o creştere surprinzătoare a apariţiilor de vampiri. Acest centru a colectat date preliminare referitoare la anumite *caracteristici*, atât pentru vampiri [deja] cunoscuţi cât şi pentru non-vampiri, şi acum ar dori să construiască un arbore de decizie ca să-i ajute pe cetăţeni să identifice noi vampiri. Datele culese sunt prezentate în tabelul următor:

V (Vampire)	G (Hates Garlic)	T (Transylvanian)	Nr. apariţii
+	t	t	9
+	t	f	4
+	f	t	3
+	f	f	0
−	t	t	1
−	t	f	3
−	f	t	1
−	f	f	6

Fiecare linie indică ce caracteristici au fost „observate“, şi de câte ori a fost observată fiecare combinaţie de caracteristici. De *exemplu*, combinaţia (+, t, t) a fost observată de 9 ori, pe când combinaţia (+, f, f) n-a fost observată niciodată. (Simbolii $'\mathrm{t}'$ şi $'\mathrm{f}'$ au fost folosiţi în locul lui True şi False, pentru a evita confuzia cu atributul T.)

a. Completaţi arborii de mai jos cu informaţii (count-uri) referitoare la partiţionarea datelor (sub forma $[+n, -m]$) în fiecare nod, precum şi cu deciziile care trebuie luate (în sens majoritar) în nodurile frunză.





[595] *Atenţie*! Puteţi elabora raţionamente perfecte fără să faceţi calcule laborioase!
Explorând cu atenţie datele de antrenament, veţi vedea că puteţi să identificaţi atribute cu putere de predicţie / „discriminare“ maximă [a valorilor variabilei de ieşire]. Cât este entropia condiţională medie a acestor atribute [în raport cu variabila de ieşire]?

b. Calculaţi entropia condiţională medie $H(V|G)$. (Toate calculele intermediare trebuie făcute cu o precizie de cel puţin 4 zecimale, pentru a ne asigura că răspunsul final are primele 3 zecimale corecte.)

c. Calculaţi entropia condiţională medie $H(V|T)$. (Din nou, toate calculele intermediare trebuie făcute cu o precizie de cel puţin 4 zecimale.)

d. Care dintre cei doi arbori de decizie de mai sus reprezintă rezultatul învăţării realizate de algoritmul ID3 pe aceste date?

e. Adevărat sau Fals: Arborele produs de către ID3 va clasifica o persoană căreia-i displace usturoiul (engl., hates garlic) dar nu este transilvănean ca fiind vampir.

***Indicaţie*: Este posibil să aveţi nevoie de următoarele valori pentru entropia ($H(p)$) unei variabile aleatoare Bernoulli de parametru p: $H(1/7) = 0.5916$, $H(4/17) = 0.7871$, $H(3/10) = 0.8812$, $H(4/13) = 0.8904$, $H(11/27) = 0.9751$.**

39. **(Algoritmul ID3: aplicare pe expresii booleene; exploatarea simetriilor operaţiilor $\vee, \wedge$ în alegerea atributelor; analiza „optimalităţii“ arborelui ID3)**

∗ ***prelucrare de Liviu Ciortuz, după Tom Mitchell, "Machine Learning", 1997, ex. 3.1.d***

Considerăm următoarea funcţie booleană: $(A \wedge B) \vee (C \wedge D)$. Valorile pe care le ia această funcţie, calculate conform diferitelor valori de adevăr atribuite variabilelor / atributelor A, B, C şi D sunt cele cunoscute din logica propoziţiilor. Dorim însă să reprezentăm această funcţie ca arbore de decizie.

a. Aplicaţi algoritmul ID3 acestei funcţii.
***Observaţie*: Dacă exploataţi simetriile, este nevoie doar de puţine calcule, altfel vă complicaţi în mod inutil.**

b. Arborele ID3 obţinut la punctul precedent este optimal?
Altfel spus, puteţi găsi alt arbore de decizie de adâncime mai mică sau cu număr mai mic de noduri (de test) pentru această funcţie? (Ţineţi cont că în fiecare nod al unui arbore de decizie se poate testa un sigur atribut.)

40. **(Algoritmul ID3: aplicare; analiza „optimalităţii“ arborelui ID3)**

• ∗ ***CMU, 2005 spring, C. Guestrin, T. Mitchell, midterm, pr. 4***

Agenţia spaţială NASA doreşte să distingă între marţieni (M) şi pământeni (H) folosind următoarele caracteristici: $Green \in \{N, Y\}$, $Legs \in \{2, 3\}$, $Height \in \{S, T\}$, $Smelly \in \{N, Y\}$.

Datele de antrenament de care dispunem sunt prezentate în tabelul alăturat.

a. Învăţaţi un arbore de decizie folosind algoritmul ID3 şi trasaţi arborele respectiv.

	Species	Green	Legs	Height	Smelly
1	M	N	3	S	Y
2	M	Y	2	T	N
3	M	Y	3	T	N
4	M	N	2	S	Y
5	M	Y	3	T	N
6	H	N	2	T	Y
7	H	N	2	S	N
8	H	N	2	T	N
9	H	Y	2	S	N
10	H	N	2	T	Y

b. Descrieţi conceptul M (marţian) ca un set de reguli conjunctive din logica propoziţiilor. Spre exemplu:

if $\textit{Green} = Y$ **and** $\textit{Legs} = 2$ **and** $\textit{Height} = T$ **and** $\textit{Smelly} = N$ **then M;**
else
if ... **then M; else H.**

c. Soluţia de la punctul b **de mai sus foloseşte cel mult 4 atribute în fiecare conjuncţie. Găsiţi un set de reguli conjunctive care folosesc doar 2 atribute pentru fiecare conjuncţie, păstrând însă eroarea la antrenare zero. Această ipoteză mai simplă poate fi reprezentată ca un arbore de decizie de adâncime 2? Justificaţi răspunsul.**

41. **(Algoritmul ID3: aplicare; cazul instanţelor de antrenament cu multiple apariţii)**

• · ***CMU, 2010 fall, Aarti Singh, HW2, pr. 5.1***

Tabelul de mai jos descrie instanţe (înregistrări) pozitive şi instanţe negative pentru persoane cărora banca le-a acordat (sau nu le-a acordat) un card de credit.

Fiecare linie din tabel indică nişte combinaţii de valori observate pentru atributele considerate (*Gender*, *Income* şi *Approved*) şi de câte ori a fost înregistrată respectiva combinaţie de valori. De exemplu, (F, Low, +) a apărut de 10 ori, iar (F, Low, −) de 80 de ori.

Gender	*Income*	*Approved*	*Counts*
F	Low	+	10
F	High	+	95
M	Low	+	5
M	High	+	90
F	Low	−	80
F	High	−	20
M	Low	−	120
M	High	−	30

a. Calculaţi entropia atributului *Approved* pe acest set de date de antrenament (folosind logaritmul cu baza 2).

b. Calculaţi de asemenea câştigurile de informaţie IG(*Approved, Gender*) şi IG(*Approved, Income*).

c. Desenaţi un arbore de decizie produs de către algoritmul ID3 (fără post-pruning) pe baza acestui set de date de antrenament.

42. **(“Decision stump” produs de ID3: raţionament calitativ pe un exemplu simplu)**

• ○ ***CMU, 2010 fall, Ziv Bar-Joseph, midterm exam, pr. 5.a***

Vrem să construim un arbore de decizie care să ne ajute să prezicem întârzierile avioanelor. Timp de câteva luni am colectat informaţii, iar un rezumat al acestora este prezentat în tabelul următor:

Atribut	Valoare = *Da*		Valoare = *Nu*	
	#Zboruri		#Zboruri	
	amânate	neamânate	amânate	neamânate
Ploaie	30	10	10	30
Vânt	25	15	15	25
Vara	5	35	35	5
Iarna	20	10	20	30
Ziua	20	20	20	20
Noaptea	15	10	25	30

a. Pe baza acestui tabel precizaţi ce atribut ar trebui să fie pus în rădăcina arborelui de decizie, folosind criteriul câştigului de informaţie. Justificaţi riguros; nu este însă necesar să elaboraţi în detaliu toate calculele.

b. Pe baza aceluiaşi tabel, precizaţi care dintre atribute ar trebui să apară pe al doilea nivel (nivelul de sub rădăcină) al arborelui de decizie.

43. **(O variantă ipotetică a algoritmului ID3)**

enunţ formulat de Liviu Ciortuz, pornind de la un slide al lui A. Moore

(vedeţi şi CMU, 2011 fall, E. Xing, HW1, pr. 1.2.4)

La curs am precizat că algoritmul ID3 în varianta standard (cea fără atribute numerice continue) respectă următoarea *restricţie*: niciun atribut de intrare (notat A) nu poate să apară de două sau mai multe ori pe vreun drum care uneşte nodul rădăcină cu un nod frunză oarecare (dar fixat).

La acest exerciţiu ne propunem să examinăm ce se întâmplă dacă eliminăm această restricţie.

a. Cum anume va fi partiţionată mulţimea de instanţe asignată nodului în care atributul A apare pentru a doua (sau a n-a oară)?

b. Cât va fi câştigul de informaţie calculat pentru un astfel de nod?

c. Ţinând cont de răspunsurile date la punctele precedente, se justifică *restricţia* formulată mai sus?

44. **(O aproximare a numărului de instanţe greşit clasificate care au fost asignate la un nod frunză dintr-un arbore ID3)**

• ○ ***CMU, 2003 fall, T. Mitchell, A. Moore, midterm exam, pr. 9.b***

Învăţăm un arbore de decizie folosind un set de date de antrenament cu atributul de ieşire ($class$) având valorile 0 sau 1.

Presupunând că pentru un nod frunză l din acest arbore,

– există M instanţe de antrenament asignate la acel nod, iar
– entropia sa este H,

schiţaţi un algoritm simplu care ia ca valori de intrare M şi H şi furnizează la ieşire numărul de exemple de antrenament clasificate greşit de către nodul frunză l.

***Sugestie*: Folosiţi o aproximare simplă (polinomială) pentru funcţia entropie $H(p)$.**

45. **(Algoritmul ID3: eroarea la antrenare)**

• ○ ***CMU, 2003 fall, T. Mitchell, A. Moore, HW1, pr. 2.1***

Un student mi-a spus următoarele:

– el poate să construiască un set de instanţe cu atributele de intrare discrete şi atributul de ieşire binar;
– mie îmi dă voie să aleg o parte din acest set de instanţe (dar nu toate!) pentru a antrena un arbore de decizie;
– indiferent de modul cum mi-aş alege datele de antrenament din setul construit de el, eroarea de clasificare pe care arborele de decizie (obţinut în urma antrenării cu algoritmul ID3) o va face pe instanţele care nu au fost incluse în setul de antrenament va fi de cel puţin 50%.

Credeţi că studentul are dreptate? Explicaţi de ce sau daţi un exemplu.

46. **(Algoritmul ID3: aplicare; calculul erorii la antrenare, respectiv la validare)**

∗ ***CMU, 2003 fall, T. Mitchell, A. Moore, midterm exam, pr. 1***

Folosiţi setul de date alăturat pentru a învăţa cu ajutorul unui arbore de decizie dacă o anumită floare este *Iris* (acesta este numele latinesc pentru *stânjenel*) sau nu, utilizând atributele discrete *Formă*, *Culoare* şi *Miros*.

Formă	*Culoare*	*Miros*	*Iris*
C	B	1	1
D	B	1	1
D	W	1	1
D	W	2	1
C	B	2	1
D	B	2	0
D	G	2	0
C	U	2	0
C	B	3	0
C	W	3	0
D	W	3	0

a. Ce atribut va alege algoritmul ID3 ca rădăcină a arborelui de decizie?

b. Elaboraţi întregul arbore de decizie care va fi învăţat din datele de mai sus (fără pruning).

c. Exprimaţi cu ajutorul unui set de reguli din calculul propoziţional clasificarea produsă de arborele de decizie obţinut. (IF ... THEN *Iris*; IF ... THEN $\neg$*Iris*.)

d. Să presupunem că avem un set de date de validare:

Formă	*Culoare*	*Miros*	*Iris*
C	B	2	0
D	B	2	0
C	W	2	1

Care va fi eroarea produsă de arborele de decizie pe mulţimea de date de antrenare respectiv pe datele de validare? (Exprimaţi răspunsul ca număr de exemple clasificate greşit.)

47. **(Extensii ale algoritmului ID3: variabile de intrare continue; determinarea celui mai bun prag de separare: o proprietate)**

□ • ***USC, 2008 fall, Sofus Macskassy, HW2, pr. 3***

Atunci când construim arbori de decizie, selecţia atributelor se face folosind de obicei criteriul câştigul de informaţie maxim.

Arătaţi că în cazul variabilelor de intrare continue, pragurile de separare (engl., split-point) pentru care de o parte şi de alta etichetele sunt de acelaşi tip nu vor conduce niciodată la câştig de informaţie maxim.

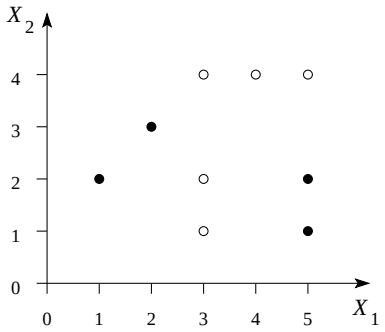
***Sugestie*: Vă recomandăm să citiţi [secţiunea 4 din] lucrarea *Technical note: On the handling of continuous-valued attributes in decision tree generation*, Usama M. Fayyad, Keki B. Irani, *Journal of Machine Learning*, nr. 8, 1992, pages 87-102.**

48. **(ID3 cu atribute continue: zone de decizie şi separatori decizionali)**

■ □ • ○ ***Liviu Ciortuz, 2017, folosind datele de la problema 24***

Fie setul de date de antrenament din figura de mai jos (partea dreaptă. X_1 şi X_2 sunt considerate atribute numerice continue. Vă readucem aminte *convenţia* noastră de notare: simbolul • desemnează instanţe pozitive, iar simbolul ○ instanţe negative.

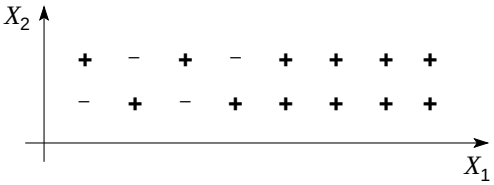
Aplicaţi algoritmul ID3 pe acest set de date. (Faceţi toate calculele necesare, în mod detaliat; precizaţi la fiecare pas care sunt pragurile de split-are pentru cele două atribute.) Desenaţi arborele de decizie rezultat. La final, reprezentaţi grafic zonele de decizie şi separatorii decizionali, marcând clar zona pozitivă (sau zonele pozitive) şi zona negativă (sau zonele negative).



49. **(Extinderea algoritmului ID3 cu atribute continue; eroarea la antrenare, eroarea la CVLOO; overfitting)**

∗ ***CMU, 2005 fall, T. Mitchell, A. Moore, midterm exam, pr. 2***

Figura alăturată prezintă un set de date cu două intrări X_1 şi X_2, variabile cu valori reale, şi o ieşire Y care poate lua valori pozitive $(+)$ sau negative $(-)$.



Testăm doi algoritmi extremi de învăţare de arbori de decizie. Algoritmul *OVERFIT* construieşte un arbore de decizie în maniera standard a algoritmului ID3, fără a face pruning. Algoritmul *UNDERFIT* refuză complet să-şi asume riscul splitării intervalelor de valori pentru X_1 şi X_2 şi construieşte un arbore de decizie alcătuit doar dintr-un singur nod (care va fi simultan şi rădăcină şi nod frunză, deci şi nod de decizie).

a. Câte noduri frunză vor fi în arborele de decizie învăţat de *OVERFIT* pe aceste date?

b. Care va fi eroarea de clasificare la cross-validare cu metoda "Leave-One-Out" pe acest set de date atunci când folosim algoritmul *OVERFIT*? (Indicaţi datele incorect clasificate.)

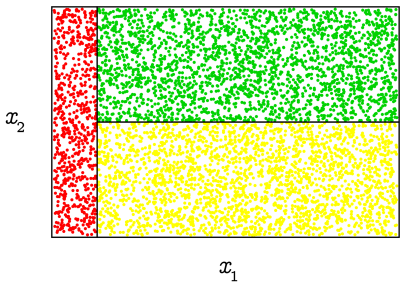
c. Similar punctului anterior, pentru cazul în care se foloseşte algoritmul *UNDERFIT*.

50. **(Arbori de decizie cu variabile continue: ID3 ca algoritm "greedy")**

• ○ ***CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, HW1, pr. 8.abcf***

Fie următoarea problemă de clasificare ternară.

Considerăm că în figura alăturată regiunea dreptunghiulară este populată în mod dens cu puncte caracterizate de două atribute numerice (continue), x_1 şi x_2. Cele trei subdreptunghiuri (roşu, verde şi galben) reprezintă trei clase de puncte, $C_1, C_2,$ şi C_3.



Dimensiunile $x_1 \times x_2$ ale dreptunghiurilor roşu, verde şi galben sunt 1×6, 7×3 şi respectiv 7×3. Dreptunghiul roşu este populat în mod uniform cu 6000 de puncte din clasa C_1. Dreptunghiul verde este populat în mod uniform cu 42000 de puncte din clasa C_2. Dreptunghiul galben este populat în mod uniform cu 42000 de puncte din clasa C_3. Pentru simplitate, nu vom considera alte puncte decât acestea.

a. Care este numărul minim de noduri de test pe care trebuie să le aibă un arbore de decizie pentru a clasifica în mod corect acest set de date?

b. Câte noduri de test are arborele de decizie obţinut în urma antrenării algoritmului ID3 pe acest set de date, folosind criteriul maximizării câştigului de informaţie?

***Indicaţie*: Pentru a determina entropiile condiţionale minime, puteţi folosi proprietatea A3 de la problema 62 de la capitolul de *Fundamente*.**

c. Avem acelaşi număr de noduri în cele două cazuri de mai sus, sau nu? Care credeţi că este explicaţia?

d. Un arbore de decizie poate să clasifice setul de date din figura de mai sus cu 100% acurateţe (presupunând că nu există zgomote la nivel de etichete). Ce condiţii trebuie să satisfacă în general un set de date de acest gen astfel încât arborele de decizie rezultat în urma antrenării să fie cât mai compact şi să producă o acurateţe de 100%?

***Indicaţie*: Fiecare nod intern al arborelui de decizie corespunde unui test bazat pe o singură trăsătură. Gândiţi-vă ce fel de clase de funcţii / graniţe de separare corespund unui astfel de arbore de decizie.**

51. **(Un exemplu de aplicare a algoritmului ID3: cazul când se folosesc atât variabile discrete cât şi variabile continue)**

□ • ∗ ***CMU, 2010 fall, Ziv Bar-Joseph, HW2, pr. 1***

Cursul de Învăţare Automată pe care l-ai urmat în acest semestru, ţi-a insuflat dorinţa ca după absolvirea facultăţii să fondezi pe cont propriu o firmă (engl., start-up company). Pentru a-ţi evalua şansele de succes — adică, mai exact, dacă vei deveni milionar sau nu —, ai colectat date de la absolvenţii de la CMU, referitoare la foşti studenţi care şi-au înfiinţat propriile lor companii start-up. Pentru fiecare start-up, ştii acum ce anume produce compania respectivă, ce fonduri de capital de investiţie a obţinut (exprimat în milioane de dolari), dacă directorul companiei a studiat la CMU o disciplină din domeniul ştiinţelor exacte şi, în final, dacă directorul a devenit milionar. Tabelul de mai jos centralizează datele pe care le-ai cules.

Produs	*Capital atras*	*Ştiinţe exacte*	*Milionar*
SiteDeSocializare	2.9	Da	Nu
SiteDeSocializare	1.7	Da	Nu
SiteDeSocializare	3.4	Da	Nu
SiteDeSocializare	2.3	Nu	Da
MaşinaAlimentatăCuCombustibilBio	3.4	Da	Da
MaşinaAlimentatăCuCombustibilBio	6.1	Da	Da
MaşinaAlimentatăCuCombustibilBio	5.6	Nu	Nu
MaşinaAlimentatăCuCombustibilBio	0.6	Nu	Nu
NanoVaccin	1.9	Nu	Nu
NanoVaccin	2.9	Nu	Da
NanoVaccin	3.1	Da	Da
NanoVaccin	0.3	Da	Nu

Acum ai vrea să construieşti un arbore de decizie pornind de la aceste date. Referitor la atributul cu valori continue *CapitalAtras*, ştim că arborele de decizie poate conţine teste (partiţionări binare) de forma *CapitalAtras* $\leq v$ şi *CapitalAtras* $> v$ şi că pot exista mai multe teste de acest fel în arbore.

a. Câte „praguri" distincte v trebuie să considerăm pentru *CapitalAtras* atunci când căutăm atributul (optim) care trebuie pus în nodul rădăcină?

b. Desenaţi arborele de decizie care va fi învăţat de către algoritmul ID3 extins cu atribute cu valori continue, aşa cum a fost prezentat la curs. Ne vom referi ulterior la acest arbore ca fiind *arborele original*. Adnotaţi fiecare nod intern (i.e., nod de test) din arbore cu câştigul de informaţie obţinut în urma aplicării testului respectiv.

***Indicaţie*: Pentru punctele c şi d de mai jos, deşi nu este neapărat necesar, este recomandabil să folosiţi o implementare a algoritmului ID3 (extins cu atribute numerice continue). Vă sugerăm să abordaţi mai întâi problemele 35 şi 58, care vă ghidează cum să construiţi propria dumneavoastră implementare.**

c. Schimbaţi eticheta (i.e., valoarea clasei binare *Milionar*) pentru o instanţă din tabelul de mai sus astfel încât arborele care va fi învăţat ulterior să conţină cel puţin încă un nod de test în raport cu arborele de la punctul b.

d. Un exemplu de antrenament este consistent cu arborele învăţat dacă este clasificat corect de către acel arbore. Întrebarea pe care v-o adresăm acum este următoarea:

Este posibil să adăugăm la setul de date de antrenament de mai sus noi exemple care sunt consistente cu arborele original, dar care fac totuşi ca algoritmul ID3 extins, executat pe noul set de date de antrenament, să înveţe un arbore cu o rădăcină diferită de cea originală şi cu mai multe noduri decât arborele original?

Dacă răspunsul dumneavoastră este afirmativ, indicaţi noile exemple pe care aţi putea să le adăugaţi, precum şi arborele de decizie rezultat. Dacă răspunsul este negativ, explicaţi de ce este aşa.

52. **(Extensii ale algoritmului ID3: cazul atributelor care au multe valori [algoritmul C4.5])**

• ∗ ***prelucrare de Liviu Ciortuz, 2020, după CMU, 2015 fall, Z. Bar-Joseph, E. Xing, HW1, pr. 4.2***

Una dintre limitările algoritmului ID3 [LC: ne referim la varianta de bază, fără extensiile pe care le-am discutat ulterior la curs] este faptul că el este prea sensibil la prezenţa atributelor care au un număr mare de valori. De exemplu, dacă fiecare instanţă de antrenament are un ID unic, atunci câştigul de informaţie va fi maxim atunci când folosim acest ID ca atribut de intrare, ceea ce nu este deloc bine pentru faza de generalizare / predicţie, în care putem întâlni adeseori instanţe având ID-uri care nu se regăsesc în datele de antrenament. Algoritmul C4.5 remediază acest aspect din funcţionarea lui ID3 folosind în locul câştigului de informaţie, pentru a evalua atributele, *raportul câştigului de informaţie* (engl., information gain ratio).

Vom nota un atribut de intrare oarecare cu X, iar eticheta sa cu Y. Vă reamintim că în algoritmul ID3, alegem pentru nodul curent acel atribut X care maximizează câştigul de informaţie, $IG(X)$:

$$IG(X) = H(Y) - H(Y|X).$$

Acum vom defini noţiunea de [LC: cantitate de] *informaţie la separare* (engl., split information) după cum urmează. Presupunem că avem $|D|$ instanţe ataşate la nodul curent, iar după ce se face testul pe baza valorii atributului X, aceste instanţe sunt repartizate la V noduri descendente din nodul curent. Presupunând că numărul de instanţe care sunt asociate la aceste noduri-fii sunt respectiv $|D_1|$, $|D_2|$, $\cdots$, $|D_V|$, vom defini *informaţia la separarea* valorilor atributului X astfel:[596]

$$\mathit{SplitInfo}\,(X) = -\sum_{j=1}^{V} \frac{|D_j|}{|D|} \log_2 \frac{|D_j|}{|D|}.$$

Ca şi în cazul entropiei, se va considera, prin convenţie, că $0 \cdot \log_2 0 = 0$.

***Raportul câştigului de informaţie* se defineşte în felul următor:**[597]

$$\mathit{GainRatio}\,(X) = \frac{IG(X)}{\mathit{SplitInfo}\,(X)}.$$

Algoritmul C4.5 (succesorul lui ID3) foloseşte *raportul câştigului de informaţie* pentru a determina „cel mai bun“ atribut de pus în nodul curent. Intuiţia spune că $\mathit{SplitInfo}\,(X)$ actionează ca un „normalizator“ (engl., normalizer), care penalizează atributele care au un număr mare de valori. De exemplu, dacă pentru $\forall i, j$ avem $|D_i| = |D_j|$, atunci $\mathit{SplitInfo}\,(X) = \log_2 |V|$ (adică, maximum posibil), aşadar atributele care au mulţimea de valori (V) mai restrânsă vor fi preferate.

În acest exerciţiu veţi elabora arborele de decizie corespunzător setului de date din tabelul de mai jos, folosind pe de o parte algoritmul ID3 şi pe de altă parte algoritmul C4.5. Atributele de intrare sunt *Outlook*, *Temperature*, *Humidity* şi *Wind*, iar eticheta este asociată cu variabila de ieşire *EnjoyTennis*. Veţi trata *Temperature* ca atribut discret.

[596] Se poate observa din formula de definiţie că $\mathit{SplitInfo}\,(X)$ este *entropia* atributului X calculată pentru cele D instanţe asociate la nodul curent. (Ea *nu* este însă nicidecum acelaşi lucru cu *entropia condiţională medie* a atributului X în raport cu variabila de ieşire Y, pe care o folosim la calcularea câştigului de informaţie.)

[597] Cazul când $\mathit{SplitInfo}\,(X) = 0$ corespunde situaţiei când X are o singură valoare (pentru instanţele din nodul în care se calculează acest *SplitInfo*), deci nu are putere discriminativă în raport cu variabila de ieşire. În consecinţă, astfel de atribute nu vor fi luate în considerare atunci când, pentru nodul curent, se va pune problema să alegem atributul cu *GainRatio* maxim.

Day	Outlook	Temperature	Humidity	Wind	EnjoyTennis
D1	**Sunny**	**26**	**High**	**Weak**	**No**
D2	**Sunny**	**25**	**High**	**Strong**	**No**
D3	**Overcast**	**25**	**High**	**Weak**	**Yes**
D4	**Rain**	**24**	**High**	**Weak**	**Yes**
D5	**Rain**	**19**	**Normal**	**Weak**	**Yes**
D6	**Rain**	**20**	**Normal**	**Strong**	**No**
D7	**Overcast**	**20**	**Normal**	**Strong**	**Yes**
D8	**Sunny**	**23**	**High**	**Weak**	**No**
D9	**Sunny**	**20**	**Normal**	**Weak**	**Yes**
D10	**Rain**	**25**	**Normal**	**Weak**	**Yes**
D11	**Sunny**	**24**	**Normal**	**Strong**	**Yes**
D12	**Overcast**	**22**	**High**	**Strong**	**Yes**
D13	**Overcast**	**23**	**Normal**	**Weak**	**Yes**
D14	**Rain**	**23**	**High**	**Strong**	**No**

a. Pentru fiecare dintre atributele *Outlook*, *Humidity*, *Wind* şi *Temperature*, faceţi reprezentări grafice pentru compaşii de decize corespunzători, astfel:

– asociaţi mai întâi la nodului rădăcină al compasului de decizie *partiţia* de instanţe corespunzătoare;

– procedaţi similar pentru fiecare dintre descendenţii lui direcţi (adică nodurile-fii);

– calculaţi entropia condiţională medie a respectivului atribut şi, în fine, câştigul de informaţie în raport cu atributul de ieşire *EnjoyTennis*.

***Indicaţie*: Următoarele valori pentru entropia distribuţiei Bernoulli vă pot fi de folos:**

p	0	1/3	2/5	3/7	1/2	1
$H(p)$	0	0.918	0.970	0.985	1	0

b. Ce atribut va selecta algoritmul ID3 pentru nodul rădăcină al arborelui pe care-l „învaţă“?

c. Calculaţi valorile $SplitInfo(\cdot)$ pentru toate atributele de intrare, relativ la întregul set de date de antrenament.

***Atenţie*! $SplitInfo(Humidity)$ şi $SplitInfo(Wind)$ se calculează foarte uşor folosind informaţiile din tabelul dat la *Indicaţia* de mai sus.**
Pentru $SplitInfo(Outlook)$ şi $SplitInfo(Temperature)$, vă cerem elaboraţi calculul cât mai complet posibil. (La calcule, puteţi folosi aproximaţiile $\log_2 3 = 1.584$, $\log_2 5 = 2.322$ şi $\log_2 7 = 2.807$.)
Presupunând că $SplitInfo(Outlook) = 1.577$, şi $SplitInfo(Temperature) = 2.646$, ce atribut va selecta algoritmul C4.5 pentru nodul rădăcină al arborelui pe care-l „învaţă“?

d. Elaboraţi complet arborele de decizie produs de algoritmul ID3. (*Atenţie*! Dacă aţi procedat corect la punctele a şi b, atunci aici nu veţi avea de făcut calcule aproape deloc!)

e. Elaboraţi complet arborele de decizie produs de algoritmul C4.5. (*Atenţie*! Dacă aţi procedat corect la punctul c, atunci aici nu veţi avea de făcut calcule

aproape deloc!)
Ce observaţi comparând arborele obţinut aici cu arborele care a fost obţinut la punctul d?

53. **(Deficienţe ale pruning-ului top-down şi respectiv bottom-up)**

□ • ***CMU, 2009 fall, Carlos Guestrin, HW1, pr. 2.4***

a. Ştim că algoritmul ID3 alege în fiecare nod de test atributul care are cel mai mare câştig de informaţie dintre toate atributele de intrare rămase disponibile pentru acel nod. O posibilă metodă de *pruning* în manieră *top-down* este următoarea: dacă toate atributele despre care am vorbit au câştig de informaţie 0 atunci oprim aplicarea agloritmului ID3 (adică elaborarea arborelui de decizie) pe ramura respectivă. Această strategie este însă problematică, întrucât ea poate conduce la arbori de decizie inconsistenţi cu datele, chiar şi atunci când datele sunt consistente / necontradictorii. Vă cerem să furnizaţi un mic set de exemple de antrenament care să demonstreze acest fapt. *Sugestie*: Este suficient să folosiţi două atribute de intrare booleene.

b. La problema 19.b, am discutat [şi] despre *pruning bottom-up* ca modalitate de control asupra „complexităţii“ arborelui creat de algoritmul ID3. Totuşi, şi această metodă are anumite neajunsuri.

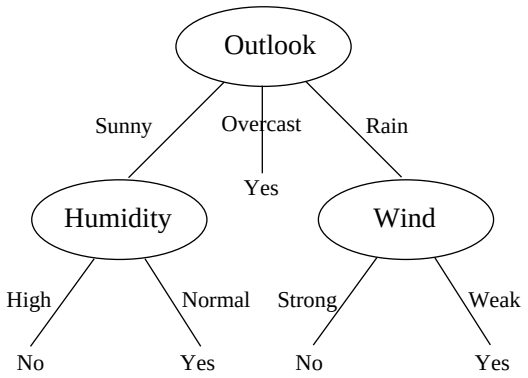
Pornind de la soluţia pe care aţi dat-o la punctul precedent, clonaţi fiecare exemplu în parte de K ori (unde K este un număr suficient de mare). Apoi adăugaţi N noi atribute de intrare care iau valorile True / False cu o probabilitate uniformă. Cum va arăta arborele de decizie care rezultă? Explicaţi acum de ce poate să eşueze pruning-ul de tip bottom-up.

54. **(Algoritmul ID3: întrebări recapitulative; compararea strategiilor de pruning: reduced-error pruning vs. rule post-pruning)**

• ***CMU, 2005 fall, T. Mitchell, A. Moore, HW1, pr. 1***

Considerăm următorul set de date de antrenament (care diferă uşor de cel de la problema 52; vedeţi pag. 605), împreună cu arborele de decizie care a fost învăţat pe aceste date de către algoritmul ID3 (fără a se aplica vreo strategie de post-pruning).

Day	Outlook	Temperature	Humidity	Wind	EnjoyTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No



a. Demonstraţi că alegerea atributului *Wind* pe nivelul secund din arbore este corectă, arătând că în cazul lui câştigul de informaţie este superior faţă de cazurile / posibilităţile alternative.

b. Orice arbore de decizie poate fi exprimat sub forma unui set de reguli, scriind câte o regulă pentru fiecare nod-frunză (de fapt, pentru întreaga cale începând cu nodul-rădăcină şi continuând până la nodul-frunză respectiv). *Pre-condiţiile* dintr-o astfel de regulă corespund secvenţei / succesiunii de atribute testate de-a lungul căii respective. De *exemplu*, nodul-frunză cel mai din stânga din arborele de mai sus corespunde regulii următoare:

```
IF (Outlook = sunny AND Humidity=High) THEN PlayTennis = No
```

Scrieţi regulile corespunzătoare celorlalte noduri-frunză. Observaţi faptul că acest set de reguli produce clasificări care sunt identice cu cele din arborele de decizie de mai sus, pentru orice instanţă posibilă [LC: de antrenament sau de test].

c. Este posibil să „traducem“ orice arbore de decizie într-un set de reguli care reprezintă un clasificator echivalent. Invers, este oare posibil să „traducem“ orice set de reguli [evident, având forma indicată mai sus] într-un arbore echivalent? Explicaţi succint sau, dacă este cazul, daţi un contraexemplu.

d. La curs am discutat despre strategii de post-pruning pentru arbori de decizie, folosind abordarea *reduced-error pruning* relativ la un set de date de validare, pentru a evita overfitting-ul. Aici vom considera o strategie alternativă, care constă în a converti / „traduce“ arborele în setul de reguli echivalent şi făcând apoi pruning pe reguli. În particular, vom considera că pruning-ul se execută în mod independent pe fiecare regulă în parte, parcurgând iterativ următorii paşi:

i. **Determină în regula respectivă acele pre-condiţii care, dacă sunt înlăturate (în mod individual) produc o îmbunătăţire a acurateţii pe setul de date de validare.**[598]

ii. **Dacă există astfel de pre-condiţii,**
identific-o pe cea care îmbunătăţeşte cel mai mult acurateţea pe setul de date de validare,
***şterge*-o din regulă şi apoi *itereză* mai departe;**
în caz contrar *opreşte* pruning-ul pe această regulă.

Consideraţi cele două strategii de pruning (adică pruning pe arbore, versus pruning pe setul de reguli echivalent). Este oare adevărat faptul că aceste două strategii de pruning produc clasificatori trunchiaţi (engl., pruned) care sunt echivalenţi? Adică, produc oare arborele de decizie trunchiat şi setul de reguli trunchiate clasificări identice (engl., equivalent) pentru orice instanţă posibilă? Explicaţi de ce da, sau de ce nu.

55. **(Post-pruning pentru arborele ID3: folosirea testului statistic χ^2 pentru identificarea partiţionărilor care sunt semnificative d.p.v. statistic)**

□ • ***CMU, 2007 fall, Carlos Guestrin, HW1, pr. 1.3.1***

La curs am menţionat că arborii de decizie pot să manifeste fenomenul de “overfitting” şi, în consecinţă, pentru a generaliza cât mai corect, trebuie să limităm „complexitatea“ arborilor pe care îi învăţăm.

O modalitate de a face pruning pe arbori de decizie — de obicei, prin parcurgere *de jos în sus*, după antrenare — este să analizăm din punct de vedere statistic instanţele asignate fiecărui nod de test din arbore. Mai precis, vom analiza etichetele acestor instanţe, cu *obiectivul* de a vedea cât de probabil este d.p.v. statistic să „observăm“ proporţia acestor etichete în cazul în care atributul de intrare ales în nodul respectiv nu se află de fapt în corelaţie cu valorile atributului-ţintă (etichetele).

În acest sens putem folosi *testul* χ^2 al lui Pearson, care este un exemplu clasic de test bazat pe o *ipoteză statistică*. Vom pleca de la ipoteza că atributul de intrare asociat nodului respectiv nu este corelat cu atributul de ieşire (reprezentat de etichete), şi vom verifica dacă datele „observate“ conduc în mod ferm la respingerea acestei ipoteze.

Mai exact, pornind de la datele de antrenament, vom calcula o anumită *mărime* statistică, despre care ştim că are o distribuţie de tipul χ^2 în cazul în care cele două entităţi (atributul şi etichetele) nu sunt corelate. Apoi, vom

[598] Acurateţea unei reguli este [o fracţie, dată de] numărul de *predicţii corecte* pe care le face această regulă, raportat la numărul total de *predicţii*.

verifica dacă acea *mărime* statistică calculată are (sau nu) o *valoare* prea puţin probabil să fi fost generată de o distribuţie χ^2. În cazul afirmativ, vom respinge ipoteza că cele două entităţi nu sunt corelate. În cazul negativ, vom înlocui nodul de test respectiv, împreună cu întregul subarbore ataşat lui, cu un nod de decizie.

Concret, să presupunem că am învăţat un arbore de decizie. Notăm cu S setul de exemple de antrenament ale căror drumuri de clasificare trec prin nodul a cărui semnificaţie statistică dorim să o testăm. Să zicem că în nodul respectiv se testează un atribut discret X care poate lua valorile $1, 2, \ldots, k$. Fie p numărul de exemple din S care au eticheta $+$, iar $n = |S| - p$ numărul de exemple din S având eticheta $-$. Fie

S_i submulţimea lui S formată din instanţele care au $X = i$,

p_i numărul de exemple din S_i având eticheta $+$,

$n_i = |Si| - p_i$ numărul de exemple din S_i având eticheta $-$.

În plus, fie $\bar{p}_i = p \cdot \frac{|S_i|}{|S|}$ şi $\bar{n}_i = |S_i| - \bar{p}_i$. Acestea sunt valorile „aşteptate“ ale lui p_i şi respectiv n_i, în ipoteza că atributul X nu este corelat cu atributul de ieşire (*clasa*).

Folosind datele de mai sus, vom calcula *mărimea numerică corespunzătoare testului statistic* χ^2:

$$\chi^2 = \sum_{i=1}^{k} \left(\frac{(p_i - \bar{p}_i)^2}{\bar{p}_i} + \frac{(n_i - \bar{n}_i)^2}{\bar{n}_i} \right)$$

În ipoteza că atributul X nu este corelat cu *clasa*, mărimea calculată mai sus urmează *distribuţia* χ^2.

Un *parametru* al distribuţiei χ^2 este aşa-numitul *grad de libertate*. Pentru X, gradul de libertate este $k-1$ atunci când se lucrează cu doar două etichete ($+$ şi $-$).[599] În particular, pentru un atribut binar gradul de libertate este 1.

După ce vom calcula valoarea mărimii statistice χ^2 pornind de la date, o vom compara cu o *valoare critică*, care reprezintă un prag astfel încât probabilitatea ca mărimea statistică calculată să depăşească acel prag este de cel mult α în cazul în care variabilele nu sunt corelate. α este *nivelul de încredere* (engl., confidence level) şi de obicei se consideră $\alpha = 0.05$, fiind astfel siguri în proporţie de 95% că o partiţionare (engl., split) care trece testul este semnificativă d.p.v. statistic.

De exemplu, atunci când gradul de libertate este 1 iar $\alpha = 0.05$, valoarea critică este 3.841; pentru 2 grade de libertate şi acelaşi α, ea este 5.991. Spunem că o partiţionare este *statistic semnificativă* dacă mărimea calculată χ^2 trece de valoarea critică.

[599] Dacă în loc de două etichete am fi considerat n etichete [de clasă] diferite, atunci am fi folosit o statistică asemănătoare, cu $(n-1)(k-1)$ grade de libertate.

Y	X_1	X_2	*Nr. apariţii*
+	T	T	3
+	T	F	4
+	F	T	4
+	F	F	1
−	T	T	0
−	T	F	1
−	F	T	3
−	F	F	5

a. Conform testului χ^2 pentru $\alpha = 0.05$, care dintre cele trei partiţionări din arborele ID3 obţinut pe datele din tabelul alăturat (Y fiind atributul-ţintă) sunt statistic semnificative?

b. Ţinând cont de rezultatul de la punctul a, care este arborele de decizie obţinut în urma pruning-ului bottom-up bazat pe testul χ^2 cu nivelul de încredere $\alpha = 0.05$?

56. **(Algoritmul ID3: Adevărat sau Fals?)**

∗ ***CMU, 2004, fall, final exam, pr. 1.b***

Învăţăm un arbore de decizie folosind algoritmul ID3 standard, fără pruning. Atributele de intrare $(X_1, X_2, ..., X_m)$ sunt categoriale, iar atributul de ieşire (Y) este de asemenea categorial.

Marcaţi cu A (adevărat) sau F (fals) fiecare din afirmaţiile de mai jos şi daţi în fiecare caz o explicaţie succintă, însoţită eventual de un exemplu sau un contraexemplu minimalist.

a. Dacă X_i şi Y văzute ca variabile aleatoare sunt independente (raportat la distribuţia probabilistă care a generat datele de antrenament), atunci X_i nu va apărea în arborele de decizie.

b. Dacă $IG(Y, X_i) = 0$, atunci atributul X_i nu va apărea în arborele de decizie.

c. Adâncimea maximă a arborelui de decizie este de cel mult m.
Notă: Dacă arborele este format doar din nodul rădăcină, ceea ce corespunde cazului în care toate exemplele de antrenament sunt identic clasificate, atunci se consideră că adâncimea arborelui este 0.

d. Dacă sunt R exemple de antrenament, atunci adâncimea maximă a arborelui de decizie este de cel mult $1 + \log_2 R$.

e. Dacă sunt R exemple de antrenament, iar unul dintre atributele de intrare are R valori distincte şi ia valori $v_1, \ldots, v_R$ în mod injectiv (pe mulţimea formată de exemplele de antrenament), atunci arborele de decizie va avea adâncimea 0 sau 1.

57. **(Adevărat sau Fals?**
ID3 vs. regresia logistică;
Acurateţe la antrenare vs. acurateţe la testare)

• ○ ***CMU, 2011 fall, T. Mitchell, A. Singh, midterm exam, pr. 1.1-3***

Presupunem că avem un set de date de imagini celulare de la pacienţi cu şi, respectiv, fără cancer.

a. Dacă ţi se cere să antrenezi un clasificator care prezice probabilitatea ca pacientul să aibă cancer, ai prefera să foloseşti mai degrabă arbori de decizie decât regresia logistică.

b. Să zicem că setul de date conţine 900 de imagini de la pacienţi fără cancer şi 100 de imagini de la pacienţi cu cancer. Dacă antrenăm un clasificator şi obţinem 85% acurateţe pe acest set de date, putem spune că acesta este un clasificator bun.

c. Un clasificator care atinge 100% acurateţe pe setul de antrenament şi 70% acurateţe pe setul de validare este mai bun decât un clasificator care atinge 70% acurateţe pe setul de antrenament şi 75% acurateţe pe setul de validare.

58. **(Implementare: algoritmul ID3)**

□ ***Liviu Ciortuz, 2016***

Pornind de la pseudo-codul algoritmului ID3 în varianta de bază (adică, folosind doar atribute cu valori discrete) dată în cartea *Machine Learning* de Tom Mitchell la pagina 56, elaboraţi o implementare, în limbajul de programare preferat. „Inima“ acestui algoritm este calculul câştigului de informaţie (a se vedea pr. 35). Programul va conţine o funcţie de antrenare şi una de testare. Ca input (la linia de comandă), programul va primi

– numele unui fişier în format SSV (engl., space-separated values), conţinând instanţele de antrenament, câte una pe fiecare linie, variabila de ieşire aflându-se pe ultima poziţie;

– numele unui al doilea fişier, tot în format SSV, conţinând instanţa de test sau instanţele de test (de asemenea, câte una pe fiecare linie).

Opţional, pe prima linie a fişierului de antrenament vor fi indicate pentru fiecare atribut numele lui, precum şi numărul de valori pe care le ia respectivul atribut (drept separatori, se vor folosi tot spaţii). În acest caz, primul caracter din această linie-comentariu va fi %.

Folosiţi o modalitate convenabilă pentru a afişa arborele ID3 obţinut, de exemplu ca program în logica propoziţiilor cu variabile cu valori multiple.

Pentru a testa programul, puteţi folosi de pildă datele de la problema 2.

Ulterior, veţi putea adăuga programului una dintre următoarele extensii (sau chiar ambele):

– variabile cu atribute continue; testaţi programul pe datele de la problema 10 din prezentul capitol şi pe cele de la problema 11.b de la capitolul *Învăţare bazată pe memorare*;[600]

– o funcţie care, în vederea contracarării fenomenului de overfitting, face trunchierea / pruning-ul arborelui ID3, în varianta "reduced-error", folosind un al treilea set / fişier de instanţe, pentru validare; a se vedea cartea *Machine Learning* de Tom Mitchell, pag. 69-70.[601]

[600] Pentru testarea variantei algoritmului ID3 care foloseşte atât atribute discrete cât şi atribute continue, puteţi folosi datele de la problema 51, precum şi cele de la problema 12.

[601] Pentru testare, vedeţi problemele CMU, 2008 spring, T. Mitchell, W. Cohen, HW1, pr. 2, CMU, 2012 spring, Roni Rosenfeld, HW3 şi / sau CMU, 2011 fall, T. Mitchell, A. Singh, HW1, pr. 2.

Indicaţie*: Pentru a lucra în mod convenabil cu atributele discrete, mai ales atunci când aceste valori nu sunt numerice, în loc să folosiţi (în timpul procesărilor) valorile lor aşa cum apar în fişierele de date, puteţi crea un *vocabular (ca vector în care se memorează valorile tuturor acestor variabile) **şi apoi să înregistraţi în tabelele de date (în locul valorile atributelor discrete) indecşii corespunzători „intrărilor“ din vocabular.**

4.2.2 Algoritmul AdaBoost

59. **(Algoritmului AdaBoost: o proprietate simplă, dar utilă)**

Liviu Ciortuz, 2026

Demonstraţi că orice funcţie de forma $H_2(x) = sign\,(\alpha_1 h_1(x) + \alpha_1 h_2(x))$,[602] **unde**

h_1 **şi** h_2 **sunt funcţii definite pe acelaşi domeniu ca şi** H_2 **şi iau valori în mulţimea** $\{-1, +1\}$,[603]

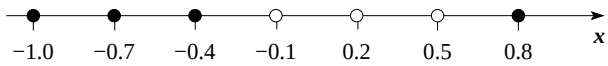
α_1 **şi** α_2 **sunt numere reale strict pozitive,**

este de fapt identică (sau, am putea spune, echivalentă) cu una dintre aceste două funcţii.

60. **(Algoritmul AdaBoost: întrebări în legătură cu aplicarea algoritmului pe un set de date din** $\mathbb{R}$**)**

prelucrare de Liviu Ciortuz, 2020, după
• ◦ CMU, 2011 fall, T. Mitchell, A. Singh, HW6, pr. 3.2-8

Considerăm setul de date de antrenament din figura următoare. Simbolurile • şi ◦ desemnează etichete pozitive şi respectiv etichete negative. Vom folosi algoritmul AdaBoost cu compaşi de decizie în rolul de ipoteze „slabe“.



a. Determinaţi în mod riguros separatorul decizional corespunzător primei ipoteze „slabe“, h_1**. Desenaţi-l pe figura de mai sus şi indicaţi [eventual printr-o mică săgeată perpendiculară pe acest separator] care este zona clasificată cu** $+$**.**

b. Calculaţi ε_1**. Cât este acurateţea obţinută de AdaBoost la antrenare dacă oprim acum algoritmul?**

c. Cât va fi valoarea noilor probabilităţi / ponderi $D_2(i)$ **pentru fiecare dintre cele şapte exemple de antrenament?**

d. Identificaţi în mod riguros separatorul decizional corespunzător celei de-a doua ipoteze „slabe“, h_2 **şi apoi includeţi-l în desen. Indicaţi iarăşi zona de decizie corespunzătoare clasei** $+$**.**

[602] Această funcţie poate fi văzută ca o [aşa-numită] *ipoteză combinată* în contextul algoritmului AdaBoost.
[603] Aceste funcţii h_t pot fi văzute ca [aşa-numite] *ipoteze slabe* în contextul algoritmului AdaBoost.

e. Care sunt exemplele de antrenament cărora le va fi asignată cea mai mică pondere / probabilitate după ce algoritmul AdaBoost va fi terminat cea de-a doua sa iteraţie?

f. Se îmbunătăţeşte oare acurateţea la antrenare obţinută de AdaBoost la a doua iteraţie în raport cu cea obţinută la prima iteraţie?

g. Este oare posibil ca algoritmul AdaBoost să obţină acurateţe 1 pe acest set de date de antrenament? Veţi justifica riguros, iar în cazul afirmativ, veţi indica separatorii decizionali şi zonele de decizie.

h. Ar putea fi învăţat acest set de date cu eroare zero la antrenare de către AdaBoost dacă nu se folosesc deloc praguri exterioare de separare (i.e., compaşi de decizie exteriori)?

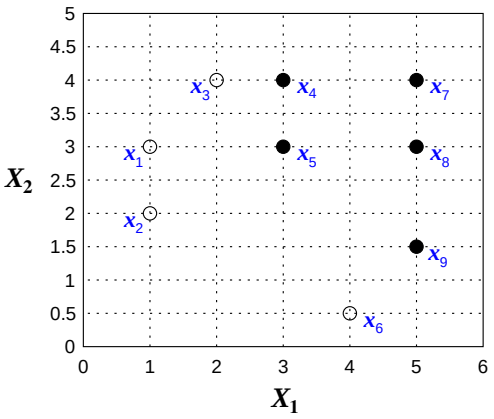
61. **(AdaBoost: aplicare pe un set de date din $\mathbb{R}^2$)**

adaptare facută de Liviu Ciortuz, după
• ○ MIT, 2004 fall, Tommi Jaakkola, final exam, pr. 1

Fie setul de exemple de antrenament (instanţe etichetate) din figura de mai jos. Simbolurile • şi ○ desemnează etichete pozitive şi respectiv etichete negative. Pentru a rezolva această problemă de clasificare, vă cerem să aplicaţi algoritmul AdaBoost folosind drept ipoteze „slabe“ compaşi de decizie (engl., decision stump). La fiecare iteraţie de boosting veţi selecta acel compas de decizie care minimizează eroarea ponderată la antrenare (engl., weighted training error). În cazul în care există mai mulţi compaşi de decizie care au [o aceeaşi!] cea mai bună eroare ponderată la antrenare, veţi putea alege unul dintre ei în mod arbitrar.

a. Pe figura dată, desenaţi primul compas de decizie şi etichetaţi-l cu h_1, indicând de asemenea cu $+/-$ cele două zone de decizie pe care le determină acest compas.

b. Pe aceeaşi figură, scrieţi în apropierea fiecărei instanţe probabilitatea asignată ei după prima iteraţie executată de algoritmul AdaBoost. De asemenea, încercuiţi instanţele care au probabilitatea cea mai mare. Justificaţi răspunsul în mod riguros.



c. Cât este eroarea ponderată la antrenare produsă de primul compas de decizie după prima iteraţie, adică după ce probabilităţile asociate instanţelor au fost recalculate? Justificaţi.

d. Desenaţi pe figura dată cel de-al doilea compas de decizie şi etichetaţi-l cu h_2, indicând de asemenea cu $+/-$ cele două zone de decizie pe care le determină acest compas.

e. Există oare instanţe de antrenament care sunt clasificate eronat de către H_2, *ipoteza combinată* produsă de AdaBoost după două iteraţii? Justificaţi răspunsul în mod riguros.

f. Dacă răspunsul pe care l-aţi dat la punctul e este pozitiv, ce se poate spune dacă se execută încă o iteraţie (adică, a treia)? Desenaţi pe figura dată cel de-al treilea compas de decizie şi etichetaţi-l cu h_3, indicând de asemenea cu $+/-$ cele două zone de decizie pe care le determină acest compas. Dacă eroarea la antrenare este acum 0, trasaţi separatorul decizional determinat de algoritmul AdaBoost.

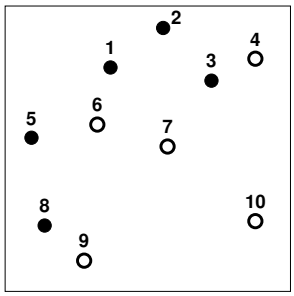
g. Care ar fi rezultatul — i.e., eroarea la antrenare produsă de ipoteza combinată şi zonele de decizie — dacă AdaBoost ar alege ca a treia ipoteză „slabă" un alt (cel mai bun!) compas de decizie (h'_3) în locul lui h_3?

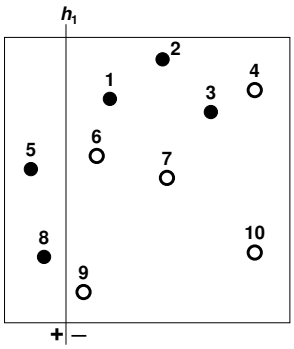
62. **(Algoritmul AdaBoost: aplicare pe un set de date din $\mathbb{R}^2$)**

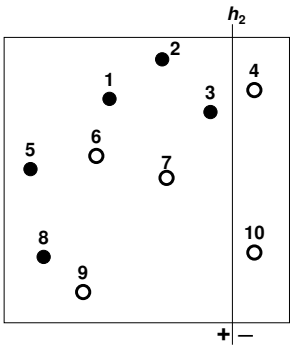
Liviu Ciortuz, 2021, pornind de la un slide al lui Virgil Pavlu, Northeastern University, USA

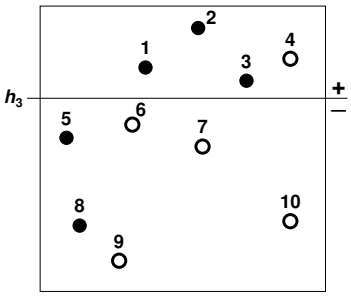
Se consideră că aplicăm algoritmul AdaBoost pe dataset-ul din figura de mai jos. (Pentru uşurinţa exprimării la calcule, am notat pe figură indicii instanţelor de antrenament, în imediata apropiere a acestora.)

Folosim convenţia noastră obişnuită de notare: simbolul $\bullet$ desemnează instanţe pozitive, iar simbolul $\circ$ instanţe negative.
La primele trei iteraţii ale algoritmului au fost selectaţi compaşii de decizie h_1, h_2 şi h_3 (în această ordine), aşa cum se indică în figurile de mai jos.









***Obiectivul* acestei probleme este să determinăm dacă la sfârşitul celor trei iteraţii algoritmul AdaBoost reuşeşte să clasifice perfect toate instanţele de antrenament.**

a. Calculaţi

– distribuţiile probabiliste corespunzătoare celor 3 iteraţii, adică $D_1(x_i)$, $D_2(x_i)$ şi $D_3(x_i)$, pentru $i = 1, \ldots, 10$;

– pentru fiecare dintre cele 3 iteraţii ($t = 1, 2, 3$): eroarea ponderată la antrenare (ε_t) produsă de compasul de decizie h_t, precum şi ponderea (α_t) asociată ipotezei / compasului de decizie h_t.

Veţi completa tabelele următoare şi veţi indica succint(!) modul în care aţi procedat pentru a ajunge la rezultatele respective.

i	1	2	3	4	5	6	7	8	9	10
$D_1(x_i)$										
$D_2(x_i)$	1/6			1/14						
$D_3(x_i)$	7/66			1/22		1/6				

t	1	2	3
ε_t			
α_t			

***Atenţie*! Pentru a vă uşura munca, am completat noi câteva dintre elementele tabelului precedent. Bazaţi-vă pe valorile indicate de noi, ca să nu faceţi calcule laborioase, păstrând însă rigurozitatea / corectitudinea raţionamentelor.**

b. Folosind ipoteza combinată obţinută de algoritmul AdaBoost la finalul celei de-a treia iteraţii, stabiliţi:

i. **eroarea la antrenare produsă (pentru calcularea ei, puteţi folosi tabelul de mai jos),**

ii. **zonele de decizie corespunzătoare acestui clasificator (veţi justifica modul în care aţi procedat!). Veţi indica aceste zone de decizie, precum şi graniţele de decizie, pe primul desen din enunţ.**

t	α_t	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}
1	$\alpha_1 = \ldots$										
2	$\alpha_2 = \ldots$										
3	$\alpha_3 = \ldots$										
	$H_3(x_i)$										

c. Verificaţi dacă alegerile ipotezelor / compaşilor de decizie h_1, h_2 şi h_3 sunt optime în sensul minimizării erorilor ponderate la antrenare la iteraţiile respective.

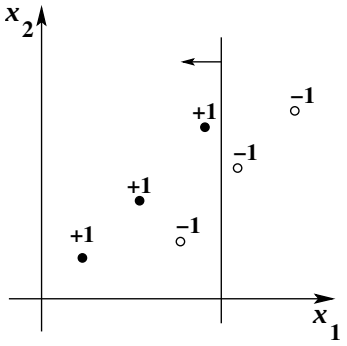
În cazul afirmativ, verificaţi dacă [la iteraţiile respective] aceste ipoteze sunt singurele soluţii optime. Dacă de data aceasta răspunsul este negativ, exploraţi o rezolvare alternativă care poate fi produsă de AdaBoost. Indicaţi separatorii decizionali şi zonele de decizie care se obţin la convergenţă (adică, atunci când eroarea la antrenare devine 0).

63. **(AdaBoost: aplicarea algoritmului pe un set de date din $\mathbb{R}^2$)**

prelucrare de Liviu Ciortuz, după
• ○ CMU, 200X spring, midterm, pr. 3
MIT, 2006 fall, Tommi Jaakkola, final, pr. 2

Folosind algoritmul AdaBoost, vrem să obţinem un ansamblu de compaşi de decizie (engl., decision stumps) h_t, de forma $H(x) = sign\left(\sum_{i=1}^{T} \alpha_t\, h_t(x)\right)$.

În figura alăturată sunt desenate câteva puncte (instanţe) etichetate în planul bidimensional, precum şi primul compas de decizie care a fost ales de către algoritmul AdaBoost. Un compas de decizie oarecare produce valori binare ± 1, ţinând cont doar de un anumit *prag* (engl., the split point). Săgeata mică din figură, care este perpendiculară pe dreapta care reprezintă compasul de decizie indică *zona de decizie* pentru care compasul de decizie va produce valoarea $+1$.



a. Încercuiţi toate acele instanţe din figură pentru care ponderea / probabilitatea [atribuită de către AdaBoost] va creşte ca urmare a incorporării [în ipoteza combinată H a] primului compas de decizie. Justificaţi răspunsul în mod riguros.

b. Desenaţi pe aceeaşi figură un compas de decizie care va putea fi selectat la următoarea iteraţie a algoritmului AdaBoost. Veţi trasa atât dreapta care reprezintă compasul de decizie cât şi [o săgeată care să indice] zona sa de decizie pozitivă. Justificaţi în mod riguros.

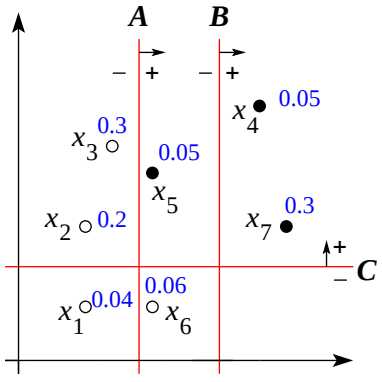
c. Va fi oare coeficientul / votul α_2, care este asociat celui de-al doilea compas de decizie mai mare decât α_1, coeficientul [din ansamblul H] pentru primul compas de decizie? Cu alte cuvinte, vom avea oare $\alpha_2 > \alpha_1$? Justificaţi în mod riguros.

d. Este oare setul acesta de date este învăţabil în sens empiric γ-slab?

64. **(Algoritmul AdaBoost: exemplu de aplicare pe date din $\mathbb{R}^2$; întrebări calitative)**

prelucrare de Liviu Ciortuz, 2021, după
• ○ MIT, 2007 fall, Tommi Jaakkola, final ex, pr. 1

Fie setul de date de antrenament din figura alăturată. Am marcat cu semnul ○ exemplele / instanţele negative ($y_i = -1$) şi cu semnul • instanţele pozitive ($y_i = +1$). Figura conţine de asemenea ponderile normalizate (adică, probabilităţile) asociate exemplelor de antrenament, aşa cum au rezultat în urma executării unui anumit număr de iteraţii ale algoritmului AdaBoost. În figură sunt trasaţi şi trei compaşi de decizie, $h(x;\theta_A)$, $h(x;\theta_B)$ şi $h(x;\theta_C)$ sau, pe scurt, A, B şi C.



a. Care dintre aceşti trei compaşi de decizie consideraţi că a fost folosit la *precedenta iteraţie* a algoritmului AdaBoost, în aşa fel încât să rezulte ponderile [asociate exemplelor] prezentate în figură? Veţi răspunde indicând A, B sau C şi veţi justifica în mod riguros alegerea pe care aţi făcut-o.

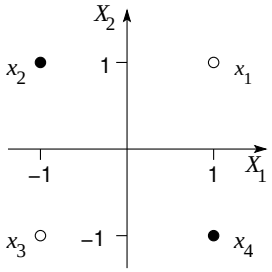
b. Pe care dintre aceşti trei compaşi de decizie consideraţi că-l va selecta algoritmul AdaBoost la *iteraţia următoare*? Veţi răspunde indicând A, B sau C şi veţi justifica riguros, prin calcule, alegerea pe care aţi făcut-o.

c. În figura dată, încercuiţi instanţele de antrenament (este posibil să nu fie niciuna!) pe care ansamblul (i.e., combinaţia liniară de compaşi de decizie) $H_2(x) = sign\,(\alpha_A h(x;\theta_A) + \alpha_C h(x;\theta_C))$, cu $\alpha_A = 0.3$ şi $\alpha_C = 0.5$ nu le poate clasifica în mod corect.

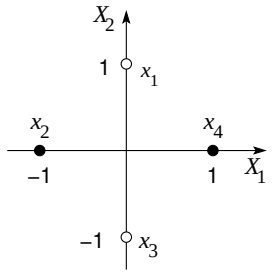
65. **(Algoritmul AdaBoost: aplicare pe dataset-uri de tip XOR)**

• ○ ***CMU, 2011 fall, T. Mitchell, A. Singh, HW6, pr. 3.1***
CMU, 2007 spring, Carlos Guestrin, HW2, pr. 2.2

a. Consideraţi setul de date XOR, reprezentat ca de obicei în planul bidimensional. Presupunând că algoritmul AdaBoost foloseşte compaşi de decizie pe post de ipoteze „slabe", va fi el oare capabil să obţină o acurateţe mai bună de 50% pe acest set de date? Justificaţi răspunsul dumneavoastră în mod riguros. (Ca de obicei, clasa $+1$ a fost reprezentată prin simbolul •, iar clasa -1 prin simbolul ○.)



În cele ce urmează veţi aplica acelaşi algoritm AdaBoost pe setul de date din figura alăturată (obţinut prin rotirea dataset-ului XOR cu $45°$ la stânga şi făcând apoi rescalarea).



b. Arătaţi cum lucrează AdaBoost pe acest [al doilea] set de date, considerând că numărul de iteraţii de efectuat este $T = 4$. Vă reamintim că pentru fiecare $t = 1, \ldots, 4$ va trebui să calculaţi numerele $\varepsilon_t, \alpha_t, Z_t$, precum şi distribuţia probabilistă $D_t(i)$ pentru $i = 1, \ldots, 4$. De asemenea, la fiecare iteraţie t veţi trasa separatorul decizional corespunzător ipotezei h_t alese de clasificatorul „slab" (A) şi veţi indica printr-o mică săgeată desenată perpendicular pe separator zona de decizie corespunzătoare etichetei $+1$.

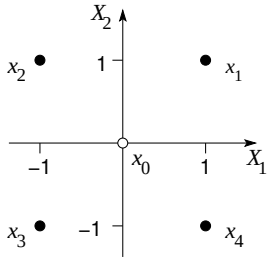
c. Cât va fi la finalul celor $T = 4$ iteraţii eroarea la antrenare produsă de combinaţia liniară H_4 livrată la ieşire de către algoritmul AdaBoost? Indicaţi pe desenul pus la dispoziţie mai sus separatorul decizional şi zonele de decizie determinate de H_4.

d. Este oare setul de date (cel folosit la punctele b şi c) liniar separabil? Explicaţi de ce algoritmul AdaBoost se comportă mai bine decât un [singur] compas de decizie pe acest set de date.

66. **(Algoritmul AdaBoost: aplicare pe un set de date din $\mathbb{R}^2$)**

prelucrare de Liviu Ciortuz, 2021, după
• ○ CMU, 2010 fall, Aarti Singh, midterm, pr. 6.1-2

În acest exerciţiu veţi aplica algoritmul AdaBoost folosind drept clasificatori „slabi“ compaşi de decizie pe setul de date de antrenament prezentat în figura de mai jos. (Atenţie! Sunt patru instanţe pozitive şi una negativă.)



a. Arătaţi că există o combinaţie liniară de doi compaşi de decizie care este consistentă cu aceste date de antrenament. Apoi, fără a face referire la aplicarea efectivă a algoritmului AdaBoost, arătaţi că acest algoritm nu poate să clasifice perfect acest set de date în mai puţin de 3 iteraţii.

b. Cât de multe iteraţii vor fi necesare pentru a atinge eroare zero la antrenare? Justificaţi elaborând toate detaliile necesare.

c. Puteţi adăuga încă un exemplu (instanţă de antrenament) la setul de date de mai sus astfel încât algoritmul AdaBoost să obţină în [doar] două iteraţii eroare la antrenare zero? Dacă nu, explicaţi de ce nu este posibil aşa ceva.

d. Reuşeşte oare algoritmul AdaBoost să înveţe perfect (i.e., cu eroare nulă la antrenare) acest set de date de antrenament? În cazul afirmativ, identificaţi separatorul decizional obţinut [atunci când eroarea la antrenare a ajuns la 0], precum şi zonele de decizie corespunzătoare.

e. Care credeţi că este motivul (principal) pentru care se folosesc clasificatori „slabi“ (şi nu clasificatori mai puternici / „tari“) în conjuncţie cu algoritmul AdaBoost?

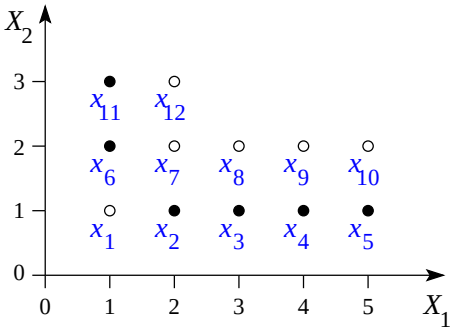
67. **(AdaBoost şi *non*-învăţabilitate γ-slabă: exemplificare pe date din $\mathbb{R}^2$)**

prelucrare de Liviu Ciortuz, după
• · CMU, 2006 spring, Carlos Guestrin, final, pr. 3

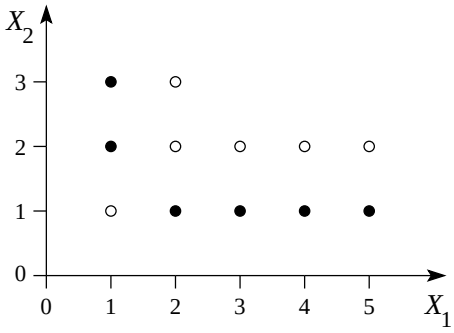
Considerând setul de date de antrenament din figura / figurile de mai jos, am dori să „învăţăm“ un clasificator care să separe instanţele pozitive de instanţele negative, folosind algoritmul AdaBoost. Fiecare instanţă x_i are o etichetă $y_i \in \{+1. - 1\}$; eticheta $+1$ corespunde simbolului •, iar eticheta -1 corespunde simbolului ○.

Vom folosi ipoteze „slabe“, ale căror graniţe de decizie (engl., decision boundaries) sunt paralele cu una din axele de coordonate, adică separatorul este fie vertical, fie orizontal. Puteţi gândi aceste ipoteze „slabe“ ca fiind compaşi de decizie, pentru care pragurile de separare (engl., threshold splits) sunt situate fie pe axa X fie pe axa Y.

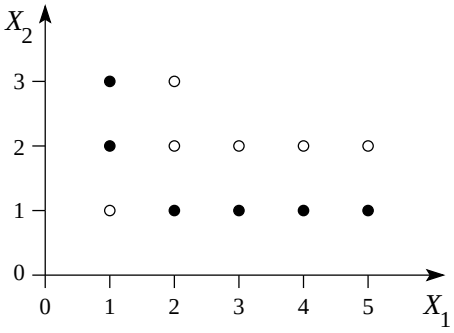
La fiecare iteraţie t, vom alege acea ipoteză „slabă“ h_t care maximizează acurateţea ponderată la antrenare în raport cu ponderile / probabilităţile curente D_t, adică alegem h_t care maximizează $\sum_i D_t(i) \cdot 1_{\{h_t(x_i)=y_i\}}$. Ca de obicei, se consideră că $h_t(x)$ ia valori doar în mulţimea $\{+1, -1\}$, după cum ipoteza h_t clasifică instanţa x ca fiind pozitivă ori negativă.



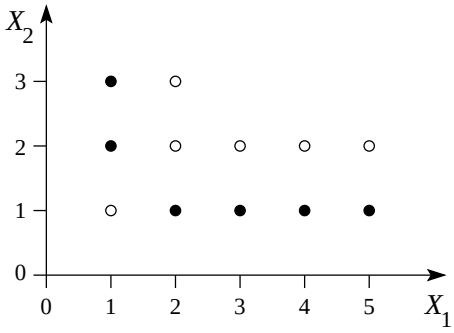
(a) Graniţa de decizie după prima iteraţie.



(b) Graniţa de decizie după a doua iteraţie.



(c) Exemplul / exemplele cu cea mai mică pondere.



(d) Exemplul / exemplele cu cea mai mare pondere.

a. Desenaţi pe figura (a) graniţa de decizie determinată de algoritmul AdaBoost după prima iteraţie (adică, după ce a fost aleasă prima ipoteză „slabă“). Nu uitaţi să indicaţi ce parte a planului euclidian este clasificată cu $'+'$ şi respectiv $'-'$.

b. Efectuăm acum cea de-a doua iteraţie de boosting. Desenaţi graniţele de decizie ale ambelor ipoteze „slabe“ în figura (b). AdaBoost combină deciziile celor două ipoteze „slabe“. Indicaţi care sunt regiunile din plan [în care sunt situate punctele] clasificate de către boosting-ul cu 2 ipoteze „slabe“ cu $'+'$ şi respectiv cu $'-'$.

c. La problema 22.i am arătat că $\alpha_i > \alpha_j \Leftrightarrow \varepsilon_i < \varepsilon_j$. Verificaţi că această relaţie este într-adevăr satisfăcută pentru primele două iteraţii ale algoritmului AdaBoost pe setul de antrenament dat.

d. Marcaţi în figura (c) exemplul / exemplele cu cea mai mică pondere / probabilitate (D_{t+1}) după primele două iteraţii de boosting ($t = 2$).

e. Marcaţi în figura (d) exemplul / exemplele cu cea mai mare pondere / probabilitate (D_{t+1}) după primele două iteraţii de boosting ($t = 2$).

f. Câte exemple de antrenament sunt clasificate eronat după primele două iteraţii de boosting?

g. Folosind acest set de date şi ipotezele „slabe“ menţionate mai sus, va obţine oare AdaBoost vreodată eroare la antrenare zero? Justificaţi în mod riguros, folosind o *implementare*.

Veţi alcătui graficele pentru ϵ_t (eroarea ponderată la antrenare pentru ipoteza h_t), pentru eroarea la antrenare produsă de ipoteza combinată H_T (eroare pe care am notat-o cu $err_S(H_T)$), precum şi pentru cele două *margini superioare* ale lui $err_S(H_T)$ (vedeţi problea 23, punctele b si d), în funcţie de numărul de iteraţii executate. (Vedeţi graficele de la finalul problemelor 24 şi 25.)

68. **(Algoritmul AdaBoost şi învăţabilitate empirică γ-slabă: o margine superioară pentru numărul de iteraţii de efectuat până la atingerea erorii empirice 0)**

□ • ○ ***CMU, 2010 fall, Aarti Singh, midterm, pr. 6.3***
CMU, 2011 fall, Eric Xing, HW5, pr. 3.2.ab

Să presupunem că rulăm algoritmul AdaBoost pe m exemple de antrenament şi, de asemenea, că la fiecare iteraţie (t) eroarea ponderată la antrenare (ε_t) produsă de ipoteza „slabă“ h_t este de cel mult $1/2 - \gamma$, unde $\gamma > 0$ nu depinde de t.

Determinaţi numărul de iteraţii (T) pe care trebuie să le execute algoritmul AdaBoost pentru ca ipoteza combinată H să fie consistentă cu cele m exemple de antrenament, adică să obţină eroare zero la antrenare. Trebuie să exprimaţi răspunsul doar în funcţie de m şi de γ.

***Sugestie*: Cât este eroarea la antrenare atunci când un singur exemplu este clasificat eronat?**

69. **(Algoritmul AdaBoost: margini de votare; aplicaţie)**

□ · ***Liviu Ciortuz, 2020***

În această problemă veţi folosi setul de date de la problema 24.

a. Pentru fiecare dintre iteraţiile $t = 1$, $t = 2$ şi $t = 3$, calculaţi $Margin_t(x_i)$, pentru $i = 1, \ldots, 9$. (Pentru definiţia noţiunii de margine de votare, vedeţi enunţul problemei 27.)

b. Verificaţi, utilizând distribuţiile probabiliste D_t care au fost calculate la problema 24, că într-adevăr are loc implicaţia $Margin_t(x_i) > Margin_t(x_j) \Rightarrow D_{t+1}(i) < D_{t+1}(j)$ — care a fost demonstrată pentru cazul general la problema 27.b — pentru cazul particular $t = 1$, $i = 1$ şi $j = 4$, precum şi pentru cazul $t = 2$, $i' = 4$ şi $j' = 8$.

c. Pentru fiecare dintre iteraţiile $t = 1$, $t = 2$ şi $t = 3$ în parte, reprezentaţi grafic: i. *distribuţia empirică* a marginilor de votare care au fost calculate la punctul a, precum şi ii. *distribuţia empirică cumulativă* a acestor margini.[604]

d. De asemenea, pentru fiecare dintre iteraţiile $t = 1$, $t = 2$ şi $t = 3$ în parte, calculaţi media marginilor [care au fost calculate la punctul a], în sensul definit/folosit la problema 28.

[604] Ca aplicaţie practică, vedeţi problema CMU, 2007 spring, Carlos Guestrin, HW2, pr. 2.3, care în Companionul practic al culegerii de exerciţii de învăţare automată, https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/ML.ex-book.Companion.pdf, este problema 38.c.

70. **(Clasificare de documente: selecţie de trăsături folosind algoritmul AdaBoost cu compaşi de decizie pentru atribute booleene)**

• ○ ***CMU, 2007 spring, midterm, pr. 3***
MIT, 2003 fall, Tommi Jaakkola, final, pr. 3.2-4

Considerăm o problemă de clasificare de texte, în care un document X este reprezentat sub forma unui vector de trăsături binare relativ la cuvintele din dicţionarul limbii în care este scris documentul respectiv. Din punct de vedere formal, putem scrie $X = [X_1, X_2, X_3, \ldots, X_m]$, unde $X_j = 1$ în cazul în care cuvântul de pe poziţia j din dicţionar este prezent în documentul X, şi 0 în caz contrar.

Vom considera acum că algoritmul AdaBoost foloseşte nişte ipoteze „slabe“ desemnate formal ca perechi (j, y) unde j este indicele unui cuvânt de dicţionar, iar y este clasa selectată, cu $y \in \{-1, +1\}$. Din punct de vedere *intuitiv*, acest lucru se poate exprima astfel: fiecare ipoteză slabă este [o pereche formată din] un cuvânt împreună cu o etichetă reprezentând clasa asociată. De *exemplu*, considerând cuvântul `football` şi clasele {`sports`, `non-sports`}, atunci vom avea două ipoteze „slabe“ relative la acest cuvânt, şi anume

- **ipoteza (`football`, $+$): dacă documentul conţine cuvântul `football`, [vom] prezice clasa `sports`; altfel, [vom] prezice clasa `non-sports`;**
- **ipoteza (`football`, $-$): dacă documentul conţine cuvântul `football`, [vom] prezice clasa `non-sports`; altfel, [vom] prezice clasa `sports`.**

a. Cât de multe ipoteze „slabe“ există în acest model?

Acest algoritm de tip boosting poate fi folosit pentru a face *selecţia de trăsături*. Rulăm algoritmul şi selectăm trăsăturile în *ordinea în care au fost identificate* de către algoritm.

b. Este oare posibil ca acest algoritm de boosting să selecteze o aceeaşi ipoteză „slabă“ mai mult decât de o singură dată? Justificaţi.

c. Să zicem că vrem să facem o ordonare (engl., ranking) a trăsăturilor în funcţie de informaţia mutuală [individuală] relativă la variabila care reprezintă clasa (y), adică $IG(y; X_j)$. Va fi oare această ordonare mai „informativă“ / bună decât ordonarea produsă de către algoritmul AdaBoost? Justificaţi.

71. **(Concepte din $\mathbb{R}$ *reprezentabile* cu ajutorul combinaţiilor liniare de compaşi de decizie)**

Liviu Ciortuz, 2021, pornind de la
• · ***CMU, 2011 spring, Roni Rosenfeld, HW10, pr. 4.a***

Presupunem că avem o problemă de clasificare a unor instanţe pe axa reală: fiecare instanţă x_i este un număr real, iar etichetele pe care urmează să le preziceţi sunt binare, $y_i \in \{-1, +1\}$.

Pentru această problemă de clasificare, veţi folosi *ansambluri*, adică nişte combinaţii liniare de separatori / ipoteze „slabe“. (Atenţie! NU trebuie să folosiţi

algoritmul AdaBoost!) Vă readucem aminte că un astfel de *clasificator* are forma următoare:

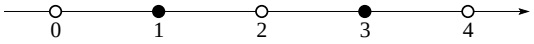
$$\hat{y} = sign\Big(\sum_{t=1}^{T} \alpha_t h_t(x)\Big), \tag{290}$$

unde $\hat{y}$ este eticheta prezisă, $sign(x)$ este $+1$ dacă $x > 0$ şi respectiv -1 în cazul contrar, α_t este o *pondere* (număr real strict pozitiv), iar $h_t(x)$ este predicţia făcută de către ipoteza „slabă“ h_t. Fiecare h_t ia una dintre următoarele forme:

$$h_t(x; s, +) = \begin{cases} -1 & \text{dacă } x < s \\ +1 & \text{dacă } x \geq s \end{cases} \qquad h_t(x; s, -) = \begin{cases} +1 & \text{dacă } x < s \\ -1 & \text{dacă } x \geq s, \end{cases}$$

pentru un anumit *prag de separare* (engl., split threshold) $s \in \mathbb{R}$.[605]

Considerăm următorul set de date, format din 5 instanţe situate pe axa reală:



a. Arătaţi că acest set de date (LC: sau, acest *concept*) este *reprezentabil* cu ajutorul unei combinaţii liniare care este formată din 4 ipoteze „slabe“. Aşadar, vă cerem să identificaţi în mod explicit 4 ipoteze „slabe“ $h_1, \ldots, h_4$, precum şi ponderile lor $\alpha_1, \ldots, \alpha_4$, astfel încât ipoteza combinată $sign\Big(\sum_{t=1}^{T} \alpha_t h_t(x)\Big)$ să fie *consistentă* cu datele de mai sus.
Puneţi în evidenţă cele 4 ipoteze „slabe“ $h_1, \ldots, h_4$ pe desenul de mai sus, folosind praguri de separare $s_1, \ldots, s_4$ precum şi linii verticale, cărora le veţi asocia o etichetare adecvată. Concret, veţi desemna cu semnele $+$ şi $-$ la stânga şi la dreapta fiecărei linii verticale — asociată unui anumit prag s_j — zonele de decizie determinate de această ipoteză „slabă“.

***Observaţie*: Ţinând cont de modul în care a fost definită funcţia $sign$ mai sus, *regula de predicţie* (290) va trata cazurile de „paritate“ etichetând cu -1 instanţele pentru care suma ponderată (engl., weighted sum) a predicţiilor făcute de ipotezele „slabe“ este 0. Acest mod de tratare a cazurilor de „paritate“ este [foarte] util pentru rezolvarea punctului a.**

***Indicaţie*: Pentru a justifica uşor consistenţa *ansamblului* ales de dumneavoastră cu datele din figura de mai sus, vă cerem să completaţi un tabel similar cu cel pe care-l folosim la algoritmul AdaBoost atunci când facem calculul erorii la antrenare produse de ipoteza combinată $sign\Big(\sum_{t=1}^{T} \alpha_t h_t(x)\Big)$.**

α_t	h_t	0	1	2	3	4
$\alpha_1 =$	$h_1(x_i)$					
$\alpha_2 =$	$h_2(x_i)$					
$\alpha_3 =$	$h_3(x_i)$					
$\alpha_4 =$	$h_4(x_i)$					
$sign\,(\sum_{t=1}^{4} \alpha_t h_t(x_i))$						

b. Demonstraţi că setul de date de mai sus NU este reprezentabil cu mai puţin de 4 ipoteze „slabe“.

[605] Remarcaţi faptul că definiţia dată aici pentru noţiunea de *ipoteză „slabă“* corespunde bine-cunoscutului *compas de decizie* (engl., decision stump), pentru cazul particular când datele sunt din $\mathbb{R}$.

c. ***Generalizaţi*** **rezultatul obţinut la punctul a, referindu-vă la posibilitatea (sau, dimpotrivă, imposibilitatea) de a *reprezenta* — folosind combinaţii liniare de ipoteze „slabe“ aşa cum au fost definite mai sus — dataset-uri arbitrare [formate din instanţe situate] pe axa reală.**

72. **(O clasă de concepte învăţabile în sens empiric γ-slab cu ajutorul compaşilor de decizie: seturile de instanţe din $\mathbb{R}$, care sunt etichetate în mod consistent)**

Liviu Ciortuz, 2023, pornind de la
■ □ ○ ***Stanford, 2016 fall, A. Ng, J. Duchi, HW2, pr. 6.abc***

Introducere**:** În această problemă, la punctele *a-c* vom demonstra câteva *proprietăţi ale compaşilor de decizie* pentru dataseturi consistente de pe axa reală. Ca o *consecinţă* a acestor proprietăţi, la punctul d vom arăta că dataseturile consistente de pe axa reală constituie *o clasă de concepte învăţabile în sens empiric γ-slab* cu ajutorul compaşilor de decizie.

Fie un set de date etichetate pe axa reală, $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, care este consistent (adică, pentru orice $x_i = x_j$ cu $i \neq j$ rezultă cu necesitate că $y_i = y_j$). Fără a reduce generalitatea, vom presupune că $x_1 \leq x_2 \leq \ldots \leq x_m$. Ba chiar mai mult, pentru a facilita redactarea demonstraţiilor corespunzătoare afirmaţiilor de mai jos, vom presupune că $x_1 < x_2 < \ldots < x_m$.

Unui *prag* $s \in \mathbb{R}$ îi putem asocia doi *compaşi de decizie*, pe care aici îi vom defini folosind următoarele *funcţii*:

$$\phi_{s,+}(x) = \begin{cases} 1 & \text{dacă } x \geq s \\ -1 & \text{dacă } x < s \end{cases} \quad \textbf{şi respectiv} \quad \phi_{s,-}(x) = \begin{cases} -1 & \text{dacă } x \geq s \\ 1 & \text{dacă } x < s. \end{cases}$$

Prin urmare, $\phi_{s,+}(x) = -\phi_{s,-}(x)$ pentru orice $x \in \mathbb{R}$.

Fie $p \overset{not.}{=} (p_1, \ldots, p_m)$ o distribuţie de probabilitate discretă asignată celor m instanţe (în acest context considerate fără etichete): $x_1 \mapsto p_1, \ldots, x_m \mapsto p_m$.

a. Fie un prag oarecare $s \in \mathbb{R}$, fixat. Vă reamintim faptul că *erorile ponderate la antrenare* produse de către compaşii de decizie $\phi_{s,+}$ şi $\phi_{s,-}$ în raport cu distribuţia de probabilitate p sunt definite astfel:

$$error_p(\phi_{s,+}) \overset{def.}{=} \sum_{i=1}^{m} p_i \cdot 1_{\{y_i \neq \phi_{s,+}(x_i)\}}, \text{ respectiv } error_p(\phi_{s,-}) \overset{def.}{=} \sum_{i=1}^{m} p_i \cdot 1_{\{y_i \neq \phi_{s,-}(x_i)\}},$$

unde simbolul $1_{\{\,\}}$ desemnează binecunoscuta *funcţie-indicator*.

Pentru pragul $s \in \mathbb{R}$ considerat mai sus (ales arbitrar, dar fixat) vom defini $m_0(s) \in \{0, 1, \ldots, m\}$ astfel:

$$m_0(s) = \begin{cases} 0, & \textbf{dacă } s < x_1 \\ m, & \textbf{dacă } s \geq x_m \\ i, & \textbf{cu proprietatea } s \in [x_i, x_{i+1}) \textbf{ unde } i \in \{1, \ldots, m-1\}, \\ & \textbf{dacă } x_1 \leq s < x_m. \end{cases}$$

În cele ce urmează vom considera funcţia $f : \{0, 1, \ldots, m\}$ definită prin expresia următoare:[606]

$$f(j) \stackrel{def.}{=} \sum_{i=1}^{j} y_i p_i \; - \sum_{i=j+1}^{m} y_i p_i,$$

cu observaţia că acele sume $\sum_i$ pentru care indicele i ia valori în mulţimea vidă vor fi considerate prin definiţie ca fiind nule (adică, egale cu 0). (De exemplu, $\sum_{i=1}^{0} a_i = 0$ şi $\sum_{i=m+1}^{m} a_i = 0$.)

Vă cerem să demonstraţi că

$$error_p(\phi_{s,+}) = \frac{1}{2} + \frac{1}{2} f(m_0(s)) \quad \text{şi} \quad error_p(\phi_{s,-}) = \frac{1}{2} - \frac{1}{2} f(m_0(s)). \tag{291}$$

b. Demonstraţi că pentru orice $j \in \{1, \ldots, m\}$ are loc proprietatea

$$|f(j) - f(j-1)| = 2p_j \tag{292}$$

şi, în consecinţă, există cel puţin un $j^* \in \{1, \ldots, m\}$ astfel încât

$$|f(j^*)| \geq \frac{1}{m} \quad \text{sau} \quad |f(j^* - 1)| \geq \frac{1}{m}. \tag{293}$$

***Observaţie*: Este posibil ca ambele inegalităţi (293) să aibă loc pentru un acelaşi indice j^*.**

c. Definim $k \in \{0, 1, \ldots, m\}$ astfel:
Dacă în relaţia (293) prima inegalitate are loc, atunci $k = j^*$. Dacă cea de-a doua inegalitate are loc, atunci $k = j^* - 1$. Dacă ambele inegalităţi au loc, atunci putem alege k ca fiind fie j^* fie $j^* - 1$, după cum dorim.

Demonstraţi că

$$\text{dacă } f(k) \leq -\frac{1}{m}, \text{ atunci } error_p(\phi_{s,+}) \leq \frac{1}{2}\Big(1 - \frac{1}{m}\Big), \text{ iar}$$
$$\text{dacă } f(k) \geq \frac{1}{m}, \text{ atunci } error_p(\phi_{s,-}) \leq \frac{1}{2}\Big(1 - \frac{1}{m}\Big),$$

pentru orice $s \in [x_k, x_{k+1})$ în cazul în care $k \in \{1, \ldots, m-1\}$, respectiv pentru orice $s < x_1$ în cazul în care $k = 0$ şi, în sfârşit, pentru orice $s \geq x_m$ în cazul în care $k = m$.[607]

d. [Corolar]
Demonstraţi că dacă rulăm algoritmul AdaBoost pe un astfel de dataset S, atunci putem alege la fiecare iteraţie t un compas de decizie h_t astfel încât eroarea sa ponderată la antrenare să fie de cel mult $\frac{1}{2}\Big(1 - \frac{1}{m}\Big)$. În cosecinţă, datasetul S este *învăţabil în sens empiric γ-slab*, cu garanţia de învăţabilitate $\gamma = \frac{1}{2m}$.

e. Puteţi indica o margine superioară (engl., upper bound) pentru numărul de compaşi de decizie necesari pentru ca algoritmul AdaBoost să obţină eroare la antrenare 0 pe un astfel de set de date de antrenament (S)?

[606] Pentru a fi şi mai pedanţi, ar trebui să folosim notaţia f_p, ca să arătăm că f depinde de distribuţia p, însă nu vrem sa complicăm prea mult formalismul.

[607] *Observaţie*: În cele ce urmează, vom putea ignora cazul $s \geq x_m$, fiindcă $error_p(\phi_{s,+}) = error_p(\phi_{s',-})$ şi $error_p(\phi_{s,-}) = error_p(\phi_{s',+})$ pentru orice $s' < x_1$ şi, în consecinţă, la punctul d ne vom putea limita la a alege doar compaşi de decizie exteriori situaţi la stânga lui x_1.

73. **(Algoritmul AdaBoost poate să producă overfitting)**

prelucrare de Liviu Ciortuz, 2021, după
□ • ○ ∗ ***CMU, 2011 fall, Eric Xing, HW5, pr. 3.2.cd***

Spunem că un sistem de clasificare automată este supra-specializat (sau, că produce overfitting) atunci când acurateţea sa este din ce în ce mai bună la antrenare, dar din ce în ce mai slabă la testare. Aşadar, el ajunge să modeleze / explice chiar şi detaliile accidentale („zgomotele“) din datele de antrenament, în loc să generalizeze, făcând abstracţie de ele.

Formalizând această noţiune, Tom Mitchell afirmă în cartea sa *Machine Learning* (la pag. 67) că o ipoteză h produsă de către un algoritm de clasificare automată este supra-specializată pe datele de antrenament dacă există o altă ipoteză h' astfel încât

$$error_{train}(h) > error_{train}(h'), \text{ dar } error_{test}(h) < error_{test}(h'). \tag{294}$$

În general, algoritmul AdaBoost nu suferă de overfitting. Există totuşi situaţii în care se întâmplă ca el să producă overfitting, şi anume atunci când datele de antrenare şi de testare sunt foarte mixate (altfel spus, ele conţin mult „zgomot“).[608]

a. Creaţi un set de date de antrenament şi un set de date de test pe care să arătaţi că într-adevăr algoritmul AdaBoost produce overfitting (conform relaţiei (294)). Faceţi graficele funcţiilor de eroare la antenare şi respectiv la testare în raport cu numărul (T) de iteraţii executate de AdaBoost.[609]

b. Identificaţi o strategie prin aplicarea căreia să poată fi contracarat / limitat fenomenul de overfitting la algoritmul AdaBoost.

74. **(Algoritmul AdaBoost folosind “confidence rated classifiers” în locul clasificatorilor „slabi“ simpli)**

• ○ ***MIT, 2001 fall, Tommi Jaakkola, HW3, pr. 1.1-3***

Fie $D = \{(x_1, y_1) \ldots, (x_m, y_m)\}$ un set de exemple de antrenament, cu $x_i \in \mathbb{R}^d$ şi $y_i \in \{-1, +1\}$. Ca şi în formularea clasică a algoritmului AdaBoost,[610] **vom nota „ponderile“ / probabilităţile asociate exemplelor de antrenament la începutul iteraţiei k a cestui algoritm cu $D_k(1), \ldots, D_k(m)$, unde $\sum_{i=1}^{m} D_k(i) = 1$. De asemenea, *presupunem* că există o metodă [necunoscută nouă] care produce o ipoteză „slabă“ h ca răspuns la un set de antrenament ponderat. Această ipoteză „slabă“ trebuie să fie un clasificator care să se comporte măcar un pic mai bine decât datul cu banul (engl., random guessing) pe setul de antrenament ponderat pe care a fost antrenat. Nu facem nicio presupunere cu privire**

[608] CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, midterm, pr. 11.3.

[609] *Indicaţie*: Există un applet interesant scris de Yoav Freund (http://cseweb.ucsd.edu/∼yfreund/adaboost/), care vă permite să vă creaţi propriile seturi de date de antrenament şi de testare în planul euclidian, să antrenaţi AdaBoost iar apoi să-l testaţi. Puteţi să faceţi apoi un screen-shot care să includă atât datele de antrenament şi datele de test cât şi graficele curbelor de eroare la antrenare şi respectiv la testare.

Alternativ, puteţi găsi un set de date pentru care zonele de decizie determinate de către AdaBoost să fie deja cunoscute. (Vedeţi de exemplu pr. 24.) Apoi plasaţi punctele de test în aşa fel încât o parte dintre ele să contravină semnului zonei respective (aşa cum fusese el stabilit de către AdaBoost în urma antrenării).

[610] Vedeţi problema 22.

la cât de bine se comportă acest clasificator „slab“ pe eventuale alte seturi de antrenament ponderate.

Spre deosebire de cazul algoritmului AdaBoost clasic, aici vom considera că ipoteza h produce ca output valori reale. (Din această cauză, astfel de ipoteze se numesc în limba engleză *confidence rated classifiers*.) Semnul output-ului $h(x)$ indică eticheta (± 1), iar modulul output-ului specifică gradul de „încredere“ (engl., confidence) în decizia luată de clasificator. De *exemplu*, output-ul $h(x) = 10$ ar putea fi interpretat ca fiind o predicţie a etichetei $+1$ cu o încredere destul de mare.

Pornind de la distribuţia uniformă $D_1(i) = 1/m$, AdaBoost generează o secvenţă de ipotheze $h_1, \ldots, h_t$, fiecare ipoteză h_k fiind antrenată utilizând un anumit set de ponderi asociate exemplelor. După ce, în cadrul iteraţiei k, a fost generată a ipoteza $h_k(x)$ ca „răspuns“ la ponderile $D_k(i)$, ponderile asociate exemplelor vor fi actualizate folosind relaţia următoare:

$$D_{k+1}(i) = c \cdot D_k(i) \exp(-\hat{\alpha}_k \, y_i \, h_k(x_i)), \; i = 1, \ldots, m, \tag{295}$$

unde c este *constanta de normalizare* care asigură faptul că $\sum_{i=1}^{n} D_{k+1}(i) = 1$, iar $\hat{\alpha}_k$ este *votul* asignat noii componente a clasificatorului, h_k. Vrem să asignăm aceste voturi astfel încât h_k să satisfacă relaţia următoare:

$$\sum_{i=1}^{m} D_{k+1}(i) \, y_i \, h_k(x_i) = 0, \tag{296}$$

ceea ce înseamnă că la iteraţia $k+1$ suma produselor dintre ponderi şi încrederi ($D_{k+1}(i) h_k(x_i)$) pentru exemplele corect clasificate (adică, acei x_i pentru care $y_i \, h_k(x_i) > 0$) de către ipoteza „slabă“ h_k este egală cu suma produselor dintre ponderi şi încrederi pentru exemplele incorect clasificate ($y_i \, h_k(x_i) < 0$).

a. Arătaţi că dacă ipoteza „slabă“ h_k generează doar output-uri binare (± 1), atunci *condiţia* (296) revine la a cere ca eroarea ponderată la antrenare a lui h_k calculată în raport cu noile ponderi ($D_{k+1}(i)$) să fie exact 0.5.[611]

***Comentariu*: Întrucât ipoteza „slabă“ h_t poate genera ca output valori reale, situaţia este sensibil diferită în cazul de faţă. Putem vedea *condiţia* (296) ca pe un mod de a *de-corela* predicţiile $h_k(x_i)$ de etichetele y_i, ţinând cont de noile ponderi $D_{k+1}(i)$.**

b. Arătaţi că dacă asignăm lui $\hat{\alpha}_k$ acea valoare a argumentului α pentru care se atinge minimul funcţiei[612]

$$J_k(\alpha) = \ln \left(\sum_{i=1}^{m} D_k(i) \cdot \exp(-\alpha \, y_i \, h_k(x_i)) \right), \tag{297}$$

atunci în mod necesar *condiţia de de-corelare* (296) este satisfăcută pentru noile ponderi.

***Sugestie*: Nu este cazul să încercaţi să calculaţi $\hat{\alpha}_k$. În schimb, egalaţi derivata**

[611] LC: Prin urmare, relaţia (296) reprezintă o generalizare a rezultatului care a fost demonstrat la problema 22.*vi*.

[612] LC: Observaţi că $J_k(\alpha)$ este un log-cost ponderat cu ajutorul probabilităţilor D_k. Funcţia logaritm natural (ln) este strict crescătoare, deci a minimiza un cost este echivalent cu a minimiza log-costul corespunzător.

funcţiei $J_k(\alpha)$ din relaţia (297) cu zero şi folosiţi ecuaţia care se obţine (şi pe care $\hat{\alpha}_k$ trebuie să o satisfacă), pentru a arăta că relaţia (296) este satisfăcută.

c. *Comentariu*: Rezultatul care a fost obţinut la punctul b pare să fie un pic suspicios... ca şi când ar exista o funcţie obiectiv pe care algoritmul nostru de tip boosting ar urmări să o minimizeze la fiecare iteraţie. La acest punct veţi arăta că într-adevăr aşa este.

Fie f_t ipoteza combinată care rezultă în urma efectuării a t iteraţii de boosting:

$$f_t(x) = \hat{\alpha}_1 h_1(x) + \ldots + \hat{\alpha}_t h_t(x).$$

Va trebui să arătaţi că funcţia

$$J(f_t) = \ln\left(\sum_{i=1}^{m} \exp(-y_i\, f_t(x_i))\right)$$

joacă rolul de *funcţie obiectiv* pentru noul nostru algoritm de boosting. Cu alte cuvinte, veţi arăta că de fiecare dată când adăugăm o nouă componentă (h) la clasificatorul combinat $f_t(x)$, valoarea funcţiei obiectiv descreşte.

Aşadar, demonstraţi că $J(f_t) \geq J(f_{t+1})$, unde

$$f_{t+1}(x) = f_t(x) + \hat{\alpha}_{t+1}\, h_{t+1}(x),$$

$h_{t+1}(x)$ este noua ipoteză „slabă“, iar $\hat{\alpha}_{t+1}$ a fost ales (adică, a fost optimizat) după cum am indicat mai sus.

***Sugestie*: Calculaţi ponderile $D_t(i)$ într-un mod similar cu cel în care am procedat la problema 23.a şi ţineţi cont de faptul că $\hat{\alpha}_{t+1}$ minimizează funcţia $J_k(\alpha)$ care a fost definită prin relaţia (297).**

75. **(Algoritmul AdaBoost, ca instanţă a unui algoritm mai general)**

• ○ ***CMU, 2012 fall, E. Xing, A. Singh, HW4, pr. 3.ab***

Considerăm următorul algoritm de clasificare *etapizat, aditiv* (engl., [forward] *stagewise additive* modelling), A:

Intrare: $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$, T, $\mathcal{H}$, ϕ, unde
T este numărul de iteraţii de executat,
$\mathcal{H}$ este o mulţime de „ipoteze“,
$\phi(y, y')$ este o funcţie de „cost“ / „risc“;

Procedură:

Initialize the classifier by taking $f_0(x) = 0$ (the constant function 0)
for $t = 1$ to T do:
1. Compute
$(h_t, \alpha_t) = \arg\min_{\alpha\in\mathbb{R}, h\in\mathcal{H}} J_t(\alpha, h)$
where $J_t(\alpha, h) \overset{not.}{=} \frac{1}{m}\sum_{i=1}^{m} \phi(y_i, f_{t-1}(x_i) + \alpha h(x_i))$
2. Update the classifier
$f_t(x) = f_{t-1}(x) + \alpha_t h_t(x)$
end for
return the classifier *sign* $(f_T(x))$

Intuitiv, la fiecare pas algoritmul A adaugă în mod *greedy* câte o ipoteză $h \in \mathcal{H}$ la ipoteza curentă pentru a minimiza costul / riscul definit de funcţia ϕ.

a. Cum ar trebui să definim $\phi(y, y')$ pentru ca algoritmul A să fie echivalent cu algoritmul AdaBoost?

b. Demonstraţi că într-adevăr, atunci când ϕ este funcţia pe care aţi indicat-o la punctul a, algoritmul AdaBoost este echivalent cu algoritmul A.

***Sugestie*: Considerând α fixat (la o valoare pozitivă, arbitrar aleasă), determinaţi acel $h_t \in \mathcal{H}$ care va minimiza costul / riscul $J_t(\alpha, h)$. (Atenţie: h_t nu este o funcţie de α.) Apoi, considerând această ipoteză h_t fixată, găsiţi acea valoare α_t (a lui α) care va minimiza costul / riscul $J_t(\alpha, h)$. Gândiţi-vă de asemenea care este relaţia dintre probabilităţile $D_t(i)$ din algoritmul AdaBoost şi algoritmul A.**

76. **(Algoritmul AdaBoost generalizat [în raport cu funcţia de cost]: aplicare; ilustrarea echivalenţei care a fost demonstrată la pr. 29.a)**

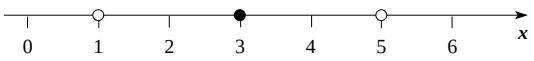
□ ● · ***Liviu Ciortuz, 2020***

La problema 29 am prezentat algoritmul AdaBoost generalizat, care poate fi parametrizat cu orice funcţie de cost / „loss“ / pierdere. La punctul a de la acea problemă am demonstrat că, la fiecare iteraţie t a algoritmului AdaBoost generalizat, identificarea celui mai bun compas de decizie (adică, minimizarea erorii ponderate la antrenare, ε_t) este echivalentă cu minimizarea criteriului

$$J_t(\alpha, \theta) = \frac{1}{m} \sum_{i=1}^{m} \text{Loss}(y_i f_{t-1}(x_i) + y_i \, \alpha \, h(x_i; \theta))$$

în raport cu parametrul θ, unde Loss este funcţia de cost cu care lucrăm.

Ne propunem ca — în cea mai mare parte a acestui exerciţiu — să ilustrăm în mod *practic* acest lucru, pe date concrete. Aşadar, vom verifica faptul că într-adevăr are loc echivalenţa despre care am vorbit mai sus atunci când i. aplicăm algoritmul AdaBoost generalizat folosind funcţia de cost [negativ] exponenţială pe datele de la problema 25 (ne referim la prima soluţie dată acolo),[613] respectiv ii. atunci când aplicăm algoritmul AdaBoost generalizat folosind funcţia de cost logistică (la punctul c) pe acelaşi set de date:



***Atenţie*! Vom folosi compaşi de decizie, *inclusiv* compaşi cu prag *exterior* (engl., outside threshold) în raport cu datele de antrenament.**

Prin urmare, ne vom uita la *valorile* pe care le ia *expresia* (264) în cursul rulării acestui algoritm pe datele pe care tocmai le-am menţionat, adică derivata

[613] La punctul b al problemei 29 am demonstrat (din punct de vedere *teoretic*) că atunci când se lucrează folosind funcţia de cost [negativ] exponenţială, respectiva „instanţă“ a algoritmului AdaBoost generalizat coincide cu algoritmul AdaBoost prezentat la problema 22.

parţială a funcţiei $J_t(\alpha,\theta)$ calculată pentru $\alpha=0$ şi pentru θ (deci o funcţie de parametrul θ):

$$\begin{aligned}\frac{\partial}{\partial\alpha}J_t(\alpha,\theta)_{|\alpha=0} &= \frac{1}{m}\sum_{i=1}^{m}\frac{\partial}{\partial\alpha}\text{Loss}(y_i f_{t-1}(x_i)+y_i\,\alpha\,h(x_i;\theta))_{|\alpha=0}\\ &= \frac{1}{m}\sum_{i=1}^{m}dL(y_i f_{t-1}(x_i))\,y_i\,h(x_i;\theta),\end{aligned}$$

unde $dL(z)\overset{not.}{=}\frac{\partial}{\partial z}\text{Loss}(z)$.

a. Arătaţi că $\frac{\partial}{\partial\alpha}J_t(\alpha,\theta)_{|\alpha=0}$ **are la iteraţia** $t=1$ **expresia**

$\frac{\partial}{\partial\alpha}J_1(\alpha,\theta)_{\|\alpha=0}=\begin{cases}-1/3 & \forall\theta\in(-\infty,1)\cup[3,5)\\ +1/3 & \forall\theta\in[1,3)\cup[5,+\infty)\end{cases}$	$\frac{\partial}{\partial\alpha}J_1(\alpha,\theta)_{\|\alpha=0}=\begin{cases}+1/3 & \forall\theta\in(-\infty,1)\cup[3,5)\\ -1/3 & \forall\theta\in[1,3)\cup[5,+\infty)\end{cases}$
dacă semnele asociate compasului de decizie (h_1) sunt $+\|-$;	**dacă semnele asociate compasului de decizie (h_1) sunt $-\|+$.**

***Notă*: Exact acestei expresii îi corespunde graficul care a fost prezentat la pagina 573.**

Verificaţi apoi că $D_2(i)=\tilde{W}_i^{(2)}$ **pentru** $i\in\{1,2,3\}$.[614]

***Observaţii*:**

1. Vă readucem aminte că parametrul θ desemează *pe lângă* pragul compasului de decizie *şi* semnele asociate acestui compas de decizie. (Aceste semne sunt $+|-$ pentru graficul din stânga şi respectiv $-|+$ pentru graficul din dreapta.)

2. Valorile lui θ pentru care se obţine minimul lui $\frac{\partial}{\partial\alpha}J_t(\alpha,\theta)_{|\alpha=0}$ sunt eligibile de către algoritmul AdaBoost la iteraţia curentă.[615] Astfel, la iteraţia $t=1$, după cum se observă în graficul de la pagina 573, putem alege $\theta\in(-\infty,1)\cup[3,5)$ cu semnele asociate $+|-$ sau, alternativ, $\theta\in[1,3)\cup[5,+\infty)$ cu semnele asociate $-|+$. Observaţi că una dintre aceste posibilităţi, şi anume, pragul $\theta=0$ cu semnele sociate $+|-$, a fost ales la rezolvarea problemei 25, şi anume prima soluţie.

b. Procedând similar, calculaţi expresiile $\frac{\partial}{\partial\alpha}J_t(\alpha,\theta)_{|\alpha=0}$ pentru iteraţiile $t=2$ şi $t=3$ şi apoi obţineţi graficele corespunzătoare.

***Notă*: În fiecare din cele două cazuri veţi presupune că la iteraţia / iteraţiile precedente s-au selectat compaşii de decizie exact ca la rezolvarea problemei 25, prima soluţie.**

c. Aplicaţi algoritmul AdaBoost generalizat folosind funcţia de cost logistică pe datele de la problema 25; executaţi 3 iteraţii, utilizând inclusiv compaşi de decizie exteriori. Pentru calcularea voturilor α_t, veţi folosi formula care a fost dedusă la pr. 29.g, adică $\alpha_t=\ln\sqrt{\frac{1-\varepsilon_t}{\varepsilon_t}}$. La final veţi compara rezultatele obţinute aici cu cele care au fost prezentate la rezolvarea problemei 25.

[614] Vedeţi rezolvarea problemei 25, prima soluţie.

[615] Mai general, se poate alege pentru θ o valoare pentru care expresia $\frac{\partial}{\partial\alpha}J_t(\alpha,\theta)_{|\alpha=0}$ este negativă. Aici însă, ca şi la problema 25, nu folosim această posibilitate.

77. **(Algoritmul AdaBoost multi-class: fundamentare teoretică)**

□ · ***Liviu Ciortuz, 2023, pornind de la articolul Multi-class AdaBoost, by J. Zhu, H. Zou, S. Rosset, T. Hastie, 2006***[616]

În această problemă vom arăta cum este posibil să generalizăm algoritmul AdaBoost clasic, în aşa fel încât să realizăm clasificare n-ară[617] Această generalizare va implica adaptarea funcţiei de cost / pierdere negativ exponenţiale[618] — pe care algoritmul AdaBoost clasic o foloseşte pentru a realiza clasificare binară — la cazul clasificării n-are:

$$L(y, f) = \exp\Big(-\frac{1}{K}(y^{(1)}f^{(1)} + \ldots + y^{(K)}f^{(K)})\Big) = \exp\Big(-\frac{1}{K}y^\top f\Big),$$

unde $f \overset{not.}{=} (f^{(1)}, \ldots, f^{(K)})^\top$, $f^{(k)}$ corespunde clasei k, iar $f^{(1)} + \ldots + f^{(K)} = 0$ (aceasta este numită *restricţia de simetrie*),[619] şi $y \overset{not.}{=} (y^{(1)}, \ldots, y^{(K)})^\top$, cu

$$y^{(k)} = \begin{cases} 1, & \text{când } y \text{ codifică clasa } k, \\ -\dfrac{1}{K-1}, & \text{în cazul contrar.} \end{cases}$$

Noul algoritm va fi formulat în maniera optimizării secvenţiale.[620]

0. Input $(x_1, y_1), \ldots, (x_n, y_n)$, with $x_i \in \mathbb{R}^p$ and $y_i \in \mathcal{Y}$, for $i = 1, \ldots, n$.
1. Initialize $f_0(x) = 0$.
2. For $m = 1$ to M:
   (a) Compute
   $(\beta_m, g_m) = \arg\min_{\beta, g} \sum_{i=1}^n L(y_i, f_{m-1}(x_i) + \beta g(x_i))$;
   (b) Set
   $f_m(x) = f_{m-1}(x_i) + \beta_m g_m(x)$.
3. Output
   $\text{argmax}_{k \in \{1,\ldots,K\}}(f_m^{(1)}(x), \ldots, f_m^{(K)}(x))$.

Mulţimea $\mathcal{Y} \subset \mathbb{R}^K$ este constituită din „codificările" claselor / etichetelor $1, \ldots, K$:

$$\mathcal{Y} = \Big\{\Big(1, -\frac{1}{K-1}, -\frac{1}{K-1}, \ldots, -\frac{1}{K-1}\Big)^\top,$$
$$\Big(-\frac{1}{K-1}, 1, -\frac{1}{K-1}, \ldots, -\frac{1}{K-1}\Big)^\top,$$
$$\ldots$$
$$\Big(-\frac{1}{K-1}, -\frac{1}{K-1}, \ldots, -\frac{1}{K-1}, 1\Big)^\top\Big\}.$$

Funcţiile g_m (deci şi g) sunt definite pe $\mathbb{R}^p$ şi iau valori în mulţimea $\mathcal{Y}$.

***Observaţie*: Pasul (2a) din acest algoritm se poate scrie în mod echivalent astfel:**

$$(\beta_m, g_m) = \arg\min_{\beta, g} \sum_{i=1}^n w_i \exp\Big(-\frac{1}{K}\beta y_i^\top g(x_i)\Big), \tag{298}$$

[616] Acest articol a fost publicat într-o versiune revizuită şi extinsă în revista *Statistics and its Interface*, 2009, vol. 2, pag. 349-360.
[617] Engl., multi-class classification.
[618] Vedeţi problemele 23 şi 26.
[619] Observaţi că în cazul $K = 2$ acest cost negativ exponenţial coincide cu costul negativ exponenţial de la clasificare binară.
[620] Această strategie este numită "stagewise additive modeling" în terminologia [de limbă engleză] folosită de autorii acestui algoritm, care sunt statisticieni.

unde

$$w_i = \exp\Big(-\frac{1}{K} y_i^\top f_{m-1}(x_i)\Big)$$

sunt ponderile nenormalizate care au fost calculate la iteraţia $m-1$ pentru instanţele / „observaţiile“ x_i.

a. Arătaţi că[621]
***i*. fixând β, a minimiza funcţia obiectiv din partea dreaptă a relaţiei** (298) **revine la a afla funcţia g care minimizează eroarea ponderată $\sum_{i=1}^n w_i \cdot 1_{\{y_i \neq g(x_i)\}}$.**[622]
***ii*. fixând g, a minimiza funcţia obiectiv din partea dreaptă a relaţiei** (298) **revine la a seta β la valoarea**

$$\beta_m = \frac{(K-1)^2}{K}\Big[\underbrace{\ln\frac{1-\varepsilon_m}{\varepsilon_m} + \ln(K-1)}_{not.:\alpha_m}\Big], \tag{299}$$

unde

$$\varepsilon_m = \sum_{i=1}^n w_i \cdot \underbrace{1_{\{y_i \neq g_m(x_i)\}}}_{1_{\{c_i \neq T_m(x_i)\}}} \Big/ \sum_{i=1}^n w_i.$$

b. Vom (re)nota cu $w_{i,m}$ ponderea nenormalizată calculată pentru instanţa / „observaţia“ x_i la iteraţia m, Arătaţi că regula de actualizare a acestor ponderi nenormalizate w se poate scrie astfel:

$$w_{i,m} = w_{i,m-1} \cdot \exp\Big(-\frac{1}{K}\beta_m y_i^\top g_m(x_i)\Big) \tag{300}$$

$$= \begin{cases} w_{i,m-1} e^{-\frac{K-1}{K}\alpha_m} & \textbf{dacă } y_i = g_m(x_i), \\ w_{i,m-1} e^{\frac{1}{K}\alpha_m} & \textbf{dacă } y_i \neq g_m(x_i) \end{cases} \tag{301}$$

$$= w_{i,m-1} \cdot \exp\Big(-\frac{(K-1)^2}{K^2}\alpha_m y_i^\top g_m(x_i)\Big). \tag{302}$$

c. Considerăm următorul pseudo-cod (algoritmul SAMME):

0. Input $(x_1, c_1), \ldots, (x_n, c_n)$, with $x_i \in \mathbb{R}^p$ and $c_i \in \{1, \ldots, K\}$, for $i = 1, \ldots, n$.
1. Initialize the observation weights $w_i = 1/n$, $i = 1, 2, \ldots, n$.
2. For $m = 1$ to M:
   (a) Using weights w_i, find the „best“ classifier $T_m(x)$ of the training data w.r.t. the weighted training error, which is defined as follows:
   (b) $\varepsilon_m = \sum_{i=1}^n w_i \cdot 1_{\{c_i \neq T_m(x_i)\}}$;
   (c) Compute
   $\alpha_m = \ln\frac{1-\varepsilon_m}{\varepsilon_m} + \ln(K-1)$;
   (d) Set
   $w_i \leftarrow w_i \cdot \exp\Big(\alpha_m \cdot 1_{\{c_i \neq T_m(x_i)\}}\Big)$
   for $i = 1, \ldots, n$;
   (e) Re-normalize w_i.
3. Output
   $C(x) = \text{argmax}_{k \in \{1,\ldots,K\}} \sum_{j=1}^m \alpha_j \cdot 1_{\{T_j(x)=k\}}$.

[621] Punctele *i* şi *ii* care urmează să fie formulate [în enunţ] constituie *Lema 1* din articolul *Multi-class AdaBoost* de J. Zhu, H. Zou, S. Rosset, T. Hastie, 2009.

[622] Corespondentul funcţiei g în algoritmul SAMME care va fi introdus la punctul c este ipoteza „slabă“ / arborele T. Erorii ponderate $\sum_{i=1}^n w_i \cdot 1_{\{y_i \neq g(x_i)\}}$ îi va corespunde în SAMME expresia $\sum_{i=1}^n w_i \cdot 1_{\{c_i \neq T(x_i)\}}$.

Veţi arăta că algoritmul SAMME este echivalent cu precedentul algoritm (de tip *forward stagewise additive modeling*) atunci când acesta utilizează funcţia de cost / pierdere negativ exponenţială multi-class.[623]

Concret, arătaţi că
i. **după normalizare, ponderile** (302) **coincid cu ponderile obţinute la pasul (2e) din algoritmul SAMME;**
ii. **regulile de decizie ale celor doi algoritmi coincid, adică are loc următoarea egalitate:**

$$\underset{k}{\operatorname{argmax}}(f_m^{(1)}(x), \ldots, f_m^{(K)}(x)) = \underset{k}{\operatorname{argmax}} \sum_{m=1}^{M} \alpha_m \cdot 1_{\{T_m(x)=k\}}.$$

***Sugestie*: Pentru a demonstra cerinţele de la acest punct al problemei, ţineţi cont de faptul că există o corespondenţă bijectivă între funcţiile $g : \mathbb{R}^p \to \mathcal{Y}$ de la primul algoritm din enunţ şi clasificatorii multi-class T de la cel de-al doilea algoritm (SAMME):**[624]

$$T(x) = k \text{ dacă } g^{(k)}(x) = 1$$

şi invers:

$$g^{(i)}(x) = \begin{cases} 1 & \text{dacă } T(x) = k, \\ -\dfrac{1}{K-1} & \text{dacă } T(x) \neq k. \end{cases}$$

78. **(Algoritmul AdaBoost: Adevărat sau Fals?)**

□ • ○ ***CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, final, pr. 8.f***
a. Eroarea la antrenare produsă de clasificatorul H_T obţinut de AdaBoost — bazat pe combinaţia liniară determinată de ipotezele „slabe“ — descreşte monoton pe măsură ce creşte numărul de iteraţii executate de algoritm.

MIT, 2002 fall, Tommi Jaakkola, midterm, pr. 5.3
b. Ponderile / „voturile“ α_t asignate de către AdaBoost ipotezelor „slabe“ h_t au tendinţa să scadă în timpul execuţiei algoritmului, pentru că eroarea ponderată produsă la antrenare (engl., weighted training error) de ipotezele „slabe“ în general creşte.

MIT, 2001 fall, Tommi Jaakkola, midterm, pr. 4.2
c. Ponderile / „voturile“ α_t pe care algoritmul AdaBoost le asignează ipotezelor „slabe“ h_t sunt optimale, în sensul că ele asigură o *acurateţe la antrenare* mai bună decât în cazul oricăror altor valori asignate acestor voturi / ponderi α_t.

CMU, 2014 spring, B. Poczos, A. Singh, midterm, pr. 1.8
d. Folosind în algoritmul AdaBoost compaşi de decizie (cu rol de ipoteze „slabe“), este posibil să obţinem graniţe de decizie sub formă de parabolă (determinate, deci, de polinoame de ordinul al doilea).

MIT, 2001 fall, Tommi Jaakkola, final, pr. 1.4
e. Unul dintre avantajele algoritmului AdaBoost este că nu produce niciodată supra-specializare (engl., overfitting).

[623] Numele SAMME este acronim pentru *Stagewise Additive Modeling with Multi-class Exponential loss.*
[624] Notăm $g(x)$ cu $(g^{(1)}(x), \ldots, g^{(K)}(x))$

CMU, 2010 fall, Aarti Singh, midterm, pr. 1.7

f. Învăţăm un clasificator f folosind algoritmul AdaBoost cu ipoteze „slabe“ h. Forma funcţională a separatorului decizional determinat de f este aceeaşi cu forma funcţională a ipotezelor h, însă cu parametri diferiţi. De exemplu, dacă ipotezele h au fost clasificatori liniari, atunci şi f este un clasificator liniar.



© *M. Romanică*

5 Maşini cu vectori-suport

Sumar

Noţiuni preliminare

– **elemente [simple] de *calcul vectorial*; proprietăţi elementare ale produsului scalar al vectorilor din $\mathbb{R}^n$, norma euclidiană (L_2) şi norma L_1 în $\mathbb{R}^n$: ex. 1, ex. 35;**

– **elemente [simple] de *geometrie analitică*: ecuaţia unei drepte din planul euclidian, ecuaţia unui plan din $\mathbb{R}^3$, ecuaţia unui hiper-plan din $\mathbb{R}^n$; ecuaţia dreptei care trece prin două puncte date în planul euclidian: ex. 5.c; panta unei drepte perpendiculare pe o dreaptă dată: ex. 8.d, ex. 36, ex. 38;**

– **distanţa (cu sau fără semn) de la un punct la o dreaptă (respectiv la un plan, sau mai general la un hiperplan): ex. 1, ex. 5, ex. 6.c;**

– ***separabilitate* liniară, separator optimal, *margine* geometrică: ex. 6.abc, ex. 8, ex. 38.**

– **proprietăţi de bază din *calculul matriceal*;**

– **calculul *derivatelor parţiale* [pentru funcţii obţinute prin compuneri de funcţii elementare];**

– ***metoda lui Lagrange* pentru rezolvarea problemelor de *optimizare convexă cu restricţii*:**
ex. 35 din prezentul capitol, precum şi
ex. 82, ex. 83, ex. 84, ex. 85, ex. 86, ex. 87, ex. 174, ex. 175, ex. 176, ex. 177, ex. 180, ex. 181 de la capitolul de *Fundamente*;

– ***funcţii-nucleu*: vedeţi secţiunea corespunzătoare de la capitolul de *Fundamente*.**

SVM cu margine "hard"

– **(•) deducerea *formei primale* pentru problema SVM (cu margine "hard"), pornind de la principiul maximizarii marginii geometrice: ex. 2;** [625]

– **(P0) o formă [simplă] echivalentă cu *forma primală* a *problemei de optimizare SVM*: ex. 7;**

– ***exemplificarea* identificării *separatorului optimal* şi a *vectorilor-suport*, pornind de la condiţiile din forma primală: ex. 3-5, ex. 36, ex. 38, ex. 39 şi CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW4, ex. 4.1-4;**

– **calcularea *erorii* la cross-validare "leave-one-out" atunci când se foloseşte o SVM liniară cu margine "hard": ex. 4.d, ex. 37;**

– **exemple de [funcţii de] *mapare a atributelor*, cu scopul de a obţine separabilitate liniară în spaţiul de trăsături: ex. 6.d, ex. 8, ex. 40.bd, ex. 41.a;**
rezolvarea directă a problemei SVM primale în [noul] spaţiu de trăsături; identificarea separatorului neliniar din spaţiul iniţial [de trăsături]: ex. 8.de, ex. 40.ce, ex. 41.b-e;

[625] Vedeţi şi Andrew Ng (Stanford), Lecture Notes, part V, section 3.

– **(•) deducerea *formei duale* pentru problema SVM cu margine "hard": ex. 9;**

– ***exemplificarea* a două modalităţi de găsire a soluţiei formei duale a problemei SVM cu margine "hard" pentru învăţarea unui concept [reprezentat de un set de date de antrenament neseparabil în spaţiul „iniţial" de trăsături], folosind o funcţie de mapare Φ dată: prin optimizare directă (ex. 10, ex. 42), respectiv prin folosirea relaţiilor de legătură cu soluţia problemei primale: ex. 11;**

– **(P1) dacă datele de antrenament sunt liniar separabile, atunci problema de optimizare SVM în formă primală are soluţie unică, întrucât funcţia obiectiv a acestei probleme este strict convexă: ex. 2.a (vedeţi *Observaţia* de acolo).**

– **(P2) efectul *multiplicării valorilor atributelor* cu o constantă pozitivă asupra separatorului obţinut de SVM), atunci când datele de antrenament sunt liniar separabile cu ajutorul unui hiperplan care trece prin originea sistemului de coordonate: ex. 27.a;**[626]

– **(P3) efectul unui *atribut irelevant* — în sensul că nu afectează satisfacerea *restricţiilor* de separabilitate liniară a datelor de antrenament şi, în plus, nu măreşte marginea de separare — asupra rezultatelor clasificatorului SVM (şi respectiv C-SVM): ex. 49, ex. 59;**

– **vedeţi proprietăţile (P6) şi (P8) enunţate mai jos (la secţiunea despre C-SVM), care sunt valabile şi în cazul SVM [cu margine "hard"];**

– ***comparaţii* între SVM [având, eventual, diferite funcţii-nucleu] şi alţi clasificatori: ex. 45, ex. 46, ex. 59; vedeţi şi ex. 12 de la capitolul *Învăţare bazată pe memorare*.**

SVM cu margine "soft" (C-SVM):

– **(•) deducerea *formei duale* pentru problema SVM cu margine "soft" (C-SVM): ex. 12.a-d;**

– **(P4) legătura dintre valorile (şi intervalele de valori) pentru multiplicatorii Lagrange α_i pe de o parte, şi valorile (şi intervalele de valori) pentru *marginile funcţionale* $y_i(\bar{w} \cdot x_i + \bar{w}_0)$ pe de altă parte: ex. 12.e; exemplificare la ex. 13.b;**

– **(P5) exprimarea / calcularea distanţei geometrice de la un vectori-suport x_i pentru care $\bar{\alpha}_i = C$ la hiperplanul-margine corespunzător etichetei y_i, cu ajutorul variabilei de „destindere" ξ_i: ex. 13.a;**

– ***exemplificarea* noţiunilor de bază: ex. 47, ex. 48 şi CMU, 2008 fall, Eric Xing, final, ex. 2.2;**
un *exemplu* de calculare a valorii optime pentru funcţia obiectiv a problemei de optimizare C-SVM: ex. 13.b;

– **exemplificarea poziţionării separatorului optimal determinat de C-SVM (pentru diferite valori ale parametrului C), în prezenţa unui outlier: ex. 15;**

– **exemplificarea efectului pe care îl are creşterea valorii parametrului de „destindere" C [asupra marginii şi asupra excepţiilor la clasificare]: ex. 16, CMU, 2010 fall, Aarti Singh, HW3, ex. 3.2;**

[626] Rezultatul *nu* se menţine şi în cazul C-SVM: ex. 27.b.

– **(P1′) spre deosebire de SVM (vedeţi proprietatea (P1)), în cazul C-SVM *unicitatea* soluţiei problemei de optimizare C-SVM în forma primală *nu este garantată* [chiar şi] atunci când datele sunt liniar separabile: ex. 17 furnizează un *exemplu* de situaţie în care forma duală a problemei de optimizare C-SVM are soluţie unică, dar forma sa primală *nu* are soluţie unică;**

– **(P6) o proprietate pentru C-SVM (dar şi pentru SVM): ex. 14.**
Dacă în setul de date de antrenament două trăsături (engl., features) sunt duplicate ($x_{ij} = x_{ik}$ pentru $i = 1, \ldots, m$), atunci ele vor primi ponderi identice ($\bar{w}_j = \bar{w}_k$) în soluţia optimală calculată de clasificatorul [C-]SVM;

– **(P7) o *margine superioară* (engl., upper bound) pentru numărul de *erori* comise la *antrenare* de către C-SVM: ex. 18;**

$$\mathbf{err}_{train}\textbf{(C-SVM)} \leq \frac{1}{m} \sum_i \xi_i$$

– **(P8) o proprietate pentru C-SVM (dar şi pentru SVM):**
La CVLOO numai vectorii-suport pot fi (eventual!) clasificaţi eronat: ex. 19; aşadar avem [şi] o *margine superioară* pentru numărul de *erori* comise la *CVLOO* de către C-SVM:

$$\mathbf{err}_{CVLOO}\textbf{(C-SVM)} \leq \frac{\#SVs}{m}$$

– **(P9) chestiuni legate de *complexitatea computaţională* privind clasificatorul C-SVM: ex. 56;**

– **(•) deducerea *formei duale* pentru problema SVM cu margine "soft" (C-SVM) de normă $\mathcal{L}_2$: ex. 50;**

– **(•) o formă echivalentă a problemei de optimizare C-SVM, în care nu apar deloc restricţii asupra variabilelor, dar în care se foloseşte funcţia de pierdere / cost *hinge*: ex. 20;**
exemplificare / aplicare (şi comparare cu regresia logistică): ex. 51; exemplu de calculare a costurilor *hinge*: ex. 57.B; algoritmul Pegasos: ex. 21 ;

– **(•) algoritmul SMO (Sequential Minimal Optimization):**
deducerea relaţiilor de actualizare a variabilelor Lagrange: ex. 22;
***exemple* de aplicare a algoritmului SMO simplificat: ex. 23, ex. 52;**
[SMO pentru *one-class*, *Max Margin* SVM: ex. 31.c;]

– **o *comparaţie* asupra efectului *atributelor irelevante* (aici, în sensul că odată eliminate / adăugate, n-ar trebui să afecteze rezultatele clasificării) asupra clasificatorilor 1-NN şi C-SVM: ex. 59.**

SVM / C-SVM şi funcţiile-nucleu — câteva proprietăţi

– ***exemplificarea* corespondenţei dintre forma (primală sau duală) a problemei C-SVM şi alegerea valorii parametrului de „destindere" C şi a *funcţiei-nucleu* pe de o parte şi alura şi poziţionarea separatorului optimal pe de altă parte: ex. 24, ex. 53, ex. 54, ex. 55;**

– ***exemplificarea* efectului pe care îl are translatarea datelor în raport cu o axă (Oy) asupra poziţiei separatorului optimal pentru SVM (în raport cu *funcţia-nucleu* folosită): ex. 43;**

– **C-SVM: condiţii suficiente asupra parametrului de „destindere" C şi asupra valorilor funcţiei-nucleu pentru ca toate instanţele de antrenament să fie vectori-suport: ex. 25;**

– **SVM cu nucleu RBF: câteva proprietăţi remarcabile**

  ◦ **pentru SVM pe un set de date [separabil liniar în spaţiul de trăsături] instanţe foarte depărtate de separatorul optimal pot fi vectori-suport: ex. 44;**

  ◦ **(P10) pentru orice set de instanţe distincte şi pentru orice etichetare a acestora, există o valoare a hiper-parametrului nucleului RBF (σ) astfel încât SVM obţine la antrenare eroare 0: ex. 26. Rezultatul *nu* este valabil şi pentru C-SVM;**

  ◦ **(P11) pentru orice set de instanţe distincte, pentru orice etichetare a acestora şi pentru *orice* valoare a hiper-parametrului nucleului RBF (σ), problema de tip SVM care impune ca toate instanţele să fie corect clasificate şi la distanţa $1/\|w\|$ faţă de separatorul optimal are soluţie: ex. 29;**

– **avantaje şi dezavantaje ale folosirii metodelor de clasificare liniară de tipul SVM, Perceptron etc. şi versiunile lor kernel-izate: ex. 58;**

– **chestiuni recapitulative: ex. 27, ex. 28, ex. 60, ex. 68.**

Alte probleme [de optimizare] de tip SVM

- **probleme de tip C-SVM cu valori distincte pentru hiperparametrul C, corespunzător exemplelor negative şi respectiv exemplelor pozitive: ex. 61.de;**
- **SVM pentru clasificare n-ară (SVM multiclass): ex. 30, ex. 62;**
- **deducerea *formei duale* pentru problema *one-class SVM*, versiunea *Max Margin*: ex. 31 (varianta cu margine "hard"), ex. 64 (varianta cu margine "soft", folosind ν-SVM);**

– **legătura dintre soluţiile problemei *one-class SVM*, versiunea *Max Margin*, cu margine "hard" şi respectiv cele ale problemei SVM (cu şi respectiv fără termen liber (engl., bias)), tot cu margine "hard": ex. 63;**

- **deducerea *formei duale* pentru problema *one-class SVM*, versiunea *minimum enclosing ball* (MEB): ex. 32 (varianta cu margine "hard"), ex. 65 (varianta cu margine "soft", folosind *ν-SVM*);**

– **o condiţie suficientă pentru ca variantele cu margine "hard" pentru cele două tipuri de probleme de optimizare *one-class SVM*, şi anume *Max Margin* şi *minimum enclosing ball* (MEB), în forma kernel-izată, să fie echivalente: ex. 32;**

- **deducerea *formei duale* pentru problema *ν-SVM*: ex. 33;**
- **deducerea *formei duale* pentru problema SVR (*Support Vector Regression*), folosind funcţie de cost / pierdere *ε-senzitivă*: ex. 34 (cu margine "hard"), ex. 67 (cu margine "soft" şi (echivalent) cu funcţie de cost ε-senzitivă); exemplificare / aplicare: ex. 66.**

5.1 Maşini cu vectori-suport — Probleme rezolvate

5.1.1 SVM cu margine "hard"

1. **(O proprietate a vectorului w care apare în ecuaţia unui hiperplan; distanţa de la un hiperplan din $\mathbb{R}^d$ la un punct oarecare din acelaşi spaţiu: deducerea formulei)**

prelucrare de Liviu Ciortuz, după
■ □ • ○ ***CMU, 2010 fall, Ziv Bar-Joseph, HW4, pr. 1.1***

Fie un punct oarecare x_0 din $\mathbb{R}^d$ şi un hiperplan de ecuaţie $w \cdot x + b = 0$, unde w şi x sunt de asemenea din $\mathbb{R}^d$, în vreme ce $b \in \mathbb{R}$, iar simbolul $\cdot$ indică produsul scalar al vectorilor din $\mathbb{R}^d$.

a. Demonstraţi că vectorul w este ortogonal pe acest hiperplan.
(*Sugestie*: Fie x_1 şi x_2 două puncte situate pe acest hiperplan. Ce puteţi spune despre valoarea produsului scalar $w \cdot (x_1 - x_2)$?)

b. Demonstraţi că în cazul în care $w \neq 0 \in \mathbb{R}^d$, distanţa [măsurată pe perpendiculara] de la x_0 la hiperplanul $w \cdot x + b = 0$ este

$$\frac{|w \cdot x_0 + b|}{\|w\|},$$

unde simbolul $\| \ \|$ desemnează norma euclidiană.[627]

Răspuns:

a. Fie H hiperplanul de ecuaţie $w \cdot x + b = 0$ (în scriere vectorială, $w^\top x + b = 0$, unde am considerat w şi x vectori-coloană din $\mathbb{R}^d$). Considerăm x_1 şi x_2 două puncte oarecare distincte situate pe hiperplanul H. Aşadar, vom avea

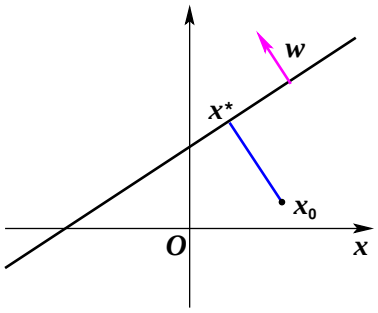
$$w \cdot x_1 + b = 0 \text{ şi } w \cdot x_2 + b = 0,$$

de unde, prin scădere rezultă $w \cdot (x_1 - x_2) = 0$, ceea ce înseamnă că vectorii w şi $x_1 - x_2$ sunt perpendiculari.[628] **Întrucât vectorul $x_1 - x_2$ are direcţia hiperplanului H, rezultă că w este perpendicular pe H.**

[627] $\|w\|^2 = w \cdot w$ sau $\|w\|^2 = w^\top w$, unde simbolul $\top$ indică operaţia de transpunere de vectori / matrice. Precizăm că norma euclidiană se desemnează de fapt prin $\| \ \|_2$, însă putem renunţa la indice atunci când nu există posibilitatea de a se face confuzie cu vreo altă normă în contextul respectiv.

[628] Vă readucem aminte că pentru doi vectori oarecare v_1 şi v_2, considerând că unghiul format de aceşti doi vectori este θ, avem: $v_1 \cdot v_2 = \|v_1\| \, \|v_2\| \cos(\theta)$.

b. Notăm cu x^* „piciorul“ perpendicularei coborâte din punctul x_0 pe hiperplanul H. Prin urmare, vectorul $x^* - x_0$ are aceeaşi direcţie cu vectorul w. Ştim că pentru oricare doi vectori paraleli u şi u' este satisfăcută proprietatea următoare: există o constantă $\lambda \in \mathbb{R}$ astfel încât $u' = \lambda u$. Aşadar, există $t \in \mathbb{R}$ cu propriteatea $x^* - x_0 = tw$. Altfel spus,



$$x^* = x_0 + tw. \tag{303}$$

Ştim de asemenea că punctul x^* aparţine hiperplanului H. Prin urmare,

$$w \cdot x^* + b = 0. \tag{304}$$

Din relaţiile (303) şi (304) rezultă

$$w \cdot (x_0 + tw) + b = 0 \Leftrightarrow w \cdot x_0 + tw^2 + b = 0 \Leftrightarrow w \cdot x_0 + t\|w\|^2 + b = 0 \Leftrightarrow$$
$$t\|w\|^2 = -(w \cdot x_0 + b) \Leftrightarrow t = -\frac{w \cdot x_0 + b}{\|w\|^2}.$$

În concluzie, distanţa de la x_0 la hiperplanul H este

$$\|x^* - x_0\| = \|tw\| = |t|\ \|w\| = \frac{|w \cdot x_0 + b|}{\|w\|^2}\|w\| = \frac{|w \cdot x_0 + b|}{\|w\|}.$$

2. **(Exerciţiu teoretic: deducerea formei primale a problemei de optimizare SVM (cu margine “hard”, pornind de la principiul maximizării marginii geometrice)**

prelucrare de Liviu Ciortuz, după
■ □ • ***CMU, 2016 fall, N. Balcan. M. Gormley, HW4, pr. 1.1***
CMU, 2006 spring, Carlos Guestrin, midterm, pr. 1.4

În acest exerciţiu veţi vedea cum anume se deduce *problema de optimizare SVM cu margine “hard”* pornind de la *principiul marginii* [*geometrice*] *maxime*.

Presupunem că avem setul de date de antrenament $D = (X, y)$, unde $X \in \mathbb{R}^{d \times m}$, iar $y \in \{-1, 1\}^m$. Coloana i a matricei X este x_i, vectorul de trăsături al celui de-al i-lea exemplu de antrenament, iar y_i este eticheta acestui exemplu.

Pentru clasificare, vom folosi o funcţie liniară, de forma

$$f(x) = w \cdot x, \text{ unde } w \in \mathbb{R}^d,$$

iar operatorul $\cdot$ reprezintă produsul scalar al vectorilor.

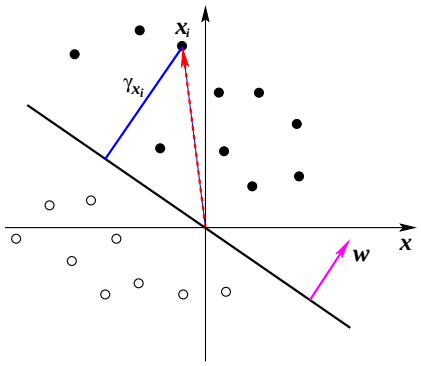
Observăm că, pentru simplitate, în expresia lui f nu a fost adăugat termenul liber (notat îndeobşte cu w_0 sau cu b, ultimul provenind de la termenul englezesc *bias*). Aşadar, vom trata aici doar cazul separabilităţii liniare prin originea sistemului de coordonate.[629]

[629] Cazul separabilităţii liniare cu termenul liber w_0 luând o valoare oarecare se tratează extinzând în mod corespunzător demonstraţia de la punctul a.

O instanţă oarecare x va fi clasificată de către funcţia f în clasa 1 dacă $f(x) > 0$, respectiv în clasa -1 în caz contrar. Hiperplanul de ecuaţie $f(x) = 0$ va funcţiona ca *separator* al instanţelor de antrenament, sau *graniţă de decizie* (engl., decision boundary). Aşadar, f este funcţia pe care vrem s-o învăţăm (adică funcţia "target").

Presupunem că datele de antrenament sunt liniar separabile prin origine, cu ajutorul funcţiei f. Atunci pentru orice exemplu de antrenament (x, y) vom avea $yf(x) > 0$. În consecinţă, *distanţa geometrică* de la x la graniţa de decizie — distanţă despre care, conform problemei 1, ştim că este egală cu $\frac{|f(x)|}{\|w\|}$ — poate fi scrisă ca

$$\gamma_x = \frac{yf(x)}{\|w\|}.$$



În contextul învăţării automate, această distanţă se numeşte *marginea geometrică* (sau, pe scurt, *marginea*) dintre instanţa x şi separatorul determinat de ecuaţia $f(x) = 0$.

În vederea reducerii *riscului de clasificare eronată* la faza de *generalizare*, este natural să determinăm f (deci, de fapt, vectorul de ponderi w) astfel încât minimul acestor margini γ_x să fie cât mai mare. Altfel spus, vrem să maximizăm distanţa de la *separatorul optimal* până la instanţele de antrenament cele mai apropiate. Aşadar, *funcţia* noastră *obiectiv* va fi:

$$\max_w \min_{i=1,\ldots,m} \frac{y_i f(x_i)}{\|w\|}. \tag{305}$$

a. Arătaţi că problema de optimizare fără restricţii (305) este *echivalentă* cu următoarea problemă de optimizare convexă cu restricţii liniare, pe care o vom numi *problema SVM* (în aşa-numita *formă primală*):[630]

$$\min_w \frac{1}{2}\|w\|^2 \tag{306}$$
$$\textbf{a. î. } y_i(w \cdot x_i) \geq 1, \textbf{ pentru } i = 1, \ldots, m.$$

***Sugestie*: Are oare vreun efect scalarea vectorului de ponderi $w \to kw$, cu $k \in \mathbb{R}$, $k > 0$?**

***Observaţie*: Are loc următoarea *proprietate*:**[631] **dacă mulţimea S este liniar separabilă, atunci problema de optimizare SVM în formă primală are soluţie unică, întrucât funcţia obiectiv a acestei probleme este strict convexă.**

b. Presupunând că datele de antrenament sunt liniar separabile, precizaţi ce se va întâmpla cu separatorul obţinut prin rezolvarea problemei SVM dacă

[630] LC: Este vorba de o echivalenţă *slabă* (engl., soft): optimul — adică, valoarea funcţiei obiectiv — pentru cele două probleme este acelaşi, iar orice soluţie a primei probleme poate fi pusă în corespondenţă cu o soluţie a celei de-a doua probleme şi invers. (Echivalenţa ar fi *tare* (engl., hard) dacă valoarea funcţiei obiectiv pentru cele două probleme ar fi aceeaşi, iar orice soluţie optimă a primei probleme ar fi soluţie optimă şi pentru cea de-a doua problemă şi invers.)

[631] Vedeţi articolul *Uniqueness of the SVM solution*, de Christopher Burges şi David Crisp, 1998.

se renunţă la unul dintre exemplele de antrenament: se va deplasa oare separatorul *înspre* exemplul / punctul respectiv, se va retrage *dinspre* punctul respectiv, ori poziţia lui va rămâne neschimbată? Justificaţi.

Răspuns:

a. Vom demonstra echivalenţa dintre problemele de optimizare date în enunţ (305) **şi** (306) **construind o succesiune de câteva probleme de optimizare echivalente, obţinute prin aplicarea unor transformări succesive, pornind de la problema** (305) **şi ajungând în final la problema** (306)**.**

Fie, aşadar, următoarele probleme de optimizare:

$$\max_{w} \min_{i=1,\ldots,m} \frac{y_i\, w \cdot x_i}{\|w\|} \tag{307}$$
$$\textbf{a. î. } y_i(w \cdot x_i) > 0, \ \textbf{pentru } i = 1, \ldots, m.$$

$$\max_{w} \frac{1}{\|w\|} \min_{i=1,\ldots,m} y_i\, w \cdot x_i \tag{308}$$
$$\textbf{a. î. } y_i(w \cdot x_i) > 0, \ \textbf{pentru } i = 1, \ldots, m.$$

$$\max_{w} \frac{1}{\|w\|} \min_{i=1,\ldots,m} y_i\, w \cdot x_i \tag{309}$$
$$\textbf{a. î. } y_i(w \cdot x_i) \geq 1, \ \textbf{pentru } i = 1, \ldots, m.$$

$$\max_{w} \frac{1}{\|w\|} \tag{310}$$
$$\textbf{a. î. } y_i(w \cdot x_i) \geq 1, \ \textbf{pentru } i = 1, \ldots, m.$$

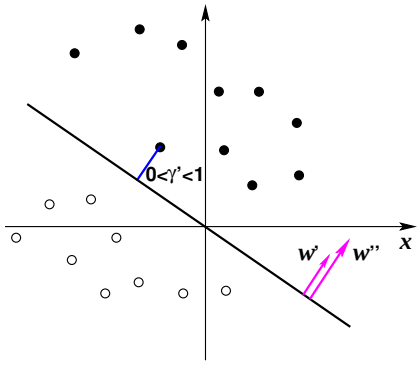
• Se observă că problema de optimizare (307) **diferă de problema iniţială** (305) **prin introducerea unor restricţii liniare. Mai mult, aceste restricţii corespund hiperplanelor care sunt separatori liniari pentru mulţimea de antrenament dată (care este, conform enunţului, separabilă liniar). Aşadar, aceste restricţii nu schimbă cu nimic soluţia problemei care a fost dată iniţial.**[632]

• Problema de optimizare (308) **a fost obţinută din problema** (307) **modificând doar funcţia obiectiv:** $\frac{1}{\|w\|}$ **a fost scos în faţa operatorului** $\min_{i=1,\ldots,m}$ **fiindcă** w **nu depinde de** i**. (Observaţi că expresia care urmează acestui operator depinde de** w**, dar acolo acest** w **este considerat fixat şi variază doar** i**.)**

• Problema de optimizare (309) **a fost obţinută din problema** (308) **schimbând în restricţii relaţia** > 0 **cu** ≥ 1**. În sine, această modificare restrânge mulţimea acelor valori ale lui** w **peste care se aplică operatorul** $\max_w$**.**

[632] Pentru un separator liniar oarecare al mulţimii de antrenament, expresia $\min_{i=1,\ldots,m} \frac{y_i f(x_i)}{\|w\|}$ are o valoare strict pozitivă. Pentru hiperplanele care nu separă mulţimea de antrenament, expresia $\min_{i=1,\ldots,m} \frac{y_i f(x_i)}{\|w\|}$ are valoare negativă sau 0.

Să considerăm însă un w' arbitrar care este eliminat la trecerea de la forma (308) la forma (309). Notăm $\gamma' = \min_{i=1,\ldots,m} y_i \underbrace{w' \cdot x_i}_{f(x_i)}$; ştim că $\gamma' \in (0,1)$. Rezultă că



$$\frac{y_i w' \cdot x_i}{\gamma'} = y_i \frac{w'}{\gamma'} \cdot x_i \geq 1 \text{ pentru } i = 1, \ldots, m.$$

Observaţi că egalitatea chiar se produce, şi anume atunci când indexul i ia valoarea pentru care se atinge minimul expresiei $y_i w' \cdot x_i$.

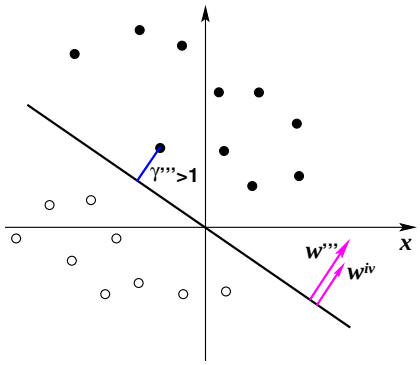
Aşadar, $w'' \overset{not.}{=} \frac{w'}{\gamma'}$ satisface restricţiile problemei (309). În plus,

$$\frac{1}{\|w''\|} = \frac{1}{\|w'\|}\gamma' \text{ şi, echivalent, } \frac{1}{\|w''\|} \underbrace{\min_{i=1,\ldots,m} y_i w'' \cdot x_i}_{1} = \frac{1}{\|w'\|} \underbrace{\min_{i=1,\ldots,m} y_i w' \cdot x_i}_{\gamma'}.$$

Prin urmare, putem spune că rolul lui w' în relaţia (308) este jucat de w'' în relaţia (309). În consecinţă, la trecerea de la o formă la cealaltă optimul rămâne acelaşi.[633]

• Problema de optimizare (310) a fost obţinută din problema (309) renunţând în funcţia obiectiv la componenta $\min_{i=1,\ldots,m} y_i\, w \cdot x_i$.

Aceasta revine de fapt la a renunţa la acele valori ale lui w pentru care $\min_{i=1,\ldots,m} y_i w \cdot x_i$ este strict mai mare decât 1. (Am văzut mai sus că există valori ale lui w (şi anume, w'') pentru care minimul respectiv este chiar 1.) Să considerăm un w''' cu proprietatea $\gamma''' \overset{not.}{=} \min_{i=1,\ldots,m} y_i \underbrace{w''' \cdot x_i}_{f(x_i)} > 1$.



Rezultă că

$$\min_{i=1,\ldots,m} y_i \frac{w'''}{\gamma'''} \cdot x_i = 1.$$

În continuare, notând $w^{iv} = \frac{w'''}{\gamma'''}$, vom avea:

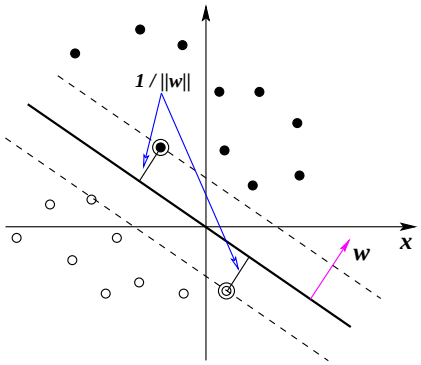
$$\frac{1}{\|w^{iv}\|} \underbrace{\min_{i=1,\ldots,m} y_i w^{iv} \cdot x_i}_{1} = \frac{1}{\|w^{iv}\|} = \frac{\gamma'''}{\|w'''\|} = \frac{1}{\|w'''\|} \underbrace{\min_{i=1,\ldots,m} y_i w''' \cdot x_i}_{\gamma'''}.$$

Prin urmare, [ca şi mai sus,] putem spune că rolul lui w''' în relaţia (309) este jucat de w^{iv} în relaţia (310). În consecinţă, [ca şi mai sus,] nu se modifică

[633] Din punct de vedere geometric, trecerea de la (308) la (309) poate fi asociată cu înmulţirea cu o constantă pozitivă (chiar supra-unitară) a ecuaţiilor acelor separatori liniari [ai mulţimii de antrenament] pentru care minimul distanţelor geometrice până la instanţele de antrenament este strict *mai mic* decât 1, astfel încât în urma înmulţirii această distanţă minimă să devină 1.

valoarea optimă a funcţiei obiectiv dacă se renunţă [şi] la aceşti w'''.[634]

• Se constată imediat că problema (310) este echivalentă cu problema de optimizare SVM care a fost dată în enunţ (306). Forma problemei de optimizare SVM ne spune că separatorul optimal se alege dintre acei separatori $w \cdot x$ ai mulţimii de antrenament care au exact valoarea 1 pentru $\min_{i=1,\ldots,m} y_i w \cdot x_i$.



Concret, alegerea se face maximizând $1/\|w\|$, adică distanţa geometrică de la separator(i) până la cele mai apropiate instanţe de antrenament. Intuitiv, maximizarea aceasta va implica faptul că distanţele de la separatorul optimal până la *toate* instanţele de antrenament cele mai apropiate (adică, *vectorii-suport*, cei pozitivi şi respectiv cei negativi) vor fi $1/\|w\|$. (În figura alăturată, în care am pus în evidenţă acest fapt, vectorii-suport sunt încercuiţi.)

Altfel spus, valoarea funcţiei $w \cdot x$ pentru w optim va fi $+1$ pentru vectorii-suport pozitivi şi -1 pentru vectorii-suport negativi.

b. La eliminarea unei instanţe de antrenament x_j, marginea $\max_w \min_i \dfrac{y_i f(x_i)}{\|w\|}$ nu poate să scadă; poate doar să crească sau, eventual, să rămână neschimbată. Într-adevăr, pentru un w oarecare (fixat), cantitatea $\min_i \dfrac{y_i f(x_i)}{\|w\|}$ fie creşte fie rămâne aceeaşi la eliminarea unui x_j, fiindcă acum operatorul $\min$ este aplicat pe o mulţime mai puţin amplă. În consecinţă, lăsându-l apoi pe w să varieze, cantitatea $\max_w \min_i \dfrac{y_i f(x_i)}{\|w\|}$ creşte şi ea sau cel puţin rămâne aceeaşi.

Ţinând cont de rezultatul de echivalenţă de la punctul a, rezultă că la eliminarea unei instanţe de antrenament x_j marginea de separare optimală $(1/\|w\|)$ creşte sau rămâne, eventual, neschimbată. Aşadar, poziţia separatorului optimal (*i*) va fi deplasată înspre instanţa eliminată sau (*ii*) va rămâne neschimbată. Cazul (*i*) este *posibil* să apară doar dacă x_j este vector-suport, adică se afla la distanţă minimală faţă de separatorul optimal.[635] **Cazul (*ii*) se produce atunci când x_j este vector-suport şi există mai mulţi vectori-suport care au etichete identice cu eticheta lui x_j sau atunci când x_j nu este vector-suport.**

***Observaţie importantă*:**

Până la problema 9 — unde vom prezenta deducerea *formei duale* a problemei de optimizare SVM —, vom lucra cu o *definiţie în sens larg* (geometric) pentru noţiunea de *vector-suport*: spunem că un vector-suport este o instanţă x_i care se află la distanţa geometrică de $1/\|w\|$ faţă de separatorul optimal (w, w_0). Odată cu introducerea formei duale a problemei de optimizare SVM, vom folosi pentru noţiunea de vector-suport *definiţia clasică* (în sens (analitic)

[634] Din punct de vedere geometric, trecerea de la (309) la (310) corespunde înmulţirii cu o constantă pozitivă (sub-unitară) a ecuaţiilor acelor separatori liniari [ai mulţimii de antrenament] pentru care minimul distanţelor geometrice până la instanţele de antrenament este strict *mai mare* decât 1, astfel încât în urma înmulţirii această distanţă minimă să devină 1.

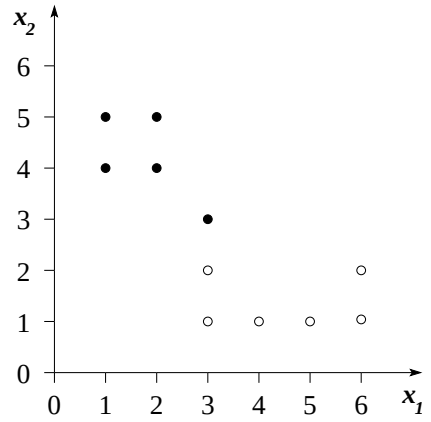
[635] Cum distanţa până la vectorii-suport de semn opus rămâne aceeaşi (1), rezultă că este *posibil* ca separatorul optimal să se deplaseze în direcţia vectorului-suport care a fost eliminat. Totuşi, nu este *obligatoriu* ca în această situaţie poziţia separatorului optimal să se modifice. Vedeţi cazul (*ii*).

restrâns în raport cu cel precedent): o instanţă x_i este vector-suport dacă în soluţia optimă a problemei duale, variabila / multiplicatorul Lagrange $\bar{\alpha}_i$ are valoare nenulă (deci strict pozitivă, pentru că toţi multiplicatorii Lagrange sunt acolo nenegativi). În această a doua accepţiune, unele instanţe situate la distanţa (geometrică) de $1/\|w\|$ faţă de separatorul optimal (w, w_0) pot să nu fie vectori-suport. Pentru SVM cu margine "soft" (C-SVM), va fi util să vedeţi *Rezumatul* de la rezolvarea punctului e de la problema 12. Se va constata acolo că este posibil să avem vectori-suport aflaţi la distanţă (geometrică) mai mică de $1/\|w\|$ faţă de separatorul optimal (w, w_0); aceştia sunt aşa-numiţii (în engl.) *non-bound support vectors*.

3. **(Separabilitate în $\mathbb{R}^2$; SVM liniară, forma primală)**

CMU, 2003 fall, T. Mitchell, A. Moore, final exam, pr. 5

Desenul alăturat prezintă un set de date cu două atribute de intrare x_1 şi x_2, şi un atribut de ieşire y, ale cărui valori sunt reprezentate prin culoarea punctului (alb/− şi negru/+).

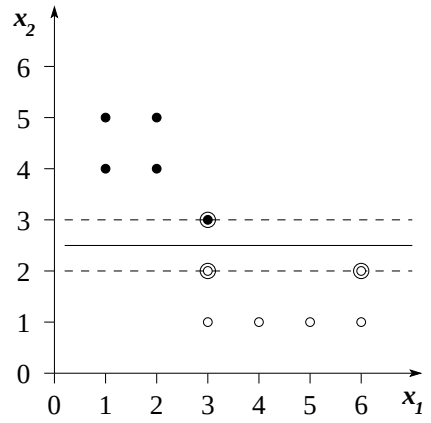


a. Presupunând că aceste date sunt corecte (adică fără 'zgomote') şi că folosim o SVM liniară, trasaţi pe acest desen trei drepte care să indice linia de *separare optimală* şi cele două „margini" (paralelele la separatorul optimal care trec prin vectorii-suport). Încercuiţi vectorii-suport.

b. Plecând de la forma primală a unei SVM liniare (având regula de decizie $y = sign(w \cdot x + w_0)$),[636] calculaţi valorile corespunzătoare parametrilor w şi w_0 care determină hiperplanul de separare optimală de la punctul a.

Răspuns:

a. În figura alăturată, linia continuă reprezintă dreapta de separare, iar cele două linii punctate reprezintă marginile maximale. Vectorii-suport sunt punctele $(3, 3)$, $(3, 2)$ şi $(6, 2)$.



b. În forma primală a problemei SVM liniare se cere minimizarea valorii $\frac{1}{2} \cdot ||w||^2$, simultan cu satisfacerea restricţiilor $y_i(w \cdot x + w_0) \geq 1$, unde $w \stackrel{not.}{=} (w_1, w_2)$ şi $x \stackrel{not.}{=} (x_1, x_2)$. Ţinând cont de faptul că aceste inegalităţi sunt îndeplinite cu egalitate [doar] în cazul vectorilor-suport, obţinem următorul sistem de ecuaţii:

[636] Vedeţi la problema 2.a (sau la problema 9) cum se defineşte *forma primală* a acestei SVM liniare.

$$\left\{\begin{array}{rcl} -((w_1,w_2)\cdot(3,2)+w_0) & = & 1 \\ -((w_1,w_2)\cdot(6,2)+w_0) & = & 1 \\ (w_1,w_2)\cdot(3,3)+w_0 & = & 1 \end{array}\right. \Rightarrow \left\{\begin{array}{rcl} 3w_1+2w_2+w_0 & = & -1 \\ 6w_1+2w_2+w_0 & = & -1 \\ 3w_1+3w_2+w_0 & = & 1 \end{array}\right. \Rightarrow$$

$$\left\{\begin{array}{rcl} w_1 & = & 0 \\ 2w_2+w_0 & = & -1 \\ 3w_2+w_0 & = & 1 \end{array}\right. \Rightarrow \left\{\begin{array}{rcl} w_1 & = & 0 \\ w_2 & = & 2 \\ w_0 & = & -5 \end{array}\right.$$

Am obţinut deci $w = (w_1, w_2) = (0, 2)$ şi $w_0 = -5$, ceea ce conduce la ecuaţia $2x_2 - 5 = 0 \Leftrightarrow x_2 - \frac{5}{2} = 0$ pentru separatorul optimal.

***Alternativ*, valorile parametrilor w_0, w_1 şi w_2 pot fi determinate folosind cunoştinţe de geometrie analitică. Ecuaţia dreptei care este reprezentată prin linia continuă din figura de mai sus este**

$$x_2 = \frac{5}{2} \Leftrightarrow x_2 - \frac{5}{2} = 0$$

sau, mai general, $\alpha\left(x_2 - \frac{5}{2}\right) = 0$, cu $\alpha \in \mathbb{R},\ \alpha \neq 0$. Valoarea lui α corespunzătoare separatorului optimal se obţine impunând restricţiile specifice vectorilor-suport:

$\alpha\left(x_2 - \frac{5}{2}\right) = 1$ pentru $x_2 = 3$ sau / şi, respectiv, $\alpha\left(x_2 - \frac{5}{2}\right) = -1$ pentru $x_2 = 2$.

Rezultă că $\alpha = 2$, deci ecuaţia separatorului optimal este $2\left(x_2 - \frac{5}{2}\right) = 0 \Leftrightarrow 0 \cdot x_1 + 2 \cdot x_2 - 5 = 0$, de unde, identificând pe componente, rezultă: $w_1 = 0, w_2 = 2$ şi $w_0 = -5$.

4. **(Separabilitate în $\mathbb{R}$; SVM liniară, forma primală; calculul erorilor la antrenare şi CVLOO)**

CMU, 2001 fall, Andrew Moore, final exam, pr. 11

Fie următorul set de date:

x_i	1	2	3.5	4	5
y_i	-1	-1	-1	1	1

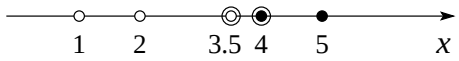
a. Care sunt valorile parametrilor w si w_0 învăţate de o SVM liniară pornind de la aceste date? (Vă readucem aminte că regula de decizie a unei astfel de SVM este $y = sign(w \cdot x + w_0)$.)

b. Care sunt vectorii-suport?

c. Care este eroarea de clasificare pe mulţimea de date de antrenament de mai sus?

d. Care este eroarea de clasificare pe acelaşi set de date, folosind metoda de cross-validare "Leave-One-Out"? (Veţi exprima eroarea sub forma procentajului de instanţe clasificate eronat.)

Răspuns:



În mod *direct*, reprezentarea grafică ne arată că separatorul optimal va fi plasat în dreptul punctului 3.75, folosind vectorii-suport $x_3 = 3.5$ şi $x_4 = 4$, şi conducând la eroare de antrenare 0 (deoarece datele de antrenament sunt liniar separabile). Este imediat că instanţele $x_3 = 3.5$ şi $x_4 = 4$ vor fi clasificate eronat la CVLOO.

a. *Analitic*, din forma primală a problemei SVM rezultă că trebuie satisfăcute următoarele restricţii:

$$\begin{cases} -(w\cdot 1 + w_0) \geq 1 \\ -(w\cdot 2 + w_0) \geq 1 \\ -(w\cdot 3.5 + w_0) \geq 1 \\ +(w\cdot 4 + w_0) \geq 1 \\ +(w\cdot 5 + w_0) \geq 1 \end{cases} \Leftrightarrow \begin{cases} w + w_0 \leq -1 \\ 2w + w_0 \leq -1 \\ 3.5w + w_0 \leq -1 \\ 4w + w_0 \geq 1 \\ 5w + w_0 \geq 1 \end{cases}$$

Impunând ca restricţiile corespunzătoare vectorilor-suport ($x_3 = 3.5$ şi $x_4 = 4$) să fie satisfăcute cu egalitate — ceea ce va conduce în mod implicit şi la satisfacerea celorlalte inegalităţi din sistem —, rezultă:

$$\begin{cases} 3.5w + w_0 = -1 \\ 4w + w_0 = 1 \end{cases} \Leftrightarrow \begin{cases} -3.5w - w_0 = 1 \\ 4w + w_0 = 1 \end{cases} \Leftrightarrow \begin{cases} 0.5w = 2 \\ 4w + w_0 = 1 \end{cases} \Leftrightarrow \begin{cases} w = 4 \\ w_0 = -15 \end{cases}$$

Am obţinut deci soluţia $w = 4$, $w_0 = -15$.

b. După cum am menţionat mai sus, vectorii-suport sunt $x_3 = 3.5$ şi $x_4 = 4$. *Alternativ*, însă tot *analitic*, punctele a şi b de mai sus se pot rezolva astfel:

Se observă în mod direct că inegalitatea $x \geq 3.75$ este satisfăcută pentru toate instanţele pozitive (şi doar pentru ele). Aşadar, funcţia $sign(x - 3.75)$ *clasifică* perfect atât instanţele pozitive cât şi pe cele negative. Mai mult, instanţele $x_3 = 3.5$ (negativă) şi $x_4 = 4$ (pozitivă) sunt cele mai apropiate faţă de „pragul" $x = 3.75$ şi sunt situate la distanţe egale faţă de acest prag, deci sunt vectorii-suport. Ecuaţia $x = 3.75$ se scrie echivalent ca $\alpha(x = 3.75)$, cu $\alpha \neq 0$. Impunând condiţia $\alpha(x - 3.75) = -1$ pentru $x = 3.5$ şi respectiv $\alpha(x - 3.75) = 1$ pentru $x = 4$, rezultă $\alpha = 4$ şi deci, prin identificare pe componente, $w = 4$ şi $w_0 = -15$.

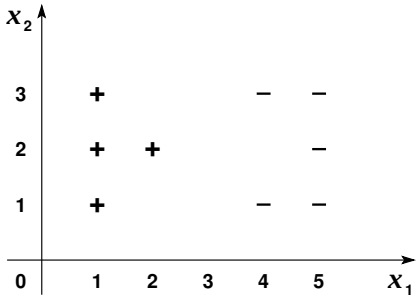
c. Judecând *calitativ*, eroarea de clasificare pe mulţimea datelor de antrenament este 0, fiindcă datele sunt liniar separabile. *Analitic*, am arătat deja că $sign(w \cdot x_i + w_0) = y_i$, pentru $i \in \{1, 2, 3, 4, 5\}$.

d. La cross-validare cu metoda "Leave-One-Out", instanţele 3.5 şi 4 vor fi clasificate eronat pentru că pragul de separare optimală devine $x = 3$ şi respectiv $x = 4.25$. Eroarea este deci de 40%. De notat, ca un *principiu general*: atunci când se lucrează cu SVM, erorile la cross-validare se *pot* produce doar la eliminarea vectorilor-suport.[637]

[637] Vedeţi legătura cu problema 2.b.

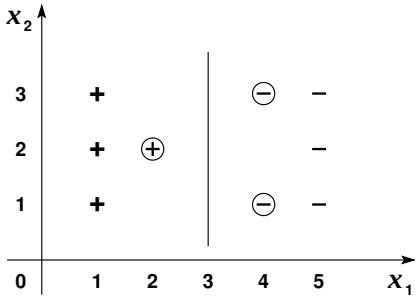
5. **(SVM liniară: exemplificare pe date din $\mathbb{R}^2$; modificarea eventuală a poziţiei separatorului optimal la eliminarea unei instanţe de antrenament)**

În imaginea alăturată, încercuiţi fiecare punct care are proprietatea că odată ce este eliminat din setul de date de antrenament, la re-antrenarea maşinii cu vectori-suport vom obţine un alt separator optimal decât cel rezultat în cazul antrenării pe întreaga mulţime de date. Justificaţi răspunsul.



<u>Răspuns:</u>

În figura alăturată am reprezentat *grafic* separatorul liniar învăţat de SVM din datele de antrenament şi am încercuit vectorii-suport. După cum ştim deja din rezolvarea problemei 4.d, eliminarea datelor care nu sunt vectori-suport nu modifică rezultatul procesului de învăţare. Aşadar, rămâne de verificat comportamentul sistemului la eliminarea câte unui vector-suport.

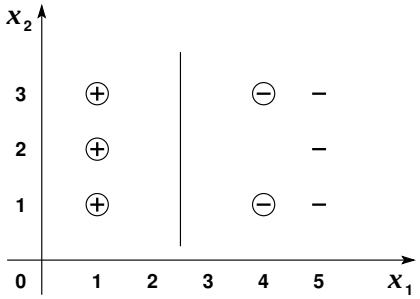


***Analitic*, putem vedea că separatorul învăţat este unic determinat de sistemul:**

$$\begin{cases} +(w_1,w_2)\cdot(2,2)+w_0 &= 1 \\ -((w_1,w_2)\cdot(4,1)+w_0) &= 1 \\ -((w_1,w_2)\cdot(4,3)+w_0) &= 1 \end{cases} \Rightarrow \begin{cases} 2w_1+2w_2+w_0 &= 1 \\ 4w_1+w_2+w_0 &= -1 \\ 4w_1+3w_2+w_0 &= -1 \end{cases} \Rightarrow \begin{cases} w_1=-1 \\ w_2=0 \\ w_0=3 \end{cases}$$

În consecinţă, *ecuaţia separatorului optimal* este $-x_1+3=0 \Leftrightarrow x_1=3$, iar *marginea geometrică* (distanţa de la vectorii-suport până la hiperplanul de separare) este $d=\dfrac{1}{||w||}=\dfrac{1}{\sqrt{(-1)^2+0^2}}=1$, unde după cum ştim, $w=(w_1,w_2)$.

La eliminarea punctului (2,2), hiperplanul de separare optimal este altul, vectorii-suport devenind (1,1), (1,2), (1,3), (4,1) şi (4,3). Noile valori ale parametrilor (notate w_0', w_1', w_2') se calculează rezolvând sistemul:

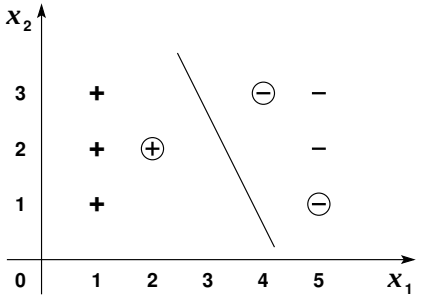


$$\begin{cases} w_1'+w_2'+w_0' &= 1 \\ w_1'+2w_2'+w_0' &= 1 \\ w_1'+3w_2'+w_0' &= 1 \\ 4w_1'+w_2'+w_0' &= -1 \\ 4w_1'+3w_2'+w_0' &= -1 \end{cases} \Rightarrow \begin{cases} w_1'=-\dfrac{2}{3} \\ w_2'=0 \\ w_0'=\dfrac{5}{3} \end{cases}$$

Aşadar, ecuaţia separatorului optimal este acum $-\dfrac{2}{3}x_1+\dfrac{5}{3}=0 \Leftrightarrow x_1=\dfrac{5}{2}$, iar marginea geometrică este $d'=\dfrac{1}{||w'||}=\dfrac{1}{\sqrt{(-2/3)^2+0^2}}=\dfrac{3}{2}$.

La eliminarea punctului (4,1), vectorii-suport devin (2,2), (5,1) şi (4,3). Determinarea noilor valori ale parametrilor se face prin rezolvarea sistemului:

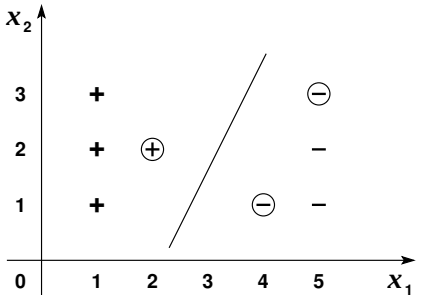
$$\left\{\begin{array}{rcr} 2w_1'' + 2w_2'' + w_0'' & = & 1 \\ 5w_1'' + w_2'' + w_0'' & = & -1 \\ 4w_1'' + 3w_2'' + w_0'' & = & -1 \end{array}\right. \Rightarrow \left\{\begin{array}{l} w_1'' = -\frac{4}{5} \\ w_2'' = -\frac{2}{5} \\ w_0'' = \frac{17}{5} \end{array}\right.$$



În consecinţă, în acest caz ecuaţia separatorului optimal este $-\frac{4}{5}x_1 - \frac{2}{5}x_2 + \frac{17}{5} = 0 \Leftrightarrow 4x_1 + 2x_2 - 17 = 0 \Leftrightarrow x_2 = -2x_1 + \frac{17}{2}$**, iar marginea geometrică până la hiperplan este** $d'' = \frac{1}{||w''||} = \frac{1}{\sqrt{(-4/5)^2 + (-2/5)^2}} = \frac{5}{2\sqrt{5}} = \frac{\sqrt{5}}{2}$**.**

La eliminarea punctului (4,3), vectorii-suport devin (2,2), (4,1) şi (5,3). Vom obţine următoarele valori ale parametrilor:

$$\left\{\begin{array}{rcr} 2w_1''' + 2w_2''' + w_0''' & = & 1 \\ 4w_1''' + w_2''' + w_0''' & = & -1 \\ 5w_1''' + 3w_2''' + w_0''' & = & -1 \end{array}\right. \Rightarrow \left\{\begin{array}{l} w_1''' = -\frac{4}{5} \\ w_2''' = \frac{2}{5} \\ w_0''' = \frac{9}{5} \end{array}\right.$$



Prin urmare, aici ecuaţia separatorului optimal este $-\frac{4}{5}x_1 + \frac{2}{5}x_2 + \frac{9}{5} = 0 \Leftrightarrow -4x_1 + 2x_2 + 9 = 0 \Leftrightarrow x_2 = 2x_1 - \frac{9}{2}$**, iar marginea geometrică este** $d''' = \frac{1}{||w'''||} = \frac{1}{\sqrt{(-4/5)^2 + (2/5)^2}} = \frac{5}{2\sqrt{5}} = \frac{\sqrt{5}}{2}$**.**

6. **(Separabilitate liniară în $\mathbb{R}^2$:**
tratare în raport cu un parametru dat;
Separabilitate neliniară în $\mathbb{R}$:
folosirea unei funcţii simple pentru maparea atributelor)

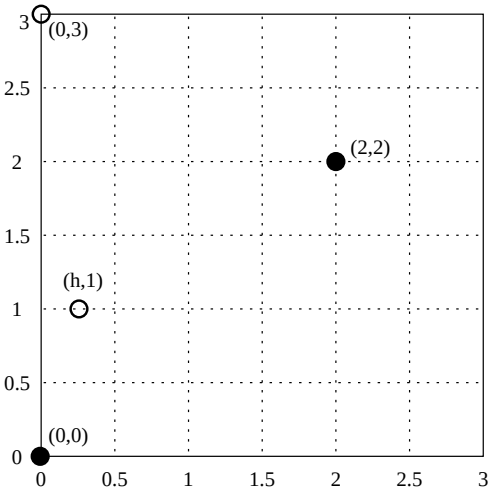
CMU, 2009 spring, Ziv Bar-Joseph, final exam, pr. 4

Presupunem că avem doar patru exemple de antrenament în spaţiul euclidian bidimensional, şi anume: $x_1 = (0,0)$**,** $x_2 = (2,2)$ **sunt exemple pozitive, iar** $x_3 = (h,1)$**,** $x_4 = (0,3)$ **sunt exemple negative,** h **fiind un parametru cu proprietatea** $0 \leq h \leq 3$**.**

a. Cât de mare poate fi valoarea lui $h \geq 0$ cu condiţia ca punctele de antrenament să rămână liniar separabile?

b. Se schimbă direcţia[638] separatorului optimal ca funcţie de h atunci când punctele sunt separabile? (Da / Nu; justificare.)

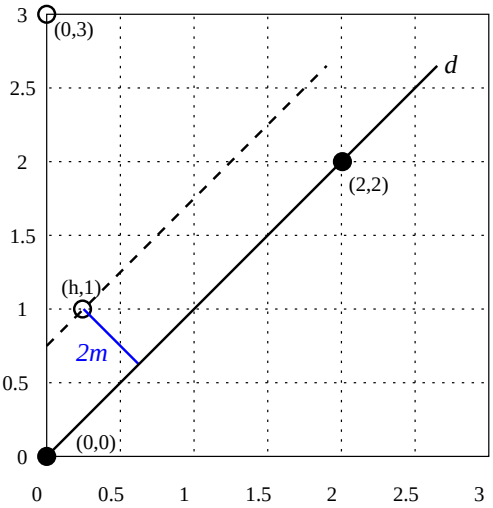
c. Cât este *marginea* [*geometrică*] corespunzătoare separatorului optimal (adică, distanţa de la separator până la vectorii-suport) ca funcţie de h?



d. Presupunem că putem observa doar a doua componentă a instanţelor de antrenament. Fără cealaltă componentă, datele de antrenament etichetate se reduc la $(0,+)$, $(1,-)$, $(2,+)$ şi $(3,-)$. Care este gradul minim p al unei funcţii polinomiale care, aplicată acestor date, ne permite să le clasificăm corect?

Răspuns:

a. Dreapta d, care este determinată de instanţele pozitive $x_1 = (0,0)$ şi $x_2 = (2,2)$ are ecuaţia $y = x$. Pentru ca punctele de antrenament să rămână liniar separabile trebuie ca punctul x_3 să fie situat de aceeaşi parte a dreptei d cu instanţa negativă $x_4 = (0,3)$. Această condiţie se traduce analitic prin inegalitatea $h < 1$.



b. Hiperplanul de separare optimală va fi o dreaptă paralelă cu dreapta d (care are ecuaţia $y = x$), situată la jumătatea distanţei dintre aceasta şi punctul $x_3 = (h,1)$. Dreapta d este determinată de vectorii-suport pozitivi $x_1 = (0,0)$ şi $x_2 = (2,2)$ pe de o parte şi de vectorul-suport negativ $x_3 = (h,1)$ de cealaltă parte. Panta dreptei d este complet determinată de cei doi vectori-suport pozitivi, deci este 1; ea nu depinde de valoarea parametrului h. În concluzie, direcţia separatorului optimal nu se schimbă în funcţie de parametrul h, atâta timp cât cele patru instanţe rămân separabile.

[638] Direcţia unei drepte din planul euclidian este dată de *panta* ei, adică tangenta unghiului format de acea dreaptă cu axa Ox. O definiţie alternativă este următoarea: direcţia unei drepte este reprezentată de mulţimea tuturor dreptelor paralele cu ea.

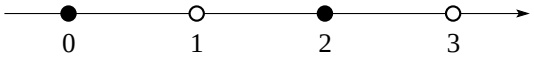
c. Marginea separatorului optimal este $m = \frac{1}{2} dist(x_3, d)$. Dreapta d are ecuaţia $-x + y = 0$, iar punctul x_3 este $(h, 1)$. Ţinând cont de formula care ne dă distanţa de la un punct oarecare la o dreaptă de ecuaţie cunoscută,[639] rezultă că

$$m = \frac{1}{2} \cdot \frac{|(-1)h + 1|}{\sqrt{(-1)^2 + 1^2}} = \frac{1}{2} \cdot \frac{|1 - h|}{\sqrt{2}}$$

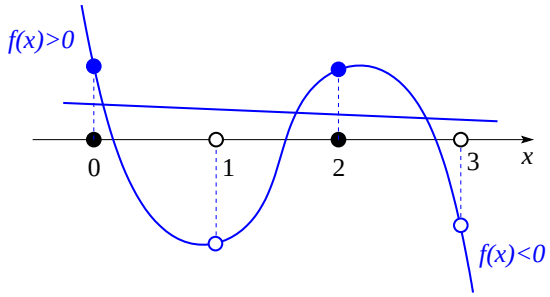
Ca urmare a combinării restricţiei $h \in [0, 3]$ din enunţ cu restricţia $h < 1$ rezultată la punctul a, vom avea $h \in [0, 1)$, deci expresia lui m de mai sus devine:

$$m = \frac{1 - h}{2\sqrt{2}} = \frac{(1 - h)\sqrt{2}}{4}$$

d. Punctele de antrenare $(0, +)$, $(1, -)$, $(2, +)$ şi $(3, -)$ se reprezintă pe axa reală după cum urmează:



Pentru a clasifica corect aceste puncte este nevoie de o funcţie polinomială de forma reprezentată în figura alăturată. Se observă că „proiecţiile“ instanţelor de antrenament pe curba polinomială sunt separabile liniar în planul euclidian. Aşadar, gradul minim p al unei funcţii polinomiale care ne permite să clasificăm corect aceste puncte este 3. De pildă, putem considera funcţia $-(x - 0.5)(x - 1.5)(x - 2.5)$.



7. **(O formă echivalentă cu forma primală a problemei de optimizare SVM)**

University of Utah, 2008 spring, Hal Daumé III, HW1C, pr. 3

Forma primală a problemei de optimizare SVM cu margine "hard" este definită astfel:[640]

$$\min_{w, w_0} \ \frac{1}{2} \mid\mid w \mid\mid^2$$
$$y_i(w \cdot x_i + w_0) \geq 1 \ \ (\forall 1 \leq i \leq m)$$

Arătaţi că soluţia acestei probleme rămâne efectiv neschimbată atunci când numărul 1 din partea dreaptă a restricţiei este înlocuit cu o constantă oarecare pozitivă a.

Răspuns:

[639] Vedeţi problema 1.
[640] Vedeţi problema .a.

Setul de inegalităţi $y_i(w \cdot x_i + w_0) \geq 1$, cu $i = 1, \ldots, m$ constituie o condiţie suficientă pentru separabilitatea liniară a instanţelor de antrenament $x_i \in \mathbb{R}^d$. Această inegalitate devine egalitate exclusiv pentru punctele care sunt vectori-suport. Hiperplanul de separare optimală are ecuaţia $w \cdot x + w_0 = 0$. Distanţa dintre acest hiperplan şi oricare dintre vectorii-suport este deci $1/ \mid\mid w \mid\mid$.

Dacă numărul 1 din partea dreaptă a restricţiei este înlocuit cu o constantă oarecare pozitivă a, vom avea următoarea echivalenţă între probleme:

$$\begin{array}{l} \min_{w,w_0} \ \frac{1}{2} \mid\mid w \mid\mid^2 \\ y_i(w \cdot x_i + w_0) \geq a \ \ (i = 1, \ldots, m) \end{array} \Leftrightarrow \begin{array}{l} \min_{w,w_0} \ \frac{1}{2} \mid\mid w \mid\mid^2 \\ y_i(\frac{1}{a} w \cdot x_i + \frac{1}{a} w_0) \geq 1 \ \ (i = 1, \ldots, m) \end{array} \Leftrightarrow$$

$$\begin{array}{l} \min_{w,w_0} \ \frac{1}{2} \mid\mid \frac{1}{a} w \mid\mid^2 \\ y_i(\frac{1}{a} w \cdot x_i + \frac{1}{a} w_0) \geq 1 \ \ (i = 1, \ldots, m) \end{array} \Leftrightarrow \begin{array}{l} \min_{w',w'_0} \ \frac{1}{2} \mid\mid w' \mid\mid^2 \\ y_i(w' \cdot x_i + w'_0) \geq 1 \ \ (i = 1, \ldots, m) \end{array}$$

unde $w' \stackrel{not.}{=} \frac{1}{a} w$ şi $w'_0 \stackrel{not.}{=} \frac{1}{a} w_0$. Echivalenţa se datorează pozitivităţii constantei a. Aşadar, până la un factor pozitiv $(1/a)$, soluţia (w, w_0) rămâne neschimbată.

Evident, hiperplanul de separare optimală rămâne acelaşi. La fel, mulţimea vectorilor-suport este neschimbată. Distanţa dintre hiperplan şi vectorii-suport devine $a \ / \mid\mid w \mid\mid$, iar egalitatea $y_i(w \cdot x_i + w_0) = a$ va fi adevărată doar pentru punctele x_i care sunt vectori-suport.

8. **(Exemplu de folosire a unei funcţii de mapare a atributelor cu scopul de a obţine separabilitate liniară; rezolvarea directă a problemei SVM primale în spaţiul [nou] de trăsături; identificarea separatorului neliniar din spaţiul iniţial [de trăsături])**

• ***CMU, 2009 fall, Carlos Guestrin, HW3, pr. 2.1***

Se dă un set de date de antrenament D, caracterizate de atributul X care ia valori în $\mathbb{R}$ şi de eticheta corespunzătoare $y \in \{+1, -1\}$. Acest set de date este constituit din trei exemple pozitive şi anume pentru $X \in \{-3, -2, 3\}$ şi trei exemple negative pentru $X \in \{-1, 0, 1\}$.

a. Putem separa acest set de date (în spaţiul de trăsături specificat mai sus) folosind un separator liniar? De ce da, sau de ce nu?

b. Definim funcţia $\Phi(x) = (x, x^2)$ care transformă / „mapează“ punctele din $\mathbb{R}$ în $\mathbb{R}^2$. Aplicaţi funcţia Φ datelor din D şi trasaţi imaginea lor în $\mathbb{R}^2$, noul spaţiu de trăsături. Poate un separator liniar să separe în mod perfect punctele din noul spaţiu de trăsături, obţinut prin funcţia Φ? De ce da, sau de ce nu?

c. Găsiţi forma analitică a funcţiei-nucleu $K(x, x')$ care corespunde transformării Φ. Vă reamintim că $K(x, x') \stackrel{def.}{=} \Phi(x) \cdot \Phi(x')$.

d. Identificaţi un *hiperplan de separare optimală* pentru mulţimea $\Phi(D)$. Acest hiperplan este o dreaptă în planul euclidian, caracterizată de o ecuaţie de forma $w_1 Y_1 + w_2 Y_2 + w_0 = 0$, unde $\Phi(X) = (Y_1, Y_2)$. Găsiţi valorile lui w_0, w_1 şi w_2; conform problemei de optimizare SVM, pentru fiecare vector-suport X având eticheta y trebuie să avem satisfăcută restricţia $w_1 Y_1 + w_2 Y_2 + w_0 = y$.

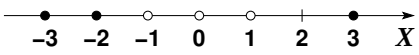
Desenaţi hiperplanul de separare optimală şi încercuiţi vectorii-suport. De asemenea, calculaţi *marginea* hiperplanului, adică distanţa de la hiperplan la vectorii-suport.

e. Desenaţi în $\mathbb{R}$ corespondentul hiperplanului de separare optimală din $\mathbb{R}^2$.

f. Dacă adăugăm încă un punct pozitiv ($y=+1$) la mulţimea de antrenament, şi anume $X = 5$, se va schimba hiperplanul de separare optimală sau marginea lui? De ce da, sau de ce nu?

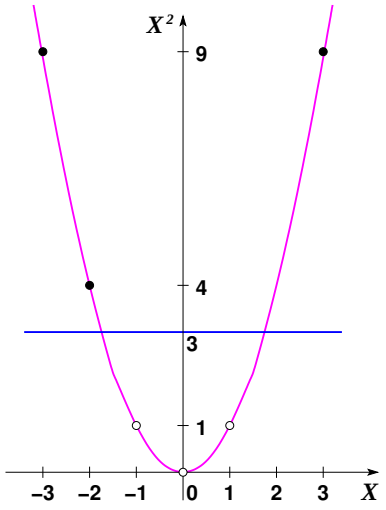
Răspuns:

a. Figura de mai jos reprezintă datele de antrenament în spaţiul original, $\mathbb{R}$, folosind puncte negre pentru exemplele pozitive şi puncte albe pentru exemplele negative.



Conform *definiţiei*, un separator liniar în mulţimea numerelor reale este un punct de pe axă care are de o parte a sa toate exemplele pozitive şi de cealaltă parte toate exemplele negative. Se observă foarte uşor că nu există niciun punct pe axa reală care să satisfacă această condiţie. Prin urmare, mulţimea datelor de antrenament nu este separabilă liniar în spaţiul de trăsături iniţial.

b. Figura alăturată reprezintă imaginea setului de antrenament în noul spaţiu de trăsături, $\mathbb{R}^2$. Se observă uşor că aici datele sunt separabile liniar. Putem lua, de exemplu, dreapta de ecuaţie $y - 3 = 0$, reprezentată în figura alăturată. Avem deci $w_1 = 0, w_2 = 1, w_0 = -3$. *Analitic*, se observă imediat că sunt îndeplinite inegalităţile de forma $sign(w_1 u_i + w_2 u_i^2 + w_0) = y_i$:

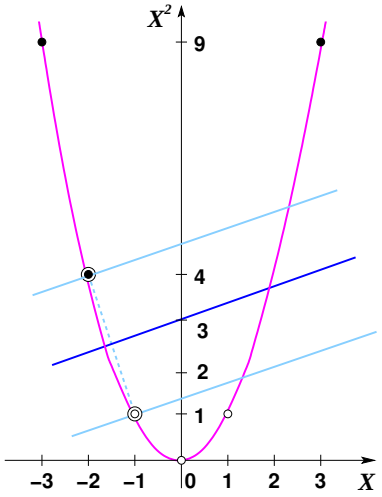


$$\begin{cases} 0\cdot(-3)+1\cdot 9-3 &= 6>0 \\ 0\cdot(-2)+1\cdot 4-3 &= 1>0 \\ 0\cdot(-1)+1\cdot 1-3 &= -2<0 \\ 0\cdot 0+1\cdot 0-3 &= -3<0 \\ 0\cdot 1+1\cdot 1-3 &= -2<0 \\ 0\cdot 3+1\cdot 9-3 &= 6>0 \end{cases}$$

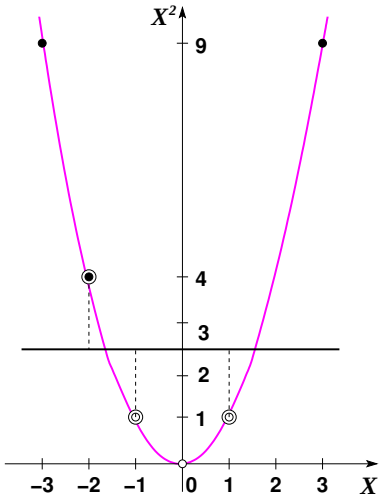
c. Conform *definiţiei* funcţiei-nucleu,

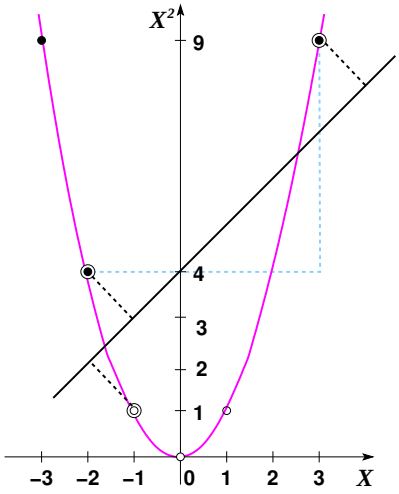
$$K(x,x') = \Phi(x)\cdot\Phi(x') = (x,x^2)\cdot(x',x'^2) \Rightarrow K(x,x') = xx' + x^2x'^2.$$

d. Hiperplanul de separare optimală pentru setul de date de antrenament D din enunţ este determinat de punctele $(-2,4)$ şi $(-1,1)$. Se observă că aceste două puncte constituie perechea de puncte de semne / etichete diferite aflate la distanţă minimă, între toate perechile de puncte de semne contrare din D. Mediatoarea punctelor $(-2,4)$ şi $(-1,1)$ este hiperplanul de separare maximală pentru mulţimea D, iar cele două puncte sunt vectorii-suport. Marginea, adică distanţa dintre orice vector-suport şi hiperplanul de separare optimală este $\frac{\sqrt{10}}{2} = \sqrt{\frac{5}{2}}$.



***Observaţie*: Se poate verifica [şi] în mod *analitic* că, dintre toţi separatorii liniari ai mulţimii D, dreapta de mai sus maximizează distanţa până la cele mai apropiate instanţe pozitive şi respectiv negative. În particular, comparând marginea corespunzătoare mediatoarei punctelor $(-2,4)$ şi $(-1,1)$ cu marginile determinate respectiv de către cele două drepte din figurile de mai jos, vom obţine: $\sqrt{\frac{5}{2}} > \frac{3}{2}$ pentru figura din partea stângă şi $\sqrt{\frac{5}{2}} > \sqrt{2}$ pentru figura din partea dreaptă.**[641]





Ecuaţia separatorului optimal se determină astfel:

[641] Mai mult, cele două drepte menţionate au poziţii extreme în raport cu toate dreptele care

- separă instanţele din D,
- trec prin punctul de la jumătatea segmentului determinat de punctele $(-2,4)$ şi $(-1,1)$, care sunt cele mai apropiate puncte de semne / etichete contrare din D,
- nu se apropie de punctul $(3,9)$ şi respectiv $(1,1)$ mai mult decât este distanţa până la fiecare din punctele $(-2,4)$ şi $(-1,1)$.

– **ecuaţia dreptei determinate de punctele** $(-2,4)$ **şi** $(-1,1)$ **este**

$$\frac{x-(-1)}{-2-(-1)}=\frac{y-1}{4-1}\Leftrightarrow 3(x+1)=-(y-1)\Leftrightarrow y=-(3x+2),$$

aşadar, panta acestei drepte este -3;

– **orice dreaptă perpendiculară pe dreapta de mai sus are panta** $-\frac{1}{-3}=\frac{1}{3}$ **şi deci este determinată de o ecuaţie de forma** $y=\frac{1}{3}x+c$**, unde** c **este o constantă reală;**

– **mijlocul segmentului determinat de punctele** $(-2,4)$ **şi** $(-1,1)$ **este** $\left(-\frac{3}{2},\frac{5}{2}\right)$**;**

– **mediatoarea segmentului determinat de punctele** $(-2,4)$ **şi** $(-1,1)$ **va determina valoarea corespunzătoare pentru constanta** c**, prin faptul că punctul** $\left(-\frac{3}{2},\frac{5}{2}\right)$ **aparţine acestei mediatoare, deci:**

$$\frac{5}{2}=\frac{1}{3}\left(-\frac{3}{2}\right)+c\Leftrightarrow c=\frac{5}{2}+\frac{1}{2}=3;$$

– **aşadar, ecuaţia separatorului optimal este:**

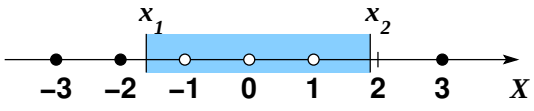
$$y=\frac{1}{3}x+3\Leftrightarrow x-3y+9=0\Leftrightarrow -\frac{1}{5}x+\frac{3}{5}y-\frac{9}{5}=0.$$

În legătură cu ultima formă a ecuaţiei de mai sus se poate observa că expresia $-\frac{1}{5}x+\frac{3}{5}y-\frac{9}{5}$ **satisface condiţia — exprimată în definiţia problemei SVM, forma primală — de a produce valorile** 1 **şi respectiv** -1 **pentru punctele** $(-2,4)$ **şi** $(-1,1)$**. Aşadar, funcţia care defineşte separatorul optimal este** $-\frac{1}{5}x+\frac{3}{5}y-\frac{9}{5}$**, iar soluţiile sunt** $\bar{w}=\left(-\frac{1}{5},\frac{3}{5}\right)$ **şi** $\bar{w}_0=-\frac{9}{5}$**. Rezultă că marginea este** $\frac{1}{||\bar{w}||}=\frac{5}{\sqrt{1+9}}=\sqrt{\frac{5}{2}}$**.**

e. Clasificarea unui punct $x\in\mathbb{R}$ **cu separatorul determinat la punctul** d **se face în felul următor:**

$$y=sign(\bar{w}\cdot\Phi(x)+\bar{w}_0)=sign\left(\left(-\frac{1}{5},\frac{3}{5}\right)\cdot(x,x^2)-\frac{9}{5}\right)=sign\left(\frac{3}{5}\,x^2-\frac{1}{5}\,x-\frac{9}{5}\right)$$

Prin urmare, vom avea eticheta $y=-1$ **dacă şi numai dacă** $3x^2-x-9<0$**, adică** x **aparţine intervalului** (x_1,x_2)**, unde** $x_1=\frac{1-\sqrt{1+12\cdot 9}}{6}\approx -1.57$ **iar** $x_2=\frac{1+\sqrt{1+12\cdot 9}}{6}\approx 1.90$**. Corespondentul hiperplanului de separare din** $\mathbb{R}^2$ **în spaţiul iniţial de trăsături este ilustrat în figura de mai jos, sub forma „tăieturilor“ reprezentate de** x_1 **şi** x_2**:**



***Observaţie*: Rezultatul de mai sus corespunde (intuitiv) cu rezultatul de la punctul precedent (d): instanţele (x, x^2) situate de partea „pozitivă“ a dreptei de ecuaţie $-\frac{1}{5}x+\frac{3}{5}y-\frac{9}{5}=0 \Leftrightarrow y=\frac{1}{3}x+3$ satisfac condiţia $x^2 > y \Leftrightarrow x^2 > \frac{1}{3}x+3 \Leftrightarrow 3x^2-x-9>0$. Similar, instanţele (x, x^2) situate de partea „negativă“ a dreptei de ecuaţie $-\frac{1}{5}x+\frac{3}{5}y-\frac{9}{5} \Leftrightarrow y=\frac{1}{3}x+3$ satisfac condiţia $x^2 < y \Leftrightarrow x^2 < \frac{1}{3}x+3 \Leftrightarrow 3x^2-x-9<0$.**

f. Se observă uşor (vedeţi prima figură de la punctul d) că imaginea noului exemplu de antrenament (5, 25) cade de partea „pozitivă“ a hiperplanului de separare şi, mai mult, cade în afara marginii. Prin urmare, separatorul optimal rămâne acelaşi.

9. **(Exerciţiu teoretic: deducerea formei duale pentru problema SVM cu margine “hard”)**

prelucrare de L. Ciortuz, după
■ ***CMU, 2010 fall, Ziv Bar-Joseph, HW4, pr. 1.3-5***

Se consideră vectorii de intrare $x_1, \ldots, x_m \in \mathbb{R}^d$ şi etichetele corespunzătoare $y_1, \ldots, y_m \in \{-1, 1\}$. Problema SVM cu margine “hard” — termen care desemnează cazul în care se presupune că instanţele $(x_1, y_1), \ldots, (x_n, y_n)$ sunt liniar separabile — este o *problemă de optimizare convexă*, exprimată sub *forma primală* astfel:

$$\begin{aligned} &\min_{w,w_0} \ \frac{1}{2}\|w\|^2 \\ &\textbf{a. î.} \quad (w \cdot x_i + w_0)y_i \geq 1, \ \textbf{pentru } i = 1, \ldots, m, \end{aligned} \tag{P}$$

unde $w \in \mathbb{R}^d$ şi $w_0 \in \mathbb{R}$. În urma rezolvării acestei probleme se va obţine un *model* liniar, de forma $y(x) = w \cdot x + w_0$, ce va servi ulterior pentru clasificare, conform funcţiei de decizie $sign(y(x))$.[642]

La curs am prezentat *metoda dualităţii Lagrange*,[643] **care ne permite în anumite condiţii să rezolvăm probleme de *optimizare convexă cu restricţii de asemenea convexe*, transpunând aceste probleme într-o formă mai convenabilă numită *forma duală*. Pentru început, vom defini o altă funcţie, numită *lagrangeanul generalizat*,**[644] **care combină funcţia obiectiv din (P) cu expresiile**

[642] Este uşor de constatat faptul că problema (P) este un caz particular de problemă de optimizare convexă cu restricţii (vedeţi *Introducerea* de la pr. 82 de la capitolul de *Fundamente*) şi, conform teoremei Karush-Kuhn-Tucker (vedeţi *Comentariul* de la pr. 83 tot de la capitolul de *Fundamente*), ea are soluţie atunci când *regiunea fezabilă* (engl., feasible region), adică mulţimea convexă formată din valorile lui w şi w_0 pentru care *restricţiile* $(w \cdot x_i + w_0)y_i \geq 1$ cu $i = \overline{1,m}$ sunt satisfăcute este nevidă.

Din punct de vedere *computaţional*, rezolvarea acestei probleme devine ineficientă atunci când m, numărul de instanţe de antrenament, este foarte mare. Dificultatea ţine de satisfacerea restricţiilor. Ca să depăşim această dificultate, *ideea* de bază este să punem problema noastră sub o altă formă, cu restricţii mai simple. „Preţul“ pe care va trebui să-l plătim în schimb este „complicarea“ funcţiei obiectiv; se va dovedi ulterior că acest schimb este convenabil.

[643] Vedeţi problemele 82 şi 83 de la capitolul de *Fundamente*.

[644] Unii autori numesc funcţia L_P *lagrangeanul primal*. Pentru problema de optimizare convexă (cazul general), el se defineşte astfel:

$$L_P(x, \alpha, \beta) \overset{def.}{=} f(x) + \sum_i \alpha_i g_i(x) + \sum_j \beta_i h_j(x).$$

Vedeţi *Introducerea* de la pr. 82 de la capitolul de *Fundamente*.

care intervin în partea stângă a restricţiilor:[645]

$$L_P(w, w_0, \alpha) \stackrel{def.}{=} \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m} \alpha_i[(w \cdot x_i + w_0)y_i - 1],$$

unde $\alpha_i \geq 0$ **pentru** $i = \overline{1, m}$ **sunt aşa-numiţii multiplicatori Lagrange sau *variabilele duale*. Variabilele *primale* sunt** w **şi** w_0**.**

***Comentariu*:** Vă readucem aminte următoarele *puncte principale* [ale *teoriei dualităţii Lagrange*], care justifică de ce anume procedăm aşa cum procedăm în rezolvarea problemei noastre.

- [P→P1] Este relativ uşor de arătat[646] că problema (P) este echivalentă cu următoarea problemă de optimizare

$$\begin{array}{rl} & \min_x \max_{\alpha,\beta} L_P(x, \alpha, \beta) \\ \text{a. î.} & \alpha_i \geq 0 \text{ pentru } i = 1, \ldots, m. \end{array} \tag{P1}$$

  Se observă că în această problemă restricţiile sunt mult mai simple decât în problema (P).

- [P1→D1] Inversând operatorii min şi max în problema (P1), obţinem problema de optimizare

$$\begin{array}{rl} & \max_{\alpha,\beta} \min_x L_P(x, \alpha, \beta) \\ \text{a. î.} & \alpha_i \geq 0 \text{ pentru } i = 1, \ldots, m. \end{array} \tag{D1}$$

  Dacă notăm cu p^* optimul problemei (P1) (deci şi al problemei (P)) şi cu d^* optimul problemei (D1), este relativ uşor de demonstrat[647] proprietatea de *dualitate slabă*:

$$p^* \geq d^*. \tag{311}$$

  Proprietatea de dualitate slabă este valabilă şi în cazul general al problemelor de optimizare convexă [cu restricţii].

- Este imediat că pentru problema (P1) este satisfăcută aşa-numita *condiţie a lui Slater*:[648]

$$\exists w \in \mathbb{R}^d \text{ şi } w_0 \in \mathbb{R} \text{ astfel încât } (w \cdot x_i + w_0)y_i - 1 > 0, \text{ pentru } i = 1, \ldots, m. \tag{312}$$

  Într-adevăr, faptul că instanţele $(x_1, y_1), \ldots, (x_n, y_n)$ sunt liniar separabile implică imediat satisfacerea condiţiei de mai sus. Se demonstrează că în general, adică pentru orice problemă de optimizare convexă cu restricţii, dacă este îndeplinită condiţia lui Slater (sub forma $\exists x$ a. î. $g_i(x) < 0$ pentru $i = 1, \ldots, n$ şi $h_j(x) = 0$ pentru $j = 1, \ldots, p$), atunci are loc egalitatea

$$p^* = d^*,$$

  care reprezintă proprietatea de *dualitate tare*. Aşadar, optimul problemei (D1) este optim şi pentru problema (P1) şi, corespunzător, pentru problema (P).

[645] Semnul minus (−) din faţa simbolului de sumare ($\sum$) din funcţia L_P corespunzătoare problemei de optimizare SVM în forma primală (P) apare din cauza faptului că restricţiile din cadrul acestei probleme sunt de tip $\geq$, în timp ce restricţiile din cadrul formulării generale a problemei de optimizare convexă (vedeţi *Introducerea* de la problema 82 de la capitolul de *Fundamente*) sunt de tip $\leq$.

[646] Vedeţi documentul *Convex Optimization Overview (cont'd)*, de Chuong B. Do, 2009, pag. 4-5 sau documentul *Support Vector Machines* de Andrew Ng, (Stanford University, CS229 Lecture Notes, Part V), pag. 8-9.

[647] Vedeţi problema 82 de la capitolul de *Fundamente* şi / sau documentul *Convex Optimization Overview (cont'd)*, de Chuong B. Do, 2009, pag. 5-6.

[648] Vedeţi *Learning with Kernels*, B. Schölkopf, A. Smola, MIT Press, 2002, pag. 167.

O *teoremă* importantă din teoria problemelor de optimizare convexă[649] demonstrează următoarea *implicaţie*: în cazul în care proprietatea de dualitate tare este satisfăcută, notând cu $\bar{\alpha}$, $\bar{\beta}$ o soluţie a problemei (D1), urmează că

$$\bar{\alpha}_i g_i(\bar{x}) = 0 \text{ pentru } i = 1, \ldots, n.$$

Aceste egalităţi se numesc *condiţiile de complementaritate duală Karush-Kuhn-Tucker*.

Pentru problema SVM cu margine "hard" din enunţ, considerând $\bar{w}$, $\bar{w}_0$ şi $\bar{\alpha}$ soluţiile problemei (D1) (deci şi ale problemei (P1)), condiţiile de complementaritate duală Karush-Kuhn-Tucker se vor exprima astfel:

$$\bar{\alpha}_i((\bar{w} \cdot x_i + \bar{w}_0)y_i - 1) = 0 \text{ pentru } i = 1, \ldots, m, \tag{313}$$

adică, pentru fiecare valoare admisibilă a lui i, avem fie $\bar{\alpha}_i = 0$, fie $\bar{\alpha}_i > 0$ şi, în consecinţă, $(\bar{w} \cdot x_i + \bar{w}_0)y_i - 1 = 0$.

- [D1→D] În fine, pentru că funcţia L_P este derivabilă în raport cu w şi respectiv w_0, vom putea rescrie expresia lui $\min_{w,w_0} L_P(w, w_0, \alpha)$ din cadrul funcţiei obiectiv a problemei (D1) calculând în prealabil rădăcinile derivatelor parţiale ale lui L_P. (Se poate arăta[650] că funcţia $\min_{w,w_0} L_P(w, w_0, \alpha)$ este concavă.) Se va ajunge astfel la a asocia problemei (P) forma duală (D) din enunţ; vedeţi punctul *b*.

  Raţionamentul acesta este valabil în general pentru problemele de optimizare convexă în care toate funcţiile f, g_i şi h_j sunt convexe şi derivabile.

  Condiţiile ca derivatele parţiale ale lui $L_P(w, w_0, \alpha)$ în raport cu w şi respectiv w_0 să se anuleze se numesc *condiţiile de staţionaritate* (sau: *optimalitate*) *Karush-Kuhn-Tucker*.

a. Calculând derivatele parţiale ale funcţiei L_P în raport cu variabilele primale w şi w_0, arătaţi că între valorile $\bar{w}$, $\bar{w}_0$ şi $\bar{\alpha}$ pentru care aceste derivate parţiale se anulează există relaţiile:

$$\bar{w} = \sum_{i=1}^{m} \bar{\alpha}_i x_i y_i, \tag{314}$$

$$\sum_{i=1}^{m} \bar{\alpha}_i y_i = 0. \tag{315}$$

De asemenea, arătaţi că din *condiţia de complementaritate Karush-Kuhn-Tucker* se poate deduce relaţia:

$$\bar{w}_0 = y_i - \bar{w} \cdot x_i \text{ pentru orice } i \text{ astfel încât } \bar{\alpha}_i > 0. \tag{316}$$

***Notă*: Instanţele x_i pentru care $\bar{\alpha}_i > 0$ sunt numite vectori-suport [în sens *clasic*, analitic].[651] Aceasta este definiţia pe care o vom folosi de acum încolo pentru această noţiune. Observaţi că în relaţia (314) soluţia $\bar{w}$ se scrie ca o combinaţie liniară de vectorii-suport!**

b. Calculaţi funcţia $L_D(\alpha)$ — numită *lagrangeanul dual* — care se obţine din expresia lagrangeanului generalizat $L_P(w, w_0, \alpha)$ substituind variabila w cu

[649] Vedeţi documentul *Convex Optimization Overview (cont'd)*, de Chuong B. Do, 2009, pag. 6 sau *Learning with Kernels*, B. Schölkopf, A. Smola, MIT Press, 2002, pag. 165, Teorema 6.21.
[650] Vedeţi documentul *Convex Optimization Overview (cont'd)*, de Chuong B. Do, 2009, pag. 5.
[651] Vedeţi *Observaţia importantă* de la pag. 645.

$\sum_{i=1}^m \alpha_i x_i y_i$ şi folosind egalitatea $\sum_{i=1}^m \alpha_i y_i = 0$, conform relaţiilor (314) şi (315) de la punctul precedent.

***Observaţie* (1): Este evident acum că problemei (P) îi poate fi asociată următoarea formă (numită „duală"):**

$$\begin{aligned} & \max_\alpha L_D(\alpha) \\ \textbf{a. î.} \quad & \alpha_i \geq 0, \ \textbf{pentru } i = 1, \ldots, m \\ & \textstyle\sum_{i=1}^m \alpha_i y_i = 0 \end{aligned} \qquad \textbf{(D)}$$

în care *restricţiile* sunt mult mai *simple* decât erau în forma primală (P). Este de reţinut faptul că relaţiile (314) şi (316) de la punctul precedent constituie legătura dintre soluţia problemei (D) şi soluţia problemei (P).

c. Dacă se dă o instanţă nouă (de test) x_{new}, cum veţi decide clasa ei?

Răspuns:

a. Calculăm mai întâi derivatele parţiale ale funcţiei L_P în raport cu w şi respectiv w_0:[652]

$$\frac{\partial}{\partial w} L_P(w, w_0, \alpha) = w - \sum_{i=1}^m \alpha_i x_i y_i$$
$$\frac{\partial}{\partial w_0} L_P(w, w_0, \alpha) = -\sum_{i=1}^m \alpha_i y_i.$$

Atunci când se atinge optimul funcţiei L_P (considerând argumentul său α fixat), aceste derivate parţiale devin egale cu 0. Din $\frac{\partial}{\partial w} L_P(\bar{w}, \bar{w}_0, \bar{\alpha}) = 0$ rezultă că

$$\bar{w} = \sum_{i=1}^m \bar{\alpha}_i x_i y_i.$$

Similar, $\frac{\partial}{\partial w_0} L_P(\bar{w}, \bar{w}_0, \bar{\alpha}) = 0$ implică relaţia $\sum_{i=1}^m \bar{\alpha}_i y_i = 0$.

În fine, din condiţia de complementaritate Karush-Kuhn-Tucker, care se exprimă aici sub forma

$$\bar{\alpha}_i[(\bar{w} \cdot x_i + \bar{w}_0) y_i - 1] = 0 \ \textbf{pentru } i = 1, \ldots, m$$

[652] În notaţia matriceală, considerând w şi x_i pentru $i = 1, \ldots, m$ vectori-coloană, putem scrie lagrangeanul L_P astfel:

$$L_P(w, w_0, \alpha) = \frac{1}{2} w^\top w - \sum_{i=1}^m \alpha_i ((w^\top x_i + w_0) y_i - 1).$$

Pentru derivarea lui L_P în raport cu vectorul w, se folosesc regulile (5a) şi (5b) din documentul *Matrix Identities* de Sam Roweis (New York University, June 1999), pe care le-am folosit (de exemplu) la problema 88 de la capitolul *Fundamente*, problema 3.d de la capitolul *Metode de regresie*, precum şi la problema 24 de la capitolul *Clusterizare*.

Revenind la formula iniţială, care foloseşte notaţia vectorială şi produsul scalar din $\mathbb{R}^d$, se poate constata că într-un astfel de caz (simplu!), regulile de derivare sunt similare cu cele din $\mathbb{R}$.

rezultă că pentru orice $i \in \{1, \ldots, m\}$ cu $\bar{\alpha}_i > 0$ avem $(\bar{w} \cdot x_i + \bar{w}_0)y_i - 1 = 0$.[653] Această relaţie este echivalentă cu $\bar{w} \cdot x_i + \bar{w}_0 = y_i$ fiindcă $y_i \in \{-1, 1\}$. Din această ultimă egalitate rezultă:

$$\bar{w}_0 = y_i - \bar{w} \cdot x_i.$$

b. Substituind $w = \sum_{i=1}^{m} \bar{\alpha}_i x_i y_i$ în expresia lui L_P şi ţinând cont că $\sum_{i=1}^{m} \bar{\alpha}_i y_i = 0$, vom obţine:

$$\begin{aligned} L_D(\alpha) &= \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j - \sum_{i=1}^{m} \alpha_i [(\sum_{j=1}^{m} \alpha_j y_j x_j \cdot x_i + w_0) y_i - 1] \\ &= \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \end{aligned} \tag{317}$$

c. Mai întâi vom calcula

$$\begin{aligned} f(x_{new}) &= \bar{w} \cdot x_{new} + \bar{w}_0 \\ &= \Big(\sum_{i=1}^{m} \bar{\alpha}_i y_i x_i\Big) \cdot x_{new} + \bar{w}_0 = \sum_{i=1}^{m} \bar{\alpha}_i y_i x_i \cdot x_{new} + \bar{w}_0, \end{aligned} \tag{318}$$

unde $\bar{\alpha}$ este soluţia problemei duale (D), iar $\bar{w}$ şi $\bar{w}_0$, soluţiile problemei primale (P) sunt calculate conform relaţiilor (314) şi (316) de la punctul a.

După aceea, dacă $f(x_{new}) \geq 0$ atunci x_{new} va fi clasificat pozitiv, iar în caz contrar va fi clasificat negativ.

***Observaţie* (2): Remarcăm faptul că atât în funcţia obiectiv a problemei de optimizare SVM în formă duală (D) cât şi în funcţia f care serveşte la clasificarea instanţelor noi, operaţiile care se execută asupra instanţelor sunt doar de tip produs scalar: $x_i \cdot x_j$ şi respectiv $x_i \cdot x_{new}$. Acest fapt face posibilă folosirea *funcţiilor-nucleu* în contextul SVM (aşa cum vom exemplifica la problema 10), ceea ce este convenabil atât din punctul de vedere al obţinerii (eventuale) a separabilităţii, cât şi din punctul de vedere al executării eficiente a calculelor.**

10. (Învăţarea conceptului ¬XOR folosind forma duală a problemei SVM şi o mapare particulară a trăsăturilor)

CMU, 2006 fall, E. Xing, T. Mitchell, midterm exam, pr. 5

Fie o problemă de învăţare supervizată în care exemplele de antrenare se află în spaţiul euclidian bidimensional. Exemplele pozitive sunt $x_1 = (1, 1)$ şi $x_3 = (-1, -1)$ iar exemplele negative sunt $x_2 = (-1, 1)$ şi $x_4 = (1, -1)$.

[653] Dacă $\bar{\alpha}_i = 0$ pentru $i = \overline{1, m}$, din relaţia $\bar{w} = \sum_{i=1}^{m} \bar{\alpha}_i y_i x_i$ rezultă că $\bar{w} = 0$. În consecinţă, funcţia $f(x) = \bar{w} \cdot x + \bar{w}_0$ care dă ecuaţia separatorului optimal ($f(x) = 0$) nu va avea putere de discriminare între instanţele pozitive şi instanţele negative (indiferent de valoarea atribuită lui $\bar{w}_0$, care este o constantă).

De fapt, atunci când s-a introdus problema SVM ca o problemă de optimizare a marginii / distanţei $\frac{1}{||w||}$, s-a considerat în mod implicit că se caută soluţii $w \neq 0$.

Aşadar, în cazul în care în urma rezolvării problemei primale, respectiv a celei duale — urmată în ultimul caz de aplicarea relaţiilor de legătură între cele două tipuri / seturi de soluţii — obţinem $\bar{w} = 0$, căutarea lui $\bar{w}_0$ (valoarea optimă pentru w_0) pur şi simplu nu are sens.

a. Sunt exemplele pozitive separabile liniar faţă de exemplele negative?

b. Consideră transformarea $\Phi : \mathbb{R}^2 \to \mathbb{R}^4$, definită astfel: $\Phi(u) = (1,\, u_1,\, u_2,\, u_1u_2)$, unde $u = (u_1, u_2)$. Funcţia de predicţie pe care vrem s-o obţinem este de forma $y(x) = w \cdot \Phi(x) + w_0$, unde $x \in \mathbb{R}^2$, $w \in \mathbb{R}^4$, $w_0 \in \mathbb{R}$ (cu x variabil şi w şi w_0 fixaţi), iar $\cdot$ reprezintă produsul scalar. Indicaţi coeficienţii w şi w_0 corespunzători unui hiperplan de separare cu margine maximă pentru imaginea punctelor de antrenament din enunţ în „spaţiul de trăsături" ($\mathbb{R}^4$).

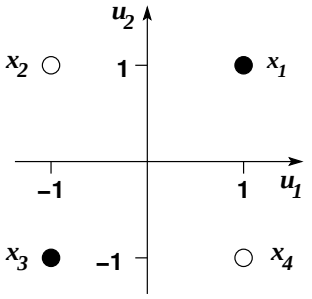
***Indicaţie*: Vă recomandăm să rezolvaţi problema SVM în *formă duală*,[654] iar apoi să exprimaţi *soluţia formei primale* folosind *relaţiile* dintre cele două tipuri de soluţii, care au fost date la curs.[655] (*Observaţie*: Se poate constata în prealabil că mulţimea $\{\Phi(x_1), \Phi(x_2), \Phi(x_3), \Phi(x_4)\}$ este liniar separabilă şi (ca atare) în *spaţiul de trăsături* determinat de maparea Φ este satisfăcută condiţia lui Slater.)**

c. Adăugaţi un nou exemplu la mulţimea de date de antrenament, astfel încît această nouă mulţime să nu mai fie separabilă liniar în spaţiul corespunzător transformării Φ de la punctul anterior.

d. Cărei funcţii-nucleu (engl., kernel function) îi corespunde transformarea Φ?

Răspuns:

a. Reprezentând grafic datele într-un sistem de coordonate bidimensional, se poate observa imediat că ele nu sunt separabile liniar. *Analitic*, putem demonstra acest lucru după cum urmează:
Să presupunem prin reducere la absurd că exemplele negative sunt separabile liniar de cele pozitive în $\mathbb{R}^2$. Atunci ar exista $\overline{w} = (w_1, w_2) \in \mathbb{R}^2$ şi $w_0 \in \mathbb{R}$ astfel încât:



$$\left.\begin{array}{l} w_1 + w_2 + w_0 > 0 \\ -w_1 - w_2 + w_0 > 0 \\ -w_1 + w_2 + w_0 < 0 \\ w_1 - w_2 + w_0 < 0 \end{array}\right\} \begin{array}{l} \Rightarrow \quad 2w_0 > 0 \Rightarrow w_0 > 0 \\ \\ \Rightarrow \quad 2w_0 < 0 \Rightarrow w_0 < 0 \end{array} \Bigg\} \Rightarrow \textbf{ contradicţie!}$$

Aşadar, presupunerea făcută este falsă. În consecinţă, cele 4 puncte nu sunt liniar separabile în $\mathbb{R}^2$.

b. Conform formei duale a problemei SVM, avem de maximizat funcţia „lagrangeană" L_D, unde:

$$L_D(\alpha_1, \alpha_2, \alpha_3, \alpha_4) \quad = \quad \sum_{i=1}^{4} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{4} \alpha_i \alpha_j y_i y_j \Phi(x_i) \cdot \Phi(x_j) \ ,$$

ţinând de asemenea cont de restricţiile $\sum_{i=1}^{4} y_i \alpha_i = \alpha_1 - \alpha_2 + \alpha_3 - \alpha_4 = 0$ şi $\alpha_i \geq 0$ pentru $i = \overline{1,4}$.

[654] Vedeţi problema de optimizare (D) de la pagina 660 şi relaţia (317).
[655] Adică relaţiile (314) şi (316).

Conform definiţiei funcţiei de mapare Φ, se observă că $\Phi(x_i) \cdot \Phi(x_i) = 4$ pentru $i = \overline{1,4}$ şi $\Phi(x_i) \cdot \Phi(x_j) = 0$ pentru orice $i, j = \overline{1,4}$ cu $i \neq j$. Înlocuind aceste produse în expresia lui L_D de mai sus, obţinem:

$\Phi(x_1)$	$\Phi(x_2)$	$\Phi(x_3)$	$\Phi(x_4)$
1	1	1	1
1	−1	−1	1
1	1	−1	−1
1	−1	1	−1

$$\begin{aligned} L_D(\alpha_1,\alpha_2,\alpha_3,\alpha_4) &= \alpha_1+\alpha_2+\alpha_3+\alpha_4-\frac{1}{2}(4\alpha_1^2+4\alpha_2^2+4\alpha_3^2+4\alpha_4^2) = \\ &= \alpha_1+\alpha_2+\alpha_3+\alpha_4-2\alpha_1^2-2\alpha_2^2-2\alpha_3^2-2\alpha_4^2. \end{aligned}$$

Vom încerca să rezolvăm problema duală căutând punctul de optim al lui L_D,[656] adică determinând $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ astfel încât derivatele parţiale ale lui L_D în funcţie de aceste variabile să se anuleze:

$$\left.\begin{aligned} \frac{\partial}{\partial\alpha_1}L_D(\alpha_1,\alpha_2,\alpha_3,\alpha_4) &= 1-4\alpha_1 = 0 \\ \frac{\partial}{\partial\alpha_2}L_D(\alpha_1,\alpha_2,\alpha_3,\alpha_4) &= 1-4\alpha_2 = 0 \\ \frac{\partial}{\partial\alpha_3}L_D(\alpha_1,\alpha_2,\alpha_3,\alpha_4) &= 1-4\alpha_3 = 0 \\ \frac{\partial}{\partial\alpha_4}L_D(\alpha_1,\alpha_2,\alpha_3,\alpha_4) &= 1-4\alpha_4 = 0 \end{aligned}\right\} \Rightarrow \bar{\alpha}_1=\bar{\alpha}_2=\bar{\alpha}_3=\bar{\alpha}_4=\frac{1}{4}.$$

Se observă că aceste soluţii satisfac restricţiile din forma duală a problemei SVM: $\bar{\alpha}_i \geq 0$ pentru $i = \overline{1,4}$ şi $\sum_{i=1}^{4} y_i\bar{\alpha}_i = \bar{\alpha}_1 - \bar{\alpha}_2 + \bar{\alpha}_3 - \bar{\alpha}_4 = 0$.

Pentru a determina vectorul $\overline{w}$ din soluţia formei primale a problemei SVM, vom folosi formula (314), care leagă soluţia formei duale de soluţia formei primale:

$$\begin{aligned} \overline{w} &= \sum_{i=1}^{4} y_i\bar{\alpha}_i\Phi(x_i) \\ &= \frac{1}{4}(1,1,1,1)-\frac{1}{4}(1,-1,1,-1)+\frac{1}{4}(1,-1,-1,1)-\frac{1}{4}(1,1,-1,-1)=(0,0,0,1). \end{aligned}$$

Aflarea valorii lui $\bar{w}_0$ se face folosind una dintre ecuaţiile (316), adică $\alpha_i(y_i(\overline{w} \cdot \Phi(x_i) + \bar{w}_0) - 1) = 0$ cu $i = \overline{1,4}$. Luând, spre exemplu, $i = 1$ obţinem:

$$\begin{aligned} &\frac{1}{4}(1((0,0,0,1)\cdot\Phi((1,1))+\bar{w}_0)-1)=0 \Leftrightarrow \\ &(0,0,0,1)\cdot(1,1,1,1)+\bar{w}_0-1=0 \Leftrightarrow 1+\bar{w}_0=1 \Leftrightarrow \bar{w}_0=0. \end{aligned}$$

Aşadar, ecuaţia separatorului optimal este

$$\overline{w}\cdot\Phi(u)+\bar{w}_0=0 \Leftrightarrow (0,0,0,1)\cdot(1,u_1,u_2,u_1u_2)+0=0 \Leftrightarrow u_1u_2=0,$$

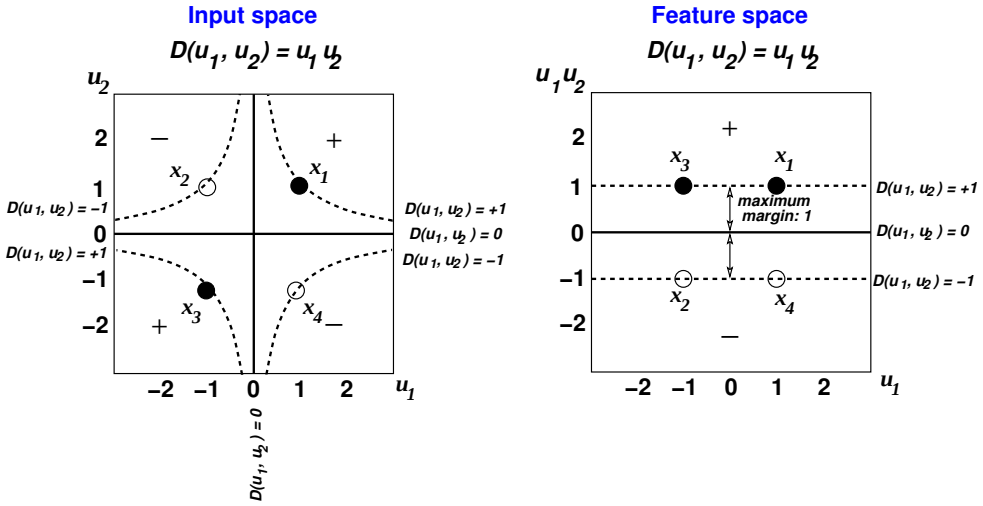
ceea ce corespunde mulţimii de puncte (u_1, u_2) formate din cele două axe de coordonate. Cele două „margini" au ecuaţiile $u_1u_2 = -1$ şi respectiv $u_1u_2 = 1$ şi sunt reprezentate grafic sub forma celor două hiperbole din figura de mai

[656] *Observaţie importantă*: Se poate constata imediat că *hessianul* lui $L_D(\alpha)$ este o matrice negativ definită, prin urmare lagrangeanul $L_D(\alpha)$ este funcţie concavă şi, deci, admite un maxim. A priori, este posibil ca acest punct de maxim să fie în afara intervalului de fezabilitate, adică să nu satisfacă restricţiile $\alpha_i \geq 0$, şi / sau să nu satisfacă condiţia $\sum_i y_i\alpha_i = 0$ din forma duală a problemei SVM. Vom vedea că, din fericire, în problema de faţă punctul de optim al lui $L_D(\alpha)$ satisface aceste două restricţii.

jos, partea stângă. Clasificarea unei instanţe oarecare de test $u = (u_1, u_2)$ se va face conform expresiei:

$$y = sign(\overline{w} \cdot \Phi(u) + \bar{w}_0) = sign(u_1 u_2).$$

Punctele din primul şi din al treilea cadran vor fi clasificate pozitiv, iar punctele din al doilea şi al patrulea cadran vor fi clasificate negativ.



Se constată uşor acum că în „spaţiul de trăsături“ rezultat prin mapare avem separabilitate liniară, după cum indică figura de mai sus, partea dreaptă. (Pentru convenienţă, am reţinut doar coordonatele a doua şi a patra, adică u_1 şi u_1u_2. Deşi nu era necesar, am fi putut adăuga şi coordonata u_2, dar atunci ar fi trebuit să facem graficul în 3D.) În acest spaţiu, separatorul liniar ($u_1u_2 = 0$) reprezintă o dreaptă; la fel şi marginile ($u_1u_2 = -1$ şi $u_1u_2 = 1$). Semiplanul superior corespunde punctelor clasificate pozitiv, iar semiplanul inferior corespunde punctelor clasificate negativ.

c. Se poate observa uşor că, dacă se consideră adiţional instanţa de antrenament $x_5 = (0.5, 4)$ clasificată negativ, mulţimea $\{\Phi(x_1), \Phi(x_2), \Phi(x_3), \Phi(x_4), \Phi(x_5)\}$ din „spaţiul de trăsături“ este liniar neseparabilă.

d. Considerând $u = (u_1, u_2)$ şi $u' = (u'_1, u'_2)$, urmează că

$$\Phi(u) \cdot \Phi(u') = (1, u_1, u_2, u_1u_2) \cdot (1, u'_1, u'_2, u'_1u'_2) = 1 + u_1u'_1 + u_2u'_2 + u_1u_2u'_1u'_2 = K(u, u').$$

11. **(Un [alt] exemplu de rezolvare a problemei duale SVM)**

• ***CMU, 2009 fall, Carlos Guestrin, HW3, pr. 2.1.8***
CMU, 2011 fall, T. Mitchell, A. Singh, HW6, pr. 2.1

Calculaţi soluţia $\bar{\alpha}$ pentru forma duală a problemei de optimizare SVM pentru setul de date D şi transformarea Φ (aşadar, în spaţiul de „trăsături“ determinat de Φ) de la problema 8.

Răspuns:

Putem rezolva această problemă fie (i.) folosind soluţiile problemei primale (din rezolvarea problemei 8) şi relaţiile dintre soluţiile celor două forme ale problemei de optimizare SVM (fără însă a folosi efectiv lagrangeanul dual), fie (ii.) în mod combinat, adică rezolvând problema în forma duală după ce în prealabil am simplificat expresia lagrangeanului dual ţinând cont de soluţia problemei în forma primală.

i. Vom arăta că soluţia $\bar{\alpha} = (\bar{\alpha}_1, \ldots, \bar{\alpha}_6)$ a problemei duale se poate determina folosind relaţiile dintre soluţiile celor două forme ale problemei SVM:

$$\bar{w} = \sum_{i=1}^{6} \bar{\alpha}_i y_i \Phi(x_i) \quad \textbf{şi} \quad \bar{\alpha}_i(y_i(\bar{w} \cdot \Phi(x_i) + \bar{w}_0) - 1) = 0 \textbf{ pentru } i = \overline{1,6}.$$

De la rezolvarea problemei în forma primală (vedeţi pr. 8) ştim că $\bar{\alpha}_1 = \bar{\alpha}_4 = \bar{\alpha}_5 = \bar{\alpha}_6 = 0$, fiindcă vectorii-suport sunt (doar) instanţele x_2 şi x_3. În consecinţă, relaţia $\bar{w} = \sum_{i=1}^{6} \bar{\alpha}_i y_i \Phi(x_i)$ devine:

$$\begin{pmatrix} -1/5 \\ 3/5 \end{pmatrix} = \bar{\alpha}_2 \begin{pmatrix} -2 \\ 4 \end{pmatrix} - \alpha_3 \begin{pmatrix} -1 \\ 1 \end{pmatrix} \Leftrightarrow \begin{cases} -2\bar{\alpha}_2 + \bar{\alpha}_3 = -\dfrac{1}{5} \\ 4\bar{\alpha}_2 - \bar{\alpha}_3 = \dfrac{3}{5} \end{cases} \Leftrightarrow \bar{\alpha}_2 = \bar{\alpha}_3 = \frac{1}{5}.$$

Soluţia astfel obţinută, $\overline{\alpha} = \left(0, \frac{1}{5}, \frac{1}{5}, 0, 0, 0\right)$, verifică şi restricţia Karush-Kuhn-Tucker $\sum_{i=1}^{6} \bar{\alpha}_i y_i = 0$ din forma duală a problemei SVM.

În consecinţă, funcţia de decizie obţinută de SVM se poate exprima ca

$$\begin{aligned} y(x) &= sign\Big(\sum_i \bar{\alpha}_i y_i \Phi(x_i) \cdot \Phi(x) + \bar{w}_0\Big) = sign\left(\frac{1}{5}K(x,-2) - \frac{1}{5}K(x,-1) - \frac{9}{5}\right) = \\ &= sign\,(-2x + 4x^2 + x - x^2 - 9) = sign\,(3x^2 - x - 9). \end{aligned}$$

Aşadar, regăsim rezultatul de la punctul e de la problema 8.

ii. Lagrangeanul dual este

$$\begin{aligned} L_D(\alpha) &= \sum_{i=1}^{6} \alpha_i - \frac{1}{2}\sum_{i=1}^{6}\sum_{j=1}^{6} \alpha_i \alpha_j y_i y_j \Phi(x_i) \cdot \Phi(x_j) \\ &= \sum_{i=1}^{6} \alpha_i - \frac{1}{2}(90\alpha_1^2 + 20\alpha_2^2 + 2\alpha_3^2 + 2\alpha_5^2 + 90\alpha_6^2 + \\ &\quad 2(42\alpha_1\alpha_2 - 12\alpha_1\alpha_3 - 6\alpha_1\alpha_5 + 72\alpha_1\alpha_6 \\ &\quad -6\alpha_2\alpha_3 - 2\alpha_2\alpha_5 + 30\alpha_2\alpha_6 - 6\alpha_3\alpha_6 - 12\alpha_5\alpha_6)), \end{aligned}$$

iar condiţia de optimalitate Karush-Kuhn-Tucker asociată este $\sum_{i=1}^{6} y_i\alpha_i = 0$,[657] adică $\alpha_1 + \alpha_2 - \alpha_3 - \alpha_4 - \alpha_5 + \alpha_6 = 0$.

Ţinând cont de relaţiile $\alpha_1 = \alpha_4 = \alpha_5 = \alpha_6 = 0$ care rezultă (ca o consecinţă) din soluţia problemei în forma primală (aşa cum am arătat la i) şi implică $\alpha_2 = \alpha_3 \overset{not.}{=} \alpha > 0$, urmează că putem scrie lagrangeanul dual astfel:

$$L_D(\alpha) = 2\alpha - \frac{1}{2}(20\alpha^2 + 2\alpha^2 - 12\alpha^2) = 2\alpha - 5\alpha^2 = \alpha(2 - 5\alpha).$$

[657] Vedeţi relaţia (315) de la problema 9.

Maximul funcţiei $L_D(\alpha)$ se obţine pentru $\alpha = -\frac{2}{2\cdot(-5)} = \frac{1}{5} > 0$. Aşadar, regăsim rezultatul de la i.

***Observaţie importantă*:**

Dacă am fi încercat să rezolvăm această problemă în forma duală *în mod de sine stătător*, adică făcând abstracţie de rezolvarea problemei în forma primală (obţinută la pr. 8), deci similar cu modul cum am procedat la problema 10,[658] se va constata că $\frac{\partial L}{\partial \alpha_4} = 1 \neq 0$![659] În concluzie, această variantă de rezolvare *analitică* a problemei duale SVM nu este „fezabilă" pe aceste date, iar soluţia ar trebui căutată folosind o metodă numerică de optimizare, care să ia în calcul chiar de la început restricţiile asupra multiplicatorilor $\alpha_1, \ldots, \alpha_m$. O astfel de metodă este algoritmul SMO (pentru rezolvarea problemei de optimizare C-SVM), care este prezentat la problema 22 şi a cărui aplicare o vom exemplifica la problema 23.

5.1.2 SVM cu margine "soft"

12. **(Exerciţiu teoretic: deducerea formei duale pentru problema de optimizare SVM cu margine "soft" (C-SVM))**

prelucrare de Liviu Ciortuz, după
■ ***CMU, 2012 spring, Ziv Bar-Joseph, HW3, pr. 3.2***

Antrenăm o SVM pe un set de date de intrare $\{x_i\}$ cu $i = 1, \ldots, n$, considerate împreună cu setul de etichete asociate $\{y_i\}$, cu $y_i \in \{-1, 1\}$. Se [poate] presupune că aceste date sunt neseparabile liniar. Permiţând ca la finalul antrenării unele (în genere puţine) dintre datele de antrenament să fie clasificate eronat, *obiectivul* pe care ni-l propunem aici este ca, simultan cu maximizarea *marginii geometrice*, să „controlăm" efectul cumulativ determinat de aceste excepţii.

Aşadar, vom considera problema de optimizare

$$\begin{aligned} & \min_{w, w_0, \xi} \ \left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^m \xi_i\right) \\ \textbf{a. î.} \quad & (w \cdot x_i + w_0) y_i \geq 1 - \xi_i, \ \textbf{pentru } i = 1, \ldots, m \\ & \xi_i \geq 0, \ \textbf{pentru } i = 1, \ldots, m, \end{aligned} \tag{P'}$$

unde $\xi \overset{not.}{=} (\xi_1, \ldots, \xi_m)$, iar $C > 0$ este un parametru („cost") care controlează compromisul (engl., trade-off) pe care urmărim să-l facem între mărimea *marginii* pe de o parte,[660] şi *penalizările pentru „destindere"* (engl., slack penalty) reprezentate prin variabilele $\xi_i \geq 0$ pe de altă parte.

[658] Concret, aceasta revine la a rezolva sistemul obţinut prin egalarea derivatelor parţiale ale lui $L_D(\alpha)$ cu 0, după care trebuie să verificăm dacă soluţia $\bar{\alpha}$ satisface restricţiile $\alpha_i \geq 0$ pentru $i = 1, \ldots, 6$, precum şi condiţia de optimalitate Karush-Kuhn-Tucker.

[659] Aşadar, ar fi mai bine ca mai întâi să înlocuim în L_D variabila α_4 cu $\alpha_1 + \alpha_2 - \alpha_3 - \alpha_5 + \alpha_6$ şi apoi să rescriem sistemul de ecuaţii formate cu ajutorul derivatelor parţiale ale noului L_D. Vă puteţi convinge, făcând calculele, că soluţia acestui sistem $\bar{\alpha} = (\bar{\alpha}_1, \ldots, \bar{\alpha}_6)$ nu satisface restricţiile $\alpha_i \geq 0$.

[660] Prin *margine* aici vom înţelege distanţa dintre hiperplanul de separare optimală definit de soluţia (w, w_0) a problemei (P′) şi oricare dintre instanţele x_i pentru care $y_i(w \cdot x_i + w_0) = 1$. Această distanţă este egală cu $1/\|w\|$.

a. Verificaţi faptul că pentru problema (P′) este satisfăcută *condiţia lui Slater*.[661]

b. Folosind variabile duale (adică, multiplicatori Lagrange), scrieţi expresia *lagrangeanului generalizat* care corespunde problemei (P′). Specificaţi pentru fiecare multiplicator în parte care este restricţia care-i corespunde lui în problema (P′).

c. Scrieţi *condiţiile de complementaritate Karush-Kuhn-Tucker* corespunzătoare problemei (P′).

d. Calculând derivatele parţiale ale lagrangeanului generalizat $L_P(w, w_0, \xi, \alpha, \beta)$ în raport cu variabilele w, w_0 şi respectiv ξ, arătaţi că forma duală a problemei (P′) este:

$$\begin{array}{ll} & \max_\alpha \left(\sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \right) \\ \textbf{a. î.} & 0 \leq \alpha_i \leq C \textbf{ pentru } i = 1, \ldots, m \\ & \sum_{i=1}^m \alpha_i y_i = 0. \end{array} \qquad (\mathbf{D}')$$

Faceţi toate calculele în mod detaliat. Analizaţi deosebirile dintre (D′) şi forma duală a problemei SVM cu margine "hard" (a se vedea definiţia (D) de la problema 9, pag. 660).

e. Dată fiind o soluţie a problemei duale (D′), cum identificăm vectorii-suport?

f. Cum va fi clasificată o instanţă nouă x'?

Răspuns:

a. Condiţia lui Slater pentru problema (P′) se formulează astfel:

$\exists w \in \mathbb{R}^d, w_0 \in \mathbb{R}$ **şi** $\xi \in \mathbb{R}^m$ **a. î.** $(w \cdot x_i + w_0) y_i - 1 + \xi_i > 0$ **şi** $\xi_i > 0$, **pentru** $i = 1, \ldots, m$.

Luând $w = 0$, $w_0 = 0$ şi $\xi_i = 2$ pentru $i = 1, \ldots, m$,[662] **se constată că această condiţie se verifică imediat. În consecinţă, optimul problemei primale (P′) va coincide cu optimul problemei duale (de la punctul d).**

b. Lagrangeanul generalizat care corespunde problemei de optimizare (P′) este:

$$L_P(w, w_0, \xi, \alpha, \beta) = \frac{1}{2}\|w\|^2 + C \sum_{i=1}^m \xi_i - \sum_{i=1}^m \alpha_i ((w \cdot x_i + w_0) y_i - 1 + \xi_i) - \sum_{i=1}^m \beta_i \xi_i.$$

Multiplicatorii $\alpha_i \geq 0$ corespund restricţiilor $(w \cdot x_i + w_0) y_i \geq 1 - \xi_i$, în vreme ce multiplicatorii $\beta_i \geq 0$ corespund restricţiilor $\xi_i \geq 0$.

c. $\alpha_i ((w \cdot x_i + w_0) y_i - 1 + \xi_i) = 0$ şi $\beta_i \xi_i = 0$, pentru $i = 1, \ldots, m$.

d. Calculând derivatele parţiale indicate în enunţ, vom avea:

[661] Pentru formalizarea acestei condiţii — ca şi pentru diverse noţiuni implicate la punctele următoare —, a se vedea *Comentariul* de la problema 9 (pag. 658), care tratează cazul formei duale a problemei SVM cu margine "hard".

[662] De fapt, este suficient ca — pe lângă $w = 0$, $w_0 = 0$ — să considerăm $\xi_i > 1$ pentru $i = 1, \ldots, m$.

$$\frac{\partial}{\partial w}L_P(w,w_0,\xi,\alpha,\beta)=0 \quad\Leftrightarrow\quad w-\sum_{i=1}^{m}\alpha_i y_i x_i=0\Leftrightarrow w=\sum_{i=1}^{m}\alpha_i y_i x_i$$

$$\frac{\partial}{\partial w_0}L_P(w,w_0,\xi,\alpha,\beta)=0 \quad\Leftrightarrow\quad -\sum_{i=1}^{m}\alpha_i y_i=0\Leftrightarrow\sum_{i=1}^{m}\alpha_i y_i=0$$

$$\frac{\partial}{\partial \xi_i}L_P(w,w_0,\xi,\alpha,\beta)=0 \quad\Leftrightarrow\quad C-\alpha_i-\beta_i=0\Leftrightarrow\alpha_i+\beta_i=C \text{ pentru } i=1,\ldots,m.$$

Se poate constata că primele două egalităţi care tocmai au fost obţinute, şi anume $w=\sum_{i=1}^{m}\alpha_i y_i x_i$ şi $\sum_{i=1}^{m}\alpha_i y_i=0$, coincid cu relaţiile (314) şi (315) derivate din *condiţia de staţionaritate / optimalitate* corespunzătoare problemei de optimizare SVM cu margine "hard" (vedeţi problema 9).

Substituind aceste două rezultate în expresia de definiţie a lui L_P vom obţine:

$$\begin{aligned}
L_D(\alpha,\beta) &\overset{def.}{=} \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i\cdot x_j + C\sum_{i=1}^{m}\xi_i - \\
&\quad -\sum_{i=1}^{m}\alpha_i\Big(\Big(\Big(\sum_{j=1}^{m}\alpha_j y_j x_j\Big)\cdot x_i + w_0\Big)y_i - 1 + \xi_i\Big) - \sum_{i=1}^{m}\beta_i\xi_i \\
&= \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i\cdot x_j + C\sum_{i=1}^{m}\xi_i - \\
&\quad -\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i\cdot x_j - w_0\underbrace{\sum_{i=1}^{m}\alpha_i y_i}_{0} + \sum_{i=1}^{m}\alpha_i - \sum_{i=1}^{m}\alpha_i\xi_i - \sum_{i=1}^{m}\beta_i\xi_i \\
&= -\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i\cdot x_j + C\sum_{i=1}^{m}\xi_i + \sum_{i=1}^{m}\alpha_i - \sum_{i=1}^{m}\underbrace{(\alpha_i+\beta_i)}_{C}\xi_i \\
&= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i\cdot x_j \overset{not.}{=} L_D(\alpha).
\end{aligned}$$

Observaţi că la scrierea argumentelor lagrangeanului L_D, am renunţat în final la β, întrucât β_i (pentru $i=1,\ldots,m$) a fost eliminat în timpul efectuării calculului. Se constată de asemenea că acest lagrangean este identic cu cel care a fost obţinut în cazul SVM cu margine „hard" (vedeţi relaţia (317) de la problema 9).

Ţinând cont că $\beta_i=C-\alpha_i\geq 0$ implică $\alpha_i\leq C$, rezultă imediat că forma duală asociată problemei (P′) este cea indicată în enunţ.

Remarcaţi faptul că forma duală (D′) pentru problema C-SVM (cu margine "soft") diferă de forma duală (D) pentru problema SVM cu margine "hard" doar prin restricţia suplimentară $\alpha_i\leq C$. Aceasta înseamnă că „importanţa" care revine fiecărei instanţe etichetate (x_i,y_i) cu privire la determinarea separatorului optimal devine limitată.

Facem aici menţiunea că legătura dintre soluţiile problemelor (P′) şi (D′) este dată de relaţiile care au fost obţinute la punctele c şi d: mai întâi

$$\bar{w}=\sum_{i=1}^{m}\bar{\alpha}_i y_i x_i, \tag{319}$$

iar apoi $\bar{w}_0$ se obţine din relaţia $(\bar{w} \cdot x_i + \bar{w}_0)y_i = 1 - \bar{\xi}_i$ pentru un $i \in \{1, \ldots, m\}$ astfel încât $\bar{\alpha}_i > 0$.[663] Aşadar,

$$\bar{w}_0 = -\bar{w} \cdot x_i + y_i(1 - \bar{\xi}_i), \text{ cu } \bar{\xi}_i = 0 \text{ dacă în plus } \bar{\alpha}_i < C. \tag{320}$$

Această ultimă egalitate este implicată de relaţia $\alpha_i + \beta_i = C$ dedusă mai sus şi de condiţia de complementaritate Karush-Kuhn-Tucker $\beta_i \xi_i = 0$.[664]

e. Fie $\bar{\alpha}$ o soluţie a problemei (D′), iar $\bar{w}$ şi $\bar{w}_0$ stabiliţi ca mai sus (vedeţi relaţiile (319) şi (320)). Vectorii-suport sunt instanţele x_i pentru care $\bar{\alpha}_i > 0$. Mai sus (vedeţi rezolvarea de la punctul d, la final) am arătat că $\bar{\alpha}_i \in (0, C) \Rightarrow \bar{\xi}_i = 0$, deci $y_i(\bar{w} \cdot x_i + \bar{w}_0) = 1$. Alternativ, pentru $\bar{\alpha}_i = C$ rezultă $\bar{\beta}_i = C - \bar{\alpha}_i = 0$, deci $\bar{\xi}_i \geq 0$ şi, ţinând cont de egalitatea $y_i(\bar{w} \cdot x_i + \bar{w}_0) = 1 - \bar{\xi}_i$, rezultă că $y_i(\bar{w} \cdot x_i + \bar{w}_0) \leq 1$.[665] Similar, pentru instanţele care nu sunt vectori-suport, $\bar{\alpha}_i = 0 \Rightarrow \bar{\beta}_i = C \Rightarrow \bar{\xi}_i = 0$, deci $y_i(\bar{w} \cdot x_i + \bar{w}_0) \geq 1$.[666]

***Observaţie*: Cele trei relaţii deduse mai sus, rescrise sintetizat ca**

$$\begin{cases} \bar{\alpha}_i \in (0, C) \Rightarrow y_i(\bar{w} \cdot x_i + \bar{w}_0) = 1 \\ \bar{\alpha}_i = C \Rightarrow y_i(\bar{w} \cdot x_i + \bar{w}_0) \leq 1 \\ \bar{\alpha}_i = 0 \Rightarrow y_i(\bar{w} \cdot x_i + \bar{w}_0) \geq 1 \end{cases} \tag{321}$$

se întâlnesc uneori în literatura de specialitate sub numele de *condiţiile Karush-Kuhn-Tucker*, dar acesta este impropriu. Ele sunt *condiţii necesare* implicate de *condiţiile de optimalitate şi complementaritate Karush-Kuhn-Tucker*. Însă ceea ce este important de reţinut este faptul că relaţiile de mai sus sunt folosite ca o *condiţie de oprire* pentru algoritmul SMO,[667] care rezolvă în practică

[663] Vedeţi *condiţiile de complementaritate Karush-Kuhn-Tucker*, care au fost formulate la punctul c: $\bar{\alpha}_i[(\bar{w} \cdot x_i + \bar{w}_0)y_i - 1 + \bar{\xi}_i] = 0$ şi $\bar{\beta}_i\bar{\xi}_i = 0$ pentru $i = 1, \ldots, m$. Dacă $\bar{\alpha}_i > 0$, atunci $(\bar{w} \cdot x_i + \bar{w}_0)y_i - 1 + \bar{\xi}_i = 0$, deci $(\bar{w} \cdot x_i + \bar{w}_0)y_i = 1 - \bar{\xi}_i$. Dacă $\bar{\alpha}_i < C$, atunci $\bar{\beta}_i = C - \bar{\alpha}_i > 0$ şi deci $\bar{\xi}_i = 0$.

[664] Dacă $\bar{\alpha}_i = 0$ pentru $i = \overline{1, m}$, obţinem $\bar{w} = 0$, ceea ce reprezintă o soluţie inadmisibilă.

Rămâne de tratat cazul în care $\exists \bar{\alpha}_i = C$ şi pentru orice alt $\bar{\alpha}_j$ avem fie $\bar{\alpha}_j = 0$ fie $\bar{\alpha}_j = C$. Este de notat mai întâi faptul că din relaţia $\sum_{i=1}^{m} \alpha_i y_i = 0$ va rezulta că jumătate dintre vectorii-suport sunt instanţe pozitive, iar restul (cealaltă jumătate) sunt instanţe negative. Apoi, $\bar{\alpha}_j = C > 0$ implică $y_j(\bar{w} \cdot x_j + w_0) = 1 - \bar{\xi}_j$ şi $\bar{\beta}_j = 0$ şi deci $\bar{\xi}_j \geq 0$. Pe de altă parte, $\bar{\alpha}_j = 0$ implică $\bar{\beta}_j = C$ şi deci $\bar{\xi}_j = 0$, şi $y_j(\bar{w} \cdot x_j + w_0) \geq 1$.

În consecinţă,

$$\begin{aligned} \sum_i \xi_i &= \sum_{i:\bar{\alpha}_i = C} \xi_i = \sum_{i:\bar{\alpha}_i = C} (1 - y_i(\bar{w} \cdot x_i + \bar{w}_0)) = \sum_{i:\bar{\alpha}_i = C} (1 - y_i \bar{w} \cdot x_i) \\ &= |\{i : \bar{\alpha}_i = C\}| - \sum_{i:\bar{\alpha}_i = C} y_i \bar{w} \cdot x_i = |\{i : \bar{\alpha}_i = C\}| - \bar{w} \cdot \sum_{i:\bar{\alpha}_i = C} y_i x_i \\ &= |\{i : \bar{\alpha}_i = C\}| - \frac{1}{C} \bar{w} \cdot \sum_{i:\bar{\alpha}_i = C} \bar{\alpha}_i y_i x_i = |\{i : \bar{\alpha}_i = C\}| - \frac{1}{C} \bar{w}^2 \end{aligned}$$

Aceasta ne arată că optimul problemei primale (P′) depinde doar de $\bar{w}$ (nu şi de $\bar{w}_0$).

Notând în mod generic prin x_+ instanţele pozitive (şi cu $\bar{\xi}_+$ valoarea variabilei de „destindere“ corespunzătoare), iar prin x_- instanţele negative (şi, corespunzător, $\bar{\xi}_-$), vom avea de satisfăcut restricţiile $\bar{w} \cdot x_+ + \bar{w}_0 \geq 1 - \bar{\xi}_+$ şi $\bar{w} \cdot x_- + \bar{w}_0 \leq -1 + \bar{\xi}_-$. Ţinând cont că $\bar{\xi}_+ \geq 0$ şi $\bar{\xi}_- \geq 0$, considerăm că din punct de vedere practic se poate lua pentru $\bar{w}_0$ valoarea $-\frac{1}{2}(\min_{x_+}\{\bar{w} \cdot x_+\} + \max_{x_-}\{\bar{w} \cdot x_-\})$. De remarcat că într-o astfel de situaţie, marginea va reprezenta distanţa dintre separatorul optimal şi cele mai depărtate instanţe pozitive / negative clasificate corect. Evident, aceasta este o situaţie (destul de) extremă în raport cu ideea cu care s-a plecat la drum în formalizarea clasificării cu margine maximală.

[665] Dacă $0 < \bar{\xi}_i \leq 1$, atunci instanţa x_i este situată în interiorul „marginii“ de separare, iar dacă $\bar{\xi}_i > 1$, instanţa x_i este clasificată eronat.

[666] Vectorii pentru care are loc egalitatea $y_i(\bar{w} \cdot x_i + \bar{w}_0) = 1$ se mai numesc *vectori-margine* (engl., margin vectors).

[667] Vedeţi problemele 22 şi 23.

problema de optimizare cu margine "soft" (desemnată în continuare prin C-SVM).

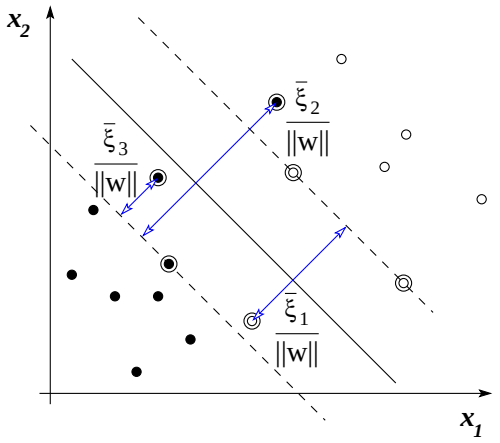
***Rezumat*: Pentru SVM cu margine "soft" (C-SVM), *vectorii-suport* sunt instanţele pentru care $\bar{\alpha}_i > 0$ (pentru acestea, avem $y_i(\bar{w} \cdot x_i + \bar{w}_0) \leq 1$), aşadar inclusiv instanţele pentru care $y_i(\bar{w} \cdot x_i + \bar{w}_0) < 1$ (deci cu $\bar{\alpha}_i = C$).**[668]

f. Păstrând notaţiile de mai sus, instanţa de test x' va fi clasificată pozitiv dacă $\bar{w} \cdot x' + \bar{w}_0 \geq 0$, şi negativ în caz contrar.

13. (C-SVM: calculul distanţei faţă de hiperplanul-margine corespunzător, pentru acei vectori-suport x_i pentru care $\bar{\alpha}_i = C$; calculul valorii optime pentru funcţia obiectiv)

□ ***UAIC Iaşi, 2018, Sebastian Ciobanu***

a. Folosind notaţiile pentru forma primală şi forma duală de la problema de optimizare C-SVM, — vedeţi problema 12 — demonstraţi următoarea afirmaţie: pentru toate acele instanţe de antrenament x_i pentru care $\bar{\alpha}_i = C$ (adică, pentru acei x_i pentru care, conform rezolvării problemei 12.e, $y_i(\bar{w} \cdot x_i + \bar{w}_0) \leq 1$ şi, deci $\bar{\xi}_i \geq 0$), *distanţa geometrică* de la x_i la *hiperplanul-margine*, care este paralel cu hiperplanul de separare optimală şi are ecuaţia $\bar{w} \cdot x + \bar{w}_0 = y_i$, este $\frac{\bar{\xi}_i}{\|w\|}$.



Credit: cf. http://efavdb.com/svm-classification/.

Altfel spus,

$$\bar{\alpha}_i = C \Rightarrow \frac{\bar{\xi}_i}{\|\bar{w}\|} = d(x_i,\ \bar{w} \cdot x + \bar{w}_0 = y_i),\ \forall i \in \{1, \ldots, m\}.$$

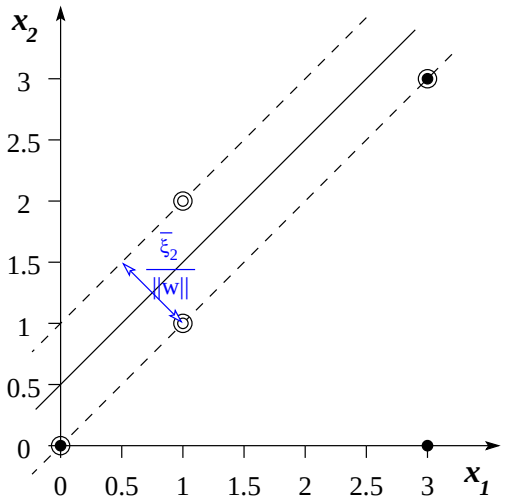
[668] Din punct de vedere *geometric*, vectorii-suport care *nu* sunt vectori-magine (numiţi şi *vectorii-suport excepţie*), adică acei vectori-suport pentru care $y_i(\bar{w} \cdot x_i + \bar{w}_0) < 1$ sunt fie instanţe etichetate (x_i, y_i) incorect clasificate de către separatorul optimal (adică, ele satisfac inegalitatea $y_i(\bar{w} \cdot x_i + \bar{w}_0) \leq 0$), fie instanţe situate între separatorul optimal şi hiperplanul „margine" de ecuaţie $\bar{w} \cdot x_i + \bar{w}_0 = y_i$.

b. Fie următorul set de date de antrenament:

i	1	2	3	4	5
x_i	$(0,0)$	$(1,1)$	$(3,3)$	$(1,2)$	$(3,0)$
y_i	$+1$	-1	$+1$	-1	$+1$

Valorile multiplicatorilor Lagrange învăţate de către o C-SVM liniară pentru $C = 10$, pornind de la aceste date sunt:

i	1	2	3	4	5
α_i	8.66	10	5.33	3.99	0



***i.* Verificaţi numeric relaţia $\frac{\bar{\xi}_2}{\|\bar{w}\|} = d(x_2,\ \bar{w}\cdot x + \bar{w}_0 = y_2)$.**

***ii.* Calculaţi valoarea funcţiei obiectiv de la problema primală C-SVM pentru acelaşi $C = 10$.**

***Sugestii*:**
1. Folosiţi condiţiile [de complementaritate] Karush-Kuhn-Tucker pentru problema de optimizare C-SVM, precum şi relaţiile de legătură dintre soluţiile formei primale şi soluţiile formei duale ale problemei C-SVM. (Vedeţi rezolvarea problemei 12, subpunctele c şi d.)
2. Pentru uşurinţa calculelor, se pot face aproximări. (De exemplu, 1.99 poate fi aproximat cu 2.)

Răspuns:

a. Întrucât $\bar{\alpha}_i = C > 0$, din condiţia de complementaritate Karush-Kuhn-Tucker $\bar{\alpha}_i[y_i(\bar{w}\cdot x_i + \bar{w}_0) - (1 - \bar{\varepsilon}_i)] = 0$ (vedeţi rezolvarea problemei 12.c), rezultă

$$y_i(\bar{w}\cdot x_i + \bar{w}_0) = 1 - \bar{\xi}_i \Rightarrow \bar{w}\cdot x_i + \bar{w}_0 = y_i(1 - \bar{\xi}_i) \Rightarrow \bar{w}\cdot x_i + \bar{w}_0 - y_i = -y_i\bar{\xi}_i.$$

Prin urmare, ţinând cont şi de formula dată la problema 1, distanţa de la punctul x_i la hiperplanul de ecuaţie $\bar{w}\cdot x + \bar{w}_0 - y_i = 0$ este

$$\frac{|-y_i\bar{\xi}_i|}{\|\bar{w}\|} \overset{\bar{\xi}_i \geq 0}{=\!=} \frac{\bar{\xi}_i}{\|\bar{w}\|}.$$

b.*i.* Conform problemei 12.d, mai precis conform relaţiei (319), $\bar{w} = \sum_{i=1}^{5} \bar{\alpha}_i y_i x_i$. Deci, în cazul nostru,

$$\bar{w} = 8.66\begin{pmatrix}0\\0\end{pmatrix} - 10\begin{pmatrix}1\\1\end{pmatrix} + 5.33\begin{pmatrix}3\\3\end{pmatrix} - 3.99\begin{pmatrix}1\\2\end{pmatrix} + 0 = \begin{pmatrix}-10 + 15.99 - 3.99\\-10 + 15.99 - 7.98\end{pmatrix} = \begin{pmatrix}2\\-1.99\end{pmatrix}.$$

Pentru uşurinţa calculelor vom aproxima:

$$\bar{w} \approx \begin{pmatrix}2\\-2\end{pmatrix}.$$

Aşadar,

$$\|\bar{w}\| = \sqrt{2^2 + (-2)^2} = \sqrt{4+4} = \sqrt{8} = 2\sqrt{2}.$$

Conform problemei 12.d, putem aproxima termenul liber $\bar{w}_0$ folosind ecuaţiile / proprietăţile corespunzătoare unui vector-suport x_i pentru care $\bar{\xi}_i = 0$:

$$\bar{\alpha}_1 = 8.66 \in (0, C) \Rightarrow \bar{\beta}_1 > 0 \overset{\textbf{KKT}}{\Rightarrow} \bar{\xi}_1 = 0 \Rightarrow x_1 \textbf{ este vector-suport}.$$

$$y_1(\bar{w} \cdot x_1 + \bar{w}_0) = 1 - \bar{\xi}_1 \Rightarrow (+1)\left(\begin{pmatrix} 2 \\ -2 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \end{pmatrix} + \bar{w}_0\right) = 1 - 0 \Rightarrow \bar{w}_0 = 1.$$

Acum putem calcula $\bar{\xi}_2$ folosind prima parte a raţionamentului de la punctul a:

$$\bar{\alpha}_2 = 10 = C \Rightarrow y_2(\bar{w} \cdot x_2 + \bar{w}_0) = 1 - \bar{\xi}_2$$
$$\Rightarrow \bar{\xi}_2 = 1 - y_2(\bar{w} \cdot x_2 + \bar{w}_0) = 1 - (-1)\left(\begin{pmatrix} 2 \\ -2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 1\right) = 2.$$

În consecinţă,

$$d(x_2,\ \bar{w} \cdot x + \bar{w}_0 = y_2) = d\left(\begin{pmatrix} 1 \\ 1 \end{pmatrix},\ \begin{pmatrix} 2 \\ -2 \end{pmatrix} \cdot x + 1 = -1\right)$$
$$= \frac{\left|\begin{pmatrix} 2 \\ -2 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} + 1 - (-1)\right|}{\|\bar{w}\|} = \frac{|2|}{2\sqrt{2}} = \frac{1}{\sqrt{2}}.$$

Aşadar, se verifică egalitatea

$$\frac{\bar{\xi}_2}{\|\bar{w}\|} = d(x_2,\ \bar{w} \cdot x + \bar{w}_0 = y_2).$$

***Observaţie*: Se poate verifica şi din punct de vedere pur geometric, lucrând pe figura din enunţ, că distanţa dintre punctul x_2 şi hiperplanul $\bar{w} \cdot x + \bar{w}_0 = y_2$ este de (aproximativ[669]) $1/\sqrt{2}$.**

b.*ii.* Trebuie să calculăm valoarea

$$\frac{1}{2}\|\bar{w}\|^2 + C\sum_{i=1}^{5} \bar{\xi}_i.$$

De la subpunctul precedent ştim că $\|w\| \approx 2\sqrt{2}$ şi $\bar{\xi}_2 \approx 2$. Pentru a determina celelalte valori ale variabilelor de „destindere“ $\bar{\xi}_i$, vom face din nou apel la *condiţiile Karush-Kuhn-Tucker*:

$$0 \leq \bar{\alpha}_1, \bar{\alpha}_3, \bar{\alpha}_4, \bar{\alpha}_5 < C \overset{\alpha_i + \beta_i = C}{\Rightarrow} 0 < \bar{\beta}_1, \bar{\beta}_3, \bar{\beta}_4, \bar{\beta}_5 \leq C \overset{\beta_i \bar{\xi}_i = 0}{\Rightarrow} \bar{\xi}_1, \bar{\xi}_3, \bar{\xi}_4, \bar{\xi}_5 = 0.$$

În consecinţă,

$$\frac{1}{2}\|\bar{w}\|^2 + C\sum_{i=1}^{5} \bar{\xi}_i \approx \frac{1}{2}(2\sqrt{2})^2 + 10 \cdot (0 + 2 + 0 + 0 + 0) = \frac{1}{2} \cdot 8 + 10 \cdot 2 = 24.$$

[669] Din cauza „rotunjirii“ aplicate mai sus asupra lui $\bar{w}$.

14. **(SVM şi C-SVM: o proprietate simplă a soluţiei ($\bar{w}$), indusă de o caracteristică particulară a datelor de antrenament)**

prelucrare de Liviu Ciortuz, după
• ◦ MIT, 2008 fall, Tommi Jaakkola, midterm exam, pr. 1.1
CMU, 2010 fall, Aarti Singh, HW3, pr. 3.3.a

Considerăm instanţele de antrenament $x_1 = (1, 1)$, $x_2 = (2, 2)$, $x_3 = (-1.5, -1.5)$ şi $x_4 = (4, 4)$. Antrenăm o maşină cu vectori-suport pe aceste date.

Arătaţi că

– indiferent care este etichetarea celor patru instanţe de antrenament, şi

– indiferent dacă maşina cu vectori-suport lucrează cu margine "hard" ori cu margine "soft",

vectorul $\bar{w} = (\bar{w}_1, \bar{w}_2)$ care constituie soluţia problemei de optimizare [C-]SVM pe aceste date are următoarea proprietate: $\bar{w}_1 = \bar{w}_2$.

Răspuns:

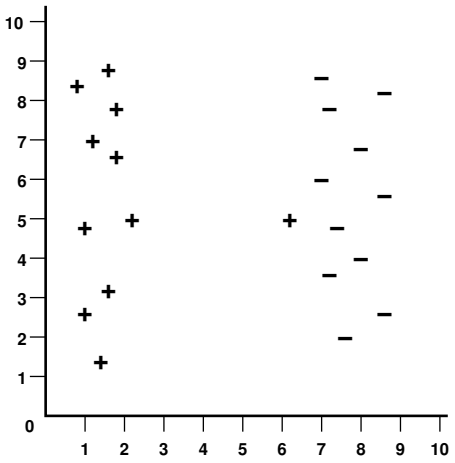
Ştim că soluţia problemei [C-]SVM în forma primală este $\bar{w} = \sum_{i=1}^{m} \bar{\alpha}_i y_i x_i$, unde $\bar{\alpha}_i$, cu $i = 1, \ldots, m$, constituie soluţia problemei în forma duală.[670] Întrucât toate instanţele de antrenament au proprietatea $x_{i1} = x_{i2}$, rezultă imediat că $\bar{w}_1 = \bar{w}_2$.

***Observaţie*: Exerciţiul acesta a ilustrat de fapt următoarea *proprietate importantă*: dacă în setul de date de antrenament două trăsături (engl., features) sunt duplicate ($x_{ij} = x_{ik}$ pentru $i = 1, \ldots, m$), atunci ele vor primi ponderi identice ($\bar{w}_j = \bar{w}_k$) în soluţia optimală calculată de clasificatorul [C-]SVM.**

15. **(C-SVM, cazul [datelor separabile] liniar: aplicare în prezenţa unui "outlier")**

CMU, 2005 spring, C. Guestrin, T. Mitchell, HW3, pr. 2.2

Se dau datele prezentate în figura alăturată. Lucrând cu clasificatorul C-SVM, parametrul C — pentru penalizarea aferentă „destinderii"; engl., slack penalty — va determina poziţia hiperplanului de separare. Presupunem că se lucrează în varianta liniară, adică fără funcţii-nucleu. Răspundeţi succint, în manieră *calitativă*, la întrebările de mai jos.



a. Unde va fi situat separatorul optimal în cazul în care C ia valori foarte mari (adică, $C \to \infty$)? Indicaţi pe figura dată.

b. Pentru $C \approx 0$, indicaţi pe figura dată unde va fi situat separatorul optimal.

[670] Vedeţi relaţiile (314) şi (319) deduse la problemele 9 şi respectiv 12.

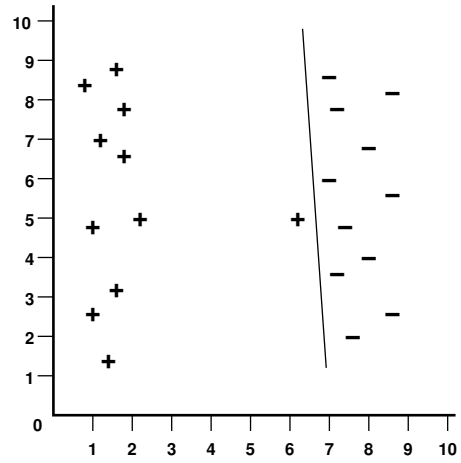
c. Care din cele două situaţii de mai sus credeţi că este mai adecvată pentru clasificare? De ce?

Răspuns:

Putem face mai întâi două *observaţii* în legătură cu acest set de date de antrenament:

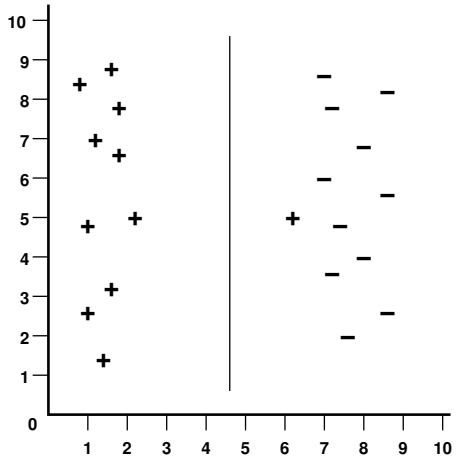
- **grupul de date clasificate negativ, precum şi grupul de date clasificate pozitiv (din partea stângă) sunt foarte compacte;**
- **există un singur exemplu pozitiv care este foarte apropiat de grupul exemplelor negative. Acest exemplu-excepţie constituie o „anomalie“ (engl., outlier).**

a. O valoare mare a parametrului C semnifică o penalizare mai mare a erorilor de clasificare ξ_i. Aşadar, pentru valori ale lui C tinzând la $+\infty$ hiperplanul de separare optimală va corespunde celei mai mici valori posibile pentru suma $\sum_i \xi_i$, care reprezintă eroarea totală în raport cu marginile. Întrucât exemplele date sunt separabile liniar, separatorul optimal învăţat este cel reprezentat în figura alăturată. Poziţia sa este foate mult influenţată de poziţia outlierului (+), care este situat în imediata proximitate a instanţelor negative.



***Observaţie*: În general, dacă avem încredere în corectitudinea datelor de antrenament, putem alege $C \to \infty$ pentru a obţine o separare cât mai precisă a celor două clase.**

b. În cazul $C \approx 0$, erorile ξ_i sunt penalizate foarte puţin. În consecinţă, separatorul optimal tinde să maximizeze marginea — adică distanţa de la hiperplanul de separare $w \cdot x + w_0 = 0$ la vectorii-suport x_i pentru care $y_i(w \cdot x_i + w_0) = 1$ —, chiar dacă asta înseamnă că unele instanţe vor fi clasificate eronat. Separatorul determinat în acest caz va fi poziţionat ca în figura alăturată.



***Observaţie*: În general, este indicat să alegem o valoare mică pentru parametrul C atunci când datele de antrenament pe care le avem sunt afectate de „zgomote“ / perturbaţii / anomalii. Într-o astfel de situaţie am putea avea**

de-a face cu un set de date neseparabile liniar, în care câteva exemple pozitive să fie amestecate printre cel negative şi invers.

c. Separatorul determinat la punctul b este alegerea cea mai bună pentru generalizare / testare în prezenţa outlier-erelor, întrucât minimizează riscul de a face clasificări eronate.

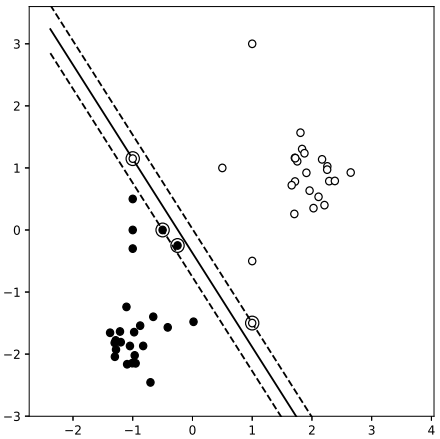
16. **(C-SVM, cazul [datelor separabile] liniar: efectul alegerii diverselor valori pentru parametrul C)**

CMU, 2010 fall, Ziv Bar-Joseph, midterm exam, pr. 7.c

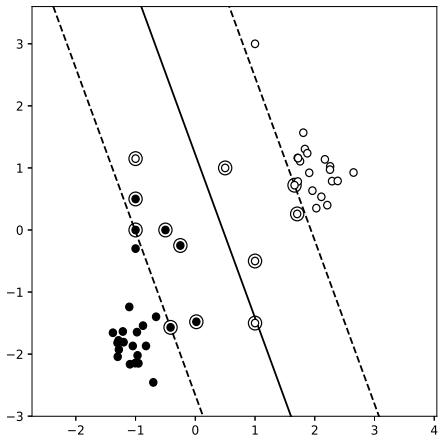
În următoarele patru figuri, se consideră că s-a aplicat o C-SVM liniară — adică o maşină cu vectori-suport cu margine "soft" şi cu parametru de „destindere" C, fără funcţie-nucleu — pe un set de date din $\mathbb{R}^2$, folosind de fiecare dată una din următoarele valori pentru parametrul C: 0.1, 1, 10 sau 100.

Sub fiecare figură scrieţi valoarea corespunzătoare pentru parametrul C.

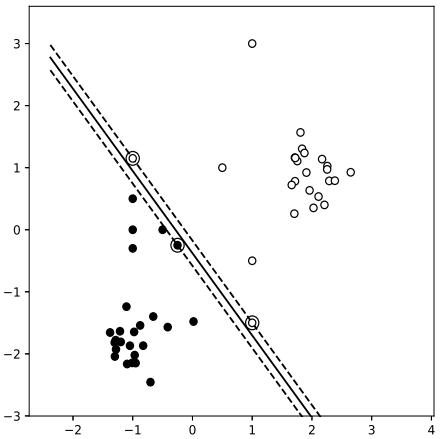
Justificaţi alegerea pe care aţi făcut-o.



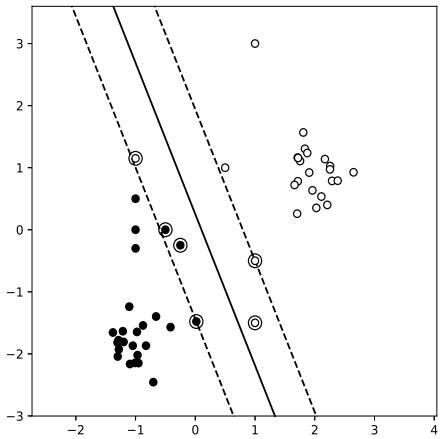
A.



B.



C.



D.

Răspuns:

Considerând datele de antrenament (x_i, y_i) cu $i = \overline{1,m}$, funcţia obiectiv pentru forma primală a problemei C-SVM este

$$\min_{w\in\mathbb{R}^d,\,\xi} \left(\frac{1}{2} \parallel w \parallel^2 + C \sum_{i=1}^{m} \xi_i \right),$$

unde $\xi_i \geq 0$ sunt variabile de „destindere“ ($y_i(w \cdot x_i + w_0) \geq 1 - \xi_i$ pentru $i = 1, \ldots, m$). Din expresia funcţiei obiectiv decurge imediat că la valori mari ale parametrului C — vă reamintim, $C > 0$ — sunt permise eventuale erori ξ_i mici, şi invers: la valori mici ale parametrului C sunt permise erori ξ_i mari. De asemenea, ne aşteptăm ca, pe măsură ce valoarea parametrului C creşte, marginea[671] să scadă.

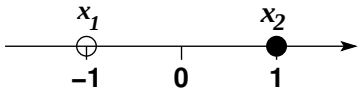
Dintre cele patru grafice din enunţ, în cazul celui de-al treilea (C) avem eroare la antrenare 0; în toate celelalte trei cazuri avem în mod evident $\sum_i \xi_i > 0$.[672] Aşadar, graficul C corespunde celei mai mari valori dintre cele patru valori propuse pentru parametrul C: 100.

În cazul graficului B sunt mult mai mulţi vectori-suport decât în cazul graficelor A sau D, iar suma $\sum_i \xi_i$ este cea mai mare dintre toate aceste trei cazuri rămase în discuţie (A, B şi D). Aşadar, graficului B îi corespunde cea mai mică dintre valorile lui C rămase: 0.1. În fine, marginea este mai mare în cazul D decât în cazul A. Prin urmare, lui A îi corespunde $C = 10$, iar lui D valoarea rămasă: $C = 1$.

17. **(Rezolvarea unei probleme de optimizare C-SVM pentru un set simplu de date din $\mathbb{R}$; rezolvarea mai întâi a formei duale a problemei de optimizare C-SVM, iar apoi aflarea soluţiilor formei primale folosind relaţiile dintre cele două seturi de soluţii Studierea unicităţii soluţiei problemei de optimizare C-SVM.)**

• ∗ *Liviu Ciortuz, 2024, după “Uniqueness of the SVM solution”, C. Burges, D. Crisp, 1998*

Fie instanţele de antrenament $(x_1 = -1, y_1 = -1)$ şi $(x_2 = 1, y_2 = 1)$. Vrem să antrenăm o maşină cu vectori-suport cu margine “soft” (deci, C-SVM) pe acest set de date.



a. Scrieţi problema de optimizare C-SVM în forma primală, corespunzătoare *acestui* set de date de antrenament.

***Observaţie*: Explicitaţi fiecare restricţie (de tip inegalitate) în funcţie de datele concrete.**

b. Scrieţi forma duală a problemei de optimizare C-SVM, pentru acelaşi set de date.

***Observaţie*: Veţi aduce lagrangeanul dual la forma cea mai simplă.**

[671] Definită ca $1/\|w\|$, adică distanţa de la hiperplanul de separare optimală la instanţele (vectorii-suport) x_i pentru care $(w \cdot x_i + w_0)y_i = 1$.

[672] Inegalitatea $\sum_i \xi_i > 0$ se datorează faptului că avem (cel puţin!) câte o excepţie la clasificare în raport cu cele două margini (hiperplanele de ecuaţie $w \cdot x_i + w_0)y_i = \pm 1$), în fiecare dintre aceste ultime trei cazuri.

c. Rezolvaţi problema de optimizare C-SVM în formă duală pe care aţi determinat-o la punctul b.

***Atenţie*: Veţi avea de tratat două cazuri: i. $C \in \left(0, \frac{1}{2}\right)$ şi ii. $C \in \left[\frac{1}{2}, +\infty\right)$, unde C este *parametrul de destindere*.**

În fiecare dintre cele două cazuri veţi indica vectorii-suport.

d. Calculaţi soluţiile problemei de optimizare C-SVM în forma primală pe care aţi determinat-o la punctul a. Pentru fiecare dintre soluţiile determinate veţi identifica separatorul decizional.

***Sugestie*: Folosiţi relaţiile de legătură dintre soluţiile formei primale şi soluţiile formei duale (precum şi condiţiile de complementaritate Karush-Kuhn-Tucker).**

e. Ce puteţi spune relativ la unicitatea soluţiilor formei duale a problemei C-SVM pentru acest set de date?
Dar relativ la unicitatea soluţiilor formei primale a aceleiaşi probleme C-SVM?
***Observaţie*: În acest context, chestiunea *unicităţii soluţiilor* se pune / discută în raport cu o valoare (oarecare) fixată a parametrului $C > 0$. Aşadar, vă veţi referi la cazurile menţionate la punctul c.**

Răspuns:

a. Forma primală a problemei de optimizare C-SVM este — în cazul general — următoarea:

$$\begin{aligned} & \min_{w, w_0, \xi} \ \left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^m \xi_i\right) \\ \textbf{a. î.} \quad & y_i(w \cdot x_i + w_0) \geq 1 - \xi_i, \ \textbf{pentru } i = 1, \ldots, m \\ & \xi_i \geq 0, \ \textbf{pentru } i = 1, \ldots, m, \end{aligned}$$

unde m este numărul de instanţe de antrenament ($x_i \in \mathbb{R}^d$, pentru $i = 1, \ldots, m$), $w \in \mathbb{R}^d$, $w_0 \in \mathbb{R}$, $\xi \overset{not.}{=} (\xi_1, \ldots, \xi_m)$, iar $C > 0$.
În cazul nostru, această formă se particularizează astfel:

$$\begin{aligned} & \min_{w, w_0, \xi_1, \xi_2} \ \left(\frac{1}{2}w^2 + C(\xi_1 + \xi_2)\right) \\ \textbf{a. î.} \quad & -(-w + w_0) \geq 1 - \xi_1, \\ & w + w_0 \geq 1 - \xi_2, \\ & w, w_0, \xi_1, \xi_2 \in \mathbb{R}; \xi_1 \geq 0, \xi_2 \geq 0. \end{aligned}$$

b. Forma duală a problemei de optimizare C-SVM este — în cazul general — următoarea:

$$\begin{aligned} & \max_{\alpha} \left(\sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j\right) \\ \textbf{a. î.} \quad & 0 \leq \alpha_i \leq C \ \textbf{pentru } i = 1, \ldots, m \\ & \sum_{i=1}^m \alpha_i y_i = 0. \end{aligned}$$

În cazul nostru, această formă se particularizează astfel:

$$\begin{aligned} & \max_{\alpha_1, \alpha_2} \left(\sum_{i=1}^2 \alpha_i - \frac{1}{2}\sum_{i=1}^2 \sum_{j=1}^2 \alpha_i \alpha_j y_i y_j x_i \cdot x_j\right) \\ \textbf{a. î.} \quad & 0 \leq \alpha_i \leq C \ \textbf{pentru } i = 1, 2 \\ & \sum_{i=1}^2 \alpha_i y_i = 0, \end{aligned}$$

deci

$$\begin{aligned} & \max_{\alpha_1,\alpha_2} \left(\alpha_1+\alpha_2-\frac{1}{2}\sum_{i=1}^{2}\sum_{j=1}^{2}\alpha_i\alpha_j y_i y_j x_i x_j\right) \\ \textbf{a. î.} \quad & 0\le\alpha_i\le C \textbf{ pentru } i=1,2 \\ & \alpha_1=\alpha_2. \end{aligned}$$

Datorită egalităţii $\alpha_1=\alpha_2$, funcţia obiectiv din problema duală devine:

$$\begin{aligned} L_D(\alpha_1,\alpha_2) &\overset{not.}{=} \alpha_1+\alpha_2-\frac{1}{2}[\alpha_1^2(-1)^2(-1)^2+\alpha_2^2\cdot 1^2\cdot 1^2+2\alpha_1\alpha_2(-1)\cdot 1\cdot(-1)\cdot 1] \\ &= 2\alpha_1-\frac{1}{2}(2\alpha_1^2+2\alpha_1^2)=2\alpha_1-2\alpha_1^2=2\alpha_1(1-\alpha_1). \end{aligned}$$

Aşadar, forma duală a problemei noastre de optimizare C-SVM este

$$\begin{aligned} & \max_{\alpha_1} \alpha_1(1-\alpha_1) \\ \textbf{a. î.} \quad & 0\le\alpha_1\le C \textbf{ şi } \alpha_2=\alpha_1. \end{aligned}$$

c. Evident, maximul funcţiei obiectiv din problema duală (funcţie polinomială de gradul al doilea) se atinge pentru $\alpha_1=1/2$.

Soluţiile optime pentru problema duală C-SVM sunt supuse restricţiilor $0\le\bar{\alpha}_i\le C$.[673] **Aşadar, vom avea două cazuri: $\bar{\alpha}_1=\bar{\alpha}_2=1/2$ atunci când $C\ge 1/2$ şi, respectiv, $\bar{\alpha}_1=\bar{\alpha}_2=C$ atunci când $C<1/2$.**

În ambele cazuri, x_1 şi x_2 sunt vectori-suport, fiindcă valorile multiplicatorilor $\bar{\alpha}_1$ şi $\bar{\alpha}_2$ sunt strict pozitive.

d+e. Conform relaţiei (319), **$\bar{w}=\sum_i y_i\alpha_i x_i$, vom avea $\bar{w}=2\bar{\alpha}_1$.**

<u>Cazul 1</u> ($C\ge 1/2$)**: $\bar{\alpha}_1=\bar{\alpha}_2=1/2$, $\bar{w}=2\bar{\alpha}_1=1$.**

Calculăm $\bar{w}_0$, $\bar{\xi}_1$ şi $\bar{\xi}_2$ folosind relaţiile de complementaritate Karush-Kuhn-Tucker:

$$\begin{cases} \bar{\alpha}_1[y_1(\bar{w}\cdot x_1+\bar{w}_0)-(1-\bar{\xi}_1)]=0 \\ \bar{\alpha}_2[y_2(\bar{w}\cdot x_2+\bar{w}_0)-(1-\bar{\xi}_2)]=0 \end{cases} \overset{\bar{\alpha}_i\neq 0}{\Rightarrow} \begin{cases} y_1(\bar{w}\cdot x_1+\bar{w}_0)=1-\bar{\xi}_1 \\ y_2(\bar{w}\cdot x_2+\bar{w}_0)=1-\bar{\xi}_2 \end{cases}$$

$$\Rightarrow \begin{cases} -1(-1+\bar{w}_0)=1-\bar{\xi}_1 \\ 1+\bar{w}_0=1-\bar{\xi}_2 \end{cases} \Rightarrow \bar{w}_0=\bar{\xi}_1=-\bar{\xi}_2.$$

Ţinând cont de faptul că $\bar{\xi}_1\ge 0$ şi $\bar{\xi}_2\ge 0$, rezultă $\bar{w}_0=\bar{\xi}_1=\bar{\xi}_2=0$.

Aşadar, pentru forma primală a problemei de optimizare C-SVM a rezultat soluţia $\bar{w}=1$, $\bar{w}_0=0$, iar separatorul optimal are ecuaţia $\bar{w}\cdot x+\bar{w}_0=0$, deci $x=0$.

Observaţie*: Pentru sub-azul ($C>1/2=\bar{\alpha}_i$) soluţiile $\bar{w}_0=\bar{\xi}_1=\bar{\xi}_2=0$ pot fi obţinute şi pe altă cale: folosind una dintre aşa-numitele *condiţii de tip Karush-Kuhn-Tucker (321),[674] **şi anume prima dintre ele (cea care corespinde vectorilor-suport margine). Întrucât i. valorile variabilelor lagrangeene $\bar{\beta}_1=$**

[673] Cf., din nou, ex. 12.d.

[674] Le reproducem aici:

$$\begin{cases} \bar{\alpha}_i\in(0,C)\Rightarrow y_i(\bar{w}\cdot x_i+\bar{w}_0)=1 \\ \bar{\alpha}_i=C\Rightarrow y_i(\bar{w}\cdot x_i+\bar{w}_0)\le 1 \\ \bar{\alpha}_i=0\Rightarrow y_i(\bar{w}\cdot x_i+\bar{w}_0)\ge 1 \end{cases}$$

$C-\bar{\alpha}_1$ şi $\bar{\beta}_2 = C-\bar{\alpha}_2$ sunt strict pozitive (vedeţi relaţia $\alpha_i+\beta_i = C$ din ex. 12.d) şi *ii.* **avem condiţia de complementaritate Karush-Kuhn-Tucker, $\bar{\beta}_i\bar{\xi}_i = 0$ pentru $i \in \{1,2\}$, rezultă că $\bar{\xi}_1 = \bar{\xi}_2 = 0$. În sfârşit, putem calcula $\bar{w}_0$:**

$$y_1(\bar{w}\cdot x_1 + \bar{w}_0) = 1 \Rightarrow -(-1+\bar{w}_0) = 1 \Rightarrow \bar{w}_0 = 0.$$

<u>Cazul 2</u> $(C < 1/2)$**:** $\bar{\alpha}_1 = \bar{\alpha}_2 = C$, $\bar{w} = 2\bar{\alpha}_1 = 2C$.

Aplicând (ca şi mai sus) relaţiile de complementaritate Karush-Kuhn-Tucker, vom obţine:

$$\begin{cases} \bar{\alpha}_1[y_1(\bar{w}\cdot x_1+\bar{w}_0)-(1-\bar{\xi}_1)] = 0 \\ \bar{\alpha}_2[y_2(\bar{w}\cdot x_2+\bar{w}_0)-(1-\bar{\xi}_2)] = 0 \end{cases} \overset{\bar{\alpha}_i\neq 0}{\Rightarrow} \begin{cases} y_1(\bar{w}\cdot x_1+\bar{w}_0) = 1-\bar{\xi}_1 \\ y_2(\bar{w}\cdot x_2+\bar{w}_0) = 1-\bar{\xi}_2 \end{cases}$$

$$\Rightarrow \begin{cases} -(-2C+\bar{w}_0) = 1-\bar{\xi}_1 \\ 2C+\bar{w}_0 = 1-\bar{\xi}_2 \end{cases} \Rightarrow \begin{cases} \bar{\xi}_1 = 1-2C+\bar{w}_0 \\ \bar{\xi}_2 = 1-2C-\bar{w}_0 \end{cases}$$

Ultimul sistem obţinut mai sus este un sistem liniar de două ecuaţii cu trei necunoscute. Soluţiile $\bar{\xi}_1$ şi $\bar{\xi}_2$ se scriu în funcţie de $\bar{w}_0$. În final, ţinând cont de restricţiile $\bar{\xi}_1 \geq 0$ şi $\bar{\xi}_2 \geq 0$, rezultă $\bar{w}_0 \geq 2C-1$ şi $\bar{w}_0 \leq 1-2C$, deci în concluzie $\bar{w}_0 \in [2C-1, 1-2C]$.[675]

Prin urmare, în acest caz, forma primală a problemei C-SVM are o infinitate de soluţii — deşi forma duală are o singură soluţie —, şi anume câte una pentru fiecare valoare posibilă a lui $\bar{w}_0$ în intervalul $[2C-1, 1-2C]$. Ecuaţia separatorului optimal este $\bar{w}\cdot x + \bar{w}_0 = 0$, adică $2Cx + \bar{w}_0 = 0$, ceea ce implică $x = -\bar{w}_0/2C$.

***Observaţie*: Este imediat că**
$\bar{w}_0 \in [2C-1, 1-2C] \Rightarrow -\bar{w}_0 \in [2C-1, 1-2C] \Rightarrow -\bar{w}_0/2C \in \Big[-\frac{1}{2C}+1, \frac{1}{2C}-1\Big]$**, iar** $0 < C < 1/2 \Rightarrow 0 < 2C < 1 \Rightarrow \frac{1}{2C} > 1 \Rightarrow -\frac{1}{2C} < -1 \Rightarrow -\frac{1}{2C}+1 < 0$**, iar** $\frac{1}{2C}-1 > 0$.
Remarcaţi faptul că separatorul optimal $(-\bar{w}_0/2C)$ se poate afla oriunde pe axa reală, dacă parametrul C este lăsat să varieze în intervalul $(0, 1/2)$!

18. **(C-SVM: o margine superioară pentru numărul de erori la antrenare)**

□ • ***CMU, 2017 fall, Nina Balcan, HW4, pr. 4.Q12***

Fie $\mathbf{S} = \{(x_1, y_1), \ldots (x_m, y_m)\}$ un set de m instanţe etichetate din $\mathbb{R}^d$, cu etichete din mulţimea $\{1, -1\}$. Vă readucem aminte că forma primală a problemei C-SVM este

$$\min_{w, w_0 \xi} \Big(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i\Big),$$
$$\textbf{a. î. } \forall i,\ y_i(w\cdot x_i + w_0) \geq 1-\xi_i, \textbf{ cu } \xi_i \geq 0.$$

Fie $\bar{w}, \bar{w}_0, \bar{\xi}$ soluţia acestei probleme. Stabiliţi o margine superioară (engl., upper bound) pentru numărul de instanţe din S care sunt clasificate în mod eronat la antrenare, folosind $\bar{\xi}_i$ pentru $i = 1, \ldots, m$.

[675] Observaţi că $2C-1 < 0$, iar $1-2C > 0$.

Răspuns:

Ţinând cont de semnificaţia [analitică a] variabilelor de destindere ξ_i, rezultă că o instanţă x_i este clasificată eronat la antrenare dacă $\bar{\xi}_i > 1$. Aşadar, numărul total de erori comise de C-SVM la antrenare poate fi scris ca $\sum_i 1_{\{\bar{\xi}_i>1\}}$, unde am folosit variabila-indicator

$$1_{\{\bar{\xi}_i>1\}} = \begin{cases} 1 & \text{dacă } \bar{\xi}_i > 1, \\ 0 & \text{în caz contrar.} \end{cases}$$

Prin urmare,

$$\sum_i 1_{\{\bar{\xi}_i>1\}} < \sum_{i\,:\,\bar{\xi}_i>1} \bar{\xi}_i \le \sum_i \bar{\xi}_i.$$

Dacă definim eroarea comisă de C-SVM la antrenare ca raportul dintre numărul de erori şi m, numărul total de instanţe de antrenament, atunci marginea superioară cerută este $\frac{1}{m}\sum_i \bar{\xi}_i$.

19. **(C-SVM: o margine superioară pentru eroarea de tip CVLOO)**

prelucrare de Liviu Ciortuz, după
• ◦ CMU, 2010 fall, Aarti Singh, midterm exam, pr. 5.2

Considerăm un C-SVM, adică o maşină cu vectori-suport cu margine "soft", care foloseşte parametrul de „destindere" C. Arătaţi că — pentru orice valoare a lui C fixată (în mod arbitrar) — eroarea produsă de acest clasificator la cross-validare cu metoda "Leave-One-Out" este mai mică sau egală cu

$$\frac{\#\text{SVs}}{m},$$

unde m este numărul exemplelor de antrenament, iar #SVs este numărul de vectori-suport obţinuţi (pentru respectiva valoare a lui C) la antrenarea acestui clasificator pe întreg setul de exemple.

***Observaţii*:**

1. Vom demonstra în mod riguros (i.e., analitic) următoarea *proprietate importantă*: La CVLOO cu C-SVM, numai vectorii-suport pot fi (eventual!) clasificaţi eronat. Altfel spus, orice instanţă x_k care nu este vector-suport nu produce eroare la CVLOO.

2. Vă readucem aminte că în contextul clasificatorului C-SVM, vectorii-suport sunt acei x_i pentru care $\bar{\alpha}_i > 0$. Sunt posibile două cazuri: fie $\bar{\alpha}_i \in (0, C)$, iar în acest caz avem $y_i(\bar{\alpha}_i \cdot x_i + \bar{w}_0) = 1$, fie $\bar{\alpha}_i = C$, iar atunci $y_i(\bar{\alpha}_i \cdot x_i + \bar{w}_0) \le 1$ (aceşti x_i constituie erori la antrenare dacă $\xi_i > 1$). Vedeţi *Rezumatul* de la rezolvarea problemei 12.e, pag. 670.

3. Se poate demonstra că rezultatul din enunţ este valabil şi pentru SVM (adică maşina cu vectori-suport cu margine "hard").

4. Eroarea de tip CVLOO este un bun indicator pentru capacitatea de generalizare a unui clasificator automat. Numărul vectorilor-suport fiind în general relativ mic în raport cu numărul total de instanţe de antrenament, marginea superioară indicată mai sus pentru eroarea CVLOO produsă de C-SVM „atestă" într-un anumit sens calitatea acestui clasificator.

Răspuns:

Folosind notaţiile de la problema 12, vom considera problemele de optimizare

$$\begin{array}{ll} & \min_{w,w_0,\xi} \left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^m \xi_i\right) \\ \textbf{a. î.} & (w\cdot x_i + w_0)y_i \geq 1-\xi_i, \textbf{ pentru } i=1,\ldots,m \\ & \xi_i \geq 0, \textbf{ pentru } i=1,\ldots,m \end{array} \qquad (\mathbf{P}')$$

şi

$$\begin{array}{ll} & \min_{w,w_0,\xi} \left(\frac{1}{2}\|w\|^2 + C\sum_{i=1,\ i\neq k}^m \xi_i\right) \\ \textbf{a. î.} & (w\cdot x_i + w_0)y_i \geq 1-\xi_i, \textbf{ pentru } i=1,\ldots,m,\ i\neq k \\ & \xi_i \geq 0, \textbf{ pentru } i=1,\ldots,m,\ i\neq k. \end{array} \qquad (\mathbf{P}'_k)$$

unde $k \in \{1,\ldots,m\}$**.**

Ştim (tot de la problema 12) că problemele $(\mathbf{P}')$ **şi** $(\mathbf{P}'_k)$ **sunt în relaţie de *dualitate tare* cu cu dualele** $(\mathbf{D}')$ **şi respectiv** $(\mathbf{D}'_k)$**:**[676]

$$\begin{array}{ll} & \max_{\alpha} \left(\sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i\cdot x_j\right) \\ \textbf{a. î.} & 0\leq \alpha_i \leq C \textbf{ pentru } i=1,\ldots,m \\ & \sum_{i=1}^m \alpha_i y_i = 0 \end{array} \qquad (\mathbf{D}')$$

şi

$$\begin{array}{ll} & \max_{\alpha} \left(\sum_{i=1,\ i\neq k}^m \alpha_i - \frac{1}{2}\sum_{i,j;\ i,j\neq k}\alpha_i\alpha_j y_i y_j x_i\cdot x_j\right) \\ \textbf{a. î.} & 0\leq \alpha_i \leq C \textbf{ pentru } i=1,\ldots,m,\ i\neq k \\ & \sum_{i=1,\ i\neq k}^m \alpha_i y_i = 0. \end{array} \qquad (\mathbf{D}'_k)$$

În plus, relaţia dintre soluţiile problemelor $(\mathbf{P}')$ **şi** $(\mathbf{D}')$ **este**

$$\bar{w} = \sum_{i=1}^m \bar{\alpha}_i y_i x_i,$$

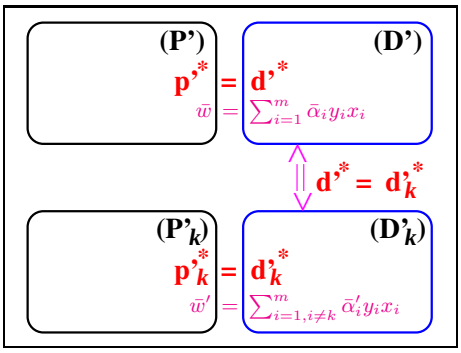
iar $\bar{w}_0$ **se obţine din relaţia** $(\bar{w}\cdot x_i + \bar{w}_0)y_i = 1-\bar{\xi}_i$ **pentru un** $i\in\{1,\ldots,m\}$ **cu proprietatea** $\bar{\alpha}_i > 0$**. Relaţia dintre soluţiile problemelor** $(\mathbf{P}'_k)$ **şi** $(\mathbf{D}'_k)$ **se exprimă în mod similar.**

Dacă în urma rezolvării problemei $(\mathbf{P}')$**, instanţa** x_k ***nu*** **este vector-suport, adică** $\bar{\alpha}_k = 0$**, atunci urmează că** $\bar{\beta}_k = C - \bar{\alpha}_k = C \neq 0$**, deci din condiţia de complementaritate Karush-Kuhn-Tucker** $\bar{\beta}_k\bar{\xi}_k = 0$ **rezultă** $\bar{\xi}_k = 0$ **şi, prin urmare**

$$y_k(\bar{w}\cdot x_k + \bar{w}_0) \geq 1-\bar{\xi}_k = 1. \qquad (322)$$

[676] Pentru definiţia dualităţii tari, vedeţi *Comentariul* de la problema 9 (pag. 658).

Lucrând în continuare cu presupoziţia că x_k *nu* este vector-suport, se poate arăta — vedeţi mai jos — că soluţiile optime $(\bar{w}, \bar{w}_0)$ pentru problemele (P′) şi (P′$_k$) sunt aceleaşi. În consecinţă, folosind relaţia (322), va rezulta că x_k este corect clasificat la CVLOO.



Demonstraţia faptului că soluţiile optime ale problemelor (P′) şi (P′$_k$) sunt aceleaşi se poate face astfel:

- **se ţine cont de relaţiile de dualitate tare menţionate mai sus, desemnate prin notaţiile (P′)≡(D′) şi (P′$_k$)≡(D′$_k$). Referindu-ne la valorile optime ale funcţiilor-obiectiv respective, putem scrie — folosind notaţii inspirate de proprietatea de dualitate tare — $p'^* = d'^*$ şi $p'^*_k = d'^*_k$;**
- **se arată că soluţia optimă pentru problema (D′) coincide cu soluţia optimă pentru problema (D′$_k$), cu singura „diferenţă“ că $\bar{\alpha}_k = 0$. (Aşadar, $d'^* = d'^*_k$.) Pe scurt, putem justifica aceasta în modul următor:**

> **Valoarea optimă pentru funcţia obiectiv a problemei (D′) este mai mare sau cel puţin egală cu valoarea optimă pentru funcţia obiectiv a problemei (D′$_k$), pentru că spaţiul de *valori admisibile* pentru variabilele α_i (i.e, valorile care satisfac restricţiile) este mai amplu. Mai departe, ştiind că $\bar{\alpha}_k = 0$, rezultă că valorile optime ale celor două probleme coincid, iar soluţia optimă $\bar{\alpha}$ pentru problema (D′) este (abstracţie făcând de $\bar{\alpha}_k$) şi soluţie optimă a problemei (D′$_k$);**[677]

[677] În manieră riguroasă, considerând (în principal de dragul simplificării exprimării) problema

$$\begin{array}{ll} & \max_\alpha \left(\sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \right) \\ \text{a. î.} & 0 \le \alpha_i \le C \text{ pentru } i = 1, \ldots, m \text{ cu } i \neq k \text{ şi } \alpha_k = 0 \\ & \sum_{i=1}^m \alpha_i y_i = 0 \end{array} \qquad (\text{D}'')$$

veţi observa mai întâi că problemele (D′) şi (D″) sunt *echivalente* — adică orice soluţie optimă a lui (D′) este şi soluţie optimă a lui (D″) — şi apoi că problemele (D″) şi (D′$_k$), deşi diferite din punct de vedere „sintactic“, sunt practic identice, întrucât pe de o parte α_k (componenta k a vectorului generic α) este 0 în problema (D″) şi lipseşte în problema (D′$_k$), iar pe de altă parte substituind $\alpha_k = 0$ în funcţia obiectiv a problemei (D″) obţinem funcţia obiectiv a problemei (D′$_k$).

Pentru a demonstra că problemele (D′) şi (D″) sunt echivalente, observaţi mai întâi că orice soluţie admisibilă a problemei (D″) este şi soluţie admisibilă pentru problema (D′). Aşadar, mulţimea de soluţii admisibile pentru problema (D′) este mai amplă decât mulţimea de soluţii admisibile pentru problema (D″). Notăm cu F funcţia obiectiv a problemei (D′), adică $F(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j x_i \cdot x_j$. Fie $\bar{\alpha}$ soluţie optimă a problemei (D′). Componenta sa de pe poziţia k este nulă (adică, $\bar{\alpha}_k = 0$), fiindcă x_k nu este vector-suport. Urmează că $F(\bar{\alpha}) \ge F(\alpha)$ pentru orice soluţie adimisibilă α a problemei (D′), deci

$$F(\bar{\alpha}) = \max\{F(\alpha) \mid \alpha \text{ este soluţie adimisibilă a problemei (D}')\}.$$

Întrucât mulţimea de soluţii admisibile pentru problema (D′) este mai amplă decât mulţimea de soluţii admisibile pentru problema (D″), rezultă că

$$F(\bar{\alpha}) \ge \max\{F(\alpha) \mid \alpha \text{ este soluţie adimisibilă a problemei (D}'')\}.$$

Însă ţinând cont pe de o parte că $\bar{\alpha}$ este şi soluţie admisibilă pentru problema (D″), iar pe de altă parte că $\bar{\alpha}_k = 0$, rezultă că inegalitatea precedentă devine

$$F(\bar{\alpha}) = \max\{F(\alpha) \mid \alpha \text{ este soluţie adimisibilă a problemei (D}'')\}.$$

Aşadar, $\bar{\alpha}$ este soluţie optimă şi pentru problema (D″).

• se utilizează în final relaţiile de legătură de forma $\bar{w} = \sum_{i=1}^{m} \bar{\alpha}_i y_i x_i$ (cu $\bar{w}_0$ determinat corespunzător).

***Observaţie*: Demonstraţia proprietăţii din enunţ pentru SVM [cu margine "hard"] se poate face fie urmând „tipicul" demonstraţiei de mai sus — adică, făcând apel la legătura dintre forma primală şi forma duală a problemei de optimizare SVM —, fie (considerabil mai simplu!) folosind o formă uşor generalizată a proprietăţii pe care am demonstrat-o la problema 2, care face legătura dintre problema de optimizare SVM şi *interpretarea ei geometrică*.**

20. **(Problema de optimizare C-SVM: o formulare echivalentă, dar fără restricţii, folosind în schimb funcţia de cost / pierdere *hinge*)**

prelucrare de Liviu Ciortuz, după
■ • CMU, 2008 fall, Eric Xing, HW2, pr. 1.2
CMU, 2017 fall, Nina Balcan, HW4, pr. 4.1

Considerăm m instanţe de antrenament $\{x_i, y_i\}_{i=1}^{m}$. Vă readucem aminte că problema SVM cu margine "soft" şi parametru de „destindere" $C > 0$ poate fi formulată ca o problemă de optimizare (pătratică) cu restricţii:

$$\begin{array}{ll} & \min_{w,w_0,\xi} \ \left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i\right) \\ \textbf{a. î.} & (w\cdot x_i + w_0)y_i \geq 1-\xi_i, \ \textbf{pentru } i = 1,\ldots,m \\ & \xi_i \geq 0, \ \textbf{pentru } i = 1,\ldots,m \end{array} \qquad \textbf{(P}')$$

a. Demonstraţi că formularea de mai sus este *echivalentă* cu o problemă de optimizare (tot pătratică) *fără* restricţii, de forma:

$$\min_{w,w_0} \ \left(\|w\|^2 + \lambda\sum_{i=1}^{m}\max(1-y_i(w\cdot x_i+w_0),\,0)\right), \qquad \textbf{(L)}$$

unde λ este un parametru real pozitiv fixat.

b. Exprimaţi valoarea noului parametru λ în funcţie de parametrul de „destindere" C.

c. Cum apreciaţi din punct de vedere „calitativ" această nouă formulare a problemei de optimizare C-SVM?

Răspuns:

a. Din restricţiile din forma primală a problemei C-SVM, avem

$$\xi_i \geq 1 - y_i(w\cdot x_i + w_0) \ \textbf{şi} \ \xi_i \geq 0,$$

ceea ce echivalează cu $\xi_i \geq \max(1-y_i(w\cdot x_i+w_0),\,0)$ pentru $i = 1,\ldots,m$.

Operatorul $\min$ din formularea *obiectivului* problemei de optimizare C-SVM (P$'$) implică faptul că $\bar{\xi}_i$ din soluţia optimă a acestei probleme $(\bar{w}, \bar{w}_0, \bar{\xi})$ va fi setată chiar la valoarea $\max(1-y_i(\bar{w}\cdot x_i+\bar{w}_0),\,0)$.

Prin urmare, este naturală transformarea funcţiei obiectiv a problemei (P$'$) în funcţia obiectiv a problemei (L). Ceea ce nu apare a fi la fel de uşor de explicat este renunţarea la restricţiile formulate în cadrul problemei (P$'$) atunci când se face trecerea la problema (L). De aceea, în cele ce urmează este necesar

să arătam că aceste două probleme sunt *echivalente*, dovedind că orice soluţie optimă a problemei (P′) corespunde unei soluţii optime a problemei (L) şi invers, orice soluţie optimă a problemei (L) corespunde unei soluţii optime a problemei (P′).

Considerăm deci mai întâi $\bar{w}, \bar{w}_0, \bar{\xi}$ o soluţie optimă a problemei C-SVM (P′). Luând $\lambda = 2C$, vom arăta că $\bar{w}, \bar{w}_0$ este şi soluţie optimă a noii probleme de optimizare (L). Presupunem prin *prin reducere la absurd* că $\bar{w}, \bar{w}_0$ nu este soluţie optimă a noii probleme de optimizare (L). Rezultă că există w^*, w_0^* o soluţie optimă a problemei (L), care este mai bună decât soluţia $\bar{w}, \bar{w}_0$, adică

$$L(w^*, w_0^*) < L(\bar{w}, \bar{w}_0),$$

unde prin L am notat funcţia obiectiv a problemei de optimizare (L). Notând $\xi_i^* \stackrel{not.}{=} \max(1-y_i(w^* \cdot x_i + w_0^*), 0)$, rezultă că w^*, w_0^*, ξ^* este pentru problema C-SVM (P′) o soluţie fezabilă, adică satisface sistemul de restricţii $y_i(w \cdot x_i + w_0) \geq 1 - \xi_i$ şi $\xi_i \geq 0$, pentru $i = 1, \ldots, m$. Pe lângă aceasta, notând cu P' funcţia obiectiv a problemei de optimizare C-SVM (P′), rezultă

$$\underbrace{P'(w^*, w_0^*, \xi^*)}_{=\frac{1}{2}L(w^*, w_0^*)} < \underbrace{P'(\bar{w}, \bar{w}_0, \bar{\xi})}_{=\frac{1}{2}L(\bar{w}, \bar{w}_0)}.$$

În această relaţie, egalitatea $P'(w^*, w_0^*, \xi^*) = \frac{1}{2}L(w^*, w_0^*)$ are loc datorită modului în care am definit L, P' şi ξ^*, iar egalitatea $P'(\bar{w}, \bar{w}_0, \bar{\xi}) = \frac{2}{L}(\bar{w}, \bar{w}_0)$ are loc datorită operatorului $\min$ din cadrul obiectivului problemei (P′). Inegalitatea $P'(w^*, w_0^*, \xi^*) < P'(\bar{w}, \bar{w}_0, \bar{\xi})$ implică faptul că soluţia w^*, w_0^*, ξ^* constituie pentru problema (P′) o soluţie mai bună decât soluţia optimă $\bar{w}, \bar{w}_0, \bar{\xi}$, ceea ce este absurd. Prin urmare, $\bar{w}, \bar{w}_0$ este soluţie optimă a problemei (L).

Invers, trebuie să arătăm acum că luând $\lambda = 2C$, orice soluţie optimă $\bar{w}$, $\bar{w}_0$ a problemei (L) este – printr-o extensie naturală — soluţie optimă a problemei C-SVM (P′). Este imediat (vedeţi raţionamentul de mai sus) că orice soluţie optimă $\bar{w}$, $\bar{w}_0$ a problemei (L), augmentată cu $\bar{\xi}_i \stackrel{not.}{=} \max(1 - y_i(\bar{w} \cdot x_i + \bar{w}_0), 0)$ satisface restricţiile problemei (P′). Presupunem, ca şi mai sus, prin reducere la absurd că $\bar{w}$, $\bar{w}_0$, $\bar{\xi}$ nu este soluţie optimă a problemei (P′). Considerând w^*, w_0^*, ξ^* o soluţie optimă a problemei (P′), rezultă:

$$\underbrace{P'(w^*, w_0^*, \xi^*)}_{=\frac{1}{2}L(w^*, w_0^*)} < \underbrace{P'(\bar{w}, \bar{w}_0, \bar{\xi})}_{=\frac{1}{2}L(\bar{w}, \bar{w}_0)}.$$

Aşadar, $L(w^*, w_0^*) < L(\bar{w}, \bar{w}_0)$, ceea ce contrazice faptul că $\bar{w}, \bar{w}_0$ este soluţie optimă pentru problema (L). Conchidem că presupunerea făcută mai sus este falsă, deci $\bar{w}, \bar{w}_0, \bar{\xi}$ este soluţie optimă pentru problema (P′).

b. $\lambda = 2C$, conform rezolvării de la punctul a.

c. Forma nou-obţinută pentru problema de optimizare C-SVM este caracterizată de i. lipsa restricţiilor asupra variabilelor şi ii. de realizarea unui echilibru (engl., trade-off) între

– simplitate,[678] reflectată de termenul $\| w \|^2$;
– o bună capacitate de predicţie / generalizare în cazul neseparabilităţii liniare a datelor de antrenament, graţie termenului $\lambda \sum_{i=1}^{m} \max(1 - y_i(w \cdot x_i + w_0), 0)$.

***Observaţie importantă*: [Relaţia cu regresia logistică]**
Este imediat că problema de optimizare (L) este echivalentă cu problema

$$\min_{w, w_0} \Big(\theta \|w\|^2 + \sum_{i=1}^{m} \max(1 - y_i(w \cdot x_i + w_0), 0) \Big), \tag{323}$$

dacă se consideră parametrul (fixat) $\theta = \dfrac{1}{\lambda} > 0$. Comparând problema de optimizare (323) cu problema de optimizare din definiţia *regresiei logistice* cu termen de regularizare L_2,[679] putem observa o mare similaritate. Diferenţa constă (doar!) în funcţia de cost / pierdere folosită: SVM foloseşte *funcţia de cost hinge*, definită prin $f(z) = \max(1 - z, 0)$, pe când regresia logistică foloseşte *funcţia de cost logistică* $\ln(1 + e^{-z})$.[680]

***Observaţie*: Pentru detalii de implementare (şi aplicare) a acestei forme a problemei de optimizare SVM folosind *metoda subgradientului* (pentru că funcţia *hinge* nu este derivabilă pe tot domeniul de definiţie),[681] vedeţi articolul *Pegasos: Primal estimated sub-gradient solver for SVM*, de Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, Andrew Cotter, publicat în revista *Mathematical programming*, 127(1):3–30, 2011.[682]**

21. **(Algoritmul Pegasos [simplificat]: justificarea regulii de actualizare; kernelizare)**

***formulare de Liviu Ciortuz, pornind de la* • · *CMU, 2017 fall, Nina Balcan, HW4, pr. 4.1*[683]**

A. Fie $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ o mulţime de m exemple de antrenament, cu $x_i \in \mathbb{R}^d$ şi $y_i \in \{-1, +1\}$ pentru $i = 1, \ldots, m$.

Considerăm următoarea formulare a problemei de optimizare C-SVM fără restricţii, folosind funcţia de cost / pierdere (engl., loss) *hinge*:[684]

$$\min_{w \in \mathbb{R}^d} \Big(\frac{\lambda}{2} \|w\|^2 + \frac{1}{m} \sum_{i=1}^{m} \max(1 - y_i(w \cdot x_i), 0) \Big), \tag{324}$$

[678] În raport cu alţi clasificatori, de exemplu reţelele neuronale artificiale.
[679] Puteţi vedea problemele 13.c, 16.a şi 17 de la capitolul *Metode de regresie*, mai precis relaţiile (183), (187), (204) şi (205).
[680] În mod similar, regresia liniară foloseşte ca funcţie de cost suma pătratelor erorilor, iar algoritmul AdaBoost foloseşte funcţia de cost [negativ-]exponenţială e^{-z}. Pentru un cadru unitar de prezentare a acestor metode de învăţare (foarte diferite!), bazat pe minimizarea costurilor / pierderilor, vă recomandăm să citiţi documentul *Supplemental lecture notes*, de John Duchi, de la Universitatea Stanford.
[681] Pentru definiţia noţiunii de *subgradient*, vedeţi pr. 81 de la capitolul de *Fundamente*.
[682] Pentru implementarea unei versiuni simplificate a acestui algoritm şi aplicarea lui, vedeţi problema 21 şi CMU, 2017 fall, Nina Balcan, HW4, pr. 4.1.
Pentru aplicarea metodei subgradientului în cazul regresiei liniare cu regularizare de normă L_1, vedeţi ex. 28 de la capitolul *Metode de regresie*.
[683] În versiunea originală — CMU, 2017 fall, Nina Balcan, HW4, pr. 4.1 —, această problemă este de tip implementare; acolo se cere să se implementeze algoritmul Pegasos simplificat (prezentat aici) şi apoi să se aplice pe un subset al setului de date MNIST.
[684] Vedeţi problema 20.

unde λ este un parametru real pozitiv fixat.[685] Remarcaţi faptul că în această versiune nu folosim termen liber (w_0).

Scopul acestui exerciţiu este să arătăm că funcţia obiectiv din relaţia (324) poate fi optimizată folosind *metoda subgradientului descendent stochastic* (engl., Stochastic Sub-Gradient Descent, SSGD).[686] Această abordare este foarte simplă şi se scalează uşor la seturi mari de date de antrenament. În cadrul metodei SSGD selectăm în mod aleatoriu un exemplu de antrenament la fiecare iteraţie şi actualizăm vectorul de ponderi w făcând un pas (engl., step) în direcţia opusă *subgradientului* funcţiei de cost / pierdere.[687] Forma algoritmului SSGD pentru problema dată este următoarea:

Input: S, λ, η, T;
initialize the weight vector $w = 0$;
for $t = 1, \ldots, T$
{
choose $i_t \in \{1, \ldots m\}$ uniformly at random;
set $\eta_t = \dfrac{1}{\lambda t}$;
if $y_{i_t} w \cdot x_{i_t} < 1$ then
set $w \leftarrow (1 - \lambda\eta_t)w + \eta_t y_{i_t} x_{i_t}$;
else
set $w \leftarrow (1 - \lambda\eta_t)w$;
}
end for;
return w;

a. Arătaţi că regula de actualizare a ponderilor w din acest algoritm corespunde într-adevăr subgradientului descendent stochastic.

b. Arătaţi că acest algoritm poate fi kernelizat. Apoi scrieţi pseudo-codul versiunii kenelizate a acestui *algoritm Pegasos* [*simplificat*].

***Sugestie*: Veţi justifica mai întâi faptul că soluţia w poate fi scrisă ca o combinaţie liniară de instanţe de antrenament x_i, iar apoi că în regula de decizie $sign(w \cdot x)$, instanţele de antrenament x_i şi instanţa de test x apar doar ca argumente ale produsului scalar (şi, în consecinţă, doar ca argumente ale funcţiei-nucleu considerate, K, atunci când se lucrează în spaţiul de trăsături). Pentru a scrie pseudo-codul versiunii kenelizate a acestui algoritm, vă puteţi inspira din pseudo-codul versiunii kenelizate a Perceptronului Rosenblatt.[688]**

B. *Comentariu*: Algoritmul care a fost dat în enunţ este o versiune simplificată / particulară în raport cu algoritmul Pegasos:[689]

Input: S, λ, T, k;
choose w_1 s. t. $\|w_1\| \leq 1/\sqrt{\lambda}$;
for $t = 1, \ldots, T$

[685] Corespondenţa dintre parametrul θ din relaţia (323) şi parametrul λ din relaţia (324) este următoarea: $\theta = \dfrac{m\lambda}{2}$.

[686] Pentru o variantă (nestochastică) a algoritmului subgradientului descendent, vedeţi problema 171 de la capitolul de *Fundamente*. Acolo se arată că algoritmul Perceptron (Rosenblatt) poate fi interpretat ca fiind o instanţă a algoritmului subgradientului descendent.

[687] Noţiunea de subgradient generalizează noţiunea de gradient în cazul funcţiilor care sunt nederivabile pe tot domeniul de definiţie. Vedeţi problema 81 de la capitolul de *Fundamente*.

[688] Vedeţi ex. 19 de la capitolul *Reţele neuronale artificiale*.

[689] Vedeţi articolul *Pegasos: Primal Estimated sub-GrAdient SOlver for SVM*, de Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, Andrew Cotter, *Mathematical programming*, 127(1):3–30, 2011.

{
choose $A_t \subseteq S$, where $|A_t| = k$;
set $A_t^+ = \{(x, y) \in A_t : y(w_t \cdot x) < 1\}$;
set $\eta_t = \frac{1}{\lambda t}$;
set $w_{t+\frac{1}{2}} = (1 - \lambda\eta_t)w_t + \frac{\eta_t}{k} \sum_{(x,y)\in A_t^+} yx$;
set $w_{t+1} = \min\left\{1, \frac{1/\sqrt{\lambda}}{\|w_{t+\frac{1}{2}}\|}\right\} w_{t+\frac{1}{2}}$;
}
end for;
return w_{T+1};

Mulţimea A_t este formată din k exemple selectate i.i.d. din setul de date de antrenament S.

Observaţii:
1. Nici în acest algoritm nu folosim termen liber (w_0). (Pentru varianta în care se foloseşte termen liber, vedeţi articolul indicat la nota de subsol 689.)
2. La elaborarea acestui algoritm, mai exact la iteraţia t, funcţia obiectiv din relaţia (324) este înlocuită cu o aproximare a ei:

$$\frac{\lambda}{2}\|w\|^2 + \frac{1}{k} \sum_{(x,y)\in A_t} \max(1 - y(w_t \cdot x), 0).$$

3. $\frac{1/\sqrt{\lambda}}{\|w_{t+\frac{1}{2}}\|} w_{t+1}$ este proiecţia lui $w_{t+\frac{1}{2}}$ pe suprafaţa hipersferei $B = \{w : \|w\| \leq 1/\sqrt{\lambda}\}$, întrucât se poate demonstra că soluţia optimă a problemei (324) se află în această hipersferă (vedeţi articolul indicat la nota de subsol 689).

c. Arătaţi că dacă în algoritmul Pegasos se setează w_1 la vectorul 0 din $\mathbb{R}^d$, atunci acest algoritm este kernelizabil. (*Atenţie*! Raţionamentul este similar cu cel de la punctul b. Însă, în plus, va trebui să arătaţi că $\|w_{t+\frac{1}{2}}\|$ se calculează folosind instanţele x_i doar ca argumente pentru funcţia-nucleu K.)

Răspuns:

a. Justificarea faptului că regula de actualizare a ponderilor din algoritmul prezentat în enunţ corespunde subgradientului descendent sochastic a fost deja făcută la rezolvarea problemei 81.b de la capitolul de *Fundamente*. Trebuie să mai precizăm doar că
***i*. „stochasticitatea" este dată de alegerea aleatorie a instanţei de antrenament care este tratată la iteraţia t;**
***ii*. faptul că înainte de a se adăuga la vectorul w componenta $\Delta \stackrel{not.}{=} \eta_t y_{i-t} x_{i_t}$ se face micşorarea mărimii vectorului w (şi anume, prin înmulţirea sa cu constanta $1 - \lambda\eta_t$) are ca efect reducerea riscului de overfitting.**

b. Pentru antrenare, vom arăta mai întâi prin inducţie completă că în „spaţiul de intrare" la orice iteraţie t vectorul w este o combinaţie liniară de instanţele de antrenament x_i.

Pentru iteraţia $t = 0$ (adică la iniţializare), avem $w = 0$, deci este verificată ipoteza inductivă. Pentru pasul inductiv, presupunem că la iteraţia t vectorul w se scrie ca o combinaţie liniară de instanţe de antrenament x_i. Ţinând cont că regula de actualizare a vectorului w se poate scrie sub forma

$$w \leftarrow (1-\lambda\eta_t)w + \begin{cases} \eta_t y_{i_t} x_{i_t}, & \textbf{dacă } y_{i_t} w \cdot x_{i_t} < 1 \\ 0, & \textbf{altfel}, \end{cases}$$

rezultă că noua valoare a vectorului w care este calculată la iteraţia t este tot o combinaţie liniară de instanţe de antrenament x_i.

Aşadar, conform principiului inducţiei complete, la orice iteraţie t vectorul w poate fi scris ca o combinaţie liniară de instanţe de antrenament x_i.

În consecinţă, scriind w calculat în „spaţiul de trăsături" sub forma $\sum_{j=1}^{m} \alpha_j \phi(x_j)$ unde $\alpha_j \in \mathbb{R}$ pentru $j = 1, \ldots, m$, devine imediat faptul că produsele $w \cdot \phi(x_{i_t})$ se pot calcula cu ajutorul funcţiei-nucleu K:

$$\left(\sum_{j=1}^{m} \alpha_j \phi(x_j) \right) \cdot \phi(x_{i_t}) = \sum_{j=1}^{m} \alpha_j \phi(x_j) \cdot \phi(x_{i_t}) = \sum_{j=1}^{m} \alpha_j K(x_j, x_{i_t}).$$

În sfârşit, pentru testare / generalizare facem un raţionament similar:

$$\left(\sum_{j=1}^{m} \alpha_j \phi(x_j) \right) \cdot \phi(x) = \sum_{j=1}^{m} \alpha_j \phi(x_j) \cdot \phi(x) = \sum_{j=1}^{m} \alpha_j K(x_j, x).$$

Versiunea kernelizată a algoritmului Pegasos simplificat care a fost dat în enunţ este următoarea:

Input: S, λ, η, T;
initialize $\alpha_j = 0$ for $j = 1, \ldots, n$;
for $t = 1, \ldots, T$
{
set $\eta_t = \dfrac{1}{\lambda t}$;
for $j = 1, \ldots, n$
set $\alpha_j \leftarrow (1 - \lambda\eta_t)\alpha_j$;
end for;
choose $i_t \in \{1, \ldots n\}$ uniformly at random;
if $y_{i_t}\left(\sum_{j=1}^{m} \alpha_j K(x_j, x_{i_t}) < 1 \right)$ then
set $\alpha_j \leftarrow \alpha_j + \eta_t y_{i_t}$;
else ;
}
end for;
return $\alpha_1, \ldots, \alpha_m$;

Condiţia din instrucţiunea if **din acest pseudo-cod sugerează faptul că este foarte util să se memoreze matricea Gram ale cărei elemente sunt de forma $K(x_j, x_i)$. Însă această chestiune pune în evidenţă şi *dezavantajul* algoritmilor kernelizaţi: necesitatea folosirii unui spaţiu de memorie de ordinul $\mathcal{O}(m^2)$, unde m este numărul instanţelor de antrenament.**

c. După cum s-a precizat deja în enunţ, justificarea kernelizării algoritmului Pegasos [simplificat] este similară cu demonstraţia de la punctul b; doar constantele din faţa instanţelor x_i care apar în scrierea vectorului w se modifică. Rămâne doar să arătăm că în „spaţiul de trăsături" $\|w_{t+\frac{1}{2}}\|$ se poate calcula în aşa fel încât [în acest calcul] instanţele x_i să apară doar ca argumente ale funcţiei-nucleu K. Într-adevăr,

$$\begin{aligned}\|w_{t+\frac{1}{2}}\|^2 = w_{t+\frac{1}{2}}^2 &= w_{t+\frac{1}{2}} \cdot w_{t+\frac{1}{2}} \\ &= \Big((1-\lambda\eta_t)w_t + \frac{\eta_t}{k}\sum_{(x,y)\in A_t^+} y\phi(x)\Big)^2 \\ &= (1-\lambda\eta_t)^2 w_t^2 + 2(1-\lambda\eta_t)\frac{\eta_t}{k}w_t \cdot \Big(\sum_{(x,y)\in A_t^+} y\phi(x)\Big) + \frac{\eta_t^2}{k^2}\Big(\sum_{(x,y)\in A_t^+} y\phi(x)\Big)^2.\end{aligned}$$

Justificarea este practic finalizată dacă folosim faptul că în „spaţiul de trăsături" vectorul w_t este o combinaţie liniară de vectorii $\phi(x_i)$ (deci w_t^2 se scrie ca o combinaţie liniară de „imagini" ale funcţiei-nucleu, $K(x_i, x_j)$), iar

$$\begin{aligned}\Big(\sum_{(x,y)\in A_t^+} y\phi(x)\Big)^2 &= \Big(\sum_{(x,y)\in A_t^+} y\phi(x)\Big)\cdot\Big(\sum_{(x',y')\in A_t^+} y'\phi(x')\Big) \\ &= \sum_{(x,y)\in A_t^+}\sum_{(x',y')\in A_t^+} yy'\phi(x)\cdot\phi(x') = \sum_{(x,y)\in A_t^+}\sum_{(x',y')\in A_t^+} yy'K(x,x').\end{aligned}$$

22. **(C-SVM: deducerea relaţiilor folosite în algoritmul SMO)**

□ • ○ ***Liviu Ciortuz, 2018, după***
■ ***Nello Cristianini, John Shawe-Taylor,***
An Introduction to Support Vector Machines,
Cambridge University Press, 2000, pp. 139-140

În această problemă veţi face deducerea relaţiilor (pentru *actualizare* şi respectiv pentru *oprire*) folosite în algoritmul SMO,[690] care rezolvă forma duală a problemei de optimizare C-SVM. Acest algoritm foloseşte ca strategie de optimizare *metoda creşterii pe coordonate* (engl., *coordinate ascent*).[691] Aceasta este o metodă iterativă, pentru optimizarea unei funcţii de mai multe variabile — în cazul nostru funcţia concavă $L_D(\alpha_1, \ldots, \alpha_m)$. În *cazul clasic* al aplicării acestei metode, la fiecare iteraţie se alege câte o variabilă care este lăsată liberă, iar celalte variabile sunt fixate; la iteraţia respectivă, optimizarea funcţiei obiectiv se face în raport cu variabila liberă. Însă, spre deosebire de cazul clasic, la aplicarea metodei creşterii pe coordonate pentru rezolvarea problemei duale C-SVM, datorită restricţiei $\sum_{i=1}^m y_i\alpha_i = 0$ va trebui ca la fiecare iteraţie să alegem două variabile pe care să le lăsăm libere, iar pe restul să le fixăm:[692]

[690] John C. Platt, *Sequential Minimal Optimization: A fast algorithm for training Support Vector Machines.* Microsoft Research, Technical report MSR-TR-98-14, 1998. O variantă simplificată a algoritmului SMO este prezentată de Andrew Ng în *Machine Learning, Lecture Notes*, Part V, section 9.

[691] Pentru folosirea *metodei descreşterii pe coordonate* (engl., *coordinate descent*) la rezolvarea regresiei liniare cu regularizare de normă L_1, vedeţi ex. 11 de la capitolul *Metode de regresie*. Algoritmii AdaBoost şi AdaBoost generalizat (ex. 26 şi 29 de la cap. *Arbori de decizie*) folosesc şi ei metoda descreşterii pe coordonate, iar una dintre cele două coordonate corespunde aşa-numitelor ipoteze „slabe" (care nu neapărat trebuie optimizate). De asemenea, metoda descreşterii pe coordonate este folosită în mod indirect la algoritmul K-means (văzut ca algoritm de minimizare a criteriului sumei celor mai mici pătrate (J)) şi în mod direct la algoritmul EM; vedeţi problema 12.a de la capitolul de *Clusterizare* şi respectiv problema 1 de la capitolul *Schema algoritmică EM*.

[692] În prezenta problemă nu includem detalii despre modul în care se aleg aceste două variabile.

Repeat until convergence{
1. Select some pair α_i and α_j to update next (using a heuristic that tries to pick the two that will allow us to make the biggest progress towards the global maximum).
2. Reoptimize $L_D(\alpha)$ with respect to α_i and α_j, while holding all the other α_k's $(k \neq i, j)$ fixed.

}

a. Folosind notaţiile standard de la problema de optimizare C-SVM (în forma primală şi respectiv forma duală; vedeţi pr. 12) şi presupunând că *variabilele libere* la iteraţia curentă sunt α_1 şi α_2, demonstraţi că lagrangeanul dual L_D are la această iteraţie soluţiile optime

$$\begin{aligned}\alpha_2^{new,\ unclipped} &= \alpha_2^{\boldsymbol{old}} + \frac{y_2(E_2 - E_1)}{\eta}\\ \alpha_1^{new,\ unclipped} &= \alpha_1^{\boldsymbol{old}} + y_1\, y_2(\alpha_2^{\boldsymbol{old}} - \alpha_2^{new,\ unclipped}),\end{aligned}$$

unde[693]

$$\begin{aligned}w &= \sum_{i=1}^{m} y_i\, \alpha_i\, x_i\\ E_k &= \underbrace{w \cdot x_k + w_0}_{not.:f(x_k)} - y_k \text{ pentru } k \in \{1,2\}\\ \eta &= -\|x_1 - x_2\|^2.\end{aligned}$$

b. Vă readucem aminte că $0 \leq \alpha_j \leq C$ pentru $j \in \{1,2\}$ (vedeţi problema 12.d). La fiecare iteraţie a algoritmului SMO trebuie calculate două *margini* (engl., bounds), L şi H, astfel încât să restricţionăm şi mai mult variabila α_2: $0 \leq L \leq \alpha_2 \leq H \leq C$.

Demonstraţi că (în contextul problemei noastre) aceste margini sunt conform următoarelor relaţii:

- **dacă $y_1 \neq y_2$, atunci $L = \max(0, \alpha_2 - \alpha_1)$, $H = \min(C, C + \alpha_2 - \alpha_1)$;**
- **dacă $y_1 = y_2$, atunci $L = \max(0, \alpha_1 + \alpha_2 - C)$, $H = \min(\alpha_1 + \alpha_2, C)$.**

În consecinţă, *regulile de actualizare* vor fi:

$$\alpha_2^{\boldsymbol{new,\ clipped}} = \begin{cases} H & \text{dacă } \alpha_2^{\boldsymbol{new,\ unclipped}} > H \\ \alpha_2^{\boldsymbol{new,\ unclipped}} & \text{dacă } L \leq \alpha_2^{\boldsymbol{new,\ unclipped}} \leq H \\ L & \text{dacă } \alpha_2^{\boldsymbol{new,\ unclipped}} < L \end{cases} \tag{325}$$

$$\alpha_1^{\boldsymbol{new,\ clipped}} = \alpha_1^{\boldsymbol{old}} + y_1 y_2\big(\alpha_2^{\boldsymbol{old}} - \alpha_2^{\boldsymbol{new,\ clipped}}\big),$$

unde calificativele *new* şi *old* desemnează noile şi respectiv vechile valori ale variabilelor α_1 şi α_2, iar calificativul *clipped* (spre deosebire de *unclipped*) corespunde „ajustării" valorii lui α_2 ca urmare a aplicării restricţiei $L \leq \alpha_2 \leq H$.

c. Algoritmul SMO foloseşte următoarea *condiţie de oprire*:

$$\textbf{NOT } [(\alpha_i < C \ \textbf{ AND } \ y_i E_i < -\boldsymbol{tol}) \textbf{ OR } (\alpha_i > 0 \ \textbf{ AND } \ y_i E_i > \boldsymbol{tol})], \ i = 1, \ldots, m,$$

[693] În scrierea lui w sub forma $\sum_{i=1}^{m} y_i\, \alpha_i\, x_i$ se va subînţelege că α_1 şi α_2 sunt de fapt α_1^{old} şi respectiv α_2^{old}.

unde *tol* este un număr pozitiv (mic, fixat în prealabil), numit *parametru de toleranţă*. Demonstraţi că această condiţie este implicată în mod natural de relaţiile de tip Karush-Kuhn-Tucker (321) **care au fost deduse pentru problema de optimizare C-SVM (vedeţi problema 12.e).**

Răspuns:

a. Ştim — conform problemei 12.d — că forma duală a problemei de optimizare C-SVM are funcţia obiectiv $L_D(\alpha) = \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i=1}^m \sum_{j=1}^m y_i y_j \alpha_i \alpha_j x_i \cdot x_j$**, cu restricţiile** $\sum_{i=1}^m y_i \alpha_i = 0$ **(atât pentru noile cât şi pentru vechile valori ale variabilelor) şi** $0 \leq \alpha_i \leq C$ **pentru** $i = 1, \ldots, m$**. Notând**

$$v_i = \Big(\underbrace{\sum_{j=3}^m y_j\, \alpha_j\, x_j}_{w-(y_1\alpha_1 x_1 + y_2 \alpha_2 x_2)} \Big) \cdot x_i \text{ pentru } i = 1, 2, \tag{326}$$

va rezulta că

$$v_i = w \cdot x_i - (y_1 \alpha_1 x_1 \cdot x_i + y_2 \alpha_2 x_2 \cdot x_i) \text{ pentru } i \in \{1, 2\} \tag{327}$$

şi

$$\begin{aligned} L_D(\alpha_1, \alpha_2) &= \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i=1}^m \sum_{j=1}^m y_i y_j \alpha_i \alpha_j x_i \cdot x_j \\ &= \alpha_1 + \alpha_2 + \sum_{j=3}^m \alpha_j - \frac{1}{2}\alpha_1^2 x_1^2 - \frac{1}{2}\alpha_2^2 x_2^2 - \frac{1}{2}\sum_{j=3}^m \alpha_j^2 x_j^2 - y_1 y_2 \alpha_1 \alpha_2 x_1 \cdot x_2 \\ &\quad \underbrace{- y_1 \alpha_1 x_1 \cdot \sum_{j=3}^m y_j \alpha_j x_j}_{y_1\alpha_1 v_1} \underbrace{- y_2 \alpha_2 x_2 \cdot \sum_{j=3}^m y_j \alpha_j x_j}_{y_2\alpha_2 v_2} - \frac{1}{2}\sum_{j=3}^m \sum_{j'=3\, j' \neq j}^m y_j y_{j'} \alpha_j \alpha_{j'} x_j \cdot x_{j'} \\ &= \alpha_1 + \alpha_2 - \frac{1}{2}\alpha_1^2 x_1^2 - \frac{1}{2}\alpha_2^2 x_2^2 - \underbrace{y_1 y_2}_{\textbf{\textit{not.}}: s} \alpha_1 \alpha_2 x_1 \cdot x_2 - y_1 \alpha_1 v_1 - y_2 \alpha_2 v_2 + \textbf{\textit{const}}_1. \end{aligned}$$

Întrucât $y_1\alpha_1 + y_2\alpha_2 = -\sum_{j=3}^m y_j \alpha_j$**, înmulţind această egalitate cu** y_1 **— despre care ştim că aparţine mulţimii** $\{-1, 1\}$ **—, vom obţine**

$$\alpha_1 + \underbrace{y_1 y_2}_{s} \alpha_2 = \underbrace{-y_1 \sum_{j=3}^m y_j \alpha_j}_{\textbf{\textit{const}}_2}.$$

Aşadar,

$$\alpha_1^{\textbf{\textit{old}}} + s\alpha_2^{\textbf{\textit{old}}} = \underbrace{\textbf{\textit{const}}_2}_{\textbf{\textit{not.}}: \gamma} = \alpha_1^{\textbf{\textit{new, unclipped}}} + s\alpha_2^{\textbf{\textit{new, unclipped}}}. \tag{328}$$

Substituind $\alpha_1 = \gamma - s\alpha_2$ **în expresia lagrangeanului** $L_D(\alpha_1, \alpha_2)$**, vom obţine — renunţând totodată la argumentul** α_1 **— următoarea expresie:**

$$\begin{aligned} L_D(\alpha_2) &= \gamma - s\alpha_2 + \alpha_2 - \frac{1}{2}(\gamma - s\alpha_2)^2 x_1^2 - \frac{1}{2}\alpha_2^2 x_2^2 - s(\gamma - s\alpha_2)\alpha_2 x_1 \cdot x_2 \\ &\quad - y_1(\gamma - s\alpha_2)v_1 - y_2 \alpha_2 v_2 + \textbf{\textit{const}}_1. \end{aligned}$$

Pentru a maximiza $L_D(\alpha_2)$, mai întâi vom calcula derivata sa în raport cu α_2, după care vom egala această derivată cu 0:

$$\begin{aligned}\frac{\partial L_D(\alpha_2)}{\partial \alpha_2} &= -s+1+\frac{1}{2}2s(\gamma-s\alpha_2)x_1^2-\frac{1}{2}2\alpha_2x_2^2-s\gamma x_1\cdot x_2+2\alpha_2x_1\cdot x_2+y_1sv_1-y_2v_2\\ &= -s+1+(s\gamma-\alpha_2)x_1^2-\alpha_2x_2^2-s\gamma x_1\cdot x_2+2\alpha_2x_1\cdot x_2+y_1sv_1-y_2v_2\\ &= -\alpha_2(x_1^2+x_2^2-2x_1\cdot x_2)-s+1+s\gamma x_1^2-s\gamma x_1\cdot x_2+y_1\underbrace{s}_{y_1y_2}v_1-y_2v_2\\ &= -\alpha_2(x_1-x_2)^2-s+1+s\gamma x_1^2-s\gamma x_1\cdot x_2+y_2v_1-y_2v_2=0.\end{aligned}$$

Întrucât $(x_1-x_2)^2>0$ pentru orice $x_1\neq x_2$, rezultă că semnele acestei derivate corespund existenţei maximului pentru $L_D(\alpha_2)$. Aşadar, soluţia este:

$$\begin{aligned}\alpha_2^{\textbf{\textit{new, unclipped}}} &= \frac{-s+1+s\gamma(x_1^2-x_1\cdot x_2)+y_2(v_1-v_2)}{\|x_1-x_2\|^2}\\ &= \frac{y_2(-y_1+y_2+y_1\gamma(x_1^2-x_1\cdot x_2)+v_1-v_2)}{\|x_1-x_2\|^2}\\ &\overset{(330)}{=} \frac{y_2(f(x_1)-y_1-f(x_2)+y_2)+\alpha_2\|x_1-x_2\|^2}{\|x_1-x_2\|^2}\\ &= \alpha_2+\frac{y_2(E_1-E_2)}{\|x_1-x_2\|^2}=\alpha_2+\frac{y_2(E_2-E_1)}{-\|x_1-x_2\|^2}.\end{aligned}\tag{329}$$

***Observaţii*:**
1. Atât în relaţia (329) şi pe linia care o succedă, α_2 trebuie înţeles ca fiind $\alpha_2^{\textbf{\textit{old}}}$.
2. Conform relaţiei (326), valoarea expresiei v_2-v_1 nu depinde de α_1 sau de α_2. Dacă în scrierea lui $w=\sum_j y_j\alpha_j x_j$ vom considera că α_1 şi α_2 apar în accepţiunea $\alpha_1^{\textbf{\textit{old}}}$ şi respectiv $\alpha_2^{\textbf{\textit{old}}}$, nimic nu ne împiedică să facem raţionamentul următor, obţinând în final relaţia (330). De fapt, relaţia (330) şi a fost invocată mai sus la deducerea relaţiei (329).

$$\begin{aligned}v_1-v_2 &\overset{(327)}{=} f(x_1)-y_1\alpha_1x_1^2-y_2\alpha_2x_1\cdot x_2-(f(x_2)-y_1\alpha_1x_1\cdot x_2-y_2\alpha_2x_2^2)\\ &= f(x_1)-y_1\underbrace{\alpha_1}_{\gamma-s\alpha_2}x_1^2-y_2\alpha_2x_1\cdot x_2-f(x_2)+y_1\underbrace{\alpha_1}_{\gamma-s\alpha_2}x_1\cdot x_2+y_2\alpha_2x_2^2\\ &= f(x_1)-f(x_2)-y_1\gamma x_1^2+\underbrace{sy_1}_{y_2}\alpha_2x_1^2-y_2\alpha_2x_1\cdot x_2+y_1\gamma x_1\cdot x_2-\underbrace{sy_1}_{y_2}\alpha_2x_1\cdot x_2+\\ &\quad y_2\alpha_2x_2^2\\ &= f(x_1)-f(x_2)-y_1\gamma x_1^2+y_2\alpha_2x_1^2-y_2\alpha_2x_1\cdot x_2+y_1\gamma x_1\cdot x_2-y_2\alpha_2x_1\cdot x_2+y_2\alpha_2x_2^2\\ &= f(x_1)-f(x_2)-y_1\gamma(x_1^2-x_1\cdot x_2)+y_2\alpha_2(x_1^2-2x_1\cdot x_2+x_2^2)\\ &= f(x_1)-f(x_2)-y_1\gamma(x_1^2-x_1\cdot x_2)+y_2\alpha_2\|x_1-x_2\|^2.\end{aligned}\tag{330}$$

În final, pentru a deduce $\alpha_1^{\textbf{\textit{new, unclipped}}}$, vom folosi egalitatea dublă

$$\alpha_1^{\textbf{\textit{old}}}+s\alpha_2^{\textbf{\textit{old}}}=\gamma=\alpha_1^{\textbf{\textit{new, unclipped}}}+s\alpha_2^{\textbf{\textit{new, unclipped}}}.$$

Aşadar,

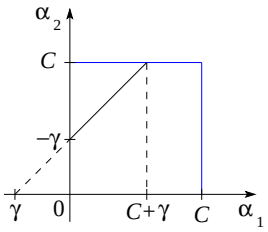
$$\begin{aligned}\alpha_1^{\textbf{\textit{new, unclipped}}} &= \gamma-s\alpha_2^{\textbf{\textit{new, unclipped}}}=\alpha_1^{\textbf{\textit{old}}}+s\alpha_2^{\textbf{\textit{old}}}-s\alpha_2^{\textbf{\textit{new, unclipped}}}\\ &= \alpha_1^{\textbf{\textit{old}}}+y_1y_2(\alpha_2^{\textbf{\textit{old}}}-\alpha_2^{\textbf{\textit{new, unclipped}}}).\end{aligned}$$

***Observaţie*: Egalitatea aceasta corespunde cu ultima relaţie care a fost inclusă [în vederea rezolvării] în enunţul de la punctul b.**

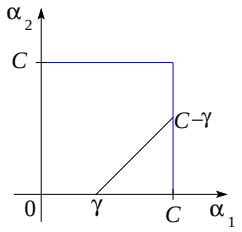
b. Vom analiza pe rând cele două cazuri: $y_1 \neq y_2$ (care implică $\gamma = \alpha_1 - \alpha_2$, conform relaţiei (328)) şi respectiv $y_1 = y_2$ (care implică $\gamma = \alpha_1 + \alpha_2$). Pentru fiecare din aceste două cazuri, în funcţie de semnul lui $\gamma \overset{not.}{=} \alpha_1 + s\alpha_2$ şi respectiv în funcţie de semnul lui $\gamma - C$, vom avea câte două subcazuri, reprezentate grafic în partea dreaptă.

$y_1 \neq y_2 \Rightarrow \alpha_1 - \alpha_2 = \gamma$
$\Rightarrow \alpha_2 = \alpha_1 - \gamma$

$-C \leq \gamma < 0$:

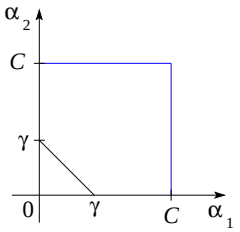


$0 \leq \gamma \leq C$:

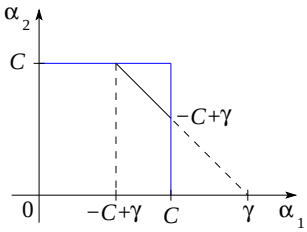


$y_1 = y_2 \Rightarrow \alpha_1 + \alpha_2 = \gamma$
$\Rightarrow \alpha_2 = -\alpha_1 + \gamma$

$0 \leq \gamma < C$:



$C \leq \gamma \leq 2C$:



Pentru cazul $y_1 \neq y_2$, se observă din cele două grafice că $\alpha_2 \geq -\gamma$ şi respectiv $\alpha_2 \geq 0$, deci $\alpha_2 \geq L = \max(-\gamma, 0)$. De asemenea, $\alpha_2 \leq C$ şi respectiv $\alpha_2 \leq C - \gamma$, de unde rezultă $H = \min(C, C - \gamma)$. Similar, pentru cazul $y_1 = y_2$ vom obţine $L = \max(0, -C + \gamma)$ şi $H = \min(\gamma, C)$.

În fine, regula de actualizare (325) decurge în mod natural din proprietăţile funcţiei de gradul al doilea aplicate pentru $L_D(\alpha_2)$, care trebuie maximizată ţinând cont de restricţiile $L \leq \alpha_2 \leq H$.[694]

c. Vom folosi egalitatea $y_i E_i = y_i(f(x_i) - y_i) = y_i f(x_i) - 1$. Conform relaţiilor (321), cazurile $\alpha_i < C$ şi $\alpha_i > 0$ vor avea câte două subcazuri:

$$\alpha_i < C: \left\{ \begin{array}{lll} \alpha_i = 0 & \Rightarrow y_i \underbrace{(w \cdot x_i + w_0)}_{f(x_i)} \geq 1 & \Rightarrow y_i E_i \geq 0 \\ \alpha_i \in (0, C) & \Rightarrow y_i f(x_i) = 1 & \Rightarrow y_i E_i = 0 \end{array} \right\} \Rightarrow y_i E_i \geq 0 \quad (331)$$

$$\alpha_i > 0: \left\{ \begin{array}{lll} \alpha_i \in (0, C) & \Rightarrow y_i f(x_i) = 1 & \Rightarrow y_i E_i = 0 \\ \alpha_i = C & \Rightarrow y_i f(x_i) \leq 1 & \Rightarrow y_i E_i \leq 0 \end{array} \right\} \Rightarrow y_i E_i \leq 0. \quad (332)$$

Notând premisele şi respectiv consecinţele relaţiilor (331) şi (332) cu nişte variabile din logica propoziţiilor, şi anume $p_1 \overset{not.}{=} (\alpha_i < C)$, $q_1 \overset{not.}{=} (y_i E_i \geq 0)$,

[694] Pentru a vă convinge, vă recomandăm să faceţi câte un grafic *generic* pentru funcţia $L_d(\alpha_2)$ pentru fiecare din cele trei cazuri ale regulii (325). (Pentru două exemple *concrete*, puteţi vedea graficele de la rezolvarea problemei 23.b.)

$p_2 \overset{not.}{=} (\alpha_i > 0)$, şi $q_2 \overset{not.}{=} (y_i E_i \leq 0)$, vom putea scrie următoarele echivalenţe, folosind mai întâi legea dublei negaţii şi apoi legile lui De Morgan:

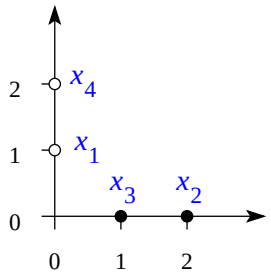
$$(p_1 \to q_1) \wedge (p_2 \to q_2) \equiv \neg\neg[(p_1 \to q_1) \wedge (p_2 \to q_2)] \equiv \neg[\neg(p_1 \to q_1) \vee \neg(p_2 \to q_2)] \equiv$$
$$\neg[\neg(\neg p_1 \vee q_1) \vee \neg(\neg p_2 \vee q_2)] \equiv \neg[(p_1 \wedge \neg q_1) \vee (p_2 \wedge \neg q_2)].$$

Expresia finală — $\neg[(p_1 \wedge \neg q_1) \vee (p_2 \wedge \neg q_2)]$ — coincide cu *condiţia de oprire* din enunţ, doar că $\neg q_1$ şi $\neg q_2$ au fost rescrise (din motive de „stabilitate" numerică la efectuarea calculelor) sub forma $y_i E_i < -tol$, şi respectiv $y_i E_i > tol$, unde *tol* este constanta pozitivă considerată în enunţ. Atunci când această condiţie este satisfăcută pentru $i = 1, \ldots, m$, algoritmul SMO poate fi oprit, întrucât [se consideră că] sunt satisfăcute condiţiile de tip Karush-Kuhn-Tucker (321).

23. **(C-SVM: exemplu de aplicare a algoritmului SMO)**

■ • ○ ***CMU, 2008 fall, Eric Xing, HW2, pr. 1.3***

Se dau patru instanţe în spaţiul euclidian bidimensional, împreună cu etichetele asignate lor: $x_1 = (0, 1)$, $y_1 = -1$; $x_2 = (2, 0)$, $y_2 = +1$; $x_3 = (1, 0)$, $y_3 = +1$ şi $x_4 = (0, 2)$, $y_4 = -1$. Vom folosi aceste exemple pentru a antrena un C-SVM (SVM liniar cu margine "soft" şi parametru de „destindere" C). Fie α_1, α_2, α_3 şi α_4 multiplicatorii Lagrange asociaţi instanţelor x_1, x_2, x_3 şi respectiv x_4. Fixăm valoarea parametrului C la 100.



a. Scrieţi forma duală a problemei de optimizare C-SVM în acest caz.

b. Vă cerem să executaţi două iteraţii ale algoritmului SMO (Sequential Minimal Optimization) pe acest set de date.[695] **Presupunem că facem iniţializările $\alpha_1 = 5$, $\alpha_2 = 4$, $\alpha_3 = 8$, $\alpha_4 = 7$. Veţi proceda astfel:**

i. **La prima iteraţie veţi actualiza multiplicatorii α_1 şi α_4 (păstrând α_2 şi α_3 fixaţi). Stabiliţi *relaţiile de actualizare* a valorilor pentru α_1 şi α_4 în funcţie de valorile multiplicatorilor α_2 şi α_3. Ce valori vor avea α_1 şi α_4 după actualizare?**

ii. **La a doua iteraţie veţi fixa α_1 şi α_4 şi veţi stabili relaţiile de actualizare pentru α_2 and α_3 în funcţie de α_1 şi α_4. Ce valori vor avea α_2 şi α_3 după actualizare?**

Răspuns:

a. Particularizând forma duală a problemei de optimizare C-SVM (vedeţi problema 12) pentru acest set de date, după efectuarea tuturor produselor scalare $x_i \cdot x_j$ vom obţine:[696]

$$\max_{\alpha_1,\alpha_2,\alpha_3,\alpha_4} \left[\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}(\alpha_1^2 + 4\alpha_2^2 + \alpha_3^2 + 4\alpha_4^2 + 4\alpha_1\alpha_4 + 4\alpha_2\alpha_3)\right]$$

a. î. $0 \leq \alpha_i \leq 100$, **pentru** $i = 1, \ldots, 4$
$-\alpha_1 + \alpha_2 + \alpha_3 - \alpha_4 = 0$

[695] Vedeţi problema 22 din prezentul capitol.
[696] Concret, $x_1^2 = x_3^2 = 1$, $x_2^2 = x_4^2 = 4$, $x_1 \cdot x_4 = x_2 \cdot x_3 = 2$, iar restul produselor $x_i \cdot x_j$ cu $i \neq j$ au valoarea 0.

b. Facem *observaţia* că, la acest punct, exerciţiul cere să se execute două iteraţii ale algoritmului SMO pe datele furnizate, fără a aplica criteriul de selecţie a variabilelor libere şi condiţiile de oprire formulate de John Platt, autorul acestui algoritm.

Vom da mai jos două rezolvări. Prima va fi mai simplă. Ea va urma *ideea* algoritmului SMO — care aplică o metodă de optimizare numită *creştere pe coordonate* (engl., coordinate ascent) — fără a recurge efectiv la formulele stabilite de John Platt. În schimb, vom proceda direct la optimizarea funcţiilor obiectiv determinate de alegerea (impusă, conform enunţului) a celor două perechi de variabile duale specificate. La a doua rezolvare, vom aplica direct formulele generale pentru „actualizarea" valorilor libere din algoritmul SMO. Facem *observaţia* că şi această rezolvare va fi utilă cititorului, pentru că vom scoate în evidenţă anumite detalii / modalităţi de calcul care nu sunt chiar „imediate" pentru cineva care nu este încă obişnuit cu algoritmul SMO.

***Prima soluţie*:**

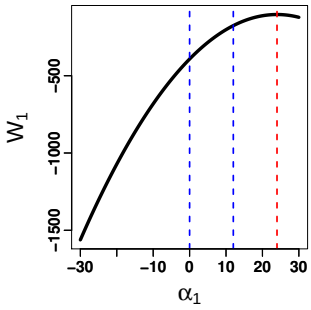
***i.* La prima iteraţie, iniţial avem $\alpha_1 = 5$, $\alpha_2 = 4$, $\alpha_3 = 8$, $\alpha_4 = 7$, iar apoi se lasă „libere" variabilele α_1 şi α_4. Datorită restricţiei $\sum_{i=1}^{4} y_i\alpha_i = 0$ din forma duală a problemei de optimizare C-SVM (a se vedea (D′) la problema 12), vom avea următoarea relaţie de legătură dintre valorile variabilelor libere: $\alpha_1^{new} + \alpha_4^{new} = \alpha_2 + \alpha_3 = 12$.**

Din restricţia $\alpha_i \geq 0$ — dar şi datorită faptului că y_1 şi y_4 au acelaşi semn — va rezulta că valorile posibile („fezabile") pentru α_1 şi α_4 vor fi limitate la intervalul $[0, 12]$, inclus în intervalul $[0, C] = [0, 100]$.

Înlocuind $\alpha_2 = 4$, $\alpha_3 = 8$ şi $\alpha_4 = 12 - \alpha_1$ în funcţia obiectiv a problemei de optimizare de la punctul a, vom obţine expresia funcţiei pe care va trebui să o maximizăm la această iteraţie:

$$\begin{aligned} W_1(\alpha_1) &\overset{not.}{=} 24 - \frac{1}{2}(\alpha_1^2 + 4 \cdot 4^2 + 8^2 + 4(12 - \alpha_1)^2 + 4\alpha_1(12 - \alpha_1) + 4 \cdot 4 \cdot 8) \\ &= 24 - \frac{1}{2}(\alpha_1^2 - 48\alpha_1 + 832). \end{aligned}$$

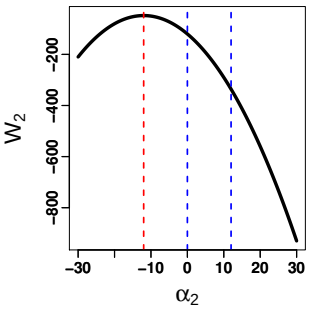
Valoarea lui α_1 pentru care se atinge optimul acestei funcţii este notată cu $\alpha_1^{new,\ unclipped}$ şi, evident, este $\frac{48}{2} = 24$. Această valoare se află în afara intervalului $[0, 12]$. Este imediat că maximul funcţiei $W_1(\alpha_1)$ pe intervalul $[0, 12]$ se atinge în punctul $\alpha_1^{new,\ clipped} = 12$. În consecinţă, α_4 va primi valoarea $\alpha_4^{\boldsymbol{new,\ clipped}} = 12 - \alpha_1^{new,\ clipped} = 0$.



***ii.* La a doua iteraţie, vom avea $\alpha_1 = 12$ şi $\alpha_4 = 0$ fixaţi, iar $\alpha_2 = 4$ şi $\alpha_3 = 8$ liberi. Din relaţia $\sum_{i=1}^{4} y_i\alpha_i = 0$ rezultă $\alpha_2^{new} + \alpha_3^{new} = \alpha_1 + \alpha_4 = 12$. Ca şi mai sus, intervalul în care vor fi permise noile valori ale variabilelor α_2 şi α_3 este $[0, 12]$. Funcţia obiectiv din problema de optimizare convexă devine:**

$$\begin{aligned} W_2(\alpha_2) &\overset{not.}{=} 24 - \frac{1}{2}(12^2 + 4\alpha_2^2 + (12 - \alpha_2)^2 + 4 \cdot 0^2 + 4 \cdot 12 \cdot 0 + 4\alpha_2(12 - \alpha_2)) \\ &= -\frac{1}{2}\alpha_2^2 - 12\alpha_2 - 120 = -\frac{1}{2}(\alpha_2^2 + 24\alpha_2 + 240) \end{aligned}$$

Maximul global al acestei funcţii se atinge în punctul $\alpha_2^{new,\ unclipped} = -\frac{24}{2} = -12$**. Acest punct se situează în exteriorul intervalului de fezabilitate** $[0, 12]$**. Maximul funcţiei** $W_2(\alpha_2)$ **pe intervalul** $[0, 12]$ **se atinge în punctul** $\alpha_2^{new,\ clipped} = 0$**. În consecinţă,** $\alpha_3^{\boldsymbol{new,\ clipped}} = 12 - \alpha_2^{new,\ clipped} = 12$**.**



***A doua soluţie*:**

Formulele date de John Platt pentru actualizarea variabilei libere α_i **(la o iteraţie oarecare a algoritmului SMO) sunt:**[697]

$$\alpha_i^{new,\ unclipped} = \alpha_i + \frac{y_i(E_i - E_j)}{\eta}$$

$$\alpha_i^{new,\ clipped} = \begin{cases} H & \textbf{dacă } \alpha_i^{new,\ unclipped} > H \\ \alpha_i^{new,\ unclipped} & \textbf{dacă } L \leq \alpha_i^{new,\ unclipped} \leq H \\ L & \textbf{dacă } \alpha_i^{new,\ unclipped} < L, \end{cases}$$

unde

$$w = \sum_{i=1}^{4} y_i \alpha_i x_i$$
$$E_k = w \cdot x_k + w_0 - y_k$$
$$\eta = - \parallel x_i - x_j \parallel^2$$
$$L = \max(0, \alpha_i - \alpha_j) \textbf{ şi } H = \min(C, C + \alpha_i - \alpha_j) \textbf{ dacă } y_i \neq y_j$$
$$L = \max(0, \alpha_i + \alpha_j - C) \textbf{ şi } H = \min(C, \alpha_i + \alpha_j) \textbf{ dacă } y_i = y_j.$$

În consecinţă, vom avea:

i. **La prima iteraţie,** $\alpha_1 = 5$**,** $\alpha_2 = 4$**,** $\alpha_3 = 8$**,** $\alpha_4 = 7$**, iar** $\eta = - \parallel x_1 - x_4 \parallel^2 = -1$**. Fără a face deocamdată calculele, explicităm erorile** $E_1 = f(x_1) - y_1 = w \cdot x_1 + w_0 - y_1$ **şi** $E_4 = f(x_4) - y_4 = w \cdot x_4 + w_0 - y_4$**, deci rezultă** $E_1 - E_4 = w \cdot (x_1 - x_4)$**. Din relaţia** $w = \sum_{i=1}^{4} y_i \alpha_i x_i$**, urmează:**

$$\begin{aligned} w &= -\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 - \alpha_4 x_4 \\ &= -5 \begin{bmatrix} 0 \\ 1 \end{bmatrix} + 4 \begin{bmatrix} 2 \\ 0 \end{bmatrix} + 8 \begin{bmatrix} 1 \\ 0 \end{bmatrix} - 7 \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 16 \\ -19 \end{bmatrix}, \end{aligned}$$

de unde rezultă că $E_1 - E_4 = w \cdot (x_1 - x_4) = (16, -19) \cdot (0, -1) = 19$ **şi vom putea calcula** $\alpha_1^{new,\ unclipped} = \alpha_1 + \frac{y_1(E_1 - E_4)}{\eta} = 5 + 19 = 24$**.**

Acum verificăm dacă $\alpha_1^{new,\ unclipped}$ **este în intervalul de „fezabilitate“: deoarece** $y_1 = y_4$**, vom avea** $L = \max(0, \alpha_1 + \alpha_4 - 100) = 0$**, pentru că** $\alpha_1 + \alpha_4 = 12$**. Similar,** $H = \min(100, 12) = 12$**.**

[697] În comparaţie cu problema 22, aici nu vom mai scrie α_i^{old}, ci direct α_i, fiindcă se poate subînţelege fără nicio dificultate, datorită contextului mult mai simplu.

Întrucât $\alpha_1^{new,\ unclipped} > H = 12$**, vom avea** $\alpha_1^{new,\ clipped} = H = 12$ **şi, în consecinţă,** $\alpha_4^{new,\ clipped} = 12 - \alpha_1^{new,\ clipped} = 0$**.**

ii. **La a doua iteraţie,** $\alpha_1 = 12$**,** $\alpha_2 = 4$**,** $\alpha_3 = 8$**,** $\alpha_4 = 0$ **şi** $\alpha_2^{new} + \alpha_3^{new} = \alpha_2 + \alpha_3 = \alpha_1 + \alpha_4 = 12$**. De asemenea,** $\eta = -\parallel x_2 - x_3 \parallel^2 = -1$ **şi** $E_2 = f(x_2) - y_2 = w \cdot x_2 + w_0 - y_2$**, iar** $E_3 = f(x_3) - y_3 = w \cdot x_3 + w_0 - y_3$**, deci** $E_2 - E_3 = w \cdot (x_2 - x_3)$**. Calculăm** w **astfel:**

$$\begin{aligned} w &= -\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 - \alpha_4 x_4 \\ &= -12 \begin{bmatrix} 0 \\ 1 \end{bmatrix} + 4 \begin{bmatrix} 2 \\ 0 \end{bmatrix} + 8 \begin{bmatrix} 1 \\ 0 \end{bmatrix} - 0 \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 16 \\ -12 \end{bmatrix}, \end{aligned}$$

deci $E_2 - E_3 = (16, -12) \cdot (1, 0) = 16$**. Aşadar,** $\alpha_2^{new,\ unclipped} = \alpha_2 + \dfrac{y_2(E_2 - E_3)}{\eta} = 4 - 16 = -12$**. Deoarece** $y_2 = y_3$**, vom avea** $L = \max(0, \alpha_2 + \alpha_3 - C) = 0$ **şi** $H = \min(C, \alpha_2 + \alpha_3) = 12$**. În consecinţă,**

$$\alpha_2^{new,\ unclipped} = -12 < L = 0,$$

deci în final vom avea $\alpha_2^{new,\ clipped} = L = 0$ **şi** $\alpha_3^{new,\ clipped} = 12 - \alpha_2^{new,\ clipped} = 12$**. Se constată imediat că am regăsit rezultatele de la prima soluţie.**

Observaţie**: Calculând valoarea funcţiei obiectiv** $W(\alpha)$**, folosind mai întâi valorile iniţiale ale parametrilor** α_i**, apoi valorile rezultate la fiecare din cele două iteraţii, obţinem valorile:** -268.5**,** -258**,** -129**. Aşa cum era de aşteptat, aceste numere sunt în ordine crescătoare. Dacă am fi calculat [şi valoarea parametrului** w_0 **care apare în forma primală a problemei de optimizare date, precum] şi valorile variabilelor de „destindere“** ξ_i**, am fi putut calcula şi valorile funcţiei obiectiv** $\dfrac{1}{2} \parallel w \parallel^2 + C \sum_i \xi_i$**. Aceste valori trebuie să fie în ordine descrescătoare şi mai mari decât valorile determinate pentru funcţia** $W(\alpha)$ **mai sus (conform relaţiei de *dualitate slabă*** $p^* \geq d^*$**; vedeţi relaţia** (311)**).**

24. **(SVM şi C-SVM, în forma primală sau forma duală: alegerea funcţiei-nucleu şi a valorii parametrului** C**)**

CMU, 2010 fall, Aarti Singh, HW3, pr. 3

Figurile date mai jos indică suprafeţele de decizie obţinute de clasificatorul SVM pe un acelaşi set de date de antrenament fie în varianta marginii "hard" (folosind diferite funcţii-nucleu), fie în varianta marginii "soft" (cu diferite valori pentru parametrul C**). Exemplele pozitive sunt reprezentate prin simbolul** $\circ$**, iar cele negative prin simbolul** $\bullet$**. Acelea care sunt încercuite desemnează vectori-suport.**

Indicaţi care dintre cele 6 figuri a fost generată de următoarele probleme de optimizare de tip SVM sau C-SVM (atenţie, sunt 6 figuri şi doar 5 probleme, deci o figură nu corespunde niciunei probleme):

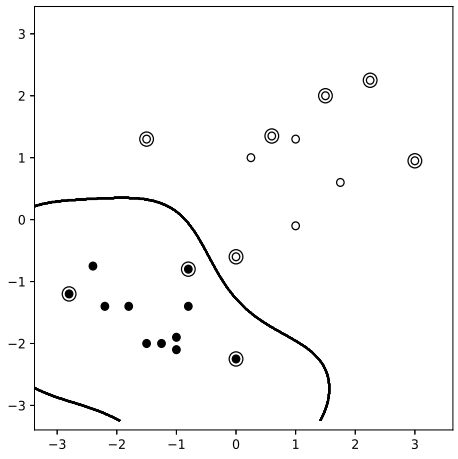
a. $\min\big(\frac{1}{2}w\cdot w + C\sum_{i=1}^{n}\xi_i\big)$
a.î. **pentru** $\forall i = 1,\cdots,n$**:**
$\xi_i \geq 0$
$(w\cdot x_i + w_0)y_i \geq 1-\xi_i$
şi $C = 0.1$**.**

b. $\min\big(\frac{1}{2}w\cdot w + C\sum_{i=1}^{n}\xi_i\big)$
a.î. **pentru** $\forall i = 1,\cdots,n$**:**
$\xi_i \geq 0$
$(w\cdot x_i + w_0)y_i \geq 1-\xi_i$
şi $C = 1$**.**

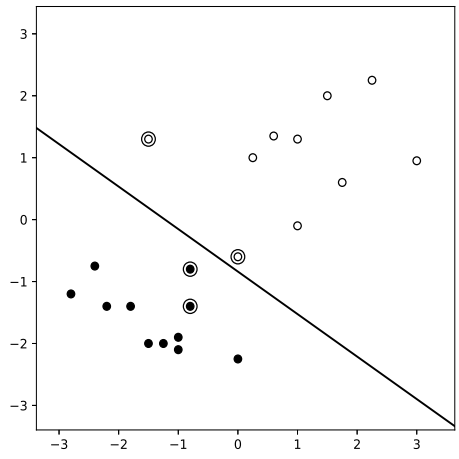
c. $\max\big(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j K(x_i,x_j)\big)$
a.î. $\sum_{i=1}^{n}\alpha_i y_i = 0$**;**
$\alpha_i \geq 0, \forall i = 1,\cdots,n$**;**
unde $K(u,v) = u\cdot v + (u\cdot v)^2$**.**

d. $\max\big(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j K(x_i,x_j)\big)$
a.î. $\sum_{i=1}^{n}\alpha_i y_i = 0$**;**
$\alpha_i \geq 0, \forall i = 1,\cdots,n$**;**
unde $K(u,v) = \exp\Big(-\frac{\|u-v\|^2}{2}\Big)$**.**

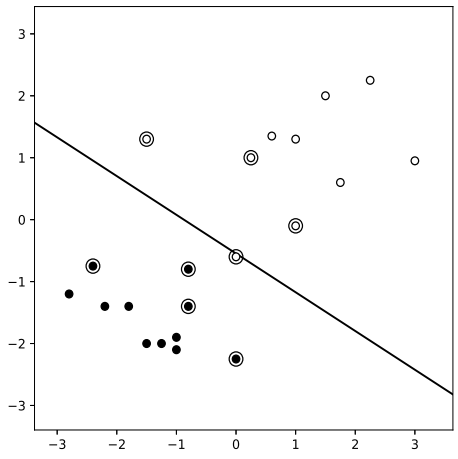
e. $\max\big(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j K(x_i,x_j)\big)$
a.î. $\sum_{i=1}^{n}\alpha_i y_i = 0$**;**
$\alpha_i \geq 0, \forall i = 1,\cdots,n$**;**
unde $K(u,v) = \exp(-\|u-v\|^2)$**.**



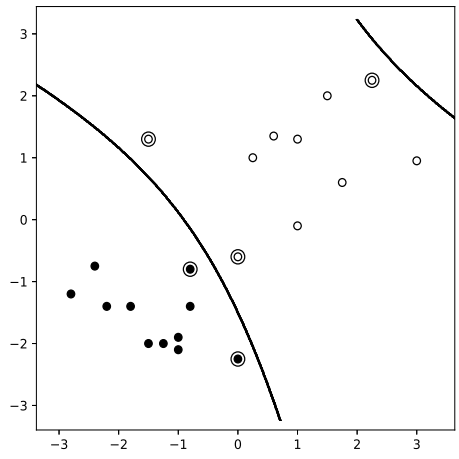
A.



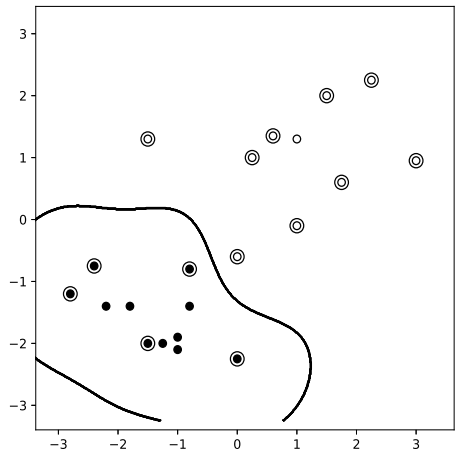
B.



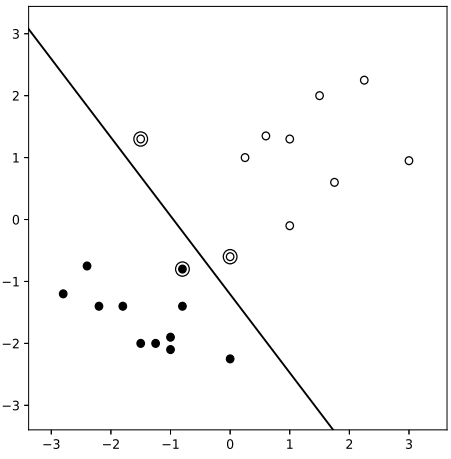
C.



D.



E.



F.

Răspuns:

Mai întâi vom „indica“ în dreptul fiecăreia dintre cele cinci SVM-uri date (a-e) care este tipul maşinii respective (adică, SVM cu margine “hard”, respectiv SVM cu margine “soft”, deci, mai precis, C-SVM), iar în primul caz ce nucleu (liniar ori neliniar, de un anumit tip) îi corespunde.

a. C-SVM liniar (i.e., fără funcţie nucleu), cu erori mari;

b. C-SVM liniar (i.e., fără funcţie nucleu), cu erori [mai] mici [decât în cazul a];

c. SVM cu margine “hard” (forma duală) cu funcţie nucleu pătratică;

d. SVM cu margine “hard” (forma duală) cu nucleu RBF (Radial Basis Function) de parametru $\sigma^2 = 1$;

e. SVM cu margine “hard” (forma duală) cu nucleu RBF de parametru $\sigma^2 = \frac{1}{2}$.

Evident, este mai uşor de tratat cazul marginii “hard”. Avem trei astfel de probleme / SVM-uri: c, d şi e. Dintre acestea, prima (c) lucrează cu nucleu pătratic, iar celelalte două cu nucleu de tip RBF. Din grafice se observă că doar separatorul din figura D are alură / formă parabolică. Aşadar, graficul D corespunde problemei c. Dintre celelalte două grafice cu separatori neliniari (A şi E), cel care se „mulează“ mai mult pe datele de antrenament este graficul E, deci lui îi va corespunde funcţia RBF cu varianţa mai mică, $\sigma^2 = \frac{1}{2}$. Aşadar, maşinii e îi corespunde graficul E, iar maşinii d îi corespunde graficul A.

În sfârşit, pentru maşinile a şi b — ambele folosind separator liniar, fără funcţie nucleu —, cea care are valoarea parametrului C mai mică (0.1) permite erori mai multe / mari în raport cu marginea. Între cele trei grafice cu separator liniar (B, C şi F), se observă că B şi F nu au erori la antrenare.[698] Dintre acestea două, se observă că în graficul F există cei mai puţini (doar trei) vectori-suport şi, aparent, eroarea totală în raport cu marginea, $\sum_i \xi_i$ este 0. Aşadar, lui F îi corespunde $C = \infty$, pentru care nu avem corespondent (apropiat) între cele cinci maşini din enunţ. Relativ la cele două grafice rămase

[698] Atenţie: noţiunea de *eroare în raport cu marginea* diferă de noţiunea de *eroare la clasificare*. Instanţa x_i constituie eroare în raport cu marginea dacă $\xi_i > 0$.

în discuţie (B şi C), se observă că marginea — şi, la fel, suma totală a erorilor în raport cu marginea — este mai mare în cazul lui C. Aşadar, graficului C îi corespunde cea mai mică dintre valorile parametrului C rămase (0.1), deci maşina a, iar graficului B îi corespunde valoarea $C = 1$, deci maşina b.

25. **(C-SVM cu funcţie-nucleu: două condiţii asupra parametrului C şi respectiv asupra funcţiei-nucleu, suficiente ca toate instanţele de antrenament să fie vectori-suport)**

prelucrare de Liviu Ciortuz, după
• ◦ MIT, 2008 fall, Tommi Jaakkola, midterm exam, pr. 1.3
CMU, 2010 fall, Aarti Singh, HW3, pr. 3.3.b

Considerăm că antrenăm o maşină cu vectori-suport cu variabile de „destindere" (engl., slack variables), care nu foloseşte variabila liberă w_0 (engl., bias variable). Se utilizează un nucleu $K(x, z)$ care are proprietatea $|K(x_i, x_j)| < 1$ dacă x_i şi x_j (din setul de antrenament) sunt în relaţia $x_i \neq x_j$ şi $|K(x_i, x_i)| \leq 1$ în caz contrar.[699] Sunt în total $m \geq 2$ puncte în setul de date de antrenament, dintre care cel puţin două sunt distincte.

Arătaţi că alegând pentru parametrul de „destindere" C o valoare astfel încât $C < \dfrac{1}{m-1}$, toate variabilele α_i din soluţia problemei duale vor fi nenule. (Aşadar, toate instanţele de antrenament devin vectori-suport.)

Răspuns:

Fie $i \in \{1, \ldots, m\}$, fixat. Prin definiţie, instanţa de antrenament x_i este vector-suport dacă $\alpha_i > 0$.

Dacă prin reducere la absurd am avea un $\alpha_i = 0$, ar rezulta

$$\begin{aligned} y_i w \cdot \Phi(x_i) &= y_i \Big(\sum_{j=1}^{m} \alpha_j y_j \Phi(x_j) \Big) \cdot \Phi(x_i) = y_i \Big(\sum_{j=1}^{m} \alpha_j y_j K(x_j, x_i) \Big) \\ &= y_i \Big(\sum_{j \neq i}^{m} \alpha_j y_j K(x_j, x_i) \Big) \leq \sum_{j \neq i}^{m} \alpha_j |K(x_j, x_i)| \\ &< \sum_{j \neq i}^{m} \alpha_j \leq (m-1)C. \end{aligned}$$

Ultima inegalitate are loc fiindcă $\alpha_j \leq C$ pentru orice j, conform demonstraţiei de la problema 12.d.

Întrucât $C < 1/(m-1)$, va rezulta $y_i w \cdot \Phi(x_i) < 1$.

Însă ştim că din *condiţiile de complementaritate duală Karush-Kuhn-Tucker* rezultă cu necesitate că dacă $\alpha_i = 0$ atunci $y_i w \cdot \Phi(x_i) \geq 1$, pentru $i = 1, \ldots, m$. (A se vedea problema 12, *observaţia* de la rezolvarea punctului e.)

Am obţinut deci o contradicţie!

[699] Nucleul RBF îndeplineşte aceste condiţii.

Prin urmare, singura posibilitate este ca α_i să fie nenul, ceea ce înseamnă că x_i este vector-suport.

***Consecinţă*: În cazul unei C-SVM cu nucleu RBF, dacă folosim $C < \frac{1}{m-1}$, va rezulta că toate instanţele sunt vectori-suport.**

26. **(Condiţii suficiente pentru ca o SVM cu nucleu RBF să producă eroare la antrenare 0)**

■ ● ○ ***Stanford, 2007 fall, Andrew Ng, HW2, pr. 3***

Considerăm o maşină cu vectori-suport care foloseşte nucleul de tip gaussian (RBF) $K(x,z) = \exp(-\parallel x - z \parallel^2 / \tau^2)$, unde am notat cu $\exp(\,)$ funcţia exponenţială. Parametrul τ determină mărimea deschiderii „clopotului" gaussian.

La punctele *a* şi *b* de mai jos vă vom ghida pas cu pas ca să demonstraţi următoarea *proprietate*: în ipoteza că setul de date de antrenament este consistent etichetat, se poate fixa o valoare a parametrului τ astfel încât eroarea la antrenare produsă de această SVM să fie zero.[700] La punctul *c* se va arăta că această proprietate *nu* este valabilă şi în cazul folosirii clasificatorului C-SVM.

a. Presupunem că setul de date de antrenament $\{(x_1, y_1), \ldots, (x_m, y_m)\}$ este alcătuit din puncte care sunt separate unele de altele de o distanţă de cel puţin ε, adică $\parallel x_j - x_i \parallel \geq \varepsilon$ pentru orice $i \neq j$.[701]

Vă readucem aminte că funcţia de decizie învăţată de către SVM cu funcţie-nucleu K se poate scrie sub forma:[702]

$$f(x) = \sum_{i=1}^{m} \alpha_i y_i K(x_i, x) + w_0, \tag{333}$$

unde $\alpha_1, \ldots, \alpha_m \in \mathbb{R}_+$, w_0 este termenul liber / bias-ul din forma primală a problemei de optimizare [C-]SVM, iar $K(x_i, x) \overset{def.}{=} \Phi(x_i) \cdot \Phi(x)$, unde Φ este o „mapare" corespunzătoare funcţiei-nucleu K.

Găsiţi valori pentru $\alpha_1, \ldots, \alpha_m$, w_0, precum şi pentru parametrul gaussian τ, astfel încât toate punctele x_i să fie clasificate corect de către clasificatorul $sign(f(x))$.

***Sugestii*:**
1. Verificaţi faptul că, lucrând cu $y_i \in \{-1, +1\}$, predicţia făcută pentru x_i de

[700] Problema 29.A arată că separabilitatea aceasta se poate obţine pentru orice valoare a parametrului nucleului RBF, dacă în problema SVM se impune condiţia suplimentară ca toate instanţele de antrenament să fie vectori-suport, în sensul că ele sunt situate la distanţa $1/\|w\|$ de separatorul optimal.

[701] Evident, dacă $x_i \neq x_j$ pentru orice $i \neq j$ şi, bineînţeles, dacă m este finit, atunci putem lua $\varepsilon \overset{not.}{=} \min_{i \neq j} \parallel x_j - x_i \parallel$.

[702] Vedeţi relaţia (318) de la problema 9.

Observaţie importantă: Această formulă presupunea *acolo* că α_i sunt soluţiile problemei SVM în forma duală. Însă *aici* nu lucrăm cu această presupoziţie. Practic, aici vom alege α_i, w_0 şi τ astfel încât să rezulte că f (de această formă) determină *separabilitate liniară* în spaţiul de „trăsături" în care sunt $\Phi(x_1), \ldots, \Phi(x_m)$, deci şi *separabilitate* (simplă, deci în general neliniară) în spaţiul original (în care sunt instanţele $x_1, \ldots, x_m$). Ulterior (la punctul *b*), folosind acelaşi f, vom arăta că problema SVM admite cel puţin o soluţie strict „fezabilă" — deci, conform condiţiei lui Slater şi o *soluţie optimă* — pentru forma primală (şi, de fapt, şi pentru cea duală, dar asta pur şi simplu *nu* are relevanţă aici) şi, în consecinţă, soluţia optimă va satisface şi ea această proprietate de separabilitate, care ne asigură că eroarea la antrenare este 0.

către $sign(f(x))$ va fi corectă dacă $|f(x_i) - y_i| < 1$. Altfel spus, verificaţi că are loc implicaţia $|f(x_i) - y_i| < 1 \Rightarrow y_i f(x_i) > 0$.[703]
2. Fixând $\alpha_i = 1$ pentru $i = 1, \ldots, m$ şi $w_0 = 0$, găsiţi o valoare a lui τ pentru care inegalitatea $|f(x_i) - y_i| < 1$ să fie satisfăcută pentru $i = 1, \ldots, m$.

b. Presupunem că rulăm o SVM cu margine "hard",[704] folosind pentru parametrul τ valoarea pe care aţi găsit-o la punctul precedent. Va obţine oare acest clasificator (în mod necesar) eroare la antrenare zero? De ce da, sau de ce nu?

c. Presupunem că antrenăm un C-SVM (adică o SVM cu variabile de „destindere") pe datele specificate mai sus, folosind pentru parametrul τ valoarea pe care aţi ales-o la punctul a, iar pentru parametrul C o valoare fixată în mod arbitrar, dar pe care nu o cunoaştem dinainte. Va obţine oare acest clasificator (în mod necesar) eroare de antrenare zero? De ce da, sau de ce nu?

Răspuns:

a. Sunt imediate următoarele echivalenţe:

$$|f(x_i) - y_i| < 1 \Leftrightarrow -1 < f(x_i) - y_i < 1 \Leftrightarrow -1 + y_i < f(x_i) < 1 + y_i.$$

Pentru $y_i = -1$, partea dreaptă a ultimei inegalităţi duble de mai sus devine $f(x_i) < 0$. Pentru $y_i = 1$, partea stângă a aceleiaşi inegalităţi duble devine $f(x_i) > 0$. Aşadar, dacă inegalitatea $|f(x_i) - y_i| < 1$ este adevărată, atunci instanţa x_i este corect clasificată de către funcţia $sign(f(x))$.

Conform *sugestiei* din enunţ, vom considera $\alpha_i = 1$ pentru $i = 1, \ldots, m$ şi $w_0 = 0$. Pentru un exemplu de antrenament oarecare (x_i, y_i), vom avea:

$$\begin{aligned}
|f(x_i) - y_i| &= \Big|\sum_{j=1}^{m} y_j K(x_j, x_i) - y_i\Big| = \Big|\sum_{j=1}^{m} y_j \exp\left(-\parallel x_j - x_i \parallel^2 / \tau^2\right) - y_i\Big| \\
&= \Big|y_i + \sum_{j\neq i} y_j \exp\left(-\parallel x_j - x_i \parallel^2 / \tau^2\right) - y_i\Big| \\
&= \Big|\sum_{j\neq i} y_j \exp\left(-\parallel x_j - x_i \parallel^2 / \tau^2\right)\Big| \\
&\leq \sum_{j\neq i} \left|y_j \exp\left(-\parallel x_j - x_i \parallel^2 / \tau^2\right)\right| = \sum_{j\neq i} |y_j| \exp\left(-\parallel x_j - x_i \parallel^2 / \tau^2\right) \\
&= \sum_{j\neq i} \exp\left(-\parallel x_j - x_i \parallel^2 / \tau^2\right) \\
&\leq \sum_{j\neq i} \exp(-\varepsilon^2/\tau^2) = (m-1)\exp(-\varepsilon^2/\tau^2).
\end{aligned}$$

Prima dintre inegalităţile de mai sus este datorată aplicării repetate a inegalităţii triunghiului ($|a + b| \leq |a| + |b|$), iar a doua inegalitate decurge din presupunerea că $||x_j - x_i|| \geq \varepsilon$ pentru orice $i \neq j$. Aşadar, pentru a avea $|f(x_i) - y_i| < 1$ pentru $i = 1, \ldots, m$ este suficient să-l alegem pe τ astfel încât

[703] Proprietatea / implicaţia aceasta este adevărată nu doar în cazul funcţiilor de forma specificată în relaţia (333), ci pentru orice funcţie $f : \mathbb{R}^d \to \mathbb{R}$, unde $\mathbb{R}^d$ este spaţiul din care au fost selectate instanţele $x_1, \ldots, x_m$.
[704] Adică, fără variabile de „destindere" (engl., slack variables).

$$(m-1)\exp(-\varepsilon^2/\tau^2) < 1,$$

sau, echivalent,[705]

$$\tau < \frac{\varepsilon}{\sqrt{\ln(m-1)}}.$$

De exemplu, putem lua $\tau = \varepsilon/\sqrt{\ln m}$**.**

***Observaţie*: Rezumând, până acum am arătat că există o instanţiere pentru variabilele duale** ($\alpha_i = 1$) **şi pentru variabila primală** w_0 **(şi anume,** $w_0 = 0$**), pentru care funcţia** $f(x) = \sum_{i=1}^m \alpha_i y_i K(x_i, x) + w_0 = \left(\sum_{i=1}^m \alpha_i y_i \Phi(x_i)\right) \cdot \Phi(x) + w_0$ **separă perfect exemplele de antrenament date în psaţiul de „trăsături determinat de „maparea“** Φ**. Punctele** b **şi** c **de mai jos vor analiza dacă *soluţiile optime* produse de SVM şi respectiv C-SVM pe acelaşi set de antrenament vor avea şi ele această proprietate. Vă reamintim că soluţiile optime pentru problema de optimizare [C-]SVM există, atât pentru forma primală cât şi pentru forma duală — şi ele sunt în relaţia** $\bar{w} = \sum_{i=1}^m \bar{\alpha}_i y_i K(x_i, x)$ **— dacă, spre exemplu, este satisfăcută condiţia lui Slater (a se vedea *Comentariul* de la pr. 9).**

b. Acum vom arăta că putem pune în corespondenţă funcţia f **din proprietatea de separabilitate liniară obţinută în spaţiul de „trăsături“ (în care sunt** $\Phi(x_1), \ldots, \Phi(x_m)$**) cu o anumită pereche** w'**,** w'_0 **care satisface condiţia lui Slater în respectivul spaţiu de „trăsături“.**

Condiţia lui Slater, relativă la problema de optimizare SVM este următoarea: există o soluţie strict „fezabilă“, adică o asignare pentru w **şi** w_0**, astfel încât restricţiile din problema primală SVM sunt satisfăcute cu inegalitate strictă:** $y_i(w \cdot \Phi(x_i) + w_0) > 1$ **pentru** $i = 1, \ldots, m$**.**

Fie $i \in \{1, \ldots, m\}$**, fixat. Luând** $\alpha_1 = 1, \ldots, \alpha_m = 1$**,** $w_0 = 0$**,** $f(x) = \sum_{j=1}^m \alpha_j y_j K(x_j, x) + w_0$ **şi** τ **ca la punctul** a**, vom avea** $|f(x_i) - y_i| < 1$**, deci** $y_i f(x_i) > 0$**. Notăm** $w = \sum_{j=1}^m \alpha_j y_j \Phi(x_j)$**. Aşadar,**

$$\begin{aligned} y_i(w \cdot \Phi(x_i) + w_0) &= y_i w \cdot \Phi(x_i) = y_i \sum_{j=1}^m \alpha_j y_j \Phi(x_j) \cdot \Phi(x_i) = y_i \sum_{j=1}^m \alpha_j y_j K(x_j, x_i) \\ &= y_i f(x_i) > 0. \end{aligned}$$

În consecinţă,

$$y_i(w \cdot \Phi(x_i) + w_0) = y_i \sum_{j=1}^m \alpha_j y_j K(x_j, x_i) > 0 \text{ pentru } i = 1, \ldots, m. \tag{334}$$

Aşadar, am demostrat că are loc separabilitatea liniară a mulţimii $\{\Phi(x_i)\}_{i=1}^m$ **din „spaţiul de trăsături“ determinat de funcţia RBF de parametru** τ**. Prin urmare, problema SVM are soluţie [optimă] în respectivul „spaţiul de trăsături“, iar această soluţie produce eroare nulă la antrenare.**

Putem multiplica toţi α_i **cu o constantă pozitivă astfel încât relaţia** (334) **să devină**

$$y_i \sum_{j=1}^m \alpha_j y_j K(x_j, x_i) > 1 \text{ pentru } i = 1, \ldots, m.$$

[705] Presupunând $m > 1$.

Aşadar, am găsit o soluţie strict „fezabilă“ (w', w_0') pentru problema noastră de optimizare. Prin urmare, condiţia lui Slater este îndeplinită. În concluzie, soluţia optimă $(\bar{w}, \bar{w}_0)$ a acestei probleme [de tip SVM cu nucleu RBF de parametru τ] poate fi găsită, ca soluţie a sistemului de condiţii Karush-Kuhn-Tucker.

c. Clasificatorul C-SVM cu nucleu RBF nu va obţine în mod neapărat eroare nulă la antrenare pe setul de date considerat, chiar dacă asignăm parametrului τ valoarea găsită la punctul a. Rezultatul depinde de valorile parametrului C.

În *general*, pentru problema de optimizare convexă C-SVM soluţia optimă obţinută $(\bar{w}, \bar{w}_0, \bar{\xi})$ pentru o valoare fixată a parametrului de „destindere“ C nu produce în mod necesar eroare la antrenare 0, chiar dacă datele de antrenament sunt liniar separabile. Se poate să existe un alt triplet (w', w_0', ξ'), pentru care eroarea la antrenare rezultată să fie 0, dar pentru care $\frac{1}{2} \parallel w' \parallel^2 + C \sum_i \xi_i' > \frac{1}{2} \parallel \bar{w} \parallel^2 + C \sum_i \bar{\xi}_i$. Vedeţi de exemplu graficele B şi C de la problema 16 (deşi, veţi remarca faptul că acolo nu se foloseşte nucelul RBF).

27. **(SVM şi C-SVM, o proprietate: Adevărat sau Fals?)**

□ • ○ ***CMU, 2017 fall, Nina Balcan, midterm, pr. 1.2.bc***

***Observaţie importantă*: În contextul formulărilor legate de [C-]SVM din această problemă, nu vom folosi termen liber (engl., bias term).**

Fie $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ o mulţime de m instanţe care sunt separabile liniar [cu ajutorul unui separator care trece] prin originea sistemului de coordonate din $\mathbb{R}^d$. Considerăm de asemenea mulţimea S', obţinută din mulţimea S astfel: $S' = \{(cx_1, y_1), \ldots, (cx_m, y_m)\}$, unde $c > 0$ este o constantă.[706]

Care dintre afirmaţiile următoare sunt adevărate?

a. Atunci când SVM (se subînţelege, cu margine “hard”) ia ca input S şi respectiv S', se obţine acelaşi separator decizional, modulo un factor constant. Adică, dacă $\bar{w}$ şi $\bar{w}'$ sunt soluţiile produse de către SVM pentru inputul S şi respectiv S', atunci $\bar{w} = c_1 \bar{w}'$, unde c_1 este o anumită constantă.

b. Atunci când C-SVM (adică, SVM cu margine “soft”) ia ca input S şi respectiv S', se obţine acelaşi separator decizional, modulo un factor constant.

Răspuns:

a. Fie $\bar{w}$ soluţia [optimă a] problemei

$$\begin{aligned} &\min_w \ \frac{1}{2}\|w\|^2 \\ \textbf{a. î.} \quad &y_i(w \cdot x_i) \geq 1, \ \textbf{pentru } i = 1, \ldots, m, \end{aligned} \tag{P}$$

şi $\bar{w}'$ soluţia [optimă a] problemei

$$\begin{aligned} &\min_{w'} \ \frac{1}{2}\|w'\|^2 \\ \textbf{a. î.} \quad &y_i(w' \cdot cx_i) \geq 1, \ \textbf{pentru } i = 1, \ldots, m. \end{aligned} \tag{P'}$$

[706] LC: În textul de la CMU apărea $c > 1$.

Subliniem (din nou!) faptul că în formularea problemelor (P) şi (P′) — contrar formulării generale a problemei de optimizare SVM de la ex. 9 — nu am folosit termen liber.

Întrucât $\bar{w}'$ este soluţie a problemei (P′), rezultă că $y_i\bar{w}'\cdot(cx_i) \geq 1 \Leftrightarrow y_i(c\bar{w}')\cdot x_i \geq 1$ pentru $i = 1,\ldots,m$. Prin urmare, $c\bar{w}'$ satisface restricţiile problemei (P), pentru care soluţia optimă este $\bar{w}$, ceea ce implică mai departe faptul că

$$\frac{1}{2}\|\bar{w}\|^2 \leq \frac{1}{2}\|c\bar{w}'\|^2 \Leftrightarrow \|\bar{w}\|^2 \leq \|c\bar{w}'\|^2 \Leftrightarrow \|\bar{w}\|^2 \leq c^2\|\bar{w}'\|^2. \tag{335}$$

Invers, $\bar{w}$ fiind soluţie a problemei (P), implică $y_i\bar{w}\cdot x_i \geq 1 \Leftrightarrow y_i\Big(\frac{1}{c}\bar{w}\Big)\cdot(cx_i) \geq 1$ pentru $i = 1,\ldots,m$. Prin urmare, $\frac{1}{c}\bar{w}$ satisface restricţiile problemei (P′), pentru care soluţia optimă este $\bar{w}'$, ceea ce implică mai departe faptul că

$$\frac{1}{2}\|\frac{1}{c}\bar{w}\|^2 \geq \frac{1}{2}\|\bar{w}'\|^2 \Leftrightarrow \frac{1}{c^2}\|\bar{w}\|^2 \geq \|\bar{w}'\|^2 \Leftrightarrow \|\bar{w}\|^2 \geq c^2\|\bar{w}'\|^2. \tag{336}$$

Din relaţiile (335) şi (336) rezultă că $\|\bar{w}\|^2 = c^2\|\bar{w}'\|^2$, deci $\|\bar{w}\| = c\|\bar{w}'\|$. În continuare vom arăta că $\bar{w} = c\bar{w}'$ (o egalitate mai „tare“ decât egalitatea pe care tocmai am demonstrat-o).

Am precizat mai sus că din faptul că $\bar{w}'$ este soluţie a problemei (P′) rezultă că $c\bar{w}'$ satisface restricţiile problemei (P), despre a cărei funcţie obiectiv ştim acum că are valoarea optimă $\frac{1}{2}\|\bar{w}\|^2 = \frac{1}{2}c^2\|\bar{w}'\|^2 = \frac{1}{2}\|c\bar{w}'\|^2$. Aşadar, $c\bar{w}'$ este soluţie optimă a problemei (P). Ţinând cont de faptul că soluţia problemei (P) este unică[707] — ea existând în cazul în care setul de date de antrenament este liniar separabil şi nedegenerat[708] —, rezultă că are loc egalitatea $c\bar{w}' = \bar{w}$. Aşadar, afirmaţia din enunţ este *adevărată*.

***Observaţie*: Separatorul ca atare, este unul şi acelaşi: $\bar{w}\cdot x = 0 \Leftrightarrow c\bar{w}'\cdot x = 0 \Leftrightarrow \bar{w}'\cdot(cx) = 0$.**

b. Răspunsul în acest caz este *negativ*, din cauza faptului că separatorii depind de valoarea parametrului C. Vom ilustra acest fapt folosind un *exemplu* simplu.

Vom considera setul de date de antrenament din enunţul problemei 17, şi anume $S = \{(x_1 = -1, y_1 = -1), (x_2 = 1, y_2 = 1)\}$; el este separabil prin originea sistemului de coordonate.

***Observaţie importantă*: Se poate arăta (făcând un raţionament similar cu cel de la problema 12) că forma duală a problemei C-SVM în cazul în care nu se foloseşte termen liber w_0 este similară cu forma duală (D′) de la problema 12.d, cu singura *diferenţă* că acum nu vom mai avea restricţia $\sum_i \alpha_i y_i = 0$.**

Procedând într-o manieră similară cu cea din prima parte a rezolvării problemei 17, se constată că

[707] Motivaţia ţine de faptul că funcţia obiectiv a problemei (P) este strict convexă. Vedeţi teorema 1 din articolul *Uniqueness of the SVM solution*, de Christopher Burges şi David Crisp, 1998.

[708] Aceasta se traduce prin: $\exists w$ a. î. $y_i w\cdot x_i > 0$ (în condiţiile acestei probleme) pentru $i = 1,\ldots,m$ şi respectiv $\exists i, j$ a. î. $y_i = 1$ şi $y_j = -1$.

- **pentru** S**, avem** $L_D(\alpha) = \alpha_1 + \alpha_2 - \frac{1}{2}(\alpha_1 + \alpha_2)^2 = s - \frac{1}{2}s^2 = \frac{1}{2}s(2-s)$,[709] **unde** $s \stackrel{not.}{=} \alpha_1 + \alpha_2$**. Această funcţie de gradul al doilea îşi atinge maximul pentru** $s = 1$**. Aşadar,** $\max_{0 \leq \alpha_i \leq C} L_D(\alpha)$ **se obţine pentru** $s = 1$ **dacă** $C \geq 1/2$**, şi respectiv pentru** $s = C$ **dacă** $C \in (0, 1/2)$**;**
- **pentru** $S' = \{(x_1 = -2, y_1 = -1), (x_2 = 2, y_2 = 1)\}$**, avem** $L_D(\alpha') = \alpha'_1 + \alpha'_2 - \frac{1}{2}4(\alpha'_1 + \alpha'_2)^2 = s' - 2s'^2 = s'(1 - 2s')$**, unde** $s' \stackrel{not.}{=} \alpha'_1 + \alpha'_2$**. Această funcţie de gradul al doilea îşi atinge maximul pentru** $s' = 1/4$**. Aşadar,** $\max_{0 \leq \alpha'_i \leq C} L_D(\alpha')$ **se obţine pentru** $s' = 1/4$ **dacă** $C \geq 1/8$ **(şi respectiv pentru** $s' = C$ **dacă** $C \in (0, 1/8)$**.**

Notând cu $\bar{\alpha}_i,\ i \in \{1, 2\}$ **soluţiile optime pentru problema duală în cazul** S **şi cu** $\bar{\alpha}'_i,\ i \in \{1, 2\}$ **soluţiile optime pentru problema duală în cazul** S'**, urmează că în funcţie de valorile parametrului** C **avem următoarele corespondenţe între** $\bar{w}$ **şi** $\bar{w}'$**:**

- $C > 1/2$**: pentru** S **avem** $\bar{\alpha}_1 + \bar{\alpha}_2 = 1 \Rightarrow \bar{w} = \bar{\alpha}_1 y_1 x_1 + \bar{\alpha}_2 y_2 x_2 = \bar{\alpha}_1 + \bar{\alpha}_2 = 1$**, iar pentru** S' **avem** $\bar{\alpha}'_1 + \bar{\alpha}'_2 = 1/4 \Rightarrow \bar{w}' = \bar{\alpha}'_1 y_1 x_1 + \bar{\alpha}'_2 y_2 x_2 = 2(\bar{\alpha}'_1 + \bar{\alpha}'_2) = \frac{1}{2} = \frac{1}{2}\bar{w}$**;**
- $C = 1/2$**: pentru** S **avem** $\bar{\alpha}_1 = \bar{\alpha}_2 = 1/2 \Rightarrow \bar{w} = \bar{\alpha}_1 + \bar{\alpha}_2 = 1$**, iar pentru** S' **avem** $\bar{\alpha}'_1 + \bar{\alpha}'_2 = 1/4 \Rightarrow \bar{w}' = 2(\bar{\alpha}'_1 + \bar{\alpha}'_2) = 1/2 = \frac{1}{2}\bar{w}$**;**
- $C \in (1/8, 1/2)$**: pentru** S **avem** $\bar{\alpha}_1 + \bar{\alpha}_2 = C \Rightarrow \bar{w} = \bar{\alpha}_1 + \bar{\alpha}_2 = C$**, iar pentru** S' **avem** $\bar{\alpha}'_1 + \bar{\alpha}'_2 = 1/4 \Rightarrow \bar{w}' = 2(\bar{\alpha}'_1 + \bar{\alpha}'_2) = 1/2$**;**
- $C = 1/8$**: pentru** S **avem** $\bar{\alpha}_1 + \bar{\alpha}_2 = C = 1/8 \Rightarrow \bar{w} = \bar{\alpha}_1 + \bar{\alpha}_2 = 1/4$**, iar pentru** S' **avem** $\bar{\alpha}'_1 = \bar{\alpha}'_2 = 1/8 \Rightarrow \bar{w}' = 2(\bar{\alpha}'_1 + \bar{\alpha}'_2) = 1/2 = 2\bar{w}$**;**
- $C \in (0, 1/8)$**: pentru** S **avem** $\bar{\alpha}_1 + \bar{\alpha}_2 = C \Rightarrow \bar{w} = \bar{\alpha}_1 + \bar{\alpha}_2 = C$**, iar pentru** S' **avem** $\bar{\alpha}'_1 + \bar{\alpha}'_2 = C \Rightarrow \bar{w}' = 2(\bar{\alpha}'_1 + \bar{\alpha}'_2) = 2C = 2\bar{w}$**.**

În concluzie,

- ***nu*** **avem o relaţie de tip** $\bar{w}' = \frac{1}{2}\bar{w}$ **(ca la punctul** a**)** ***decât*** **pentru cazurile** $C \geq 1/2$**;**
- **pentru** $C \in (0, 1/8]$ **avem chiar o relaţie „inversată“:** $\bar{w}' = 2\bar{w}$**;**
- **pentru** $C \in (1/8, 1/2)$ **nu avem practic o legătură de genul celei sugerate în enunţ între** $\bar{w}$ **şi** $\bar{w}'$**, întrucât** $\bar{w} = C$ **şi** $\bar{w}' = 1/2$**.**

28. **(SVM şi C-SVM: Adevărat ori Fals?)**

CMU, 2003 fall, T. Mitchell, A. Moore, final exam, pr. 7.c

a. Maşinile cu vectori-suport (ne referim în speţă la SVM şi C-SVM) identifică întotdeauna optimul global pentru funcţia obiectiv din problema de optimizare asociată.

CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, final exam, pr. 3.b

b. La clasificare cu ajutorul SVM, atunci când transformăm instanţele de antrenament folosind mapări (Φ) **ce corespund unor funcţii-nucleu polinomiale**

[709] Vedeţi expresia obţinută pentru $L_D(\alpha)$ în demonstraţia de la problema 12.d.

(K) având gradul $1, 2, 3, \ldots$, ne aşteptăm ca vectorii-suport să rămână în general aceiaşi.

Stanford, 2007 fall, Andrew Ng, practice midterm exam, pr. 6.f

c. Presupunem că lucrăm cu o maşină cu vectori-suport de normă L_1 folosind parametrul de „destindere" $C > 0$ şi că folosim un set de date de antrenament liniar separabile. Urmărind să minimizeze funcţia obiectiv, C-SVM-ul va asigna variabilelor de „destindere" ξ_i valoarea 0, întrucât datele sunt liniar separabile. În consecinţă, soluţia obţinută (w, w_0) va fi aceeaşi, indiferent de valoarea folosită pentru parametrul C (presupunând, bineînţeles, că această valoare este strict pozitivă). Adevărat sau fals?

Răspuns:

a. Adevărat. Învăţarea unei funcţii de clasificare prin intermediul unei maşini cu vectori-suport constă în rezolvarea unei probleme de optimizare convexă, cu funcţia obiectiv de ordin pătratic şi restricţii liniare. Algoritmii / metodele de rezolvare computaţională a acestui tip de probleme permit găsirea soluţiei optime, cu o eroare situată sub un prag oarecare, fixat $\varepsilon > 0$.

b. Fals. Vectorii de trăsături $(\Phi(x_i))$ corespunzători nucleielor polinomiale sunt rezultatul aplicării unor funcţii neliniare asupra vectorilor de intrare (x_i). Din această cauză, vectorii-suport pentru hiperplanul de separare optimală în spaţiul de trăsături pot fi foarte diferiţi de la o funcţie-nucleu la alta. (Pentru *exemplificare*, vedeţi problema 24, graficele B, C, D, şi F.)

c. Fals. Chiar dacă datele de antrenament sunt liniar separabile, totuşi poziţia separatorului optimal poate fi afectată de outlier-e. Astfel, în funcţie de valoarea parametrului C, maşina cu vectori-suport cu margine "soft" (C-SVM) poate decide să clasifice [chiar] în mod eronat unu sau mai multe exemple, dacă în acest fel se obţine o margine mai mare. Valoarea parametrului C va influenţa modul în care se face acest compromis (engl., trade-off) între mărimea marginii $(1/\|w\|)$ pe de o parte şi suma variabilelor ξ_i pe de altă parte. (Pentru *exemplificare*, vedeţi problemele 15 şi 16, iar la problema 24 graficele B, C şi F.)

5.1.3 Alte *probleme de optimizare* de tip SVM

29. **(Funcţia-nucleu RBF — o proprietate remarcabilă: pentru orice set de instanţe de antrenament distincte şi pentru orice etichetare a acestora şi, de asemenea, pentru orice valoare a parametrului funcţiei-nucleu RBF, problema de optimizare de tip SVM care impune ca toate instanţele de antrenament să fie corect clasificate şi la distanţa $1/\|w\|$ de separatorul optimal are soluţie)**

■ □ ● ○ ∗ ***MIT, 2009 fall, Tommi Jaakkola, HW2, pr. 1***

A. Expresia de definiţie a funcţiei-nucleu cu bază radială (RBF) poate fi scrisă sub forma următoare:

$$K(x, x') = \exp\Big(-\frac{1}{2\sigma^2}\|x - x'\|^2\Big), \tag{337}$$

unde x şi x' sunt elemente din $\mathbb{R}^d$, iar σ este parametrul care arată cât de repede se micşorează valoarea funcţiei-nucleu pe măsură ce punctele x şi x' se situează la distanţe din ce în ce mai mari unul faţă de celălalt.

În acest exerciţiu ne propunem să arătăm că funcţia-nucleu RBF are câteva *proprietăţi* remarcabile. În primul rând, ea poate separa în mod perfect *orice* mulţime finită de instanţe de antrenament *distincte*,[710] iar acest rezultat este valabil pentru *orice* valoare pozitivă finită a parametrului σ.[711] (Trebuie să menţionăm totuşi că valoarea lui σ afectează calitatea generalizării / predicţiei pe instanţe de test.)

a. Vom începe să demonstrăm că problema de optimizare

$$\min_w \frac{1}{2}\|w\|^2 \text{ cu restricţiile } y_i w \cdot \phi(x_i) = 1 \text{ pentru } i = 1, \ldots, n \tag{338}$$

admite soluţie, indiferent de ce valori (± 1) ar avea etichetele y_i asignate instanţelor x_i, cu $i = 1, \ldots, n$. Am notat cu $\phi(x_i)$ vectorul de trăsături (engl., feature vector) determinat de funcţia de „mapare" care corespunde funcţiei K (vedeţi definiţia dată mai sus).

***Comentariu*:** Formularea problemei de optimizare (338) diferă de formularea problemei SVM], din două motive. Mai întâi, se observă că încercăm să găsim o soluţie a acestei probleme [SVM de un tip particular] astfel încât toate instanţele etichetate date să fie vectori-suport. Evident, această proprietate nu poate fi satisfăcută pentru orice funcţie-nucleu validă, însă vom păstra pentru început o generalitate mai mare decât este nevoie — la punctele a şi b vom considera K funcţie-nucleu oarecare şi doar începând cu punctul c vom restricţiona K la funcţia-nucleu RBF, definită în relaţia (337) — şi vom vedea că astfel ne va fi mai uşor să atingem obiectivul pe care ni l-am propus iniţial. În al doilea rând, observăm că formularea noastră omite termenul liber (bias-ul, notat în general cu b sau cu w_0), din cauză că el nu este necesar pentru demonstrarea proprietăţii enunţate la acest punct (a).

Introduceţi multiplicatori Lagrange pentru restricţiile problemei de optimizare date — similar cu cele de la problema SVM duală, vedeţi problema 9 — şi indicaţi forma generală a soluţiei, w^*. Pentru simplificarea calculelor / raţionamentului, vă recomandăm / permitem să lucraţi ca şi cum vectorii w şi $\phi(x_i)$ ar avea întotdeauna dimensiune finită (deşi nu acesta este cazul funcţiilor-nucleu RBF; vedeţi problema 75 de la capitolul de *Fundamente*).

***Observaţie*: Întrucât încercăm să satisfacem restricţii de tip egalitate, multiplicatorii Lagrange pot lua orice valori reale. Aşadar, spre deosebire de problema de optimizare SVM standard, aici multiplicatorii Lagrange nu [mai] sunt constrânşi să ia valori pozitive.**

[710] La problema 26 am demonstrat un astfel de rezultat folosind SVM şi alegând în mod convenabil o anumită valoare pentru σ. Aici vom formula o problema de optimizare convexă similară, însă sensibil diferită de problema SVM.

[711] O altă variantă a acestui rezultat poate fi obţinută folosind regresia liniară kernel-izată, cu funcţie-nucleu de tip RBF. A se vedea problema 10 de la capitolul *Metode de regresie*.

Vă cerem să exprimaţi w^*, soluţia problemei de optimizare (338), în funcţie de aceşti multiplicatori Lagrange. (Aceasta n-ar trebui să implice calcule prea elaborate.)

b. Introduceţi soluţia w^* pe care aţi obţinut-o la punctul precedent în *restricţiile* de clasificare (corespunzătoare „marginilor") din problema de optimizare în formă primală (338) şi exprimaţi rezultatul sub forma unei combinaţii liniare de aplicări ale funcţiei-nucleu K.

c. Arătaţi în manieră succintă cum anume se poate folosi *teorema lui Michelli* — redată mai jos — pentru a demonstra că pentru orice funcţie-nucleu RBF K, matricea pătratică definită prin $K_{ij} = K(x_i, x_j)$ conform relaţiei (337) pentru i şi $j \in \{1, \ldots, n\}$ este inversabilă.

***Teoremă (Michelli, 1986)*: Dacă $\rho : [0, \infty) \to \mathbb{R}$ este o funcţie monotonă pe intervalul de definiţie, atunci pentru orice set de puncte distincte x_i dintr-un spaţiu $\mathbb{R}^l$, cu $i = 1, \ldots, n$, matricea pătratică — la care ne vom referi, prin abuz de notaţie, tot cu ρ — ale cărei elemente sunt $\rho_{ij} \stackrel{def.}{=} \rho(\|x_i - x_j\|)$, este inversabilă.**

d. Coroborând rezultatele obţinute la punctele precedente, demonstraţi că într-adevăr se poate găsi o soluţie a problemei de optimizare date în forma primală (338) [şi, în consecinţă, toate instanţele x_i vor fi vectori-suport].

B. Desigur, faptul că putem separa, în principiu, orice mulţime de exemple de antrenament nu implică în mod neapărat că un clasificator oarecare antrenat pe aceste date se comportă bine pe date de test. (Din contra!) Aşadar, cum se justifică faptul că folosim [pentru clasificare, în particular cu SVM] nuclee RBF? Răspunsul ţine de „marginea" maximă [LC: de separare] pe care o putem atinge atunci când variem parametrul σ.

***Observaţie*: Variind valoarea lui σ, efectul asupra marginii [LC: geometrice] nu este pur şi simplu [dat de] scalarea vectorilor de trăsături $\phi(x_i)$. Într-adevăr, este uşor să observăm că pentru [orice] nucleu RBF, adică pentru orice valoare a lui σ, avem**

$\|\phi(x)\|^2 = \phi(x) \cdot \phi(x) = K(x, x) = 1$ **(valoarea maximă posibilă pentru orice RBF).**

e. Să începem prin a atribui lui σ o valoare pozitivă foarte mică. Cât este — în acest caz — valoarea marginii geometrice $1/\|w^*\|$ pe care o obţinem ca răspuns [la rezolvarea problemei de optimizare (338)] pentru orice set de n instanţe de antrenament distincte?

f. Alegeţi un set de instanţe din spaţiul unidimensional ($\mathbb{R}$) în aşa fel încât să arătaţi că marginea [LC: geometrică, văzută ca distanţa până la hiperplanul $w^* \cdot x = 0$] poate fi mai mare decât cea obţinută ca răspuns la punctul e (pentru $\sigma \to 0$). Puteţi da orice valori pentru σ şi puteţi seta instanţele cum doriţi, astfel încât să puneţi în evidenţă modul în care ele pot determina „marginea" / distanţa maximă dintre ele.

Răspuns:

a. Funcţia lagrangeană corespunzătoare acestei probleme de optimizare este următoarea:

$$\begin{aligned} L(w,\alpha) &= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n}\alpha_i(y_i w\cdot\phi(x_i)-1) \\ &= \frac{1}{2}\|w\|^2 - w\cdot\Big(\sum_{i=1}^{n}\alpha_i y_i\phi(x_i)\Big) + \sum_{i=1}^{n}\alpha_i. \end{aligned}$$

Aşa cum s-a precizat deja, variabilele Lagrange α_i sunt *nerestricţionate*, fiindcă lucrăm doar cu constrângeri de tip egalitate.

Conform metodei multiplicatorilor lui Lagrange, forma *duală* a problemei de optimizare (338) este:

$$\max_{\alpha}\underbrace{\min_{w} L(w,\alpha)}_{\text{not.: } g(\alpha)}.$$

Observăm că atunci când fixăm [în mod arbitrar] valori pentru parametrii α_i, expresia $L(w,\alpha)$ va fi un polinom de gradul doi în raport cu variabilele w_i, iar coeficientul dominant al lui w_i este pozitiv, pentru orice $i=1,\ldots,n$. Prin urmare, putem obţine valoarea optimă w^* folosind (de exemplu) prima derivată: $\dfrac{\partial L(w,\alpha)}{\partial w}=0$. În final, procedând ca la problema 9, vom obţine soluţia următoare, scrisă sub formă vectorială (nu pe componente):

$$w^* = \sum_{i=1}^{n}\alpha_i^* y_i\phi(x_i).$$

***Convenţie de notaţie*: Pentru a simplifica redactarea, vom folosi în continuare notaţia $w^* = \Phi[y\bullet\alpha^*]$.[712] Operatorul $\bullet$ reprezintă produsul pe componente al vectorilor y şi α^*, iar Φ este o matrice de dimensiune $m\times n$, în care coloana i este vectorul $\phi(x_i)$. (Desigur, $m=\infty$ în cazul funcţiei-nucleu RBF; vedeţi problema 75 de la capitolul *Fundamente*.)**

b. Se observă uşor că restricţiile $y_i w\cdot\phi(x_i)=1$ pentru $i=1,\ldots,n$ din problema primală (338) pot fi scrise în mod echivalent (folosind notaţia matriceală) astfel:

$$\phi(x_i)^\top w = y_i,\ \text{pentru } i=1,\ldots,n. \tag{339}$$

Aceste egalităţi sunt satisfăcute pentru w^*, soluţia problemei de optimizare (338), a cărei expresie a fost dedusă la punctul precedent, $w^*=\Phi[y\bullet\alpha^*]$. Prin urmare, relaţiile (339) pot fi scrise în manieră unitară (adică, simultan pentru $i=1,\ldots,n$) astfel:

$$\begin{aligned} \Phi^\top w^* &= y \Leftrightarrow \\ \Phi^\top\Phi[y\bullet\alpha^*] &= y \Leftrightarrow \\ K[y\bullet\alpha^*] &= y, \end{aligned} \tag{340}$$

unde prin K am notat matricea $\Phi^\top\Phi$, care este matricea-nucleu (şi care se mai numeşte matricea Gram).[713]

[712] Folosind operatorul $\times$ pentru înmulţirea matricelor, egalitatea aceasta se scrie: $w^*=\Phi\times[y\bullet\alpha^*]$. Sugerăm cititorului să folosească şi mai jos această notaţie la fiecare apariţie a vectorul-coloană $[y\bullet\alpha^*]$, dacă astfel sporeşte înţelegerea textului nostru.

[713] Această matrice este pătratică, iar elementele ei sunt $\phi(x_i)\cdot\phi(x_j)=K(x_i,x_j)$, unde K desemnează funcţia-nucleu de la care am pornit.

***Observaţie*: Există acum o suprapunere la nivel de notaţie pentru funcţia-nucleu K şi matricea-nucleu corespunzătoare acestei funcţii, în raport cu o mulţime de instanţe $x_1, \ldots, x_n$. Considerăm însă că dezambiguizarea / distingerea între semnificaţiile celor două notaţii se poate face uşor, ţinând cont de context.**

c. În cazul în care K este funcţie-nucleu cu bază radială, $K(x_i, x_j)$ are prin definiţie forma $\exp\Big(-\frac{1}{2\sigma^2}\|x_i - x_j\|^2\Big)$ pentru orice x_i şi x_j din domeniul de definiţie. Funcţia $\exp\Big(-\frac{1}{2\sigma^2}t^2\Big)$ este monotonă (descrescătoare) pentru $t \in [0, \infty)$. Aplicând teorema lui Michelli, rezultă că pentru orice set de puncte distincte x_i, cu $i = 1, \ldots, n$, matricea K ale cărei elemente sunt $K_{ij} \stackrel{def.}{=} \exp\Big(-\frac{1}{2\sigma^2}\|x_i - x_j\|^2\Big)$ este inversabilă.

d. Prin ipoteză, instanţele x_i, cu $i = 1, \ldots, n$ sunt distincte, iar K este nucleu RBF. În consecinţă, conform punctului c, matricea-nucleu K este inversabilă. Vom arăta că în aceste condiţii sistemul de ecuaţii liniare (339) are soluţie. Într-adevăr, conform relaţiei (340) de la punctul b,

$$K[y \bullet \alpha^*] = y \Rightarrow [y \bullet \alpha^*] = K^{-1}y.$$

Ştim însă de la punctul a că $w^* = \Phi[y \bullet \alpha^*]$. Prin urmare, $w^* = \Phi K^{-1}y$. (Aceasta este soluţia unică a sistemului liniar (339), în necunoscutele w_i.)

e. Pe măsură ce $\sigma \to 0$, matricea-nucleu K va tinde la matricea unitate I, fiindcă

$$\lim_{\sigma \to 0} \exp\Big(-\frac{1}{2\sigma^2}\|x_i - x_j\|^2\Big) \to 0, \text{ pentru orice } i \neq j,$$

iar

$$\exp\Big(-\frac{1}{2\sigma^2}\|x_i - x_i\|^2\Big) = 1.$$

Conform relaţiei (340), $K[y \bullet \alpha^*] = y$, deci $K \to I$ implică la rândul său $y \bullet \alpha^* \to y$ şi, deci, $\alpha^* \to \bar{1}$, vectorul-coloană care are toate elementele egale cu 1.

Pentru a vedea cât este „marginea" $\frac{1}{\|w^*\|}$ atunci când $\sigma \to 0$, vom calcula mai întâi $\|w^*\|^2$, pornind de la relaţia $w^* = \Phi[y \bullet \alpha^*]$:

$$\begin{aligned} \|w^*\|^2 &= {w^*}^\top w^* = (\Phi[y \bullet \alpha^*])^\top \Phi[y \bullet \alpha^*] = [y \bullet \alpha^*]^\top \underbrace{\Phi^\top \Phi}_{K}[y \bullet \alpha^*] \\ &= [y \bullet \alpha^*]^\top K[y \bullet \alpha^*] \end{aligned} \tag{341}$$

În consecinţă, atunci când $\sigma \to 0$, vom avea

$$\|w^*\|^2 = [y \bullet \alpha^*]^\top K[y \bullet \alpha^*] \to y^\top I y = n,$$

deci „marginea" [LC: de separare] care se obţine este $\frac{1}{\|w^*\|} = \sqrt{\frac{1}{n}}$.

f. Considerăm două instanţe distincte x şi x' din $\mathbb{R}$, ambele cu eticheta $+1$. Vom nota $k = K(x, x') = \exp\Big(-\frac{1}{2\sigma^2}\|x - x'\|^2\Big)$. Se vede imediat că matricea-nucleu este:

$$K = \begin{bmatrix} 1 & k \\ k & 1 \end{bmatrix}$$

Rezolvând sistemul de ecuaţii liniare $K[\bar{1} \bullet \alpha] = \bar{1}$ implicat de relaţia (340), obţinem soluţia

$$\alpha^* = \left[\frac{1}{k+1}\ ,\ \frac{1}{k+1}\right]^\top .$$

În continuare, folosind formula (341) dedusă la punctul e, obţinem $\|w^*\|^2 = {\alpha^*}^\top K \alpha^* = \frac{2}{k+1}$. Prin urmare, „marginea" [LC: geometrică] este $\frac{1}{\|w^*\|} = \sqrt{\frac{k+1}{2}}$. Evident, $k > 0$, deci „marginea" obţinută aici este strict mai mare decât $\sqrt{\frac{1}{2}}$, valoarea marginii care a fost obţinută în cazul $\sigma \to 0$ (vedeţi punctul e).

De asemenea, atunci când apropiem cele două instanţe x şi x' (sau, la fel, când $\sigma \to +\infty$), rezultă $k \to 1$ şi, în consecinţă, obţinem „marginea" 1.

***Observaţie*:**

Pornind de la relaţia $\frac{1}{\|w^*\|} = \frac{1}{\sqrt{n}}$ stabilită la punctul e şi / sau de la relaţia $\frac{1}{\|w^*\|} = \sqrt{\frac{k+1}{2}}$ obţinută la acest punct, ne putem pune acum *întrebarea* următoare: „marginea" maximă pe care o putem obţine în cazul nucleului RBF este oare [întotdeauna] 1?

Ca să răspundem la această întrebare, observăm că pentru orice instanţă x_i avem $\|\phi(x_i)\|^2 = K(x_i, x_i) = 1$. În consecinţă, ne putem gândi la vectorii de trăsături $\phi(x_i)$ (care sunt infinit-dimensionali!) ca fiind poziţionaţi pe sfera de rază 1 (engl., unit ball) având centrul în originea sistemului de coordonate. Rezultă că, într-adevăr, valoarea maximă posibilă pentru marginea de separare este 1.

Intuitiv, ne putem gândi astfel: pe măsură ce $\sigma \to +\infty$, „distanţele" $K(x, x')$ vor tinde la 1, iar nucleele centrate pe instanţe distincte vor deveni indiscernabile.[714] Prin urmare, vectorii de trăsături vor tinde să coincidă, identificându-se la limită cu un singur punct pe sfera de rază 1 şi cu centrul în originea sistemului de coordonate. La limită „marginea" [LC: geometrică] va avea mărimea 1 (ceea ce este contra-intuitiv!).

30. **(Clasificare n-ară cu SVM: SVM multiclass, cu margine "hard")**

prelucrare de Liviu Ciortuz, după
□ • MIT, 2009 fall, Tommi Jaakkola, lecture notes 5

În practică, majoritatea problemelor de clasificare lucrează cu mai mult de două clase (de exemplu, identificarea persoanelor dintr-o imagine, prezicerea fonemelor în procesarea vorbirii, asocierea genelor la anumite procese biologice, ori prezicerea tipului de cancer pe baza analizei datelor preluate la nivel celular). În acest capitol, până aici ne-am ocupat doar de [task-uri de] clasificare binară. Totuşi, metodele de tip SVM cu care aţi făcut cunoştinţă — dar de asemenea şi alte metode, precum Perceptronul —, pot fi folosite şi pentru classificare n-ară (engl., multiclass).

[714] $\sigma \to +\infty \Rightarrow K(x, x') \to 1$ şi $K(x, x'') \to 1$ pentru orice x şi $x' \neq x''$. Deci $\phi(x) \cdot \phi(x') \approx \phi(x) \cdot \phi(x'')$ pentru orice x, x' şi x''.

Una dintre modalităţile de a face clasificare n-ară, şi anume metoda "one versus all", constă în a reduce problema de clasificare multiclass la un set de task-uri de clasificare binară. În această abordare, fiecare dintre clasificatorii binari este antrenat în mod independent faţă de ceilalţi.[715] O altă modalitate — cea care ne va interesa pe noi aici — este să încercăm să formulăm o *problemă de optimizare* per ansamblu, în aşa fel încât să putem antrena [în mod direct] un clasificator n-ar.

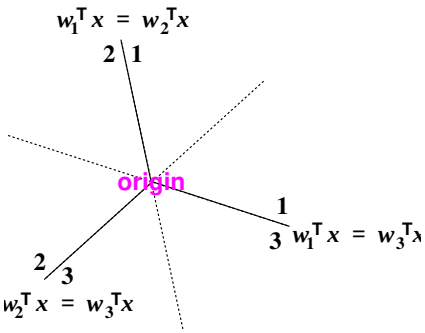
Vom folosi drept *componente* constitutive ale clasificatorului nostru n-ar mai mulţi clasificatori liniari simpli care trec prin originea sistemului de coordonate,[716] adică având forma analitică $f_j(x) = w_j \cdot x$, cu $x \in \mathbb{R}^d$ şi $w_j \in \mathbb{R}^d$, pentru $j = 1, \ldots, K$.[717] Apoi vom impune cerinţa ca [la clasificarea fiecărei instanţe x] să „câştige" acel clasificator care corespunde etichetei corecte, în sensul că valoarea *funcţiei* asociate acelui clasificator [pentru x] să fie cea mai mare dintre toate valorile funcţiilor asociate diverşilor clasificatori liniari [pentru acelaşi x], lăsând în plus un mic „spaţiu de manevră". Aşadar, problema pe care vom încerca să o rezolvăm este următoarea:

$$\begin{aligned} &\min_{w^{(1)},\ldots,w^{(K)}} \frac{1}{2} \sum_{k=1}^{K} \|w^{(k)}\|^2 \\ \textbf{a. î.} \quad &w^{(y_i)} \cdot x_i \geq w^{(y)} \cdot x_i + 1, \qquad \textbf{pentru } i = 1, \ldots, m \\ &\qquad\qquad\qquad\qquad\quad \textbf{şi orice } y \in \{1, \ldots, K\}, \textbf{ cu } y \neq y_i, \end{aligned} \tag{342}$$

unde $w^{(1)}, \ldots, w^{(K)} \in \mathbb{R}^d$, $x_i \in \mathbb{R}^d$, iar $y_i \in \{1, \ldots, K\}$ pentru $i = 1, \ldots, m$.

Observaţi că în această abordare componentele clasificatorului n-ar sunt antrenate împreună. Predicţia pentru o instanţă oarecare x este determinată de cât de mult „susţine" fiecare componentă-clasificator etichetarea corespunzătoare: $\hat{y} = \arg\max_y \{w^{(y)} \cdot x\}$.

Remarcaţi faptul că *zonele de decizie* care rezultă au o *interpretare geometrică* simplă. Putem obţine fiecare dintre aceste zone ca *intersecţie* a două *regiuni*, corespunzător mulţimilor de exemple pentru care o anumită etichetă este preferată în defavoarea celorlalte etichete. Aceste regiuni sunt determinate în mod simplu prin linii (în general, hiperplane) care trec prin originea sistemului de coordonate: $w^{(y)} \cdot x = w^{(y')} \cdot x$, ca în figura alăturată, unde $y, y' \in \{1, 2, 3\}$. Liniile punctate arată cum anume se extind aceste hiperplane în regiunile cărora le sunt asociate alte etichete. Regulile de decizie sunt desemnate prin linii continue.



[715] Vedeţi MIT, 2009 fall, Tommi Jaakkola, lecture notes 5, pag. 7–11, precum şi articolul *Reducing multiclass to binary* de Erin Allwein et al, din revista Journal of Machine Learning Research, 2000.

[716] La problema 62, care introduce versiunea SVM multiclass cu margine "soft", această cerinţă va fi relaxată, în sensul că vom permite ca aceşti clasificatori liniari să aibă termeni liberi (b_k, pentru $k = 1, \ldots, K$) eventual diferiţi între ei. Pe lângă aceasta, vom cere ca ei să se intersecteze într-un acelaşi punct (ca şi în cazul de faţă), dar nu neapărat în originea sistemului de coordonate.

[717] Pentru convenienţă, deşi am folosit mai sus termenul de clasificare n-ară, în continuare în formalismul matematic vom desemna numărul de clase cu simbolul K. (Aşadar, se poate considera $K = n$.)

Înainte de a trece la rezolvarea acestei probleme de optimizare în forma duală, ar fi util (pentru mai târziu) să scriem problema într-o formă uşor mai generală. Pentru aceasta, vom nota

$$w = \left(w^{(1)}, \cdots, w^{(y)}, \cdots, w^{(K)}\right)^\top, \ \phi(x, y) = \left(0, \cdots, x, \cdots, 0\right)^\top,$$

unde locaţia lui x din cadrul vectorului $\phi(x, y)$ garantează faptul că $w \cdot \phi(x, y) = w^{(y)} \cdot x$. În consecinţă, forma primală a problemei de optimizare *SVM multiclass* se poate rescrie în mod echivalent (în raport cu forma anterioară (342)) astfel:

$$\min_w \frac{1}{2}\|w\|^2 \tag{343}$$

$$\textbf{a. î. } w \cdot \phi(x_i, y_i) \geq w \cdot \phi(x_i, y) + 1_{\{y \neq y_i\}}, \ \textbf{pentru } i = 1, \ldots, m \ \textbf{şi } y \in \{1, \ldots, K\}.$$

a. Deduceţi forma duală a problemei de optimizare (343). Veţi proceda similar cu modul în care s-a lucrat pentru clasificatorul binar SVM (vedeţi pr. 9).

***Sugestie*: Introduceţi multiplicatorii Lagrange $\alpha_{y;i} \geq 0$ pentru restricţiile de tip inegalitate, obţineţi expresia lagrangeanului generalizat, găsiţi valoarea lui w (ca funcţie de multiplicatorii Lagrange $\alpha_{y;i}$) care minimizează acest lagrangean şi apoi substituiţi această valoare (pe care o veţi nota cu $\hat{w}(\alpha)$) în expresia lagrangeanului, pentru a obţine în sfârşit varianta duală.**

b. Folosind $\hat{w}$, expresia vectorului de ponderi care reprezintă soluţia optimă a problemei SVM multiclass, care a fost dedusă la punctul precedent în funcţie de multiplicatorii Lagrange $\alpha_{y;i}$, scrieţi *regula de predicţie* pentru eticheta unei instanţe noi x, ţinând cont că

$$\hat{y} = \arg\max_y \{\hat{w}^{(y)} \cdot x\}.$$

Răspuns:

a. Scriem mai întâi expresia lagrangeanului generalizat:

$$L_P(w, \alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{m}\sum_{y=1}^{K} \alpha_{y;i}[w \cdot \phi(x_i, y_i) - w \cdot \phi(x_i, y) - 1_{\{y \neq y_i\}}].$$

Apoi, calculăm derivata parţială a acestei funcţii în raport cu vectorul de ponderi w:

$$\frac{\partial}{\partial w} L_P(w, \alpha) = w - \sum_{i=1}^{m}\sum_{y=1}^{K} \alpha_{y;i}[\phi(x_i, y_i) - \phi(x_i, y)].$$

Conform *condiţiei de staţionaritate / optimalitate Karush-Kuhn-Tucker* care se referă la atingerea minimului lui L_P în raport cu w, vom egala această derivată parţială cu vectorul 0 şi vom obţine soluţia

$$\hat{w}(\alpha) = \sum_{i=1}^{m}\sum_{y=1}^{K} \alpha_{y;i}[\phi(x_i, y_i) - \phi(x_i, y)].$$

Acum vom substitui $\hat{w}(\alpha)$ în expresia lagrangeanului de mai sus, iar după ce vom simplifica termenii asemenea, vom obţine imediat următoarea expresie pentru lagrangeanul dual:

$$L_D(\hat{w},\alpha) = \sum_{i=1}^{m}\sum_{y=1}^{K}\alpha_{y;i}1_{\{y\neq y_i\}} - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{y=1}^{K}\sum_{y'=1}^{K}\alpha_{y;i}\,\alpha_{y';j}[\phi(x_i,y_i)-\phi(x_i,y)]\cdot[\phi(x_j,y_j)-\phi(x_j,y')].$$

Observaţi că această expresie poate fi simplificată şi mai mult, dacă ţinem cont că notaţia introdusă mai sus pentru $\phi(x,y)$ implică $\phi(x,y)\cdot\phi(x',y') = 1_{\{y=y'\}}\,(x\cdot x')$. Forma duală a problemei de optimizare SVM multiclass are ca funcţie obiectiv lagrangeanul L_D, iar ca restricţii $\alpha_{y;i}\geq 0$ pentru $y=1,\ldots,K$ şi $i=1,\ldots,m$.

b. Regula de decizie pentru clasificatorul SVM multiclass se poate scrie astfel:

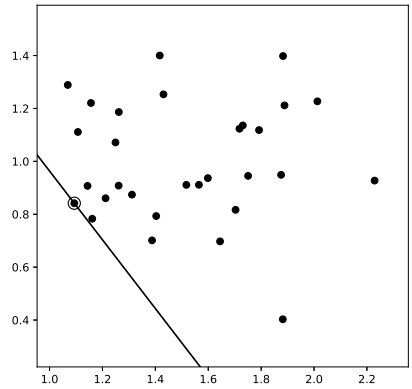
$$\hat{y} = \arg\max_y\{\hat{w}^{(y)}\cdot x\} = \arg\max_y\Big\{\sum_{i=1}^{m}\sum_{y'=1}^{K}\hat{\alpha}_{y';i}[\phi(x_i,y_i)-\phi(x_i,y')]\cdot\phi(x,y)\Big\},$$

unde $\hat{\alpha}_{y';i}$ sunt soluţiile problemei duale SVM multiclass. (Remarcaţi faptul că şi în acest caz se pot face simplificări de genul celor indicate la punctul precedent.)

31. **(Problema SVM *one-class* (varianta "Max Margin"), cazul marginii "hard")**

■ ***Stanford, 2007 fall, Andrew Ng, practice midterm exam, pr. 4***

Considerăm un set de instanţe neetichetate $\{x_1,\ldots,x_m\}\subset\mathbb{R}^d$.
Un algoritm SVM de tip *one-class* caută să identifice (dacă este posibil) o direcţie $w\in\mathbb{R}^d$ care separă în sens maximal datele de originea sistemului de coordonate, ca în figura de mai jos.[718]



Mai precis, acest algoritm rezolvă problema de optimizare (dată în formă primală):[719]

$$\min_w \frac{1}{2}\|w\|^2$$
$$\text{a. î.}\quad w\cdot x_i\geq 1,\ \text{pentru } i=1,\ldots,m.$$

O instanţă nouă (de test) va fi etichetată cu $+$ dacă $w\cdot x\geq 1$, şi cu $-$ în caz contrar.

[718] Este de remarcat faptul că *one-class SVM* este o tehnică de învăţare nesupervizată (aşadar, nu de clasificare), întrucât ea nu necesită ca instanţele să fie etichetate.

În altă ordine de idei, putem formula problema *one-class SVM* într-o formă *mai generală*:

$$\min_w \frac{1}{2}\|w\|^2$$
$$\text{a. î.}\quad w\cdot x_i + w_0\geq 1,\ \text{pentru } i=1,\ldots,m.$$

Însă, dacă extindem (cum se face de obicei la reţele nuronale artificiale) orice instanţă x_i cu o componentă $x_{i,0}=1$, ajungem la forma (mai simplă!) din enunţ.

[719] La problema 65 vom da o altă variantă a problemei *one-class* SVM, definită cu ajutorul *sferei de incluziune*

***Observaţie* (1): Un astfel de algoritm de tip SVM este util pentru detecţia anomaliilor (engl., anomaly detection). În astfel de situaţii, ni se dă mai întâi un set de date care se consideră „*normale*". Apoi ni se cere să decidem pentru alte instanţe dacă sunt (sau nu sunt) „anomalii" (engl., outliers).**

a. Scrieţi forma duală corespunzătoare problemei de optimizare *SVM one-class* (de tip *Max Margin*) de mai sus. Simplificaţi răspunsul cât mai mult posibil; evident, vectorul w nu trebuie să apară în rezultatul final. Verificaţi în prealabil dacă forma primală satisface condiţia lui Slater.

b. Am putea „kerneliza" algoritmul *SVM one-class* (de tip *Max Margin*) atât la antrenare cât şi la testare? Adică: dată fiind o funcţie-nucleu K, este posibil ca după maparea corespunzătoare, variabilele x să apară

– atât în expresia lagrangeanului (L_D) care reprezintă funcţia obiectiv a formei duale ale problemei SVM one-class,
– cât şi în funcţia de decizie / clasificare pentru o instanţă nouă x',
doar ca argumente ale funcţiei-nucleu K?

c. Concepeţi un algoritm de tip SMO[720] care să rezolve problema duală obţinută la punctul a. Ideea de bază a unui astfel de algoritm este ca la fiecare pas să se obţină soluţia optimă pe cea mai mică dintre toate submulţimile posibile de variabile. Daţi formulele analitice (engl., closed form formulas) pentru actualizarea variabilelor din această submulţime. Trebuie să justificaţi / explicaţi de ce este suficient să considerăm simultan respectivul număr de variabile la fiecare pas.

Răspuns:

a. Verificăm mai întâi faptul că problema de optimizare care a fost dată în enunţ în formă primală satisface condiţia lui Slater.[721] Dacă există un hiperplan care trece prin originea sistemului de coordonate şi „lasă" toate instanţele x_i cu $i = 1, \ldots, m$ de o aceeaşi parte a sa, aceasta înseamnă că există $w \in \mathbb{R}^d$ astfel încât $w \cdot x_i > 0$ pentru $i = 1, \ldots, m$. Întrucât m este finit, înmulţind acest w cu o anumită constantă pozitivă obţinem $w \cdot x_i > 1$ pentru $i = 1, \ldots, m$, deci condiţia lui Slater este satisfăcută.

Lagrangeanul generalizat care corespunde problemei primale date este:

$$L_P(w, \alpha) = \frac{1}{2} w \cdot w + \sum_{i=1}^{m} \alpha_i (1 - w \cdot x_i)$$

cu $\alpha_i \geq 0$ pentru $i = 1, \ldots, m$.

Egalând cu 0 derivata parţială a lui L_P în raport cu w, obţinem $w = \sum_{i=1}^{m} \alpha_i x_i$. Substituind această egalitate în expresia lui L_P, vom avea:

$$\begin{aligned} L_D(\alpha) &= \frac{1}{2}\Big(\sum_{i=1}^{m} \alpha_i x_i\Big) \cdot \Big(\sum_{j=1}^{m} \alpha_j x_j\Big) + \sum_{i=1}^{m} \alpha_i \Big(1 - \Big(\sum_{j=1}^{m} \alpha_j x_j\Big) \cdot x_i\Big) \\ &= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j x_i \cdot x_j \end{aligned}$$

minimală (engl., minimum enclosing ball, MEB). Pentru a putea distinge mai uşor cele două versiuni (una de cealaltă), vom numi varianta de aici *Max Margin*.

[720] Vedeţi problemele 22 şi 23, precum şi referinţele bibliografice indicate acolo.

[721] Pentru formalizarea acestei condiţii, vedeţi *Comentariul* de la problema 9 (pag. 658).

Aşadar, forma lui L_D este similară cu cea de la problema SVM cu margine "hard" (vedeţi problema 9.c), iar din cauză că aici nu se foloseşte termen liber problema duală este acum mai simplă: $\max_{\alpha \geq 0} L_D(\alpha)$.

b. În ce priveşte kernel-izarea, este imediat că relativ la antrenare răspunsul este afirmativ, deoarece în expresia lagrangeanului L_D care a fost dedusă la punctul precedent x_i şi x_j apar doar ca [perechi de] factori în produsele scalare. Concret, după „mapare" folosind funcţia Φ corespunzătoare nucleului K, vom avea:

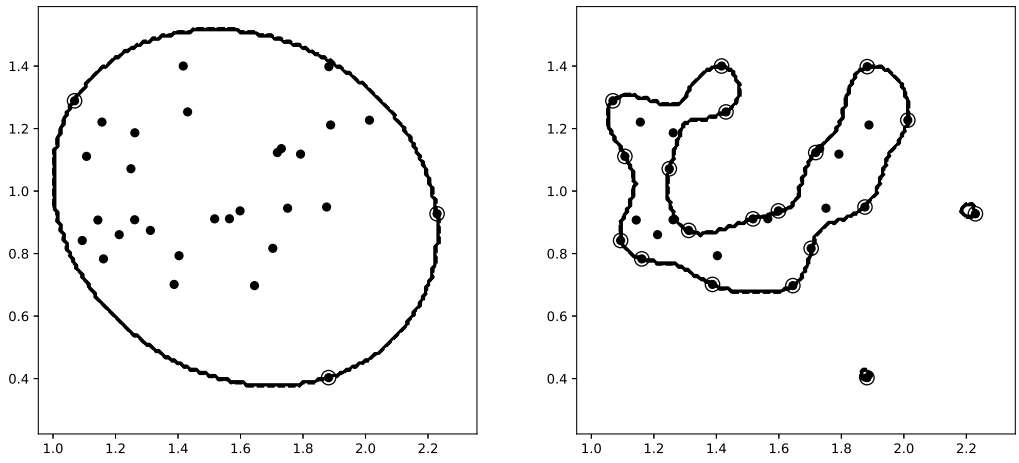
$$\begin{aligned} L_D(\alpha) &= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j \Phi(x_i) \cdot \Phi(x_j) \\ &= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j K(x_i, x_j). \end{aligned}$$

Relativ la testare, răspunsul este de asemenea afirmativ: dată fiind o instanţă de test x', ea va fi clasificată în funcţie de semnul expresiei

- $w \cdot x' = (\sum_{i=1}^{m} \alpha_i x_i) \cdot x' = \sum_{i=1}^{m} \alpha_i x_i \cdot x'$**, în cazul în care nu se face mapare**
- $w \cdot x' = \sum_{i=1}^{m} \alpha_i \Phi(x_i) \cdot \Phi(x') = \sum_{i=1}^{m} \alpha_i K(x_i, x')$**, când se face mapare.**

***Observaţie* (2): Imaginile de mai jos ilustrează rezultatul folosirii a două funcţii-nucleu de tip RBF în conjuncţie cu *SVM one-class* de tip *Max Margin* pe setul de date din enunţ. Pentru rezultatul ilustrat în partea dreaptă, s-a lucrat cu o valoare [mai] mică pentru parametrul σ^2 din definiţia nucleului RBF. Facem menţiunea că aici s-a folosit o formă uşor mai generală pentru problema SVM *one-class* (de tip Max Margin):**[722]

$$\begin{aligned} &\min_{w,\rho} \left(\frac{1}{2} \|w\|^2 - \rho \right) \\ \textbf{a. î.} \quad &w \cdot x_i \geq \rho, \ \textbf{pentru } i = 1, \ldots, m. \end{aligned}$$



[722]LC: Această formă este foarte utilă pentru a înţelege modul în care se defineşte problema de optimizare ν-SVM de la pr. 33. Vedeţi şi pr. 64, unde se elaborează varianta "soft margin" pentru problema *one-class* de tip Max Margin.

c. Întrucât în formularea dată în enunţ pentru problema *one-class SVM* nu se foloseşte termenul liber (engl., "bias") w_0, nu avem în forma duală a problemei de optimizare o restricţie de tipul $\sum_{i=1}^{n} y_i\alpha_i = 0$ (cum este cazul la problemele 22 şi 23). Aşadar, vom folosi tot metoda creşterii pe coordonate (engl., coordinate ascent) ca în algoritmul clasic SMO, însă vom alege la fiecare iteraţie doar (câte) o variabilă Lagrange (α_i).

Din expresia lagrangeanului dual $L_D(\alpha)$ pe care l-am calculat la punctul a, obţinem funcţia pe care trebuie s-o optimizăm la iteraţia curentă:

$$L(\alpha_i) = \alpha_i - \sum_{j\neq i} \alpha_i\alpha_j x_i \cdot x_j - \frac{1}{2}\alpha_i^2 x_i \cdot x_i + const.$$

Punctul de maxim este dat de soluţia derivatei de ordinul întâi a acestei funcţii:

$$\frac{\partial L(\alpha_i)}{\partial \alpha_i} = 0 \quad \Leftrightarrow \quad 1 - \sum_{j\neq i} \alpha_j x_j \cdot x_i - \alpha_i x_i \cdot x_i = 0 \Leftrightarrow \alpha_i^{new,\ unclipped} = \frac{1 - \sum_{j\neq i} \alpha_j x_j \cdot x_i}{x_i \cdot x_i}.$$

Ţinând cont de restricţia $\alpha_i \geq 0$, rezultă că noua valoare pe care o atribuim variabilei alese este $\alpha_i^{new,\ clipped} = \max\left\{0, \frac{1-\sum_{j\neq i} \alpha_j x_j \cdot x_i}{x_i \cdot x_i}\right\}$.

Mai rămân de specificat:

– criteriul de selecţie a variabilei „libere"; de preferinţă aceasta se va face în aşa fel încât să se obţină o creştere cât mai mare a valorii funcţiei obiectiv de la o iteraţie la alta;

– criteriul de oprire a algoritmului; spre exemplu, atunci când creşterea funcţiei obiectiv de la o iteraţie la alta devine nesemnificativă este inutil să mai executăm noi iteraţii.

32. **(O condiţie suficientă pentru ca cele două tipuri de probleme de optimizare *one-class SVM*, şi anume *Max Margin* şi *minimum enclosing ball* (MEB), în varianta kernel-izată, să fie echivalente)**

□ • ○ ***MIT, 2009 fall, Tommi Jaakkola, HW2, pr. 2.a***

La exerciţiul 31 am prezentat o metodă de detecţie a „anomaliilor"' dintr-un set de instanţe numită *one-class SVM*. Metoda respectivă se bazează pe separarea vectorilor de „trăsături" (engl., feature vectors) faţă de originea sistemului de coordonate. Concret, am pornit de la un separator liniar care trece prin origine şi are „margine" (adică distanţă) maximă în raport cu instanţele date. De aceea am numit această metodă problema *one-class Max Margin SVM*. Din punct de vedere matematic, varianta kernel-izată a acestei probleme a fost formulată — ca problemă de optimizare convexă cu restricţii — astfel:[723]

$$\begin{aligned} &\min_{w,\rho} \left(\frac{1}{2}\|w\|^2 - \rho\right) \\ \textbf{a. î.} \quad &w \cdot \phi(x_i) \geq \rho, \ \textbf{pentru}\ i = 1, \ldots, m. \end{aligned} \qquad \textbf{(Max Margin)}$$

[723] Vedeţi *Observaţia* (2) de la pr. 31. Veţi constata că varianta dată iniţial în enunţul acelei probleme este mai simplă; este suficient de simplă pentru introducerea modelului *Max Margin*, însă este prea simplă pentru a o putea folosi în formularea proprietăţii pe care o vom da aici.

unde $x_1, \ldots, x_m$ sunt instanţe de antrenament, iar ϕ este o aşa-numită funcţie de „mapare“ a atributelor.

O *altă modalitate* prin care putem detecta „anomalii“ este să identificăm o hiper-sferă care să includă în sens minimal (engl., minimum enclosing ball, MEB) instanţele date sau, dacă folosim o funcţie de „mapare“, imaginile instanţelor noastre în spaţiul de „trăsături“ corespunzător. Matematic, putem formula această metodă în maniera următoare: date fiind instanţele $x_1, \ldots, x_m$ şi o „mapare“ a trăsăturilor $\phi(x)$, considerăm problema de optimizare

$$\begin{array}{ll} & \min_{R,w} R^2 \\ \textbf{a. î.} & \|w - \phi(x_i)\|^2 \leq R^2, \ \textbf{pentru } i = 1, \ldots, m. \end{array} \qquad \textbf{(MEB)}$$

Vă cerem să demonstraţi că aceste două probleme devin identice în ipoteza că

$$\|\phi(x_i)\| = c \ \textbf{pentru } i = 1, \ldots, m,$$

unde c este o constantă oarecare (reală, pozitivă).[724] Cu alte cuvinte, soluţia optimă $\hat{w}$ obţinută de una (oricare) dintre cele două probleme este soluţie optimă şi pentru cealaltă problemă.

***Sugestie*: Deduceţi forma duală pentru fiecare dintre cele două probleme de optimizare de mai sus (*Max Margin* şi *MEB*). Apoi comparaţi cele două forme duale obţinute, ţinând cont de ipoteza $\|\phi(x_i)\| = c$ pentru c fixat şi $i = 1, \ldots, m$.**

***Observaţie*: Ca o consecinţă directă a proprietăţii demonstrate în această problemă, putem afirma că graficele care au fost obţinute la rezolvarea problemei 31.b (Max Margin) sunt identice cu cele care corespund soluţiilor problemei de optimizare MEB pe datele respective.**

Răspuns:

Concret, strategia de rezolvare este următoarea: Se va constata uşor că atât în cazul problemei *Max Margin* (pentru detecţia anomaliilor) cât şi în cazul problemei sferei de incluziune minimală (MEB), relaţia dintre soluţia formei primale ($\hat{w}$) şi soluţia formei duale ($\hat{\alpha}$) este aceeaşi: $\hat{w} = \sum_{i=1}^{m} \hat{\alpha}_i \phi(x_i)$. Prin urmare, dacă presupunând adevărată relaţia $\|\phi(x_i)\| = c$ pentru $i = 1, \ldots, m$ va rezulta că pentru ambele probleme de optimizare forma duală are exact aceeaşi soluţie $\hat{\alpha}$, atunci este imediat că şi soluţiile formelor primale ale celor două probleme coincid.[725]

Făcând calculele în maniera în care deja ne-am obişnuit, vom obţine pentru

[724] *Observaţie importantă*: Această condiţie este satisfăcută în cazul nucleului RBF, fiindcă $\phi(x)^2 = K(x, x) = \exp\Big(-\frac{\|x - x\|^2}{2\sigma^2}\Big) = 1$, pentru orice x şi orice σ.

[725] Ca şi în cazul problemei *Max Margin* — pentru varianta ei [mai] simplă, am văzut deja justificarea la ex. 31.a —, se constată uşor că pentru problema MEB este satisfăcută condiţia lui Slater.

problema *Max Margin* următoarea formă duală:[726]

$$\max_\alpha \big(-\sum_{i,j=1}^m \alpha_i\alpha_j K(x_i,x_j)\big)$$
$$\textbf{a. î.}\quad \alpha_i \geq 0 \textbf{ pentru } i=1,\ldots,m \textbf{ şi } \sum_{i=1}^m \alpha_i = 1.$$

În mod similar, pentru problema de optimizare MEB, forma duală va fi:[727]

$$\max_\alpha \big(\sum_{i=1}^m \alpha_i K(x_i,x_i) - \sum_{i,j=1}^m \alpha_i\alpha_j K(x_i,x_j)\big)$$
$$\textbf{a. î.}\quad \alpha_i \geq 0 \textbf{ pentru } i=1,\ldots,m \textbf{ şi } \sum_{i=1}^m \alpha_i = 1.$$

Întrucât $K(x_i,x_i)=\phi(x_i)^2=\|\phi(x_i)\|^2=c^2$ şi $\sum_{i=1}^m \alpha_i = 1$, rezultă că prima sumă ($\sum_{i=1}^m \alpha_i K(x_i,x_i) = c^2$) din cadrul funcţiei obiectiv a problemei duale MEB nu are niciun efect asupra soluţiei optime a acestei probleme. În consecinţă, forma duală a problemei MEB este — în cazul $\|\phi(x_i)\| = c$ pentru $i=1,\ldots,m$ — echivalentă cu forma duală a problemei *Max Margin*. Ambele probleme

[726] Pe scurt, se procedează astfel:

$$L_P(w,\rho,\alpha) = \frac{1}{2}w^2 - \rho - \sum_{i=1}^m \alpha_i(w\cdot\phi(x_i)-\rho)$$
$$\frac{\partial}{\partial w}L_P(w,\rho,\alpha)=0 \Leftrightarrow w - \sum_{i=1}^m \alpha_i\phi(x_i)=0 \Leftrightarrow w=\sum_{i=1}^m \alpha_i\phi(x_i)$$
$$\frac{\partial}{\partial \rho}L_P(w,\rho,\alpha)=0 \Leftrightarrow -1+\sum_{i=1}^m \alpha_i = 0 \Leftrightarrow \sum_{i=1}^m \alpha_i = 1$$
$$L_D(\alpha) = \frac{1}{2}\sum_{i,j=1}^m \alpha_i\alpha_j\phi(x_i)\cdot\phi(x_j) - \rho - \sum_{i=1}^m \alpha_i\Big(\big(\sum_{j=1}^m \alpha_j\phi(x_j)\big)\cdot\phi(x_i)-\rho\Big) =$$
$$-\frac{1}{2}\sum_{i,j=1}^m \alpha_i\alpha_j\phi(x_i)\cdot\phi(x_j) - \rho + \rho\underbrace{\sum_{i=1}^m \alpha_i}_{1} = -\frac{1}{2}\sum_{i,j=1}^m \alpha_i\alpha_j\underbrace{\phi(x_i)\cdot\phi(x_j)}_{K(x_i,x_j)} = -\frac{1}{2}\sum_{i,j=1}^m \alpha_i\alpha_j K(x_i,x_j).$$

[727] Din nou, pe scurt,

$$L_P(w,R,\alpha) = R^2 + \sum_{i=1}^m \alpha_i[(w-\phi(x_i))^2 - R^2]$$
$$\frac{\partial}{\partial w}L_P(w,R,\alpha)=0 \Leftrightarrow \sum_{i=1}^m \alpha_i\ 2(w-\phi(x_i))=0 \Leftrightarrow \sum_{i=1}^m \alpha_i w = \sum_{i=1}^m \alpha_i\phi(x_i) \Leftrightarrow w = \frac{\sum_{i=1}^m \alpha_i\phi(x_i)}{\sum_{i=1}^m \alpha_i}$$
$$\frac{\partial}{\partial R}L_P(w,\rho,\alpha)=0 \Leftrightarrow 2R - \sum_{i=1}^m \alpha_i\ 2R = 0 \overset{R\neq 0}{\Leftrightarrow} \sum_{i=1}^m \alpha_i = 1$$
$$\Rightarrow w = \sum_{i=1}^m \alpha_i\phi(x_i) \text{ şi}$$
$$L_D(\alpha) = R^2 + \sum_{i=1}^m \alpha_i\Big[\Big(\underbrace{\sum_{j=1}^m \alpha_j\phi(x_j)}_{w} - \phi(x_i)\Big)^2 - R^2\Big]$$
$$= R^2 - R^2\underbrace{\sum_{i=1}^m \alpha_i}_{1} + w^2\underbrace{\sum_{i=1}^m \alpha_i}_{1} - 2\sum_{i,j=1}^m \alpha_i\alpha_j\underbrace{\phi(x_i)\cdot\phi(x_j)}_{K(x_i,x_j)} + \sum_{i=1}^m \alpha_i\underbrace{\phi(x_i)^2}_{K(x_i,x_i)}$$
$$= \sum_{i,j=1}^m \alpha_i\alpha_j\underbrace{\phi(x_i)\cdot\phi(x_j)}_{K(x_i,x_j)} - 2\sum_{i,j=1}^m \alpha_i\alpha_j K(x_i,x_j) + \sum_{i=1}^m \alpha_i K(x_i,x_i) = -\sum_{i,j=1}^m \alpha_i\alpha_j K(x_i,x_j) + \sum_{i=1}^m \alpha_i K(x_i,x_i).$$

vor produce acelaşi $\hat{\alpha}$ (pentru forma duală), şi deci aceeaşi soluţie optimă $\hat{w}$ (pentru forma primală).

33. **(Problema ν-SVM)**

prelucrare de Liviu Ciortuz, după
■ B. Schölkopf, A. Smola, "Learning with Kernels", MIT Press, 2002, pag. 206-209

Parametrul de „destindere" $C > 0$, din forma problemei SVM cu margine "soft" (introdus de catre Cortes şi Vapnik, în *Support Vector Networks*, 1995),[728] permite realizarea unui compromis între două obiective antagoniste: maximizarea marginii[729] şi minimizarea erorii la antrenare. La valori mari ale lui C rezultă un nivel scăzut al sumei erorilor în raport cu marginea (şi anume, $\sum_{i=1}^{m} \xi_i$, în notaţia uzuală). Invers, la valori mici ale lui C rezultă un nivel ridicat al sumei erorilor. Totuşi, semnificaţia parametrului C este prea puţin intuitivă, iar valoarea sa nu poate fi determinată a priori.

De aceea, în locul acestei variante de SVM cu margine "soft", B. Schölkopf, A. Smola, R. Williamson şi P. Bartlett au propus în articlolul *New Support Vector Machines*[730] o abordare diferită, în care se foloseşte un alt parametru numeric, ν, astfel încât dacă m reprezintă numărul instanţelor de antrenament, atunci νm va limita superior numărul de erori produse la antrenare.[731] Se poate demonstra că νm este totodată o margine inferioară pentru numărul de vectori-suport.[732]

Varianta aceasta este cunoscută sub numele de ν-SVM şi este caracterizată de forma primală următoare:

$$\begin{array}{ll} & \min_{w,w_0,\xi,\rho} \left(\frac{1}{2}\|w\|^2 - \nu\rho + \frac{1}{m}\sum_{i=1}^{m}\xi_i \right) \\ \textbf{a. î.} & y_i(w \cdot x_i + w_0) \geq \rho - \xi_i, \text{ pentru } i = 1, \ldots, m \\ & \xi_i \geq 0 \text{ pentru } i = 1, \ldots, m \\ & \rho \geq 0. \end{array} \qquad (\mathrm{P}'')$$

De remarcat prezenţa variabilei suplimentare ρ, care va trebui să fie supusă procesului de optimizare la fel ca şi variabilele w, w_0 şi $\xi \overset{not.}{=} (\xi_1, \ldots, \xi_m)$.[733]

[728] Vedeţi problema 12.

[729] Definită ca distanţa de la hiperplanul de separare optimală $w \cdot x + w_0 = 0$ până la vectorii-suport x_i pentru care $(w \cdot x_i + w_0)y_i = 1$.

[730] Publicat în revista Neural Computation, 12:1207-1245, 2000. (Vedeţi şi şi *A Tutorial on ν-Support Vector Machines* de Pai-Hsuen Chen, Chih-Jen Lin, şi Bernhard Schoelkopf, în vol. Applied Stochastic Models in Business and Industry, ed. Wiley InterScience, 2005.)

[731] LC: La problema 18 am arătat că în contextul problemei de optimizare C-SVM numărul de erori comise la antrenare este mărginit superior de către suma variabilelor de „destindere" ($\sum_i \xi_i$). Similar, pentru problema (P'') de mai jos se poate arăta imediat, analizând doar(!) forma restricţiilor, că numărul de erori comise la antrenare este aici mărginit superior de $\frac{1}{\rho}\sum_i \xi_i$. Prin urmare, dacă impunem condiţia $\nu m \leq \frac{1}{\rho}\sum_i \xi_i$ rezultă $\nu\rho \leq \frac{1}{m}\sum_i \xi_i$. Aceasta explică forma funcţiei obiectiv din problema (P'') de mai jos.

[732] Vedeţi *Observaţia* de la finalul rezolvării punctului *a* de mai jos.

[733] Din forma restricţiilor liniare rezultă că distanţa de la separatorul optimal la vectorii-suport de pe margine (engl., bound support vectors) — adică acei vectori-suport pentru care multiplicatorii Lagrange corespunzători vor aparţine intervalului $(0, \frac{1}{m})$, conform problemei duale (D'') de mai jos — va fi $\frac{\rho}{||w||}$. (Vedeţi relaţia (321) de la problema 12.)

a. Derivaţi forma dală corespunzătoare problemei ν-SVM. Simplificaţi rezultatul cât mai mult posibil.

b. Stabiliţi relaţiile de legătură între $\bar{w}$, $\bar{w}_0$, $\bar{\rho}$, prin care am notat soluţiile problemei (P$''$), şi soluţiile problemei duale de la punctul a.

c. Care este regula de clasificare a unei instanţe de test oarecare x' în acest model?

Răspuns:

a. Vom urma liniile „metodologice" care au fost folosite la problemele 9 şi 12. Mai întâi, este uşor de verificat faptul că problema de optimizare (P$''$) satisface condiţia lui Slater:[734] **luând $w = 0$, $w_0 = 0$, $\rho = 0$ şi $\xi_i = 1$, restricţiile $y_i(w \cdot x_i + w_0) > \rho - \xi_i$ sunt îndeplinite, pentru $i = 1, \ldots, m$. Prin urmare, vom putea opta ca în loc să rezolvăm problema (P$''$) să rezolvăm duala ei, după care vom obţine soluţia problemei (P$''$) folosind condiţiile Karush-Kuhn-Tucker de optimalitate / staţionaritate şi respectiv complementaritate.**

Apoi, lagrangeanul generalizat pentru problema (P$''$) este

$$\begin{aligned} L_P(w, w_0, \xi, \rho, \alpha, \beta, \delta) &= \frac{1}{2}||w||^2 - \nu\rho + \frac{1}{m}\sum_{i=1}^{m}\xi_i \\ &\quad - \sum_{i=1}^{m}\alpha_i(y_i(w \cdot x_i + w_0) - \rho + \xi_i) - \sum_{i=1}^{m}\beta_i\xi_i - \delta\rho, \end{aligned}$$

unde α_i, β_i şi δ sunt multiplicatori Lagrange (corespunzători celor trei tipuri de inegalităţi din problema (P$''$)), toţi trebuind să satisfacă restricţia de nenegativitate.

Calculând derivatele parţiale ale lui L_P în raport cu variabilele primale w, w_0, ξ şi ρ şi apoi egalându-le cu 0, vom obţine imediat:

$$\begin{aligned} & w = \textstyle\sum_{i=1}^{m}\alpha_i y_i x_i \\ & \textstyle\sum_{i=1}^{m}\alpha_i y_i = 0 \\ & \alpha_i + \beta_i = \frac{1}{m} \text{ pentru } i = 1, \ldots, m \\ & \textstyle\sum_{i=1}^{m}\alpha_i - \delta = \nu. \end{aligned}$$

Din relaţia $\alpha_i + \beta_i = \frac{1}{m}$, ţinând cont că $\alpha_i \geq 0$ şi $\beta_i \geq 0$, rezultă $\alpha_i \in \left[0, \frac{1}{m}\right]$ şi $\beta_i \in \left[0, \frac{1}{m}\right]$ pentru $1, \ldots, m$. De asemenea, ştiind că $\delta \geq 0$, din relaţia $\sum_{i=1}^{m}\alpha_i - \delta = \nu$ rezultă că $\sum_{i=1}^{m}\alpha_i \geq \nu$.

Condiţiile de complementaritate Karush-Kuhn-Tucker sunt: $\alpha_i(y_i(w \cdot x_i + w_0) - \rho + \xi_i) = 0$, $\beta_i\xi_i = 0$ şi $\delta\rho = 0$.

Substituind $w = \sum_{i=1}^{m}\alpha_i y_i x_i$ în expresia lui L_P şi apoi făcând diversele simplificări posibile, va rezulta că forma dală corespunzătoare problemei (P$''$) este:

[734] Pentru formalizarea acestei condiţii, vedeţi *Comentariul* de la problema 9 (pag. 658).

$$\begin{array}{ll} & \max_\alpha \Big(-\frac{1}{2}\sum_{i,j}\alpha_i\alpha_j y_i y_j x_i \cdot x_j \Big) \\ \textbf{a. î.} & 0 \le \alpha_i \le \frac{1}{m} \textbf{ pentru } i = 1, \ldots, m \\ & \sum_{i=1}^m \alpha_i y_i = 0 \\ & \sum_{i=1}^m \alpha_i \ge \nu. \end{array} \qquad (\mathbf{D''})$$

De remarcat că

– multiplicatorii Lagrange β_i şi δ nu apar în (D″);

– în funcţia obiectiv a problemei duale (D″) nu apare termenul $\sum_{i=1}^m \alpha_i$, care era prezent în funcţia obiectiv a problemei duale atât pentru SVM cu margine "hard" cât şi pentru C-SVM;

– în schimb, în partea de restricţii apare condiţia suplimentară $\sum_{i=1}^m \alpha_i \ge \nu$.

***Observaţie*: Din relaţia $0 \le \alpha_i \le \frac{1}{m}$ rezultă imediat că $\frac{\#SV}{m} \ge \sum_{i=1}^m \alpha_i \ge \nu$, deci $\#SV \ge \nu m$, unde prin $\#SV$ am notat numărul vectorilor-suport.**

b. În ce priveşte legătura dintre soluţiile formei primale (P″) şi cele ale formei duale (D″), avem mai întâi $\bar{w} = \sum_{i=1}^m \bar{\alpha}_i y_i x_i$. Apoi, din condiţiile de complementaritate Karush-Kuhn-Tucker deducem că dacă există un $\bar{\alpha}_i$ astfel încât $0 < \bar{\alpha}_i < \frac{1}{m}$, atunci $\bar{\beta}_i > 0$ şi deci $\bar{\xi}_i = 0$. De asemenea, $\bar{\alpha}_i > 0$ implică $y_i(\bar{w} \cdot x_i + \bar{w}_0) - \bar{\rho} + \bar{\xi}_i = 0$, de unde rezultă $y_i(\bar{w} \cdot x_i + \bar{w}_0) = \bar{\rho}$.

Folosind această ultimă relaţie, valorile optime $\bar{w}_0$ şi $\bar{\rho}$ se vor determina astfel: dacă avem x_+ o instanţă pozitivă şi x_- o instanţă negativă astfel încât multiplicatorii Lagrange corespunzători sunt în intervalul $\big(0, \frac{1}{m}\big)$,[735] atunci $\bar{w} \cdot x_+ + \bar{w}_0 = \bar{\rho}$ şi $\bar{w} \cdot x_- + \bar{w}_0 = -\bar{\rho}$. Aceste ultime două ecuaţii formează un sistem din care se obţin imediat $\bar{w}_0$ şi $\bar{\rho}$:[736]

$$\begin{aligned} \bar{w}_0 &= -\frac{1}{2}\bar{w} \cdot (x_+ + x_-) \\ \bar{\rho} &= \frac{1}{2}\bar{w} \cdot (x_+ - x_-) \end{aligned}$$

***Observaţie*: În articolul despre ν-SVM citat în enunţ, autorii extind acest procedeu de calcul pentru valorile $\bar{w}_0$ şi $\bar{\rho}$ la mai multe perechi de instanţe pozitive şi respectiv negative, pentru a obţine un rezultat cât mai robust.**

c. Dată fiind o instanţă nouă (de test) x', ea va fi clasificată conform expresiei

$$sign\Big(\sum_{i=1}^m \bar{\alpha}_i y_i x_i \cdot x' + \bar{w}_0\Big).$$

[735] De remarcat că dacă există una din cele două instanţe, atunci în mod necesar există şi cea de-a doua, fiindcă $\sum_i \bar{\alpha}_i y_i = 0$.

[736] Dacă toţi $\bar{\alpha}_i$ sunt 0, atunci $\bar{w} = 0$, ceea ce este în afara discuţiei, după cum am justificat la problema 9. Rămâne de tratat cazul în care pentru orice $i \in \{1, \ldots, m\}$ avem fie $\bar{\alpha}_i = 0$ fie $\bar{\alpha}_i = \frac{1}{m}$, ştiind că există cel puţin un $\bar{\alpha}_i$ care are valoarea $\frac{1}{m}$, fiindcă $\sum_{i=1}^m \bar{\alpha}_i \ge \nu$. Acest caz se tratează similar cu cazul corespunzător de la problema 12.

34. **(SVR — Regresie cu vectori-suport, varianta "hard-margin", adică, fără variabile de „destindere")**

prelucrare de Liviu Ciortuz, după
■ • ○ *Stanford, 2014 fall, Andrew Ng, midterm, pr. 4*

Până acum, am văzut cum anume putem face clasificare cu algoritmul [C-]SVM. În acest exerciţiu vom studia o variantă a acestui algoritm, care serveşte pentru a face regresie. Aşadar, etichetele asociate aici instanţelor de antrenament vor avea valori continue, $y \in \mathbb{R}$. În mod natural, noul algoritm poartă numele de Regresie cu Vectori-Suport (engl., Support Vector Regression, SVR).[737]

Presupunem că avem un set de date de antrenament $\{(x_1, y_1), \ldots, (x_m, y_m)\}$, cu $x_i \in \mathbb{R}^{n+1}$ şi $y_i \in \mathbb{R}$ pentru $i = 1, \ldots, m$. Urmărind să găsim o ipoteză [de regresie] de forma $h_{w,b}(x) = w \cdot x + b$, vom scrie următoarea problemă de optimizare (convexă):

$$\min_{w,b} \frac{1}{2}\|w\|^2$$

$$\text{a. î. } y_i - (w \cdot x_i + b) \leq \varepsilon \text{ pentru } i = 1, \ldots, m \tag{344}$$

$$(w \cdot x_i + b) - y_i \leq \varepsilon \text{ pentru } i = 1, \ldots, m, \tag{345}$$

unde $\varepsilon > 0$ are o valoare dată, fixată.

***Observaţii*:**

1. Inegalităţile (344) şi (345) se pot scrie în mod combinat sub forma

$$-\varepsilon \leq y_i - (w \cdot x_i + b) \leq \varepsilon \Leftrightarrow |y_i - (w \cdot x_i + b)| \leq \varepsilon.$$

Semnificaţia analitică a acestei inegalităţi este imediată. Aşadar, restricţia asupra *marginii funcţionale* de la problema de optimizare SVM a fost modificată aici în aşa fel încât să se refere la diferenţa dintre valorile reale ale lui y şi outputul produs de ipoteza pe care urmărim să o învăţăm folosind SVR.

2. În mod similar cu problema [C-]SVM, în noua problemă de optimizare se cere ca valoarea lui $\|w\|$ să fie [cât mai] mică. Minimizarea lui $\|w\|$ corespunde maximizării distanţei ($\varepsilon/\|w\|$) dintre hiperplanul reprezentat de ipoteza $w \cdot x + b$ şi marginile $w \cdot x + b = \varepsilon$ şi $w \cdot x + b = -\varepsilon$ între care trebuie să se situeze toate valorile y_i.[738]

a. Scrieţi lagrangeanul corespunzător problemei de optimizare (SVR) de mai sus. Vă recomandăm să folosiţi două seturi de variabile / multiplicatori Lagrange, α_i şi α_i^*, cu $i = 1, \ldots, m$, care corespunde restricţiilor reprezentate de cele două inegalităţi (care au fost etichetate cu (344) şi (345)). Prin urmare, funcţia lagrangeană va fi de forma $L(w, b, \alpha, \alpha^*)$.

b. Obţineţi forma duală a problemei de optimizare. (Va trebui să calculaţi derivatele parţiale ale lagrangeanului în raport cu w şi respectiv cu b.)

c. Demonstraţi că acest algoritm poate fi kernel-izat. Pentru aceasta, va trebui să arătaţi că (i) funcţia obiectiv pentru forma duală a problemei de

[737] Puteţi consulta articolul *A tutorial on support vector regression* de Alex Smola şi Bernhard Schoelkopf, publicat în revista Statistics and Computing, 14:199–222, 2004.

[738] De asemenea, minimizarea lui $\|w\|$ corespunde obiectivului de a reduce overfitting-ul, aşa cum ştim şi de la celelalte metode de regresie studiate.

optimizare poate fi scrisă folosind (doar) produse scalare între instanţele de antrenament, şi (*ii*) la faza de testare, dată fiind o instanţă nouă x, ipoteza $h_{w,b}(x)$ poate fi calculată folosind de asemenea (doar) produse scalare.

Răspuns:

a. Fie $\alpha_i, \alpha_i^* \geq 0$ $(i = 1, \ldots, m)$ multiplicatorii Lagrange pentru restricţiile (344) **şi respectiv** (345)**. Atunci, lagrangeanul poate fi scris astfel:**

$$L(w, b, \alpha, \alpha^*) = \frac{1}{2}\|w\|^2 + \sum_{i=1}^{m} \alpha_i(y_i - w \cdot x_i - b - \varepsilon) + \sum_{i=1}^{m} \alpha_i^*(-y_i + w \cdot x_i + b - \varepsilon).$$

b. Mai întâi, precizăm că funcţia obiectiv pentru forma duală a problemei de optimizare SVR este definită astfel:

$$L_D(\alpha, \alpha^*) = \min_{w,b} L(w, b, \alpha, \alpha^*).$$

Acum, calculând derivatele lagrangeanului în raport cu variabilele primale şi egalându-le apoi cu 0**, vom avea:**

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)x_i = 0 \;\Rightarrow\; w = \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)x_i \tag{346}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{m}(\alpha_i^* - \alpha_i) = 0. \tag{347}$$

Substituind aceste două relaţii în $L(w, b, \alpha, \alpha^*)$, vom obţine:

$$\begin{aligned} L_D(\alpha, \alpha^*) &= \frac{1}{2}\|w\|^2 + \sum_{i=1}^{m} \alpha_i(y_i - w \cdot x_i - b - \varepsilon) + \sum_{i=1}^{m} \alpha_i^*(-y_i + w \cdot x_i + b - \varepsilon) \\ &= \frac{1}{2}\|w\|^2 - \varepsilon\sum_{i=1}^{m}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i^*) \\ &\quad - \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)w \cdot x_i - b\underbrace{\sum_{i=1}^{m}(\alpha_i - \alpha_i^*)}_{0} \\ &\overset{(346)}{=} \frac{1}{2}\|\sum_{i=1}^{m}(\alpha_i - \alpha_i^*)x_i\|^2 - \varepsilon\sum_{i=1}^{m}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i^*) \\ &\quad - \sum_{i=1}^{m}(\alpha_i - \alpha_i^*)\sum_{j=1}^{m}(\alpha_j - \alpha_j^*)x_j \cdot x_i \\ &= -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)x_i \cdot x_j - \varepsilon\sum_{i=1}^{m}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i^*). \end{aligned}$$

Aşadar, problema duală SVR poate fi formulată astfel:

$$\max_{\alpha,\alpha^*} \Big(-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)\, x_i \cdot x_j - \varepsilon\sum_{i=1}^{m}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{m} y_i(\alpha_i - \alpha_i^*)\Big)$$

a. î. $\sum_{i=1}^{m}(\alpha_i - \alpha_i^*) = 0$

$\alpha_i, \alpha_i^* \geq 0$ **pentru** $i = 1, \ldots, m$.

c. Se observă că în expresia lagrangeanului dual L_D instanţele de antrenament apar întotdeauna în produse scalare de forma $x_i \cdot x_j$. În mod similar, atunci când vrem să facem predicţie pentru o instanţă oarecare x, vom avea:

$$w \cdot x + b \overset{(346)}{=} \sum_{i=1}^{m} (\alpha_i - \alpha_i^*)\, x_i \cdot x + b.$$

Aşadar, algoritmul SVR poate fi kernel-izat.

5.2 Maşini cu vectori-suport — Probleme propuse

5.2.1 SVM cu margine "hard"

35. **(Calcularea distanţei de la un hiperplan la originea sistemului de coordonate; rezolvare cu metoda vectorială şi cu metoda multiplicatorilor lui Lagrange)**

prelucrare de Liviu Ciortuz, după
□ • ○ ***CMU, 2018 spring, Nina Balcan, HW0, pr. "Geometry"***

Considerăm hiperplanul de ecuaţie $w \cdot x + b = 0$, unde w şi x sunt din $\mathbb{R}^d$, iar $b \in \mathbb{R}$.

Demonstraţi că distanţa geometrică [deci fără semn] de la originea sistemului de coordonate până la [cel mai apropiat punct de pe] hiperplanul de ecuaţie $w \cdot x + b = 0$ este $\frac{|b|}{\|w\|}$.

Veţi face demonstraţia atât în manieră vectorială[739] cât şi cu ajutorul metodei multiplicatorilor lui Lagrange.[740]

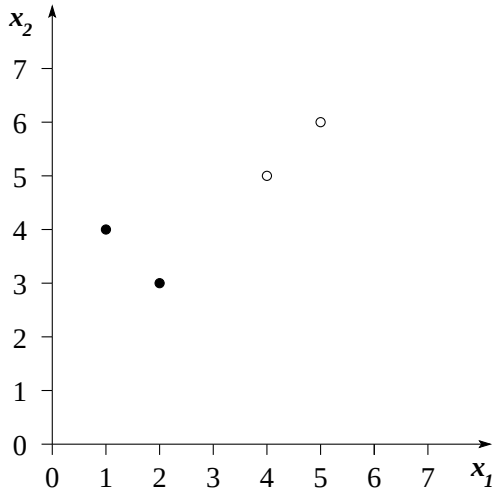
36. **(SVM liniară: aplicare pe date din $\mathbb{R}^2$)**

· ***CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, final exam, pr. 7.b***

Presupunem că antrenaţi o maşină cu vectori-suport pe setul de date din figura alăturată. Acest set de date constă din două exemple cu eticheta +1 (aceste exemple sunt marcate cu semnul •) şi două exemple cu eticheta -1 (marcate cu ○).

a. Care este ecuaţia corespunzătoare separatorului optimal?

b. Trasaţi separatorul optimal şi încercuiţi vectorii-suport.



[739] Puteţi prelua ideea rezolvării de la problema 1, însă vă cerem să nu preluaţi pur şi simplu formula care a fost obţinută acolo şi s-o aplicaţi la cazul de faţă. Dacă doriţi să lucraţi în *cazul particular* $d = 2$, puteţi folosi mijloace obişnuite de analiză matematică, în speţă proprietăţile funcţiei polinomiale de gradul al doilea.

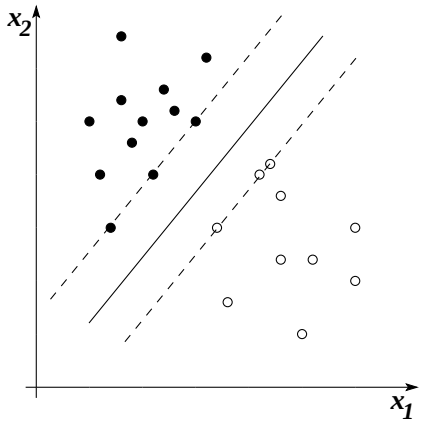
[740] Pentru o scurtă introducere la metoda multiplicatorilor lui Lagrange, vedeţi *Introducerea* de la problema 82, *Comentariul* de la problema 83, precum şi problemele 85 şi 176 din secţiunea *Metode de optimizare* de la capitolul de *Fundamente*.

Observaţie: În problema de optimizare care corespunde formulării din enunţ nu avem restricţii de tip inegalitate, deci veţi putea rezolva uşor problema de optimizare în *forma* sa *primală*, făcând apel [nu neapărat în mod explicit] la condiţiile Karush-Kuhn-Tucker.

37. **(Separabilitate în $\mathbb{R}^2$; SVM liniară; calculul erorii la CVLOO)**

$*$ ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, final exam, pr. 3.f***

Care este eroarea la cross-validare cu metoda "Leave-One-Out" atunci când folosim o maşină cu vectori-suport ca în figura următoare?



38. **(SVM liniară în $\mathbb{R}^2$, forma primală: determinarea hiperplanului de separare optimală)**

• · ***CMU, 2009 spring, Ziv Bar-Joseph, HW3, pr. 4.1***

Considerăm că în $\mathbb{R}^2$ sunt date două puncte: (x_1, y_1) cu eticheta $+1$ şi (x_2, y_2) cu eticheta -1. Care este separatorul decizional pe care îl vom obţine rulând o SVM liniară pe setul de date de antrenament compus din aceste două puncte? Calculaţi expresia analitică a acestui separator.

39. **(SVM — problema de optimizare în forma primală: familiarizare cu noţiunile de margine şi separator optimal în spaţiul de „trăsături")**

• · ***CMU, 2004 fall, T. Mitchell, Z. Bar-Jospeh, HW4, pr. 4.5***
MIT, 2004 fall, Tommi Jaakkola, HW3, pr. 1.3

Considerăm un set de date din $\mathbb{R}$, foarte simplu, constând din numai două exemple de antrenament:

$$(x_1 = 0, y_1 = -1) \text{ şi } (x_2 = \sqrt{2}, y_2 = 1).$$

Vom folosi o funcţie-nucleu polinomială de ordinul al doilea, mai precis vom pune în corespondenţă fiecare instanţă x cu vectorul

$$\Phi(x) = (1, \sqrt{2}x, x^2).$$

Vrem să găsim *soluţia* $\hat{w} = (\hat{w}_1, \hat{w}_2, \hat{w}_3)$ şi $\hat{w}_0$ a problemei de optimizare SVM [în aşa-numitul spaţiu de „trăsături" determinat de transformarea Φ]:

$$\min_{w,w_0} \frac{1}{2}\|w\|^2$$
$$\textbf{astfel încât } y_1(w \cdot \Phi(x_1) + w_0) \geq 1$$
$$y_2(w \cdot \Phi(x_2) + w_0) \geq 1.$$

a. Folosind ceea ce ştiţi despre [graniţa de] separare cu margine maximală, indicaţi un vector care are aceeaşi direcţie cu vectorul-soluţie $\hat{w}$.[741]

b. Cât este valoarea *marginii de separare* — definită în sensul distanţei de la separatorul optimal până la fiecare vector-suport — pe care o obţinem pentru aceste date (evident, în spaţiul de „trăsături" indicat mai sus)?

c. Făcând legătura dintre marginea de separare şi $\|\hat{w}\|$, indicaţi soluţia $\hat{w}$ şi deduceţi valoarea lui $\hat{w}_0$.

40. **(Neseparabilitate liniară: două exemple de mapare a trăsăturilor; rezolvarea — în manieră directă — a problemei SVM în spaţiul / spaţiile de trăsături)**

• ***CMU, 2011 fall, Eric Xing, HW4, pr. 3.1.1-7***

Se dau 6 puncte din $\mathbb{R}$: $x_1 = -1$, $x_2 = 0$, $x_3 = 1$ au etichete negative, iar $x_4 = -2$, $x_5 = 2$, $x_6 = 3$ au etichete pozitive.

a. Desenaţi cele 6 puncte pe axa reală, folosind simbolul $\circ$ pentru a reprezenta etichetele negative şi simbolul $\bullet$ pentru etichetele pozitive.

b. Aceste 6 puncte nu sunt liniar-separabile. Definiţi $f : \mathbb{R} \to \mathbb{R}$, o funcţie de transformare a trăsăturilor astfel încât punctele $f(x_1)$, $f(x_2)$, ..., $f(x_6)$ să fie liniar-separabile. Desenaţi pe axa reală cele 6 instanţe astfel obţinute. Apoi indicaţi poziţia separatorului optimal calculat de maşina cu vectori-suport liniară cu margine "hard", marcând instanţele care sunt vectori-suport.

c. Separatorul de la punctul precedent are forma analitică $w_0 + w_1 f(x) = 0$. Indicaţi valorile lui w_0 şi w_1.

d. Să presupunem acum că mapăm cele 6 puncte în spaţiul de trăsături $(x, f(x))$, unde $f(x)$ este funcţia de transformare a trăsăturilor de la punctul b. Cu alte cuvinte, acum vom avea 6 puncte în plan, $(x_1, f(x_1))$, $(x_2, f(x_2))$, ..., $(x_6, f(x_6))$. Desenaţi aceste 6 puncte în planul bidimensional, împreună cu separatorul optimal definit de către maşina cu vectori-suport liniară cu margine "hard". Indicaţi apoi vectorii-suport.

e. Separatorul de la punctul precedent are forma analitică $w_0 + w_1 x + w_2 f(x) = 0$. Calculaţi valorile parametrilor w_0, w_1 şi w_2.

[741] Veţi ţine cont că
1. într-un spaţiu de tip $\mathbb{R}^d$, egalitatea $w \cdot x = 0$ implică faptul că vectorii w şi x sunt ortogonali / perpendiculari;
2. ecuaţia unui hiperplan din $\mathbb{R}^d$ este de forma $w \cdot x + w_0 = 0$, ceea ce implică faptul că vectorul w este perpendicular pe acest hiperplan (întrucât este perpendicular pe orice vector din hiperplan, văzând un astfel de vector ca diferenţă de doi vectori de poziţie $x_1 - x_2$, unde punctele x_1 şi x_2 aparţin hiperplanului).
Aşadar, vectorul $\hat{w}$ din problema noastră este perpendicular pe hiperplanul de separare optimală.

f. Funcţia de mapare a trăsăturilor $x \longrightarrow (x, f(x))$ de la punctele d şi e este asociată cu o funcţie-nucleu $K(x, x')$, unde x şi x' sunt puncte din spaţiul original (de trăsături) unidimensional. Scrieţi expresia acestei funcţii-nucleu.

41. **(Găsirea separatorului liniar optimal, după maparea într-un „spaţiu de trăsături", conform unei funcţii-nucleu date)**

$* \circ$ ***CMU, 2008 spring, Eric Xing, midterm exam, pr. 5***
MIT, 2006 fall, Tommi Jaakkola, midterm exam, pr. 4

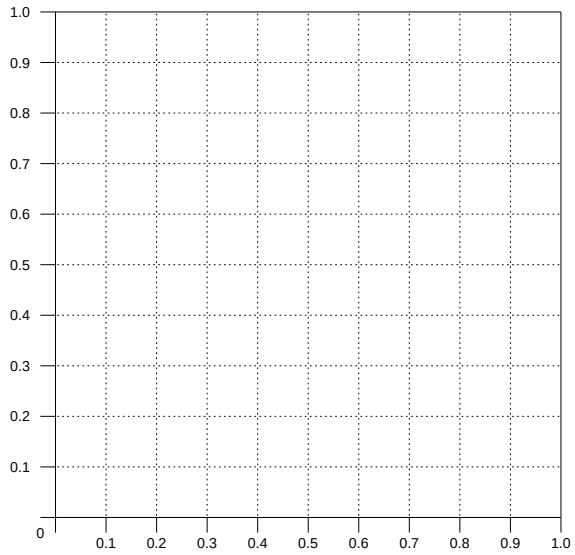
Presupunem că avem instanţele pozitive $x_1 = (0.4,\ 0.2)$, $x_2 = (0.8,\ 0.4)$, $x_3 = (0.2,\ 0.4)$, $x_4 = (0.4,\ 0.8)$ şi instanţele negative $x_5 = (0.4,\ 0.4)$ şi $x_6 = (0.8,\ 0.8)$. Se poate constata uşor că aceste date nu sunt separabile liniar.

Vom folosi o mapare Φ astfel încât

$$\Phi(x) \cdot \Phi(x') \stackrel{not.}{=} K(x, x') = \frac{x \cdot x'}{|| x || \, || x' ||}.$$

a. Care este vectorul $\Phi(x)$ corespunzător acestei funcţii-nucleu?

b. Pe gridul alăturat, reprezentaţi atât instanţele de antrenament x_i cât şi „imaginile" $\Phi(x_i)$, pentru $i = 1, \ldots, 6$. Fiecare „imagine" $\Phi(x_i)$ va avea asociată aceeaşi etichetă ca şi x_i. Veţi folosi *convenţia* noastră de notare: simbolul $\bullet$ desemnează instanţe pozitive, iar simbolul $\circ$ instanţe negative.



***Sugestie*: În loc să faceţi efectiv calcularea coordonatelor pentru „imaginile" $\Phi(x_i)$, ar fi mai util / interesant să vedeţi cât este $\|\Phi(x_i)\|$ pentru fiecare x_i, iar apoi să trageţi o *concluzie generală* cu privire la poziţia relativă a lui $\phi(x_i)$ în raport cu x_i, pentru orice i.**

c. Veţi observa că instanţele $\Phi(x_i)$ sunt liniar-separabile, deci există o dreaptă determinată de parametrii w_1, w_2 şi w_0, care le separă în mod optimal. Cât este raportul w_1/w_2?

***Indicaţie*: Nu este necesar să calculaţi efectiv w_1 şi w_2. Este suficient să exploataţi simetriile.**

d. Pe desenul de la punctul b, încercuiţi vectorii-suport $\Phi(x_i)$.

e. Indicaţi în spaţiul de origine zonele de decizie [şi separatorul decizional] care corespund separatorului optimal găsit la punctul c.

42. **(Învăţarea conceptului ¬XOR, folosind forma duală a problemei SVM şi o funcţie-nucleu polinomială de ordin 2)**

prelucrare de Liviu Ciortuz, după
∗ ***CMU, 2006 fall, E. Xing, T. Mitchell, midterm exam, pr. 5***

Fie o problemă de învăţare supervizată în care exemplele de antrenament se află în spaţiul euclidian bidimensional. Exemplele pozitive sunt $x_1 = (1, 1)$ şi $x_3 = (-1, -1)$, iar exemplele negative sunt $x_2 = (-1, 1)$ şi $x_4 = (1, -1)$.

Considerăm transformarea $\Phi(x)$ corespunzătoare funcţiei-nucleu $K(x, x') = (x \cdot x' + 1)^2$, unde x şi x' sunt din $\mathbb{R}^2$.

Funcţia de predicţie pe care vrem s-o obţinem este de forma $y(x) = w \cdot \Phi(x) + w_0$, unde $\Phi(x), w \in \mathbb{R}^n$ (cu n ales convenabil), $w_0 \in \mathbb{R}$, iar $\cdot$ reprezintă produsul scalar.

Indicaţi coeficienţii w, w_0 corespunzători suprafeţei de separare cu margine maximă pentru punctele de antrenament date. Cum va fi clasificată o instanţă oarecare de test $x \in \mathbb{R}^2$?

***Sugestie de lucru*:**

a. Mai întâi reprezentaţi datele. Determinaţi $\Phi(x)$, n, şi $\Phi(x_1)$, $\Phi(x_2)$, $\Phi(x_3)$, $\Phi(x_4)$.
b. Calculaţi apoi lagrangeanul dual

$$L_D(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j y_i y_j \alpha_i \alpha_j \Phi(x_i) \cdot \Phi(x_j)$$

c. Determinaţi soluţia problemei duale $\bar{\alpha} = \text{argmax}_\alpha L_D(\alpha)$ cu restricţiile $\alpha_i \geq 0$ pentru $i = 1, \ldots, 4$ şi $\sum_i \alpha_i y_i = 0$ şi, în final, găsiţi soluţia problemei primale $\bar{w} = \sum_i \bar{\alpha}_i y_i \Phi(x_i)$ şi $\bar{w}_0$, ştiind că $\bar{w}_0$ poate fi obţinut dintr-una din restricţiile $y_i(\bar{w} \cdot \Phi(x_i) + \bar{w}_0) = 1$.

43. **(SVM cu diferite funcţii-nucleu: efectul unei translatări a datelor de antrenament asupra poziţiei separatorului optimal)**

CMU, 2009 fall, Carlos Guestrin, HW3, pr. 2.10

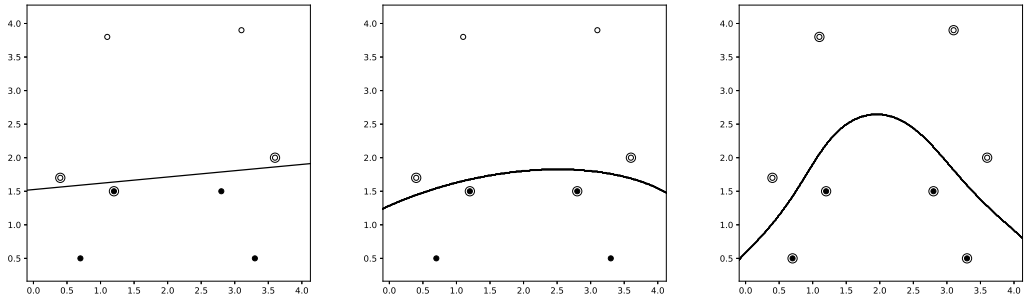
Trei maşini cu vectori-suport au fost antrenate pe un set de date din $\mathbb{R}^2$ folosind

a. **o funcţie-nucleu liniară $K(x, y) = x \cdot y$ (a se vedea figura a);**
b. **o funcţie-nucleu polinomială de gradul al doilea $K(x, y) = (x \cdot y + 1)^2$ (a se vedea figura b);**
c. **o funcţie-nucleu cu baza radială $K(x, y) = e^{-\frac{\|x-y\|^2}{2\sigma^2}}$ (a se vedea figura c).**

Presupunem că translatăm datele adăugând o constantă mare (de exemplu 10) la coordonata de pe axa verticală, pentru fiecare din datele de antrenament, adică (x, y) devine $(x, y + 10)$.

Dacă reantrenăm SVM-urile de mai sus pe noile date de antrenament, se schimbă şi poziţia separatorului optimal în raport cu datele? Trataţi pe rând cazurile a, b şi c.

Explicaţi pe scurt de ce se schimbă (sau de ce nu se schimbă) poziţia separatorului optimal în raport cu datele, pentru fiecare din cazurile a, b şi c. Trasaţi pe desen poziţia noului separator optimal, acolo unde este cazul.[742]



a. **b.** **c.**

44. (SVM cu nucleu RBF — o caracteristică surprinzătoare: instanţe foarte distanţate faţă de separatorul optimal pot fi vectori-suport)

◦ ***MIT, 2002 fall, Tommi Jaakkola, midterm exam, pr. 3.1-3***
CMU, 2011 spring, Tom Mitchell, HW6, pr. 1.2

[***Remember*:**] Folosind o funcţie-nucleu oarecare, SVM caută într-un anumit „spaţiu de trăsături" Q un hiperplan care să maximizeze distanţa dintre cele două clase. Clasificarea unei instanţe oarecare de test x se face determinând semnul expresiei

$$\bar{w} \cdot \Phi(x) + \bar{w}_0 = \Big(\sum_{i \in SV} y_i \bar{\alpha}_i \Phi(x_i) \Big) \cdot \Phi(x) + \bar{w}_0 = \sum_{i \in SV} y_i \bar{\alpha}_i K(x_i, x) + \bar{w}_0 \overset{not.}{=} f(x; \bar{\alpha}, \bar{w}_0),$$

unde

$\bar{w}$ şi $\bar{w}_0$ sunt parametrii pentru hiperplanul de clasificare în spaţiul Q,

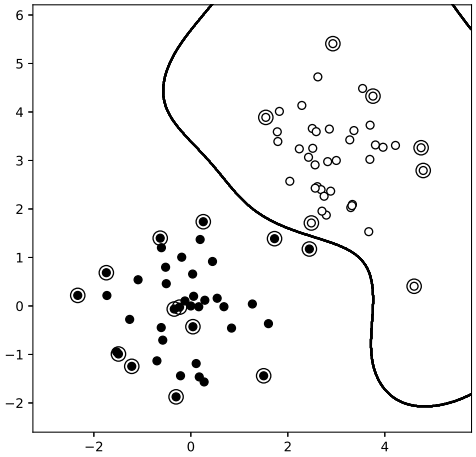
SV este setul de vectori-suport,

$\bar{\alpha}_i$ este coeficientul (multiplicatorul Lagrange) corespunzător vectorului-suport i.

În acest exerciţiu vom folosi drept nucleu funcţia cu baza radială $K(x_i, x_j) = e^{-\frac{1}{2}||x_i - x_j||^2}$. Vom presupune că instanţele de antrenament sunt liniar separabile în spaţiul Q.

[742] *Sugestie*: Puteţi consulta articolul *On In variance of Support Vector Machines*, de Shigeo Abe, http://www2.kobe-u.ac.jp/ abe/pdf/ideal2003.pdf accesat la data 12.06.2018. LC: Mulţumesc lui Sebastian Ciobanu pentru această sugestie.

***Observaţie*: Figura alăturată prezintă grafic rezultatul obţinut de SVM folosind nucleu RBF pe un set de date din planul euclidian. Vectorii-suport sunt încercuiţi. Ceea ce este curios la această clasificare este că unii vectori-suport sunt destul de departe de separatorul optimal. Şi, totuşi, ei sunt vectori-suport! Întrebările de mai jos vor încerca să contribuie la elucidarea acestei chestiuni.**



a. Arătaţi că ori de câte ori alegem un punct de test $x_{\mathbf{far}} \in \mathbb{R}^d$ foarte distanţat de orice instanţă de antrenament x_i, rezultă că $f(x_{\mathbf{far}}; \bar{\alpha}, \bar{w}_0) \approx \bar{w}_0$.

În continuare, vom presupune, pentru simplitate, că $\bar{w}_0 = 0$.

b. Din ipoteză, întrucât setul de date de antrenament este liniar separabil în spaţiul de trăsături Q, rezultă — în urma antrenării — că orice instanţă de antrenament x_i satisface inegalitatea $y_i \bar{w} \cdot \Phi(x_i) \geq 1$, unde Φ este funcţia de mapare corespunzătoare nucleului RBF specificat mai sus. Satisface şi $x_{\mathbf{far}}$ această inegalitate?

***Observaţie*: Pentru RBF, $\Phi(x)$ este un vector infinit, însă puteţi răspunde la această întrebare ţinând cont că $\bar{w} \cdot \Phi(x) = \sum_{i \in \boldsymbol{SV}} y_i \bar{\alpha}_i K(x_i, x)$, iar valorile funcţiei-nucleu $K(x_i, x)$ pot fi calculate fără nicio dificultate.**

c. Dacă am include punctul $x_{\mathbf{far}}$ în mulţimea de date de antrenament, iar aceasta ramâne liniar separabilă în spaţiul de trăsături Q (în raport cu acelaşi nucleu RBF), ar deveni oare $x_{\mathbf{far}}$ un vector-suport?

45. **(Comparaţie între SVM şi alţi clasificatori)**

∗ ∘ ***CMU, 2010 fall, Aarti Singh, midterm, pr. 1.2.1***

Fie setul de date din $\mathbb{R}^2$ din tabelul alăturat. Identificaţi dintre clasificatorii următori pe aceia care obţin eroare de antrenare 0 pe acest set de date.

X_1	X_2	Y
0	0	+
1	0	−
0	1	−
1	1	+

- **SVM cu nucleu polinomial de ordin 2, adică $(c + x \cdot x')^2$, în care metaparametrul c este la libera alegere;**
- **regresia logistică**
- **arbori ID3 de adâncime 2, adică având două nivele de test, dintre care unul este nivelul-rădăcină;**
- **3-NN.**

46. **(O legătură (simplă) între reţele neuronale şi SVM cu funcţie-nucleu polinomială)**

prelucrare de Liviu Ciortuz, după
• ***CMU, 2010 fall, Aarti Singh, HW5, pr. 4.2***

Se consideră datele de antrenament $\bar{x}_1, \ldots, \bar{x}_m \in \mathbb{R}^d$, etichetate respectiv cu $y_1, \ldots, y_m \in \{-1, +1\}$. Se consideră de asemenea o funcţie-nucleu polinomială de gradul $p \in \mathbb{N}$

$$K(\bar{x}, \bar{x}') = (\gamma \bar{x} \cdot \bar{x}' + r)^p = \Phi(\bar{x}) \cdot \Phi(\bar{x}')$$

unde γ şi r sunt numere reale [LC: pozitive].

Ştim că o SVM care foloseşte această funcţie-nucleu şi care a fost antrenată pe datele de mai sus va clasifica o instanţă de test $\bar{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ conform expresiei:

$$sign(w \cdot \Phi(\bar{x}) + w_0) = sign\Big(\sum_{i=1}^{m} y_i \alpha_i \Phi(\bar{x}) \cdot \Phi(\bar{x}_i) + w_0\Big) = sign\Big(\sum_{i=1}^{m} y_i \alpha_i K(\bar{x}, \bar{x}_i) + w_0\Big),$$

unde $w \in \mathbb{R}^n$ (n fiind dimensiunea spaţiului de trăsături asociat cu „mapa-rea“ Φ) şi $w_0 \in \mathbb{R}$ desemnează ponderile învăţate de SVM, iar α_i sunt valorile multiplicatorilor Lagrange (corespunzători instanţelor de antrenament) care constituie soluţiile formei duale a problemei SVM.

Se consideră o reţea neuronală artificială de tip feed-forward cu un singur nivel ascuns, descrisă astfel:

– intrările în reţea sunt $x_1, \ldots, x_d$ (a se vedea $\bar{x}$ de mai sus);
– reţeaua are m unităţi ascunse, şi anume câte una pentru fiecare instanţă de antrenament $\bar{x}_i = (x_{i1}, \ldots, x_{id})$, cu $i \in \{1, \ldots, m\}$;
– ponderile de pe conexiunile dintre intrări şi unităţile ascunse sunt fixate astfel:

- **pentru unitatea ascunsă i, ponderile de la intrările $x_1, \ldots, x_d$ sunt respectiv $x_{i1}, \ldots, x_{id}$ (a se vedea $\bar{x}_i$ de mai sus); aceasta are loc pentru fiecare $i \in \{1, \ldots, m\}$;**
- **nu se foloseşte termen liber / constant ($x_0 = 1$) pentru aceste unităţi ascunse;**

– ieşirea unităţii ascunse i (cu $i \in \{1, \ldots, m\}$) este definită într-o manieră nestandard, şi anume:

$$o_i = (\gamma\ net_i + r)^p, \text{ unde } net_i \text{ este } \sum_{j=1}^{d} x_j\, x_{ij}$$

– reţeaua are o singură unitate pe stratul de ieşire, pe care o vom nota cu $m + 1$; ea are funcţia de activare de tip $sign$;
– ponderile hidden-output sunt de forma $y_i \alpha_i$ pentru fiecare $i \in \{1, \ldots, m\}$;
– pentru unitatea de ieşire, ponderea intrării libere ($x_0 = 1$) este w_0.

a. Să se deseneze această reţea neuronală sub forma unui graf în care intrările şi unităţile neuronale sunt noduri, iar ponderile sunt marcate corespunzător pe arce.

b. Arătaţi că aceşti doi clasificatori — reţeaua neuronală şi SVM-ul de mai sus — sunt echivalenţi în ce priveşte faza de testare / generalizare. Adică,

pentru fiecare instanţă de test $\bar{x} \in \mathbb{R}^d$ rezultatele produse de către cei doi clasificatori (văzuţi simplu ca funcţii de clasificare) sunt identice. (Calculaţi funcţia reprezentată de outputul reţelei neuronale.)

c. O reţea neuronală similară cu cea de mai sus poate chiar să înveţe parametrii γ şi r. Indicaţi ce modificări trebuie făcute în reţea şi, corespunzător, în algoritmul (general) de retro-propagare, pentru a învăţa aceşti doi parametri.

5.2.2 SVM cu margine "soft"

47. **(C-SVM cu separator prin originea sistemului de coordonate: exemplificarea noţiunilor de bază)**

□ • ○ ***CMU, 2017 fall, Nina Balcan, midterm, pr. 1.3***

Fie următoarea afirmaţie / „propoziţie" (în sens matematic):

Atunci când *nu* folosim termen liber (engl., bias), problema de optimizare C-SVM are

***forma primală*:**

$$\min_{w,\xi} \left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^m \xi_i\right)$$
$$\textbf{a. î.} \quad y_i\, w \cdot x_i \geq 1 - \xi_i \ \textbf{şi} \ \xi_i \geq 0 \ (i = 1, \ldots, m)$$

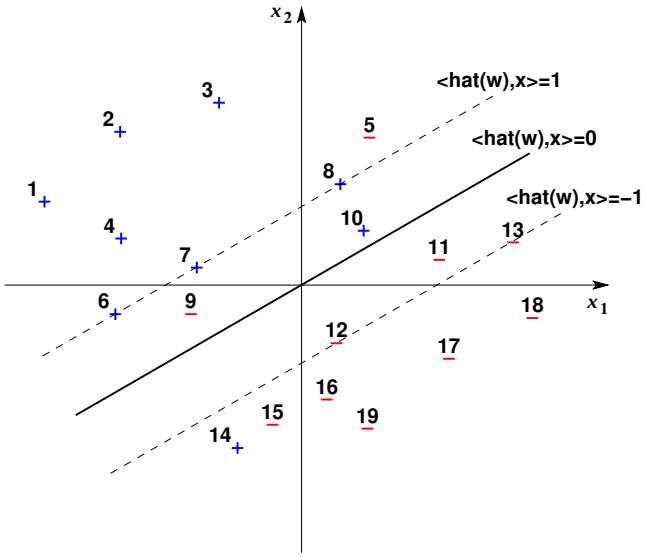
***forma duală*:**

$$\max_{\alpha \in \mathbb{R}^m} \left(\sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i=1}^m \sum_{j=1}^m y_i y_j \alpha_i \alpha_j\, x_i \cdot x_j\right)$$
$$\textbf{a. î.} \quad 0 \leq \alpha_i \leq C \ (i = 1, \ldots, m).$$

Notăm cu $\hat{w}, \hat{\xi}$ soluţia [optimă a] problemei primale şi cu $\hat{\alpha}$ soluţia [optimă a] problemei duale. (Vă readucem aminte că $\hat{\beta}_i \overset{not.}{=} C - \hat{\alpha}_i \geq 0$ este multiplicatorul Lagrange corespunzător inegalităţii $\xi_i \geq 0$.)

a. Demonstraţi că într-adevăr forma primală a problemei de optimizare C-SVM din enunţ conduce la forma duală care a fost specificată. Puneţi în evidenţă diferenţele faţă de forma duală de la pr. 12.

În figura alăturată vi se dă un set de date de antrenament din $\mathbb{R}^2$, precum şi separatorul optimal învăţat de către clasificatorul C-SVM pe acest set de date. În această figură, *notaţia* hat(w) corespunde lui $\hat{w}$, iar <hat(w),x> desemnează produsul scalar $\hat{w} \cdot x$. Numărul marcat în dreptul fiecărei instanţe reprezintă indexul (sau, ID-ul) respectivului punct.



b. Indicaţi toate instanţele pentru care $\xi_i \neq 0$.

c. Indicaţi toţi vectorii-suport. Vă readucem aminte că instanţa x_i este vector-suport dacă $\hat{\alpha}_i \neq 0$.

d. Considerăm următoarea propoziţie: pentru punctele cu indicii 10 şi 14 avem $\hat{\alpha}_i = C$. Adevărat sau Fals? Justificaţi.

48. **(SVM vs. C-SVM: comparaţie între soluţiile celor două probleme pe seturi de date liniar separabile)**

□ • ○ ***CMU, 2017 fall, Nina Balcan, HW4, pr. 4.Q11***

Fie $S = \{(x_1, y_1), \ldots (x_n, y_n)\}$ o mulţime de instanţe din $\mathbb{R}^d$, cu etichetele luând valori în mulţimea $\{1, -1\}$. Presupunem că exemplele din S sunt liniar separabile şi că rulăm C-SVM cu $C > 0$ pe setul de date S.

a. Este oare separatorul decizional obţinut de către clasificatorul C-SVM identic cu separatorul de margine maximă (adică, separatorul obţinut de către SVM)? Justificaţi.

b. Este oare adevărată afirmaţia că separatorul decizional obţinut de către clasificatorul C-SVM separă întotdeauna în mod corect cele două clase (adică, obţine eroare la antrenare 0)? Justificaţi.

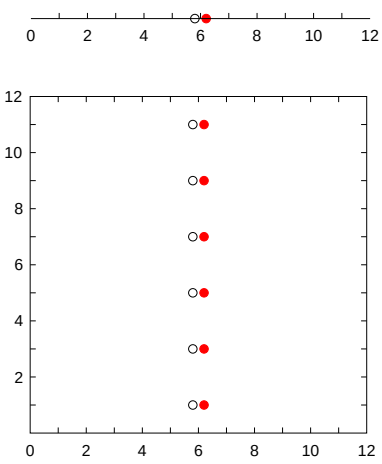
49. **(SVM şi C-SVM: efectul adăugării unui atribut irelevant (i.e., care nu măreşte marginea de separare) asupra vectorului de ponderi w)**

□ • ○ ***CMU, 2017 fall, Nina Balcan, midterm, pr. 1.4***
CMU, 2007 spring, Carlos Guestrin, midterm exam, pr. 6.3

a. Cazul SVM:[743]

[743] Se subînţelege, SVM liniară, cu margine "hard".

Fie $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ un set de date din $\mathbb{R}^d$ care sunt liniar separabile.[744] Presupunem că adăugăm la fiecare instanţă x_i un atribut $\tilde{x}_i \in \mathbb{R}$,[745] care nu conduce la mărirea marginii [geometrice, de separare a] lui S. În acest context, spunem că atributul $\tilde{x}$ este *irelevant*. Va ignora oare SVM în mod automat acest atribut (adică, în vectorul soluţie $\hat{w}$, componenta corespunzătoare noului atribut va fi 0)? Justificaţi răspunsul în mod riguros.



***Sugestie*: Pentru demonstraţie, veţi putea folosi *proprietatea* menţionată la *Observaţia* de la pr. 2.a.**

b. Cazul C-SVM:[746]

Presupunem că rulăm o C-SVM pe [un dataset S cu] atributele $X_1, \ldots, X_d$, iar apoi adăugăm [la instanţele din S] un atribut X_{d+1} care este *irelevant*, în sensul că [presupunând că păstrăm aceeaşi valoare pentru parametrul C] el nu poate creşte marginea de separare. Va ignora oare C-SVM în mod automat acest atribut? Justificaţi răspunsul în mod riguros.

50. **(C-SVM cu termeni de „destindere“ în norma L_2)**

∗ ***University of Utah, 2008 fall, Hal Daumé III, HW5, pr. 3***
CMU, 2010 fall, Aarti Singh, midterm exam, pr. 5.1

La curs am arătat că atunci când datele noastre de antrenament nu sunt liniar separabile, putem modifica forma problemei SVM introducând în fiecare restricţie câte o variabilă de „destindere“ (engl., slack variable) în raport cu marginea.[747] Mai precis, formularea pe care am dat-o este cunoscută sub denumirea *C-SVM de normă* L_1.

Acum vom considera o variantă cunoscută sub numele de *C-SVM de normă* L_2. Se pleacă de la următoarea problemă de optimizare:

$$\min_{w, w_0, \xi} \quad \left(\frac{1}{2}||w||^2 + \frac{C}{2}\sum_{i=1}^{m}\xi_i^2\right)$$
$$\text{a.î.} \qquad y_i(w \cdot x_i + w_0) \geq 1 - \xi_i, i = 1, \ldots, m.$$

Remarcaţi faptul că variabilele de „destindere“ ξ_i apar la puterea a doua în funcţia obiectiv, cu scopul de a limita posibilitatea ca valorile lor să fie excesiv de mari.[748]

[744] LC: În enunţul original se precizează — în paranteze! — că separarea se [poate] face cu un separator care trece prin originea sistemului de coordonate. Restricţia aceasta nu este neapărat necesară, deoarece [veţi constata că] demonstraţia poate fi făcută în cazul general al separabilităţii liniare.

[745] Se poate observa imediat că adăugarea acestui atribut nu afectează proprietatea de separabilitate liniară a datelor de antrenament S.

[746] Adică, SVM liniară cu margine “soft”.

[747] Am prezentat această chestiune la problema 12.

[748] Pentru convenienţa calculelor, în locul lui C — factorul care precede suma $\sum_i \xi_i$ în problema 12 —, aici lucrăm cu $\frac{C}{2}$ ca factor pentru $\sum_i \xi_i^2$. A se vedea calculele de la punctele *c* şi *d*.

a. În forma problemei de mai sus, nu apare condiţia $\xi_i \geq 0$. Arătaţi că aceste restricţii de nenegativitate pot fi într-adevăr eliminate. Aşadar, vă cerem să demonstraţi că valoarea optimă a funcţiei obiectiv va fi aceeaşi, indiferent dacă aceste restricţii ($\xi_i \geq 0$) sunt sau nu incluse în formularea problemei.

***Indicaţie*: Puteţi proceda prin reducere la absurd. Dacă ar exista o soluţie optimă $(\bar{w}, \bar{w}_0, \bar{\xi})$ a problemei de optimizare C-SVM de normă L_2 astfel încât pentru un $i_0 \in \{1, \ldots, m\}$ să avem $\xi_{i_0} < 0$, atunci se poate arăta (în mod concret / constructiv) că există o soluţie mai bună decât soluţia optimă pe care am considerat-o mai sus.**

b. Care este *funcţia lagrangeană generalizată* (notaţie: $L_P(w, w_0, \xi, \alpha)$) asociată problemei de optimizare C-SVM de normă L_2?

c. Deduceţi condiţiile de staţionaritate / optimalitate Karush-Kuhn-Tucker pentru această problemă de optimizare convexă, calculând mai întâi derivatele parţiale $\frac{\partial L_P}{\partial w}$, $\frac{\partial L_P}{\partial w_0}$ şi $\frac{\partial L_P}{\partial \xi}$, şi egalându-le apoi cu 0. Am folosit notaţia $\xi = (\xi_1, \xi_2, \ldots, \xi_m)$.

d. Care este *forma duală* a problemei de optimizare C-SVM de normă L_2? Prin ce diferă această formă duală de forma duală a problemei C-SVM (vedeţi pr. 12)?

51. **(C-SVM folosind funcţie de cost *hinge*: exemplu de aplicare (adică, rezolvarea problemei de optimizare corespunzătoare); comparaţie cu regresia logistică, relativ la efectul outlier-elor asupra poziţiei separatorului optimal)**

□ • · ***CMU, 2012 fall, E. Xing, A. Singh, HW2, pr. 4.3***

La problema 20 am arătat că putem reformula problema de optimizare C-SVM liniară (adică, nekernel-izată) astfel încât să minimizăm suma costurilor *hinge* pe setul de date de antrenament, cu termen de regularizare $\|w\|^2$:

$$\min_{w, w_0} \Big(\|w\|^2 + C \sum_i Loss_{SVM}(f(x_i), y_i) \Big),$$

unde (x_i, y_i) sunt instanţe etichetate, cu $y_i \in \{-1, +1\}$, iar costurile *hinge* sunt definite în modul următor:

$$Loss_{SVM}(f(x_i), y_i) \stackrel{def.}{=} \max(1 - (w \cdot x_i + w_0) y_i, 0).$$

Vom considera setul de date $(x_1, y_1), \ldots, (x_n, y_n)$ cu $n = 2000000$, unde pentru $i = 1, \ldots, n/2$ avem $x_i = 0$ şi $y_i = -1$, iar pentru $i = n/2 + 1, \ldots, n$ avem $x_i = 2$ şi $y_i = +1$. Cu alte cuvinte, se dau un milion de cópii ale instanţei 0 (în spaţiul unidimensional, desigur), toate etichetate cu -1, şi un milion de cópii ale instanţei 2, toate etichetate cu $+1$.

a. Găsiţi valorile lui w şi w_0 care minimizează funcţia obiectiv a problemei de optimizare C-SVM de mai sus pe aceste date, atunci când se lucrează cu $C = 1$.
Care este graniţa de decizie şi cât este „marginea“ corespunzătoare? Cum va

diferi răspunsul dacă vom considera valori din ce în ce mai mici pentru C, tinzând la 0?

b. Acum vom presupune că pe lângă cele $n = 2000000$ de instanţe de mai sus ni se mai dă încă o instanţă:

$$x_{n+1} = 100,\ y_{n+1} = -1.$$

Care sunt noile valori optimale pentru w şi w_0 (folosind din nou $C = 1$)?

c. Considerând că la punctul b în locul algoritmului C-SVM folosim regresia logistică, precizaţi (raţionând în mod intuitiv) care vor fi noile rezultate.

52. **(Algoritmul SMO pentru C-SVM: o restricţie la alegerea variabilelor libere, suficientă ca algoritmul să nu poată îmbunătăţi soluţia (α) fixată iniţial)**

● ○ ∗ ***MIT, 2008 fall, Tommi Jaakkola, midterm exam, pr. 1.2***

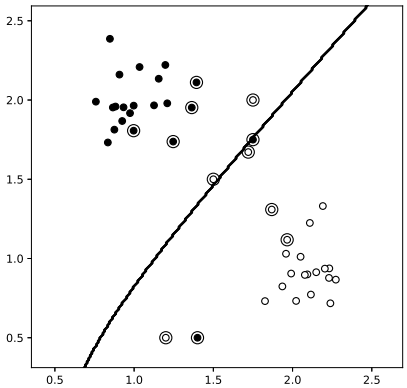
Considerăm algoritmul SMO pentru antrenarea unui C-SVM (adică, SVM cu variabile de „destindere"). Presupunem că iniţial toate variabilele duale α_i ($i = 1, \ldots, m$) sunt setate la valoarea 0. Apoi, la fiecare iteraţie a algoritmului alegem două variabile α_i şi α_j impunând spre deosebire de [sau: pe lângă criteriul de selecţie din] forma clasică a algoritmului SMO un criteriu simplu: cele două variabile trebuie să satisfacă restricţia $y_i = y_j$. După selectare, optimizarea celor două variabile se face ca în algoritmul clasic SMO.

Ce soluţie va calcula această versiune a algoritmului SMO? Justificaţi răspunsul.

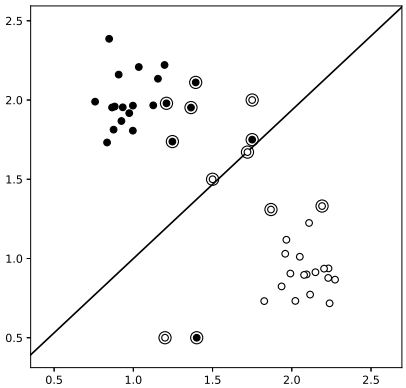
53. **(C-SVM: efectul alegerii valorii parametrului C şi / sau a funcţiei-nucleu)**

○ ***CMU, 2012 spring, Ziv Bar-Joseph, HW3, pr. 3.1***

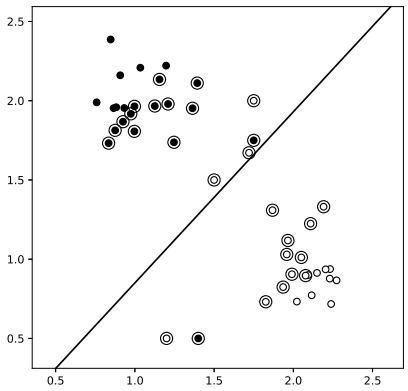
În figura de mai jos sunt ilustrate suprafeţele de decizie pentru patru maşini cu vectori-suport cu margine "soft" (adică, C-SVM-uri), care folosesc diferite funcţii-nucleu şi diferite valori pentru parametrul C pentru „destindere".



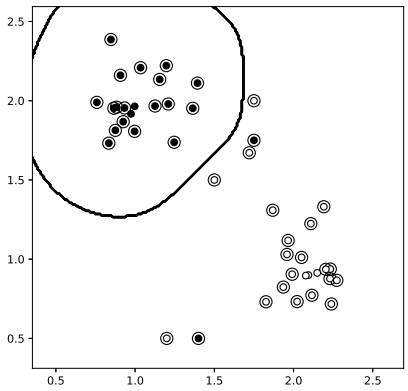
A.



B.



C.



D.

Pentru fiecare dintre aceste exemple / grafice, specificaţi ce setare — dintre cele de mai jos — credeţi că a fost folosită. Justificaţi!

a. $C = 1$ **şi nicio funcţie-nucleu;**

b. $C = 0.1$ **şi nicio funcţie-nucleu;**

c. $C = 0.1$ **şi funcţia-nucleu** $K(x_i, x_j) = e^{-10\|x_i - x_j\|^2}$**;**

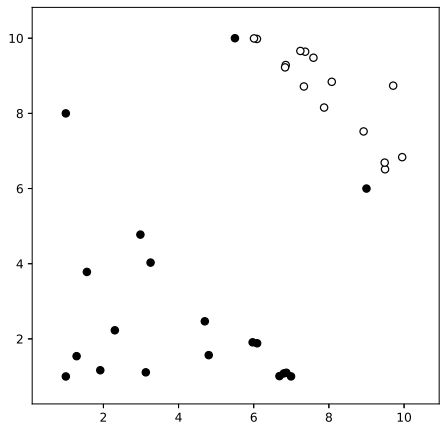
d. $C = 0.1$ **şi funcţia-nucleu** $K(x_i, x_j) = x_i \cdot x_j + (x_i \cdot x_j)^2$**.**

54. **(C-SVM cu funcţie-nucleu pătratică: determinarea graniţei de decizie în funcţie de valoarea parametrului pentru „destindere“ C)**

◦ ***CMU, 2007 spring, Carlos Guestrin, midterm exam, pr. 2***

Obiectivul acestei probleme este să clasificăm corect date de test, pornind de la un anumit set de date de antrenament.

Vom presupune că antrenăm o C-SVM, folosind ca funcţie-nucleu o funcţie polinomială de gradul al doilea. Vi se dă setul de date de antrenament din figura alăturată.



***Avertisment*: Se consideră că datele de antrenament provin de la nişte senzori care pot fi afectaţi de „zgomote“ / perturbaţii, deci ar trebui să evitaţi să vă încredeţi prea mult în vreun punct anume.**

Penalizarea de „destindere“ (engl., slack penalty) C va determina situarea hiperplanului de separare optimală.

Vi se cere să răspundeţi la întrebările de mai jos în manieră *calitativă*. Pentru fiecare dintre aceste puncte, formulaţi răspunsul sub forma unei singure fraze. Veţi reprezenta grafic soluţiile folosind în ordinea corespunzătoare figurile de la sfârşitul acestui enunţ.

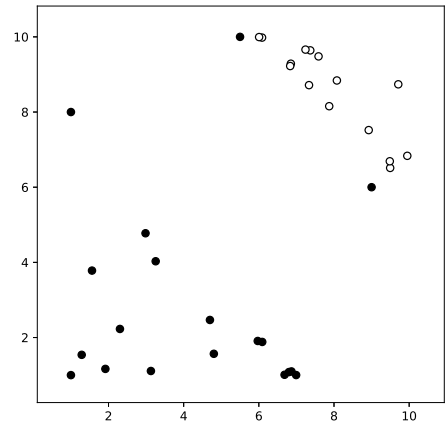
a. Unde va fi situată graniţa de separare atunci când parametrul C ia valori mari (adică $C \to \infty$)? Desenaţi răspunsul în figura a.

b. Pentru $C \approx 0$, indicaţi în figura b unde anume credeţi că va fi situată graniţa de decizie. Justificaţi răspunsul.

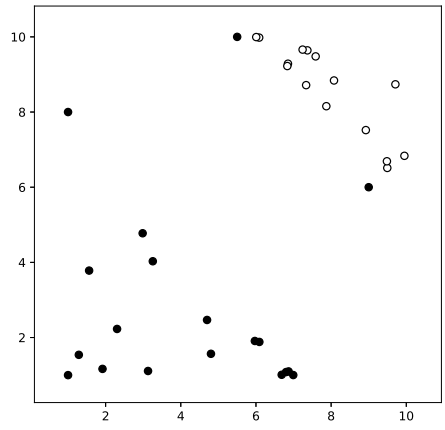
c. Care dintre cele două cazuri de mai sus credeţi că este mai adecvat pentru task-ul de generalizare / predicţie? De ce?

d. Desenaţi în figura d un punct care nu va schimba graniţa de decizie care a fost învăţată pentru valori foarte mari ale lui C. Justificaţi răspunsul.

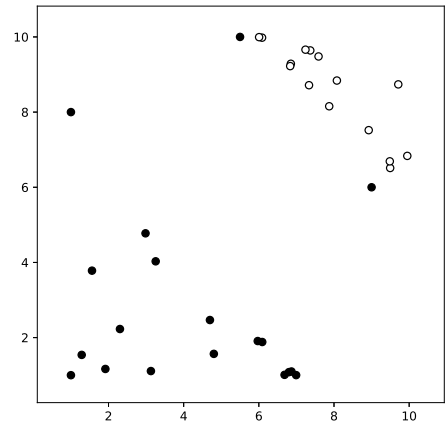
e. Desenaţi în figura e un punct care va schimba în mod considerabil graniţa de decizie care a fost învăţată pentru valori foarte mari ale lui C. Justificaţi răspunsul.



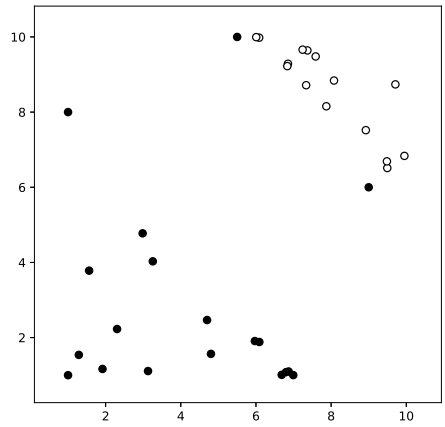
a.



b.



d.



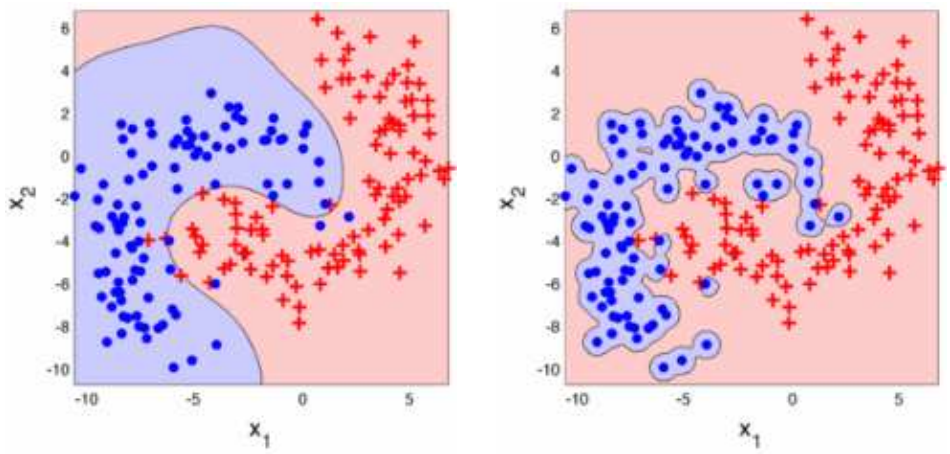
e.

55. (C-SVM: identificare de corespondenţe [cu valorile hiper-parametrilor] pe diagrame)

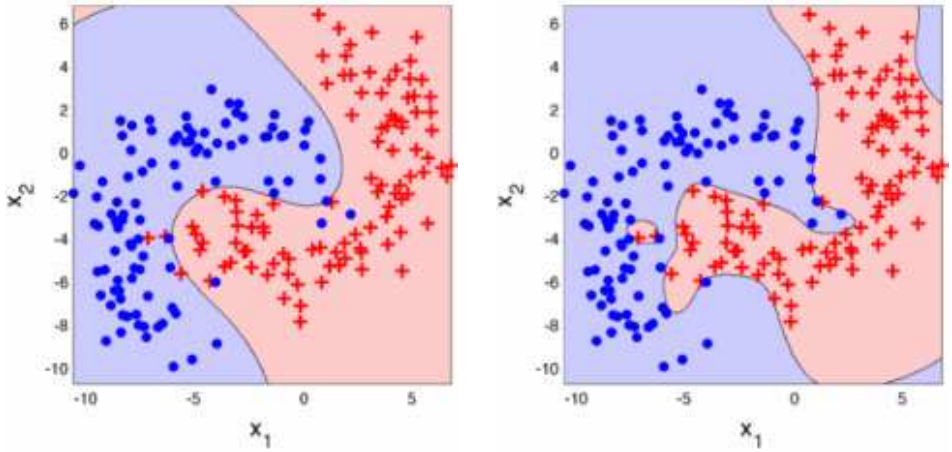
● ○ ***CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, midterm, pr. 7.BC***

a. Vă readucem aminte că nucleul RBF este definit prin formula $K(x, y) = \exp(-\gamma\|x - y\|^2)$. În figurile următoare am marcat graniţele de decizie determinate de două maşini cu vectori-suport cu margine "soft" (C-SVM) cu nucleu RBF aplicate pe setul de date Banana, folosind două valori diferite pentru

constanta γ.[749] Indicaţi care anume dintre cele două figuri corespunde valorii mai mari atribuite parametrului γ.



b. Am rulat de data aceasta două C-SVM-uri cu nucleu RBF pe setul de date Banana, însă acum am atribuit valori diferite parametrului de regularizare, C.[750] Indicaţi care anume dintre cele două figuri corespunde regularizării mai puternice, adică valorii mai mici atribuite parametrului C.



56. **(C-SVM cu funcţii-nucleu: *complexitatea* computaţională la antrenare şi respectiv la testare)**

□ • ○ ***CMU, 2014 fall, E. Xing, B. Poczos, HW2, pr. 3.6***

Presupunem că avem m exemple de antrenament $(x_i, y_i)_{i=1}^m$ din $\mathbb{R}^d \times \{-1, +1\}$ şi că dorim să antrenăm un C-SVM (adică SVM cu margine "soft") care foloseşte funcţie-nucleu. În multe cazuri practice, numărul m este foarte mare (de ordinul sutelor de milioane).

[749] Se presupune că s-a folosit aceeaşi valoare pentru C, parametrul de regularizare.
[750] Se presupune că s-a folosit aceeaşi valoare pentru γ, parametrul asociat nucleului RBF.

a. Cât este *complexitatea de spaţiu* dacă implementăm C-SVM cu funcţie nucleu în *manieră naivă*? Veţi da răspunsul presupunând că $m \gg d$.

b. Cât este costul calculării *funcţiei de decizie* pentru o instanţă oarecare x, dacă presupunem că există $\mathcal{O}(m)$ vectori-suport, iar complexitatea de timp pentru calculul valorii funcţiei-nucleu $k(x, x')$ este $\mathcal{O}(d)$?

c. Există mai multe modalităţi de a aproxima *funcţia de decizie*. Citiţi secţiunea de introducere din articolul *Fastfood – approximating kernel expansions in loglinear time*, de Quoc Le, Tamás Sarlós şi Alex Smola[751] şi scrieţi cât este complexitatea computaţională a funcţiei de decizie în respectiva abordare.

57. (Perceptronul Rosenblatt stochastic şi [C-]SVM: aplicare pe un dataset din $\mathbb{R}^2$; calcularea costului *hinge*)

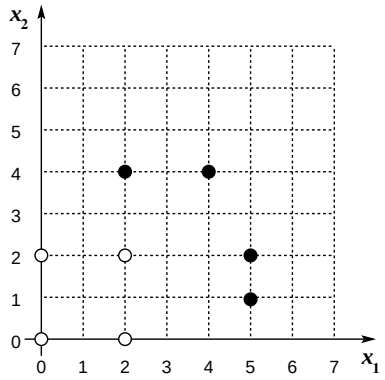
prelucrare de Liviu Ciortuz, după
• · MIT, 2018 spring, Tommi Jaakkola, midterm review, pr. 1

A. Fie setul de date de antrenament care este prezentat în tabelul şi în figura de mai jos:[752]

eticheta	-1	-1	-1	-1	$+1$	$+1$	$+1$	$+1$
coordonatele	$(0,0)$	$(2,0)$	$(0,2)$	$(2,2)$	$(5,1)$	$(5,2)$	$(2,4)$	$(4,4)$
nr. de greşeli făcute de Perceptronul Rosenblatt	1	6	1	5	3	1	2	0

Algoritmul Perceptron stochastic

initialize $w \leftarrow \bar{0}$, $w_0 \leftarrow 0$
for t=1,. . . ,T
  do
% or: do until a certain stopping condition is met
    get the example (x_t, y_t)
    if $y_t\,(w \cdot x_t + w_0) \leq 0$ then
      $w \leftarrow w + y_t x_t$;
      $w_0 \leftarrow w_0 + y_t$
    end if
  end do
end for



a. Rulăm Perceptronul Rosenblatt stochastic pe acest dataset, parcurgând exemplele de antrenament într-o ordine aleatoare până când condiţia de oprire este satisfăcută.[753] **Numărul de greşeli făcute de Perceptron pe fiecare dintre aceste exemple este menţionat pe ultima linie din tabel. Cât va fi valoarea rezultată pentru termenul liber (engl., offset parameter) w_0?**

[751] Proceedings of the 30th International Conference on Machine Learning (ICML), 2013, pp 244–252.
[752] Vă reamintim *convenţia* noastră de notare: simbolul • desemnează instanţe pozitive, iar simbolul ○ instanţe negative.
[753] Pentru o prezentare a algoritmului Perceptron [neciclic, nestochastic], vedeţi problema 16 de la capitolul *Reţele neuronale artificiale*, iar versiunea kernelizată a acelui algoritm este prezentată la problema 19 de la acelaşi capitol.

b. Greşelile făcute de Perceptron depind de ordinea în care sunt parcurse exemplele de antrenament. Este oare posibil ca instanţa $(4, 4)$, etichetată cu $'+1'$, să fi fost primul exemplu procesat de către Perceptron?

B. Presupunem acum că în loc să rulăm algoritmul Perceptron determinăm separatorul liniar care maximizează *marginea geometrică*.

c. Cât sunt valorile parametrilor $\bar{w}$ şi $\bar{w}_0$ corespunzătoare separatorului optimal?

d. Cât este valoarea marginii geometrice?

e. Cât este suma costurilor de tip *hinge* (engl., hinge losses) determinate de aceste exemple?

***Indicaţie* (1): Vă readucem aminte că funcţia de cost *hinge* este definită astfel: $hinge(z) = \max(0, 1 - z)$, iar la calculul costului hinge determinat de un exemplu oarecare (x_i, y_i) în raport cu separatorul optimal $\bar{w} \cdot x + \bar{w}_0 = 0$, acest z va fi înlocuit cu $y_i(\bar{w} \cdot x_i + \bar{w}_0)$.**

f. Presupunem că modificăm soluţia corespunzătoare marginii geometrice maxime, împărţind atât $\bar{w}$ cât şi $\bar{w}_0$ la 2. Cât devine acum suma costurilor *hinge* determinate de exemplele de antrenament în raport cu acest nou separator?

g. Daţi o interpretare geometrică pentru suma costurilor *hinge* pe care aţi calculat-o la punctul f (adică, suma costurilor *hinge* determinate de exemplele de antrenament în raport noul separator).

***Indicaţie* (2): La problema 20 am stabilit o corespondenţă între costurile *hinge* şi variabilele ecart $\bar{\xi}_i$, iar la problema 13 am demonstrat următorul rezultat:**

$$\bar{\xi}_i > 0 \Rightarrow \frac{\bar{\xi}_i}{\|\bar{w}\|} = d(x_i,\ \bar{w} \cdot x + \bar{w}_0 = y_i).$$

58. **(Clasificatori liniari — [C-]SVM, Perceptronul etc. — şi versiunile lor kernel-izate: avantaje şi dezavantaje)**

□ • ○ ***CMU, 2017 fall, Nina Balcan, HW3, pr. 3.3.2/Q17***

Care credeţi că sunt avantajele şi dezavantajele (engl., *pros* and *cons*) folosirii modelelor liniare precum SVM, Perceptronul etc., precum şi versiunile lor kernel-izate?

59. **(O comparaţie între algoritmii 1-NN şi C-SVM: efectul atributelor irelevante pentru clasificare)**

• ○ ∗ ***CMU, 2007 spring, Carlos Guestrin, midterm exam, pr. 6.1-2***

***Convenţie*: În tot acest exerciţiu, prin SVM se va înţelege o maşină cu vectori-suport cu margine "soft" (adică, C-SVM), fără funcţie-nucleu (ceea ce înseamnă că lucrăm cu separator liniar).**

a. Alcătuiţi un set de date din $\mathbb{R}^2$ astfel încât pe acest set algoritmul 1-NN să producă eroare la cross-validare de tip "leave one out" (CVLOO) mai mică decât SVM.

b. Alcătuiţi un alt set de date din $\mathbb{R}^2$ astfel încât pe noul dataset eroarea de tip CVLOO produsă de algoritmul 1-NN să fie mai mare decât cea produsă de SVM.

c. Acum veţi genera două dataset-uri care să pună în evidenţă caracterul robust al algoritmului SVM în raport cu *atributele irelevante*.[754] Veţi crea un set de date din $\mathbb{R}^2$ (câte unul pentru fiecare dintre cele două probleme date mai jos) cu atributele X_1 şi X_2 — dintre care X_2 va fi atributul irelevant —, astfel încât:

- **dacă se foloseşte doar atributul X_1, eroarea la CVLOO produsă de algoritmul 1-NN este mai mică decât cea produsă de SVM,**
- **dacă se folosesc atributele X_1 şi X_2, eroarea CVLOO produsă de SVM nu se schimbă (în raport cu cazul de mai sus), însă eroarea CVLOO produsă de 1-NN creşte în mod semnificativ.**

60. **(C-SVM: Adevărat sau Fals?)**

● ○ ∗ ***CMU, 2010 fall, Aarti Singh, HW3, pr. 3.2***

Presupunem că se lucrează cu o maşină cu vectori-suport cu margine "soft" (C-SVM), pe un anumit set de exemple, fără a folosi vreo funcţie de „mapare" a trăsăturilor. Pe măsură ce valoarea parametrului de „destindere" C creşte (pornind de la o anumită valoare de start),

a. mai multe instanţe de antrenament vor fi clasificate eronat;

b. marginea — adică $\frac{1}{\|w\|}$, distanţa de la hiperplanul de separare optimală la instanţele (vectorii-suport) x_i pentru care $(w \cdot x_i + w_0)y_i = 1$, unde y_i este eticheta asociată instanţei x_i — nu va creşte (adică fie descreşte fie rămâne aceeaşi).

5.2.3 Alte *probleme de optimizare* de tip SVM

61. **(Câteva probleme simple de optimizare de tip [C-]SVM)**

● ○ ∗ ***MIT, 2016 fall, R. Barzilay, S. Sra, Weekly Exercises, week 4, pr. 10***

După cum aţi observat, profesorului de la cursul de Învăţare automată îi place să pună în evidenţă diverse variante de metode de clasificare automată de tip SVM.

În figurile următoare (notate cu 1., 2., 3. şi 4.) sunt înfăţişate atât graniţele de decizie (engl., decision boundaries) cât şi vectorii-suport (care, după cum vedeţi, sunt încercuiţi) pentru câteva metode de de tip SVM aplicate pe un acelaşi set de date din $\mathbb{R}^2$. În toate cazurile, graniţele de decizie corespund unei ecuaţii de forma $\hat{w} \cdot x + \hat{w}_0 = 0$, unde se consideră în mod implicit că $\hat{w}_0 = 0$,

[754] Adică, acele atribute care, atunci când sunt eliminate / adăugate, n-ar trebui să schimbe rezultatul clasificării.

cu excepţia cazurilor când w_0 este inclus în mod explicit în formularea analitică a metodei de antrenare. Simbolii J_+ şi J_- din aceste formulări analitice desemnează indicii instanţelor de antrenament pozitive (reprezentate cu $\bullet$) şi respectiv negative (reprezentate cu $\circ$).

În total sunt date aici cinci metode de clasificare de tip SVM şi patru figuri/grafice. În aceste figuri, dreptele corespunzătoare separtorilor optimali $\hat{w} \cdot x + \hat{w}_0 = 0$ sunt reprezentate prin linii îngroşate. „Marginile“ / dreptele de ecuaţie $\hat{w} \cdot x + \hat{w}_0 = -1$ şi respectiv $\hat{w} \cdot x + \hat{w}_0 = +1$ sunt reprezentate prin linii simple (neîngroşate).

Pentru fiecare dintre aceste cinci metode de clasificare date mai jos, indicaţi toate(!) figurile care constituie o posibilă soluţie pentru metoda respectivă. Justificaţi riguros.
Atenţie! **Este posibil ca anumitor metode de clasificare [din cele cinci] să le corespundă mai mult decât o [singură] figură.**

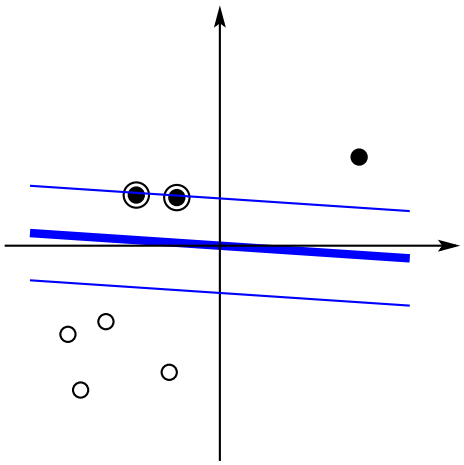
a. $\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^n \xi_i$, **a.î.** $\xi_i \geq 0,\ y_i(w\cdot x_i + w_0) \geq 1 - \xi_i,\ i = 1,\ldots,n,$ **cu** $C = \infty$.

b. $\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^n \xi_i$, **a.î.** $\xi_i \geq 0,\ y_i(w\cdot x_i) \geq 1 - \xi_i,\ i = 1,\ldots,n,$ **cu** $C = \infty$.

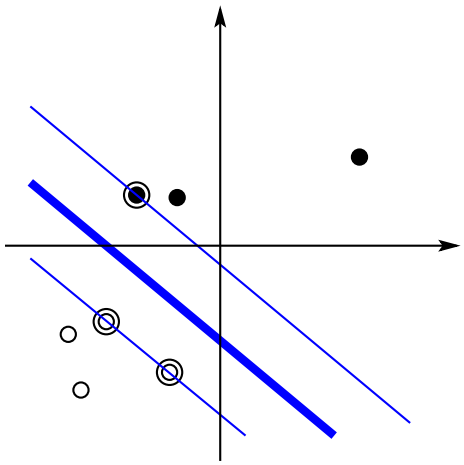
c. $\min \frac{1}{2}\|w\|^2 + C\sum_{i=1}^n \xi_i$, **a.î.** $\xi_i \geq 0,\ y_i(w\cdot x_i) \geq 1 - \xi_i,\ i = 1,\ldots,n,$ **cu** $C = 1$.

d. $\min \frac{1}{2}\|w\|^2 + C_+\sum_{i\in J_+} \xi_i + C_-\sum_{i\in J_-} \xi_i$, **a.î.** $\xi_i \geq 0,\ y_i(w\cdot x_i) \geq 1 - \xi_i,\ i = 1,\ldots,n,$ **cu** $C_+ = 1$ **şi** $C_- = 0$.

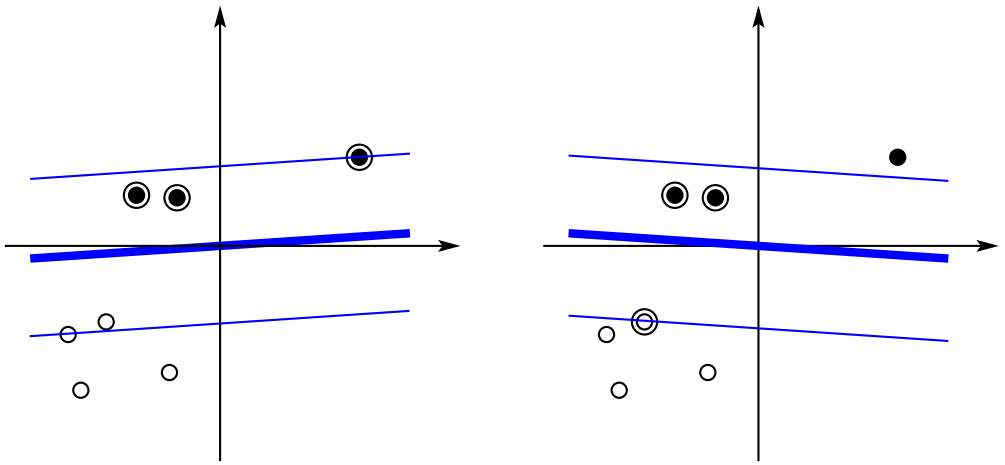
e. $\min \frac{1}{2}\|w\|^2 + C_+\sum_{i\in J_+} \xi_i + C_-\sum_{i\in J_-} \xi_i$, **a.î.** $\xi_i \geq 0,\ y_i(w\cdot x_i) \geq 1 - \xi_i,\ i = 1,\ldots,n,$ **cu** $C_+ = \infty$ **şi** $C_- = 0$.



1.



2.



3. 4.

62. **(SVM multiclass cu margine "soft": verificarea echivalenţei cu C-SVM în cazul clasificării binare)**

□ • ***MIT, 2016 spring, David Sontag, HW2, pr. 4***

Clasificatorul *SVM multiclass* constituie o generalizare a clasificatorului binar SVM de la două clase la un număr oarecare de clase, $K \geq 2$. După cum am arătat la problema 30 (unde am prezentat varianta SVM-multiclass cu margine "hard"), aceasta implică introducerea unui vector de ponderi $w^{(k)}$ şi a unui termen liber $b^{(k)}$ pentru fiecare clasă $k \in \{1, \ldots, K\}$.[755]

Ca şi la alte variante ale clasificatorului [C-]SVM, şi în cazul de faţă învăţarea / antrenarea constă în rezolvarea unei *probleme de optimizare*:

$$\begin{aligned} & \min_{w^{(k)}, b^{(k)}, \xi} \left(\frac{1}{2} \|w^{(k)}\|^2 + C \sum_i \xi_i \right) \\ \textbf{a. î.} \quad & w^{(y_i)} \cdot x_i + b^{(y_i)} \geq w^{(k)} \cdot x_i + b^{(k)} + 1 - \xi_i, \text{ pentru } i = 1, \ldots, m \text{ şi } k \neq y_i \\ & \xi_i \geq 0, \ \forall i = 1, \ldots, m. \end{aligned}$$

Observaţi că în această problemă, pentru fiecare instanţă de antrenament x_i apare — ca şi în cazul problemei C-SVM, vedeţi pr. 12 — câte o variabilă de destindere (engl., slack variable) ξ_i, însă aici asociem la fiecare instanţă [un număr de] $K - 1$ restricţii.

Pentru o instanţă nouă x, predicţia se va face folosind regula următoare:

$$y = \arg\max_k \left(w^{(k)} \cdot x + b^{(k)} \right). \tag{348}$$

În acest exerciţiu veţi compara această regulă de predicţie (de tip multiclass) în cazul particular în care se lucrează cu $K = 2$ cu regula de predicţie pe care am folosit-o pentru clasificatorul [binar] C-SVM (vedeţi pr. 12.f):

$$sign\,(w \cdot x + b). \tag{349}$$

[755] La problema 30, din considerente care ţin (doar) de simplitatea formulării, nu s-au folosit termeni liberi. Adăugarea lor se poate face în mod natural.

Concret, veţi arăta că fiecare dintre cele două reguli se reduce la [adică, este echivalentă cu] cealaltă.

a. Calculaţi w şi w_0 în funcţie de $w^{(1)}, b^{(1)}, w^{(2)}$ şi $b^{(2)}$ astfel încât pentru fiecare instanţă x rezultatul obţinut folosind noua regulă de predicţie binară (349) să fie acelaşi cu rezultatul care se obţine dacă se aplică regula de predicţie multiclass (348) care foloseşte $w^{(1)}, b^{(1)}, w^{(2)}$ şi $b^{(2)}$.

b. La acest punct veţi proceda exact invers faţă de punctul precedent: Considerând w şi w_0 ca fiind daţi, calculaţi $w^{(1)}, b^{(1)}, w^{(2)}$ şi $b^{(2)}$ (în funcţie de w şi w_0) astfel încât pentru fiecare instanţă x predicţia făcută cu ajutorul regulii de predicţie multiclass (cu $K = 2$) să fie acelaşi cu rezultatul care s-ar obţine dacă am folosi regula de predicţie binară cu respectivele valori (date) pentru w şi w_0.

63. **(O legătură între *one-class SVM* (versiunea *Max Margin*) şi SVM (cu şi respectiv fără termen liber (engl., bias))**

prelucrare de Liviu Ciortuz, după
□ • ○ ***MIT, 2009 fall, Tommi Jaakkola, midterm, pr. 3***

Un prieten ne-a spus că el poate să emuleze clasificatori de tip SVM folosind doar cod care a fost dezvoltat pentru detecţia „anomaliilor", mai precis algoritmul *one-class* SVM, versiunea *Max Margin*, pe care l-am prezentat la pr. 31. Rutinele de antrenare şi respectiv testare despre care el afirmă că sunt suficiente pentru acest scop sunt următoarele:[756]

$$(\hat{w}, \hat{\rho}) = \texttt{train}(\phi_1, \ldots, \phi_m) \ : \quad \text{Minimize}_{w,\rho} \left(\frac{1}{2}\|w\|^2 - \rho\right)$$
$$\text{subject to } w \cdot \phi_i \geq \rho,\ i = 1, \ldots, m$$
$$\text{Return } \hat{w}, \hat{\rho}$$

$$y = \texttt{test}(w, \rho, \phi) \ : \quad \text{Return } +1 \textbf{ if } w \cdot \phi \geq \rho \textbf{ else return } -1$$

unde $\phi_1, \ldots, \phi_m$ şi ϕ sunt vectori de trăsături.

Noi n-am fost siguri dacă el are dreptate sau nu, aşa că ne-am decis să verificăm.

Să începem prin a găsi clasificatorul SVM fără termen liber (engl., bias) de forma

$$\hat{y} = sign\,(w \cdot \phi(x))$$

corespunzător unui set de m instanţe de antrenament $\phi(x_1), \ldots, \phi(x_m)$ având etichetele $y_1, \ldots, y_m \in \{+1, -1\}$. Presupunem că setul de date de antrenament este liniar separabil.

a. Precizaţi care sunt vectorii de trăsături pe care ar trebui să-i transmitem [pentru antrenare] rutinei `train`?

b. Fie $\hat{w}$ şi $\hat{\rho}$ parametrii returnaţi de către rutina `train` în urma alegerii făcute la punctul precedent pentru vectorii de trăsături [pentru antrenare]. Care

[756] Vedeţi *Observaţia* (2) de la pr. 31.

sunt argumentele pe care ar trebui să le transmitem rutinei `test` astfel încât ea să clasifice instanţa de test $\phi(x)$ identic cu clasificatorul de margine maximă?

c. Cât este — în funcţie de $\hat{w}$ şi $\hat{\rho}$ — marginea geometrică pe care o obţine clasificatorul SVM pe setul de date de antrenament?

d. Încurajaţi de aceste rezultate, ne punem întrebarea dacă nu cumva ar fi posibil să antrenăm şi clasificatorul SVM care include şi termenul liber, adică $\hat{y} = sign(w \cdot \phi(x) + w_0)$, folosind [doar] cele două rutine. Este într-adevăr posibil? Justificaţi riguros.

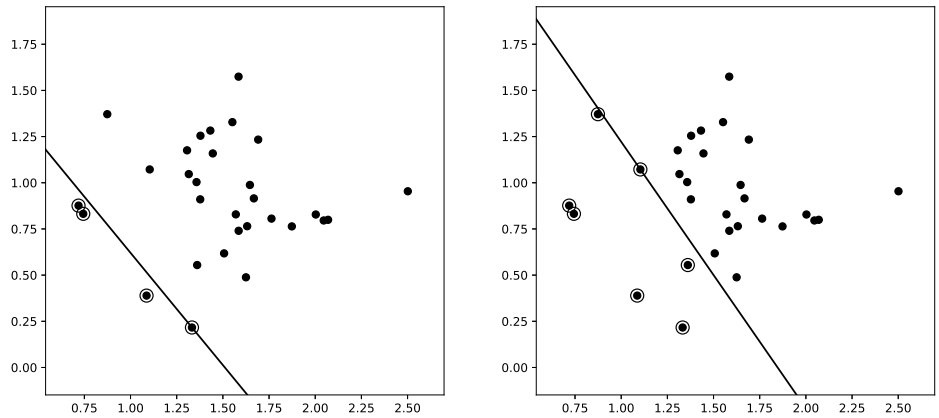
64. **(Rezolvarea problemei *one-class*, varianta *Max Margin*, cu margine "soft", folosind abordarea de la $\nu-$SVM)**

prelucrare de Liviu Ciortuz, după
■ ∗ ***MIT, 2009 fall, Tommi Jaakkola, ML course, lecture notes 5***

Vom reveni aici asupra problemei de optimizare *SVM one-class* varianta *Max Margin* (vedeţi pr. 31). Mai întâi precizăm că, în ceea ce priveşte versiunea cu margine "hard", faţă de forma primală pe care am considerat-o în problema menţionată, acum vom lucra cu o versiune mai generală:[757]

$$\min_{w,\rho} \left(\frac{1}{2}\|w\|^2 - \rho\right)$$
$$\textbf{a. î.} \quad w \cdot x_i \geq \rho, \ \textbf{pentru } i = 1, \ldots, m.$$

Forma aceasta este mai convenabilă pentru *obiectivul* pe care ni-l fixăm aici, acela de a elabora cazul marginii "soft" pentru problema *one-class SVM* (Max Margin) urmând abordarea de tip ν-SVM. (A se vedea problema 33.)



Vă reamintim că ν-SVM foloseşte un parametru numeric (ν) care va funcţiona ca margine superioară pentru proporţia de erori la antrenare din totalul instanţelor de antrenament.

[757] Distanţa de la hiperplanul de separare optimală la vectorii-suport care nu produc erori în raport cu marginea va fi $\frac{\rho}{||w||}$. Din punctul de vedere al „marginii" geometrice, noi am vrea ca [şi] ρ să fie maximizat. Aceasta echivalează cu a minimiza $-\rho$. Aşa se justifică (în mod intuitiv) expresia funcţiei obiectiv din problema de optimizare pe care urmează să o formulăm.

Forma primală pe care o vom considera aici pentru problema $\nu-$SVM *one-class* (Max Margin) este de asemenea uşor schimbată în raport cu formularea originală a problemei ν-SVM (B. Schölkopf, A. Smola, R. Williamson şi P. Bartlett, *New Support Vector Machines*, 2000):[758]

$$\begin{array}{rl} & \min_{w,\xi,\rho} \left(\frac{1}{2}\|w\|^2 - \rho + \frac{1}{\nu m}\sum_{i=1}^m \xi_i\right) \\ \textbf{a. î.} & w \cdot x_i \geq \rho - \xi_i, \textbf{ pentru } i = 1, \ldots, m \\ & \xi_i \geq 0 \textbf{ pentru } i = 1, \ldots, m. \end{array} \qquad (\mathbf{P'''})$$

Remarcaţi faptul că în ambele probleme de optimizare date mai sus n-au fost impuse restricţii asupra variabilei ρ.[759]

a. Demonstraţi că forma duală corespunzătoare formei primale ($\mathbf{P'''}$) a problemei ν-SVM *one-class* (Max Margin) este următoarea:

$$\begin{array}{rl} & \max_{\alpha} \left(-\frac{1}{2}\sum_{i,j} \alpha_i \alpha_j x_i \cdot x_j\right) \\ \textbf{a. î.} & 0 \leq \alpha_i \leq \frac{1}{\nu m} \textbf{ pentru } i = 1, \ldots, m \\ & \sum_{i=1}^m \alpha_i = 1. \end{array} \qquad (\mathbf{D'''})$$

b. Scrieţi condiţiile de complementaritate Karush-Kuhn-Tucker pentru problema primală ($\mathbf{P'''}$), apoi arătaţi cum anume se poate calcula $\bar{\rho}$, valoarea rezultată pentru variabila ρ la rezolvarea problemei duale ($\mathbf{D'''}$), în funcţie de valorile celorlalte variabile ($\bar{w}$, $\bar{\alpha}_i$, $\bar{\xi}_i$, etc).

65. **(Problema [*one-class SVM*, varianta] *sferei de incluziune minimală* (engl., minimum enclosing ball, MEB) în varianta cu margine "soft", folosind $\nu-$SVM)**

prelucrare de Liviu Ciortuz, după
■ ∗ ***MIT, 2009 fall, Tommi Jaakkola, ML course, lecture notes 5***

Date fiind instanţele $x_1, \ldots, x_m \in \mathbb{R}^d$, ne propunem să găsim o sferă care să includă toate punctele x_i şi să aibă cea mai mică rază posibilă. Pentru aceasta, formulăm următoarea problemă de optimizare convexă:

$$\min_{R,w} R^2 \textbf{ astfel încât } ||w - x_i||^2 \leq R^2 \textbf{ pentru } i = 1, \ldots, m.$$

De remarcat faptul că restricţiile din formularea acestei probleme sunt de ordin pătratic (spre deosebire de cazul problemei SVM clasice, în care restricţiile sunt liniare). La finalul rezolvării acestei probleme,

– valoarea găsită pentru w va reprezenta centrul sferei;

– vectorii-suport vor fi punctele x_i de pe suprafaţa sferei; restricţiile corespunzătoare lor vor fi satisfăcute cu egalitate ($||w - x_i||^2 = R^2$).

Ca şi în cazul problemei 64, putem impune condiţia ca maximum νm puncte (din totalul celor m) să fie lăsate în afara sferei. Corespunzător, problema de optimizare convexă de tip ν-SVM va fi:

[758] A se vedea şi problema 33.
[759] La problema 33 aveam $\rho \geq 0$.

$$\begin{aligned} &\min_{R,w,\xi} \Big(R^2 + \frac{1}{\nu m} \sum_{i=1}^{m} \xi_i \Big) \\ \textbf{astfel încât} \quad &||w - x_i||^2 \leq R^2 + \xi_i \textbf{ pentru } i = 1, \ldots, m \\ &\xi_i \geq 0 \textbf{ pentru } i = 1, \ldots, m. \end{aligned} \tag{$\mathbf{P}^{iv}$}$$

a. Demonstraţi că forma duală a problemei de tip ν-SVM de mai sus este:

$$\begin{aligned} &\max_{\alpha} \Big(-\textstyle\sum_i \sum_j \alpha_i \alpha_j x_i \cdot x_j + \sum_{i=1}^{m} \alpha_i x_i \cdot x_i \Big) \\ \textbf{a. î.} \quad &0 \leq \alpha_i \leq \frac{1}{\nu m} \textbf{ pentru } i = 1, \ldots, m \\ &\textstyle\sum_{i=1}^{m} \alpha_i = 1 \end{aligned} \tag{$\mathbf{D}^{iv}$}$$

unde α_i este multiplicatorul Lagrange corespunzător restricţiei $i \in \{1, \ldots, m\}$.

b. Indicaţi relaţia de legătură dintre soluţia $\bar{w}$ a problemei primale şi soluţia $\bar{\alpha}$ a problemei duale. (Remarcaţi semnificaţia geometrică a rezultatului!)

c. Cum se poate calcula valoarea optimă $\bar{R}$ (pentru variabila R din forma primală ($\mathbf{P}^{iv}$)) pornind de la soluţia problemei duale?

***Sugestie*: Pentru aceasta, este util ca, pentru problema de tip ν-SVM de mai sus ($\mathbf{D}^{iv}$), să enunţaţi condiţiile de complementaritate Karush-Kuhn-Tucker.**

66. **(SVR cu margine "hard", cazul liniar: exemplu de aplicare)**

□ • ○ ***CMU, 2008 fall, Eric Xing, midterm, pr. 3***

Pentru *regresie*, ni se dau m exemple de antrenament $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$, unde fiecare x_i este un vector de trăsături, iar $y_i \in \mathbb{R}$ este outputul pentru exemplul de pe poziţia i ($i = 1, 2, \ldots, m$).

Vă readucem aminte că ın cazul *regresiei liniare cu vectori-suport* (engl., support vector regression, SVR),[760] obiectivul este ca, pornind de la setul de exemple de antrenament, să învăţăm o funcţie liniară de forma $f(x) = w \cdot x + b$, unde w şi b sunt parametrii care trebuie „învăţaţi", iar w este vectorul de ponderi. Problema de *optimizare convexă* pentru SVR liniară poate fi formulată în felul următor:

$$\min_{w} \frac{1}{2} \|w\|^2 \tag{350}$$

$$\textbf{a. î.} \; y_i - (w^\top x_i + b) \leq \varepsilon \;\; \textbf{şi} \;\; (w \cdot x_i + b) - y_i \leq \varepsilon \; (i = 1, \ldots, m). \tag{351}$$

***Ideea* din spatele relaţiilor (350) şi (351) este următoarea: dorim să „învăţăm" parametrii w şi b astfel încât (i) funcţia f obţinută să fie cât mai „netedă" (engl., smooth) cu putinţă (aşadar, urmărim ca $\|w\|$ să fie cât mai mic), iar (ii) pentru fiecare exemplu de antrenament, eroarea în raport cu predicţia făcută de către funcţia f să fie de cel mult ε, unde $\varepsilon \geq 0$ este un *parametru* dat / fixat în acest algoritm.[761]**

Acum, date fiind 3 exemple de antrenament, $(1, 1)$, $(2, 2)$ şi $(3, 3)$, vrem să folosim relaţiile (350) şi (351) pentru a învăţa un model liniar SVR.

[760] Vedeţi problema 34.

[761] Remarcaţi faptul că pe măsură ce valorile lui ε cresc, funcţia f pe care o căutăm devine tot mai „netedă".

a. Cât este w atunci când $\varepsilon = 0$?

b. Cât este w atunci când $\varepsilon = 0.5$?

c. Cât este w atunci când $\varepsilon = 1$?

67. **(SVR cu margine "soft": varianta care foloseşte funcţie de cost / pierdere ε-senzitivă)**

■ □ • ○ ***CMU, 2014 spring, B. Poczos, A. Singh, HW2, pr. 1***
CMU, 2015 fall, Z. Bar-Joseph, E. Xing, HW3, pr. 2.1-4

În acest exerciţiu veţi deduce forma duală a problemei de optimizare pentru regresie cu vectori-suport (SVR) cu margine "soft", care foloseşte o funcţie de cost *ε-sensibilă* (engl., epsilon-sensitive loss), dată de expresia

$$L_\varepsilon(x, y, f) = |y - f(x)|_\varepsilon \overset{not.}{=} \max(0, |y - f(x)| - \varepsilon), \tag{352}$$

unde x este inputul, y este outputul, iar $f(x) \overset{def.}{=} w \cdot x$ este funcţia folosită pentru a predicţie.[762] Setul de date de antrenament este $(x_1, y_1), \ldots, (x_n, y_n)$, unde $x_i \in \mathbb{R}^m$, $y_i \in \mathbb{R}$.[763]

Folosind această notaţie, funcţia obiectiv pentru problema SVR este definită astfel:

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} L_\varepsilon(x_i, y_i, f),$$

unde $f(x) = w \cdot x$, iar $C > 0$ şi $\varepsilon > 0$ sunt parametri.

a. Scrieţi această problemă ca o problemă de optimizare pătratică cu restricţii liniare, folosind variabile de „destindere" (engl., slack variables). Această formă este numită *forma primală* a problemei de optimizare SVR cu margine "soft".

***Sugestie*: Expresia funcţiei de cost (352) nu trebuie să vă jeneze. Practic va trebui să procedaţi în manieră inversă faţă de cum s-a procedat la problema 20, unde, pornind de la forma primală a problemei C-SVM, se obţine o formă echivalentă care foloseşte funcţia de cost *hinge*. Alternativ, puteţi pleca de la forma primală a problemei SVR cu margine "hard" (vedeţi pr. 34) şi introduceţi variabile de destindere ξ_i similar modului în care am procedat pentru a obţine clasificatorul C-SVM din clasificatorul SVM (vedeţi pr. 12).**

b. Scrieţi *funcţia lagrangeană* pentru *forma primală* a problemei SVR pe care aţi obţinut-o la punctul precedent.

c. Folosind *condiţiile Karush-Kuhn-Tucker*, deduceţi *forma duală* a acestei probleme SVR.

d. Am putea folosi solver-e [LC: adică programe de analiză numerică] de probleme de *optimizare pătratică* pentru a rezolva problema duală SVR dedusă mai sus?

[762] Notând $z = y - f(x)$, este foarte util — pentru a înţelege mai bine ceea ce urmează — să faceţi graficul funcţiei $\max(0, |z| - \varepsilon)$.

[763] Remarcaţi faptul că funcţia de cost / pierdere *hinge* pe care am folosit-o la problema 20 (în legătură cu C-SVM) este adecvată doar pentru clasificare, nu şi pentru regresie.

e. Cum aţi putea defini *vectorii-suport* pentru această problemă SVR?

f. Scrieţi expresia care poate fi folosită pentru a face *predicţia* etichetei pentru o instanţă nouă, x.

g. Este oare posibil să kernel-izăm acest algoritm?

h. Formulaţi motivul pentru care în general rezolvăm problemele SVR şi SVM în varianta duală, nu în cea primală.

i. Ce se întâmplă atunci când modificăm valoarea parametrului ε?

j. Ce se întâmplă atunci când modificăm valoarea parametrului C?

68. **(Exrciţiu recapitulativ: corespondenţa dintre diverse funcţii de cost (respectiv diverse tipuri de graniţe de decizie) şi diferite metode de învăţare automată)**

□ • ***CMU, 2014 spring, B. Poczos, A. Singh, midterm, pr. 2.1-2***

a. Puneţi în corespondenţă fiecare dintre metodele de învăţare automată de mai jos (din coloana din stânga) cu una dintre funcţiile de cost / pierdere (engl., loss functions) din coloana din dreapta, şi anume acea funcţie de cost pe care respectiva metodă de învăţare o minimizează în cazul cel mai frecvent:

regresia liniară	**funcţia de cost pătratică**
regresia logistică	**funcţia de cost logistică**
AdaBoost	**funcţia de cost exponenţială**
k-NN	**funcţia de cost $0-1$**
SVM	**funcţia de cost *hinge***

b. Pentru fiecare dintre metodele de învăţare automată de mai jos (din coloana din stânga), încercuiţi tipul / tipurile de separator decizional (graniţă de decizie) pe care respectiva metodă îl poate produce la efectuarea unui task de clasificare binară. În undele cazuri, este posibil ca mai multe opţiuni să fie corecte. Veţi încercui toate opţiunile pe care le consideraţi corecte.

regresia logistică:	**liniar**
ID3 (cu atribute numerice continue):	**liniar, combinaţie de separatori liniari**[764]
AdaBoost:	**liniar, combinaţie de separatori liniari, pătratic**
clasificatorul Bayes Naiv gaussian:	**liniar, pătratic**
SVM (fără funcţie-nucleu):	**liniar**

[764] Engl., piecewise linear.



© *M. Romanică*

6 Reţele neuronale artificiale

Sumar

Noţiuni preliminare

- **funcţie matematică; compunere de funcţii reale;**
  calculul valorii unei funcţii pentru anumite valori specificate pentru argumentele / variabilele ei;
- **funcţie prag (sau, treaptă), funcţie liniară, funcţie sigmoidală (sau, logistică), funcţie sigmoidală generalizată;**
  separabilitate liniară pentru o mulţime de puncte din $\mathbb{R}^d$;
- **ecuaţii asociate dreptelor în plan / planelor în spaţiu / hiper-planelor în spaţiul $\mathbb{R}^d$;**
  ecuaţia dreptei în plan care trece prin două puncte date;
  semnele asociate punctelor din semiplanele determinate de o dreaptă dată în plan;
- **derivate ale funcţiilor elementare de variabilă reală; derivate parţiale**
- **vectori; operaţii cu vectori, în particular produsul scalar al vectorilor;**
- **metoda gradientului descendent (ca metoda de optimizare); avantaje şi dezavantaje; ex. 80, ex. 168 şi ex. 169 de la cap. de *Fundamente*; ex. 23;**
- **funcţii de cost / pierdere: ex. 166 de la cap. de *Fundamente*.**

Câteva noţiuni specifice

- ***unităţi* neuronale artificiale (sau, *neuroni* artificiali, *perceptroni*);**
  tipuri de neuroni artificiali: neuroni-prag, liniari, sigmoidali;
  ***componente* ale unui neuron artificial: input, componenta de sumare, componenta / funcţia de activare, output;**
  funcţia matematică reprezentată / calculată de un neuron artificial;
- ***reţea* neuronală artificială; reţele de tip feed-forward;**
  ***niveluri* / straturi de neuroni, niveluri ascunse, niveluri de ieşire;**
  ***ponderi* asociate conexiunilor dintr-o reţea neuronală artificială;**
  funcţia matematică reprezentată / calculată de o reţea neuronală artificială;
  ***graniţe şi zone de decizie* determinate de o reţea neuronală artificială;**
  funcţia de eroare / cost (engl., loss function).

Câteva proprietăţi relative la *expresivitatea* reţelelor neuronale artificiale

- **(P0) Toate cele trei tipuri de neuroni artificiali (prag, liniar, sigmoidal) produc *separatori liniari*.**

  Consecinţă: Conceptul XOR nu poate fi reprezentat / învăţat cu astfel de „dispozitive“ simple de clasificare.

– **(P0′) Reţelele neuronale artificiale pot determina graniţe de decizie neliniare (şi, în consecinţă, pot reprezenta concepte precum XOR).**

  Observaţie: Reţele de unităţi sigmoidale pot determina graniţe de decizie curbilinii: ex. 8.

– **(P1) Reţele de neuroni diferite (ca structură şi / sau tipuri de unităţi) pot să calculeze o aceeaşi funcţie: ex. 3 şi ex. 1.c vs. ex. 2.**

  (P1′) Dată o topologie de reţea neuronală (i.e., graf de unităţi neuronale al căror tip este lăsat nespecificat), este posibil ca plasând în noduri unităţi de un anumit tip să putem reprezenta / calcula o anumită funcţie, iar schimbând tipul unora dintre unităţi (sau al tuturor unităţilor), funcţia respectivă să nu mai potă fi calculată: ex. 4 vs. ex. 33.[765]

– **(P2) Orice unitate liniară situată pe un nivel ascuns poate fi „absorbită“ pe nivelul următor: ex. 32.**

– **(P3) Orice funcţie booleană poate fi reprezentată cu ajutorul unei reţele neuronale artificiale având doar două niveluri de perceptroni-prag: ex. 5.**

– **(P4) Orice funcţie definită pe un interval mărginit din $\mathbb{R}$, care este continuă în sens Lipschitz, poate fi aproximată oricât de bine cu ajutorul unei reţele neuronale care are un singur nivel ascuns: ex. 7.**

– **Corespondenţe cu regresia liniară, regresia logistică şi boosting-ul (ex. 34), respectiv cu arborii de decizie (ex. 35).**

Algoritmi de antrenare a neuronilor artificiali folosind metoda gradientului descendent

– **algoritmul de antrenare a unităţii liniare: ex. 36;**
vedeţi T. Mitchell, *Machine Learning*, p. 93, justificare: p. 91-92; convergenţa: p. 95; exemplu de aplicare: ex. 10;
varianta incrementală a algoritmului de antrenare a unităţii liniare: cartea ML, p. 93-94; despre convergenţa acestei variante (ca aproximare a variantei precedente (“batch”): cartea ML, p. 93 jos;

– **algoritmul de antrenare a perceptronului-prag şi convergenţa: cartea ML, p. 88-89; exemplu de aplicare: ex. 11;**

– **algoritmul de antrenare a perceptronului sigmoidal şi justificarea sa teoretică: cartea ML, p. 95-97;**

– **algoritmul *Perceptron* al lui Rosenblatt; exemplu de aplicare: ex. 16, ex. 38;**

– **deducerea regulii de actualizare a ponderilor pentru tipuri particulare de perceptroni: ex. 12, ex. 25.a, ex. 37, ex. 13.a;**

– **o justificare probabilistă (gen ipoteză de tip *maximum likelihood*) pentru minimizarea sumei pătratelor erorilor [la deducerea regulii de antrenare] pentru perceptronul liniar: ex. 13.b;**

– **exemple de [folosire a unei] alte funcţii de cost / pierdere / penalizare (engl., loss function) decât semisuma pătratelor erorilor: suma costurilor de tip log-sigmoidal, ex. 14 (pentru perceptronul liniar), o funcţie de tip cross-entropie, ex. 15 (pentru perceptronul sigmoidal).**

[765] Problemele 1.d şi ex. 31 au în vedere o chestiune similară, însă pentru reţele cu topologii diferite: o anumită extensie a funcţiei XOR nu poate fi reprezentată pe reţele de neuroni-prag care au un singur nivel ascuns.

Perceptronul Rosenblatt şi rezultate de *convergenţă*

– **exemplu de aplicare [adică, învăţare cu perceptronul Rosenblatt]: ex. 16.**

– **câteva *proprietăţi* simple ale perceptronului Rosenblatt: ex. 17.**

– **rezultate de convergenţă de tip "mistake bound" pentru [algoritmul de antrenare pentru] perceptronul-prag [în varianta] Rosenblatt: ex. 18;**
pentru perceptronul-prag (clasic): ex. 40;
învăţare online cu perceptronul-prag de tip Rosenblatt: ex. 39;

– ***Perceptronul* kernel-izat [dual]: ex. 19; particularizare pentru cazul nucleului RBF: ex. 41. Perceptronul Rosenblatt, cu termen liber (engl., offset), kernel-izare: ex. 42. Clasificare ternară cu perceptronul Rosenblatt, varianta kernelizată: ex. 43. (Vedeţi şi ex. 40.c.)**

Antrenarea reţelelor neuronale artificiale: algoritmul de *retro-propagare* pentru reţele feed-forward

– **T. Mitchell, *Machine Learning*, p. 98: pseudo-cod pentru reţele cu unităţi de tip sigmoidal, cu 2 niveluri, dintre care unul ascuns; pentru deducerea regulilor de actualizare a ponderilor; în cazul mai general al reţelelor feed-forward (de unităţi sigmoidale) cu oricâte niveluri, vedeţi p. 101-103;**

ex. 20: deducerea regulilor de actualizare a ponderilor în cazul reţelelor cu 2 niveluri, având însă unităţi cu funcţie de activare oarecare (derivabilă);

– **aplicare: ex. 21, ex. 45, ex. 46;**

– **prevenirea overfitting-ului:**
folosirea unei componente de tip „moment" în expresia regulilor de actualizare a ponderilor: ex. 48;
regularizare: introducerea unei componente suplimentare în funcţia de optimizat: ex. 22;

– **cazul folosirii unei funcţii de activare de tip tangentă hiperbolică: ex. 47;**

– **cazul folosirii unei funcţii de cost / penalizare / eroare de tip cross-entropie: ex. 50;**

– **execuţia manuală a unei iteraţii a algoritmului de retro-propagare în cazul unei reţele neuronale simple, având un singur nivel ascuns, cu unităţi ce folosesc funcţia de activare ReL: ex. 51.**

Reţele neuronale profunde — câteva chestiuni introductive

– **fenomenul de „dispariţie" a gradientului [în cazul aplicării algoritmului de retro-propagare] pentru reţele neuronale profunde (engl., deep neural networks) care folosesc funcţia de activare sigmoidală: ex. 26;**

– **determinarea numărului de parametri şi de conexiuni din reţeaua neuronală convolutivă LeNet: ex. 27;**

– **determinarea mărimii hărţii de trăsături de pe un anumit nivel, precum şi a numărului de operaţii în virgulă mobilă (FLOPs) executate la procesarea forward într-o reţea neuronală convolutivă: ex. 54.**

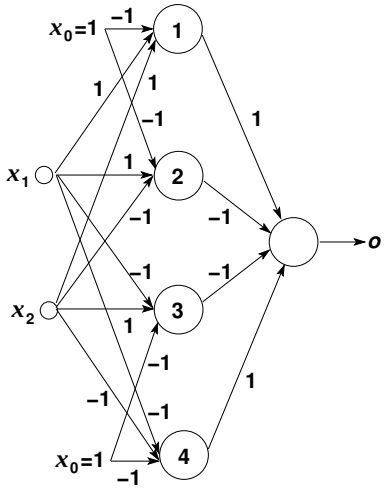
6.1 Reţele neuronale artificiale — Probleme rezolvate

6.1.1 Chestiuni introductive

1. **(Reţele de perceptroni-prag, exemplificare: calculul outputului)**

prelucrare de Liviu Ciortuz, după
■ CMU, 2010 fall, Aarti Singh, HW5, pr. 4.1.1

Considerăm reţeaua neuronală din figura alăturată. Toate unităţile acestei reţele neuronale sunt de tip prag, adică folosesc pentru activare funcţia *sign* definită prin $sign(z) = 1$ dacă $z \geq 0$ şi -1 în rest. Pentru unitatea de pe nivelul de ieşire (lăsată nenumerotată), ponderea corespunzătoare termenului liber ($x_0 = 1$) este 0.



a. Scrieţi funcţia matematică calculată de fiecare dintre unităţile reţelei, în raport cu intrările x_1 şi x_2. Veţi nota cu o_i (unde $i = 1, \ldots, 4$) ieşirile unităţilor de pe nivelul ascuns şi cu o ieşirea reţelei, adică valoarea produsă de către unitatea de pe nivelul de ieşire.

b. Calculaţi outputul reţelei atunci când intrările x_1 şi x_2 iau valori în mulţimea $\{-1, 1\}$.

c. Indicaţi funcţiile booleene reprezentate de către uităţile de pe nivelul ascuns (1, 2, 3 şi 4) atunci când $x_1, x_2 \in \{-1, 1\}$. Procedaţi similar pentru ieşirea o.

d. Specificaţi cum anume ar putea fi modificată reţeaua dată astfel încât noua variantă să calculeze funcţia de variabile reale (nu booleene ca mai înainte!) x_1 şi x_2, a cărei relaţie de definiţie este: $f(x_1, x_2) = 1$ dacă $x_1, x_2 \geq 0$ sau $x_1, x_2 < 0$, şi -1 în caz contrar.

Răspuns:

a. Sunt imediate următoarele relaţii:

$$\begin{aligned}
o_1(x_1, x_2) &= sign(x_1 + x_2 - 1),\\
o_2(x_1, x_2) &= sign(x_1 - x_2 - 1),\\
o_3(x_1, x_2) &= sign(-x_1 + x_2 - 1),\\
o_4(x_1, x_2) &= sign(-x_1 - x_2 - 1),\\
o(x_1, x_2) &= sign(o_1(x_1, x_2) - o_2(x_1, x_2) - o_3(x_1, x_2) + o_4(x_1, x_2)).
\end{aligned}$$

b. Date fiind formulele de la punctul precedent, calculele cerute sunt simple; centralizăm rezultatele sub forma tabelului următor:

x_1	x_2	o_1	o_2	o_3	o_4	o
1	1	1	−1	−1	−1	1
1	−1	−1	1	−1	−1	−1
−1	1	−1	−1	1	−1	−1
−1	−1	−1	−1	−1	1	1

c. Din tabelul obţinut la punctul precedent este imediat că, atunci când $x_1, x_2 \in \{-1, 1\}$, ieşirile calculate de către unităţile 1, 2, 3 şi 4 corespund conceptelor / funcţiilor $x_1 \wedge x_2$, $x_1 \wedge \neg x_2$, $\neg x_1 \wedge x_2$ şi respectiv $\neg x_1 \wedge \neg x_2$. Ieşirea reţelei, o, corespunde conceptului $\neg(x_1 \text{ XOR } x_2)$.

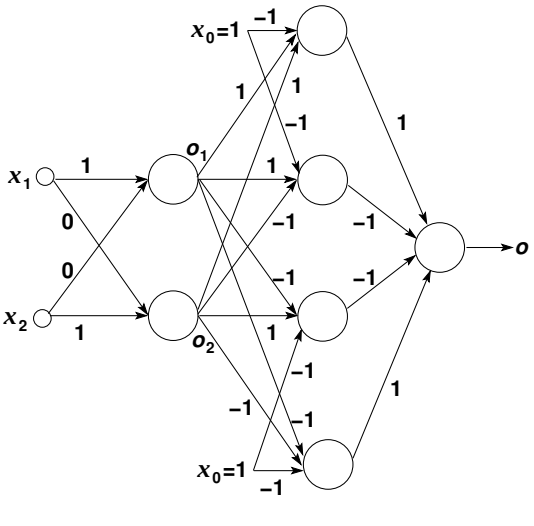
***Observaţie*: Se poate remarca faptul că în această reţea un neuron (şi doar unul!) de pe nivelul ascuns este activat la fiecare combinaţie de valori (din cele 4 posibile) pentru $x_1, x_2 \in \{-1, 1\}$.**

d. Este imediat că funcţia indicată în enunţ se poate scrie sub forma

$$f(x_1, x_2) = \begin{cases} 1 & \text{dacă } sign(x_1) \cdot sign(x_2) \geq 0 \\ -1 & \text{dacă } sign(x_1) \cdot sign(x_2) < 0. \end{cases}$$

Se observă imediat că aceasta coincide cu funcţia $\neg(sign(x_1) \text{ XOR } sign(x_2))$.

Prin urmare, este suficient să adăugăm la reţeaua dată în enunţ un nivel ascuns suplimentar, format din două unităţi cu funcţie de activare de tip prag, care să transforme intrările x_1 şi x_2 în $sign(x_1)$ şi respectiv $sign(x_2)$. Vom obţine ca rezultat reţeaua din figura alăturată.



***Observaţie*: La problema 31 vi se va cere să demonstraţi că această funcţie de variabile reale nu poate fi calculată de nicio reţea neuronală care are un singur nivel ascuns şi este compusă doar din unităţi [cu funcţie de activare] de tip prag.**

2. (Reprezentarea unor funcţii booleene cu ajutorul perceptronilor-prag sau al reţelelor de perceptroni-prag)

- ***Tom Mitchell, "Machine Learning", 1997, pr. 4.2***
  CMU, 1995 fall, Tom Mitchell, HW4, pr. 6

a. Concepeţi un perceptron care are

- **funcţia de activare de tip prag, cu valori -1 şi +1,**
- **două intrări**

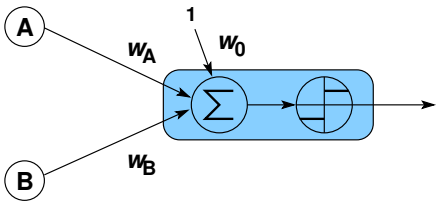
şi implementează funcţia booleană $A \wedge (\neg B)$.

b. Concepeţi o reţea neuronală formată din perceptroni cu funcţia de activare de tip prag dispuşi pe două niveluri, care implementează funcţia A XOR B.

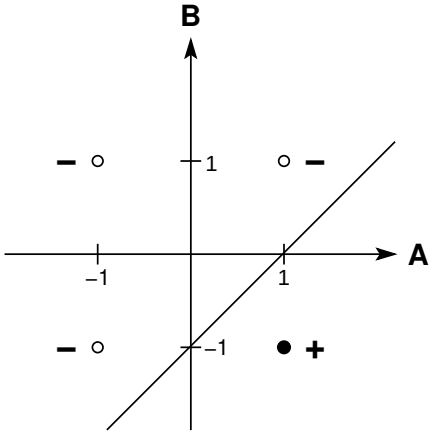
Răspuns:

***Observaţie*: Deoarece mulţimea de valori de ieşire a perceptronului este $\{-1, 1\}$, vom alege această codificare a valorilor fals / adevărat în locul celei clasice $0/1$.**

a. Perceptronul-prag are structura din figura alăturată. A îndeplini cerinţa din enunţ revine la a alege în mod convenabil valori pentru ponderile w_0, w_A şi w_B.[766]



Un perceptron consistent cu funcţia booleană $A \wedge (\neg B)$ poate fi reprezentat ca o dreaptă ce separă instanţele pozitive de cele negative, ca în desenul alăturat. O astfel de dreaptă are ecuaţia $d(A, B) = 0$, unde $d(A, B) = w_0 + w_A A + w_B B$.
Este evident că sunt o infinitate de funcţii care îndeplinesc proprietatea de separator liniar pentru cele două mulţimi, însă noi avem nevoie doar de una singură. Putem alege în mod convenabil / preferenţial două puncte prin care să treacă dreapta, de pildă ca în figura alăturată. Aşadar, *analitic*, considerăm d astfel încât $(1, 0) \in d$ şi $(0, -1) \in d$.



O astfel de alegere va impune anumite *restricţii* asupra valorilor w_0, w_A şi w_B. Într-adevăr, ţinând cont de ecuaţia dreptei d scrisă mai sus (în sens generic), va rezulta următorul sistem:

$$\begin{cases} (1,0) \in d \\ (0,-1) \in d \end{cases} \Rightarrow \begin{cases} w_0 + w_A = 0 \\ w_0 - w_B = 0 \end{cases} \Rightarrow \begin{cases} w_0 = -w_A \\ w_0 = w_B \end{cases}$$

Avem deci $w_0 = w_B \overset{not.}{=} \alpha, w_A = -\alpha$, unde $\alpha \in \mathbb{R}^*$. Prin urmare, $d(A, B) = \alpha - \alpha A + \alpha B$.

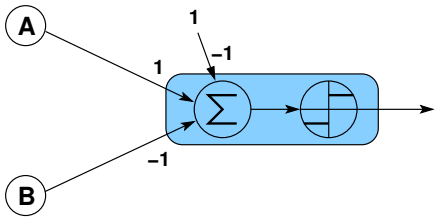
Rămâne să mai analizăm în ce condiţii dreapta d clasifică pozitiv instanţa $(1, -1)$ şi negativ instanţele $(1, 1)$ $(-1, 1)$ şi $(-1, -1)$. De fapt, ţinând cont de o proprietate din geometria analitică,[767] **este suficient să impunem una (de exemplu, prima) dintre aceste condiţii. Aceasta revine la *restricţia* ca expresia $\alpha - \alpha A + \alpha B$ să fie pozitivă pentru $A = 1$ şi $B = -1$:**

$$d(1, -1) > 0 \Leftrightarrow \alpha - \alpha - \alpha > 0 \Leftrightarrow \alpha < 0$$

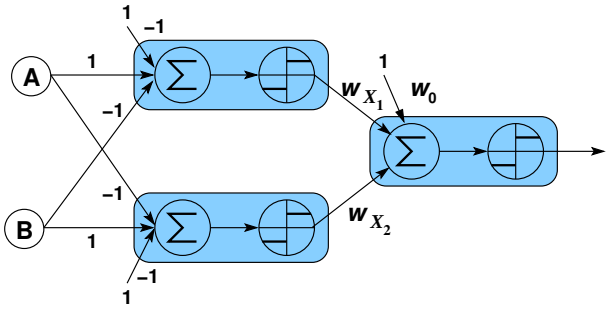
[766] Am putea satisface direct această cerinţă dacă ne raportăm la problema 1.c, în care $o_2(x_1, x_2) = x_1 \wedge \neg x_2$, deci am putea seta $w_0 = -1$, $w_A = 1$ şi $w_B = -1$. Totuşi, aici vom proceda independent de problema 1, arătând cum anume se rezolvă [în general] un exerciţiu de acest tip.

[767] Este vorba despre următoarea proprietate din geometria plană: Toate punctele situate de o parte a dreptei de ecuaţie $w_0 + w_A A + w_B B$ au acelaşi semn, în vreme ce punctele situate de cealaltă parte a dreptei au semn contrar.

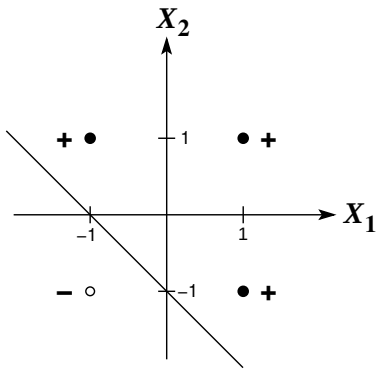
Pentru fixare, alegem $\alpha = -1$, şi vom avea $w_0 = -1, w_A = 1, w_B = -1$. Aşadar, perceptronul (sau mai bine spus: un perceptron) care implementează funcţia $A \wedge (\neg B)$ arată ca în figura alăturată.



b.[768] **Ştim că A XOR $B = (A \wedge (\neg B)) \vee (B \wedge (\neg A))$. Această formulă poate fi reprezentată printr-o reţea de perceptroni cu un nivel ascuns, ca în figura alăturată.**



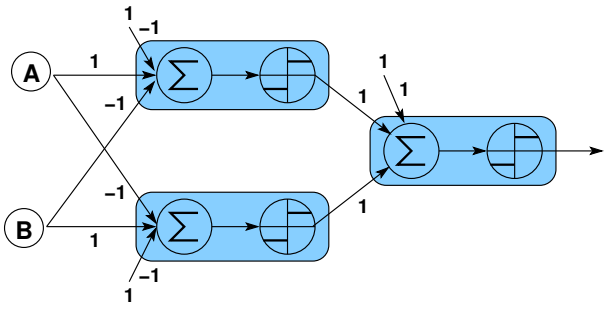
Perceptronii de pe nivelul de ascuns codifică funcţia $A \wedge (\neg B)$ (vedeţi punctul anterior) şi respectiv $(\neg A) \wedge B$, după cum se poate vedea după ponderile de pe muchii, iar perceptronul de ieşire trebuie să descrie funcţia logică $\vee$. *Suprafaţa de decizie* pentru aceasta din urmă este precum cea din figura alăturată.
Avem $d'(X_1, X_2) = w_0 + w_{X_1}X_1 + w_{X_2}X_2 = 0$. Fie d' dreapta care trece prin punctele (0,-1) şi (-1,0). Atunci:



$$\begin{cases} (0,-1) \in d' \\ (-1,0) \in d' \end{cases} \Rightarrow \begin{cases} w_0 - w_{X_2} = 0 \\ w_0 - w_{X_1} = 0 \end{cases} \Rightarrow w_0 = w_{X_1} = w_{X_2} \overset{not.}{=} \alpha' \in \mathbb{R}^* \Rightarrow$$

$$\Rightarrow d'(X_1, X_2) = \alpha' + \alpha' X_1 + \alpha' X_2 = 0.$$

Impunând condiţia referitoare la semne, vom avea $d'(-1,-1) < 0 \Leftrightarrow \alpha' - \alpha' - \alpha' < 0 \Leftrightarrow \alpha' > 0$. Pentru fixare, alegem $\alpha' = 1$, ceea ce duce la $w_0 = w_{X_1} = w_{X_2} = 1$, iar reţeaua neuronală va arăta ca în figura alăturată.



[768] *Observaţie*: Am putea obţine o soluţie imediată pentru acest punct al problemei noastre dacă preluăm reţeaua neuronală dată în enunţul problemei 1 — despre care ştim (vedeţi rezolvarea respectivei probleme) că are outputul $\neg(x_1 \text{ XOR } x_2)$ — şi schimbăm semnele tuturor ponderilor de pe arcele hidden-to-output din reţea. Vom arăta însă aici cum se poate construi „de la zero“ o astfel de reţea, pornind de la cerinţele specificate în enunţ. În plus, se va vedea la final că noua reţea este mai simplă decât cea de la problema 1. (Aşadar, reţele neuronale diferite pot codifica / reprezenta o aceeaşi funcţie reală. Evident, este de dorit ca, pentru o funcţie dată, reţeaua neuronală care o codifică să fie cât mai simplă.)

3. **(Reţele neuronale: exemplificare)**

• ***CMU, 2011 spring, Roni Rosenfeld, HW4, pr. 1.c***

Să se reprezinte expresia booleană $(A \vee \neg B)$ XOR $(\neg C \vee D)$ printr-o reţea neuronală cu două niveluri care este formată din neuroni cu funcţie de activare de tip prag.

Răspuns:

***Observaţie*: Este imediat — vedeţi rezolvările problemelor 1 şi / sau 2 — că se poate defini câte un perceptron cu funcţie de activare de tip prag (engl., threshold perceptron) care să reprezinte funcţiile $(A \vee \neg B)$ şi respectiv $(\neg C \vee D)$. Dacă ieşirile acestor doi perceptroni vor fi cuplate la intrările reţelei care reprezintă funcţia A XOR B (a se vedea exerciţiul 2, punctul b), atunci se va obţine o reţea cu *trei* niveluri care reprezintă funcţia $(A \vee \neg B)$ XOR $(\neg C \vee D)$. Exerciţiul nostru cere însă să identificăm o reţea cu (doar!) *două* niveluri care să reprezinte această funcţie.**

Explicitând definiţia funcţiei logice XOR şi apoi aplicând regulile lui DeMorgan, expresia booleană din enunţ devine:

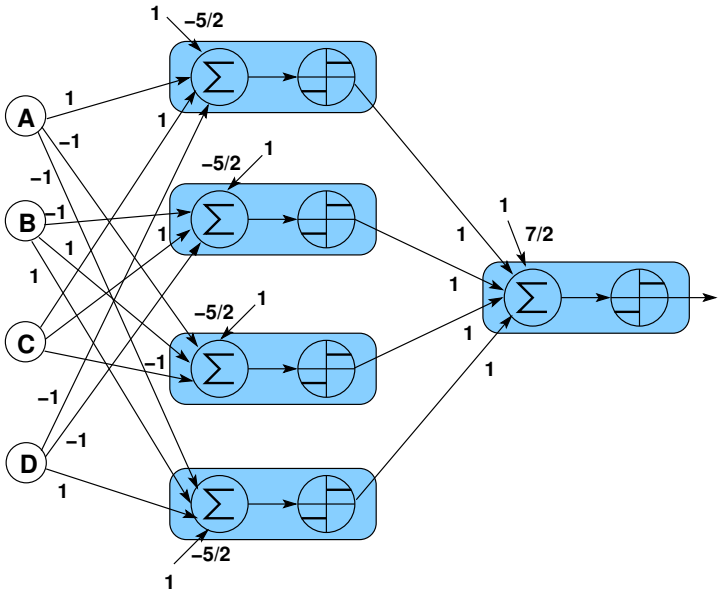
$$\begin{aligned}(A \vee \neg B) \text{ XOR } (\neg C \vee D) &= [(A \vee \neg B) \wedge \neg(\neg C \vee D)] \vee [\neg(A \vee \neg B) \wedge (\neg C \vee D)] \\ &= [(A \vee \neg B) \wedge (C \wedge \neg D)] \vee [(\neg A \wedge B) \wedge (\neg C \vee D)]\end{aligned}$$

Întrucât operatorii logici $\vee$ şi $\wedge$ sunt distributivi unul faţă de celălalt, rezultă că:

$$(A \vee \neg B) \text{ XOR } (\neg C \vee D) = (A \wedge C \wedge \neg D) \vee (\neg B \wedge C \wedge \neg D) \vee (\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge B \wedge D)$$

Fiecare din cele patru paranteze din partea dreaptă a egalităţii de mai sus poate fi reprezentată cu ajutorul unui perceptron-prag cu patru intrări, şi anume câte o intrare pentru fiecare din cele trei variabile booleene din paranteză, plus o intrare pentru termenul liber. (De exemplu, conjuncţiei $A \wedge C \wedge \neg D$ îi putem asocia inegalitatea $x + z - t > 5/2$, deci ponderile intrărilor perceptronului corespunzător vor fi $-5/2$, 1, 1 şi -1.) Cei patru perceptroni vor fi plasaţi pe primul nivel ascuns al reţelei pe care o construim.

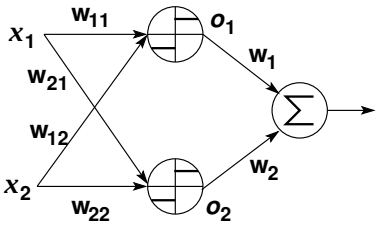
Apoi, ieşirile acestor patru perceptroni vor constitui intrări pentru un alt perceptron-prag, care să reprezinte disjuncţia a patru variabile booleene. (Acestui perceptron, situat pe nivelul de ieşire, îi putem asocia, de exemplu, inegalitatea $x + y + z + t > -7/2$.) Reţeaua neuronală rezultată este cea din figura alăturată.



4. **(Verificarea (im)posibilităţii de a reprezenta funcţia booleană XOR cu o reţea de structură / compoziţie specificată)**

CMU, 2001 fall, Andrew Moore, final exam, pr. 10.a

Presupunem că lucrăm cu reţeaua neuronală din figura alăturată,[769] **care are un nivel ascuns format din doi perceptroni cu funcţia de activare de tip prag: *sign*$(z) = 1$ dacă $z \geq 0$ şi *sign*$(z) = -1$ dacă $z < 0$.**



Folosim setul de date de antrenament din tabelul alăturat, care reprezintă funcţia booleană *sau-exclusiv* dacă interpretăm -1 ca desemnând valoarea logică *fals*, şi 1 ca *adevărat*.

X_1	X_2	Y
1	1	-1
1	-1	1
-1	1	1
-1	-1	-1

Vă cerem să indicaţi ce valori putem atribui ponderilor reţelei date, astfel încât să obţinem eroare nulă la antrenare. Dacă este imposibil să se găsească astfel de valori pentru ponderile reţelei, daţi răspunsul *Imposibil*.

***Observaţie*: În raport cu problema 2.b, remarcaţi faptul că aici s-a eliminat termenul liber $x_0 = 1$ de la toţi neuronii, precum şi funcţia de activare (de tip prag) de la neuronul de pe nivelul de ieşire.**

Răspuns:

Funcţiile calculate de către cele trei unităţi neuronale sunt:

$$\begin{aligned} o_1(x_1, x_2) &= sign(w_{11}x_1 + w_{12}x_2) \\ o_2(x_1, x_2) &= sign(w_{21}x_1 + w_{22}x_2) \\ o(x_1, x_2) &= w_1 o_1(x_1, x_2) + w_2 o_2(x_1, x_2) \\ &= w_1 sign(w_{11}x_1 + w_{12}x_2) + w_2 sign(w_{21}x_1 + w_{22}x_2) \end{aligned}$$

Aşadar,

$$o(1,1) = w_1 sign(w_{11} + w_{12}) + w_2 sign(w_{21} + w_{22}) = -1 \quad (353)$$

$$o(1,-1) = w_1 sign(w_{11} - w_{12}) + w_2 sign(w_{21} - w_{22}) = 1 \quad (354)$$

$$o(-1,1) = w_1 sign(-w_{11} + w_{12}) + w_2 sign(-w_{21} + w_{22}) = 1 \quad (355)$$

$$o(-1,-1) = w_1 sign(-w_{11} - w_{12}) + w_2 sign(-w_{21} - w_{22}) = -1 \quad (356)$$

Se poate arăta uşor că $sign(x) = -sign(-x)$ pentru orice $x \in \mathbb{R}\backslash\{0\}$. Aşadar, dacă $-w_{11} + w_{12} \neq 0$ (ceea ce este echivalent cu $w_{11} \neq w_{12}$) şi $-w_{21} + w_{22} \neq 0$ (echivalent cu $w_{21} \neq w_{22}$), relaţia (355) implică

$$w_1 sign(w_{11} - w_{12}) + w_2 sign(w_{21} - w_{22}) = -1,$$

[769] *Observaţie importantă*: Pentru comoditate, vom opta aici — dar şi în [multe dintre] problemele care urmează — pentru o reprezentare simplificată a unităţilor neuronale (comparativ cu reprezentările din problemele precedente). În figura dată, se va subînţelege că neuronii de pe nivelul ascuns au (ca întotdeauna) o componentă de sumare (care aici nu este pusă în evidenţă), iar unitatea de pe nivelul de ieşire nu are o componentă de activare (deci este unitate liniară).

ceea ce intră în contradicţie evidentă cu relaţia (354) scrisă mai sus.

Similar, dacă $w_{11} \neq -w_{12}$ şi $w_{21} \neq -w_{22}$, relaţia (356) implică

$$w_1 sign(w_{11} + w_{12}) + w_2 sign(w_{21} + w_{22}) = 1,$$

ceea ce contravine relaţiei (353).

Sumarizând cele scrise mai sus, dacă ponderile w_{ij}, cu $i, j \in \{1, 2\}$ satisfac condiţia multiplă

$$w_{11} \neq w_{12} \text{ şi } w_{21} \neq w_{22},$$
$$\text{sau}$$
$$w_{11} \neq -w_{12} \text{ şi } w_{21} \neq -w_{22},$$

răspunsul la problema noastră este: *Imposibil.*

Rămân de analizat următoarele cazuri, derivate din negarea condiţiei multiple de mai sus:

- ***Cazul particular 1*: $w_{11} = w_{12}$ şi $w_{11} = -w_{12}$ (de unde rezultă $w_{11} = w_{12} = 0$);**
- ***Cazul particular 2*: $w_{11} = w_{12}$ şi $w_{21} = -w_{22}$;**
- ***Cazul particular 3*: $w_{21} = w_{22}$ şi $w_{11} = -w_{12}$;**
- ***Cazul particular 4*: $w_{21} = w_{22}$ şi $w_{21} = -w_{22}$ (de unde rezultă $w_{21} = w_{22} = 0$).**

Cazul 4 este similar cazului 1, iar cazul 3 este similar cazului 2. Se poate arăta relativ uşor (deşi nu este chiar imediat) că în fiecare dintre aceste cazuri particulare obţinem acelaşi răspuns: *Imposibil.*

5. **(Expresivitatea reţelelor neuronale: funcţii booleene)**

■ • ○ ***CMU, 2010 fall, Aarti Singh, HW5, pr. 4.3***
CMU, 2011 spring, Roni Rosenfeld, HW4, pr. 1.d

Să se arate că orice funcţie booleană (cu n argumente / variabile booleene şi câte o singură valoare booleană pentru fiecare combinaţie posibilă de valori pentru cele n argumente), poate fi reprezentată printr-o reţea neuronală cu doar două niveluri, folosind neuroni care au componenta de activare de tip funcţie-prag.

Răspuns:

Ştim că orice funcţie booleană poate fi reprezentată în mod echivalent ca o expresie din logica propoziţiilor. La rândul ei, aceasta se poate scrie ca o conjuncţie de disjuncţii de literali (sau invers, ca o disjuncţie de conjuncţii de literali), un literal fiind fie o variabilă fie negaţia ei.[770] Pentru exemplificare, vedeţi rezolvarea problemei 3. (Demonstraţia dată aici poate fi văzută ca o generalizare naturală a exemplului particular de acolo.) Pentru a reprezenta o astfel de expresie cu ajutorul reţelelor de perceptroni-prag vom proceda astfel:

[770] Pentru fiecare combinaţie de valori $l_1, \ldots, l_n$ ale variabilelor $X_1, \ldots, X_n$ pentru care funcţia booleană dată (f) ia valoarea de adevăr T, se va considera conjuncţia $L_1 \wedge \ldots \wedge L_n$, unde L_i este X_i dacă $l_i = T$, şi L_i este $\neg X_i$ dacă $l_i = F$. Expresia booleană care reprezintă funcţia f este disjuncţia de conjuncţii astfel calculate.

• **Fiecare disjuncţie de literali va fi reprezentată prin câte un perceptron-prag plasat pe nivelul ascuns. Ponderile pentru acest perceptron vor fi:**

1 **pentru literalii pozitivi,**

-1 **pentru literalii negativi,**

$k - \frac{1}{2}$ **pentru termenul liber** ($x_0 = 1$)**, unde** k **este numărul de literali din disjuncţie.**

• **Pentru conjuncţia de disjuncţii, vom pune un perceptron-prag pe nivelul exterior şi-i vom asigna următoarele ponderi:**

1 **pentru fiecare conexiune hidden-to-output,**

$-(l - \frac{1}{2})$ **pentru termenul liber, unde** l **este numărul de disjuncţii, acelaşi cu numărul de perceptroni-prag de pe nivelul ascuns.**

***Observaţii*:**

1. În mod absolut similar se poate proceda pornind de la rezultatul, cunoscut din logica propoziţiilor, că orice expresie booleană se poate scrie ca o conjuncţie de disjuncţii.

2. Este posibil ca în reţeaua obţinută, numărul de unităţi de pe nivelul ascuns să fie exponenţial în raport cu numărul de variabile de intrare (n)**. Aşadar, în astfel de cazuri, pentru valori mari ale lui** n**, rezultatul care a fost demonstrat la acest exerciţiu, deşi are o semnificaţie teoretică importantă, nu are aplicabilitate practică.**

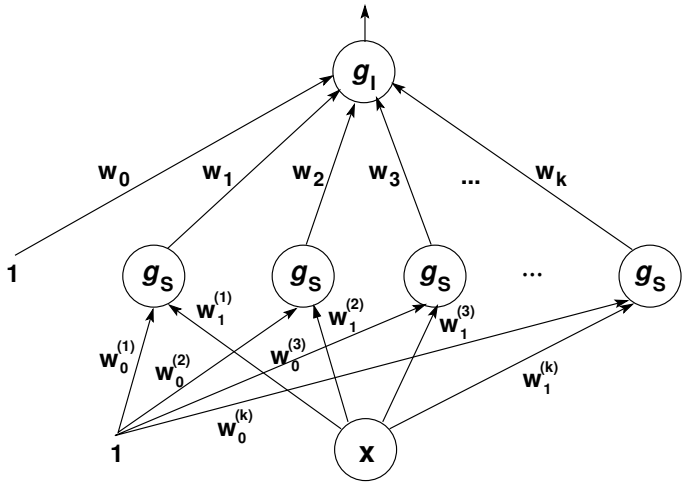
6. **(Reprezentarea unor funcţii de tip treaptă cu ajutorul unor reţele cu unităţi liniare sau de tip prag)**

CMU, 2007 fall, Carlos Guestrin, HW2, pr. 4

În această problemă presupunem că dispunem (doar) de unităţi neuronale ale căror funcţii de activare pot fi de două tipuri:

- **funcţia identitate:** $g_I(x) = x$**, şi**
- **funcţia treaptă:** $g_S(x) = 1$ **dacă** $x \geq 0$ **şi** 0 **în rest.**

De *exemplu*, reţeaua neuronală din figura alăturată are o intrare $x \in \mathbb{R}$**, un singur nivel ascuns pe care sunt** k **unităţi având funcţia de activare de tip treaptă, iar pe nivelul de ieşire un singur perceptron având funcţia de activare de tip identitate.**



Ieşirea acestei reţele poate fi descrisă astfel:

$$out(x) = g_I\left(w_0 + \sum_{i=1}^{k} w_i\, g_S\left(w_0^{(i)} + w_1^{(i)} x\right)\right) = w_0 + \sum_{i=1}^{k} w_i\, g_S\left(w_0^{(i)} + w_1^{(i)} x\right)$$

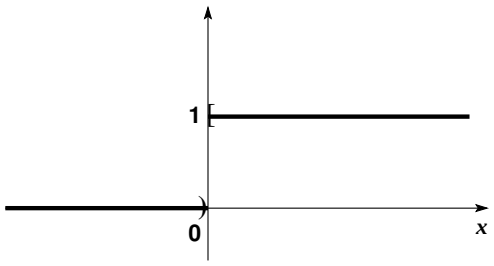
a. Considerăm funcţia de tip treaptă care este definită prin expresia $u(x) = c$ când $x < a$ şi 0 în rest (unde a şi c sunt constante reale fixate). Construiţi o reţea neuronală având un singur nivel ascuns, intrarea x şi outputul $u(x)$. Desenaţi structura reţelei neuronale, indicând funcţia de activare pentru fiecare unitate (fie g_I fie g_S), şi specificaţi valorile tuturor ponderilor (ca expresii în funcţie de a şi c).

b. Se consideră constantele reale (fixate) a, b şi c. Construiţi o reţea neuronală având o intrare x şi un nivel ascuns, al cărei output este c dacă $x \in [a, b)$ şi 0 în rest. Desenaţi structura reţelei neuronale, indicând funcţia de activare pentru fiecare unitate (fie g_I fie g_S), şi specificaţi valorile tuturor ponderilor (ca expresii în funcţie de a, b şi c).

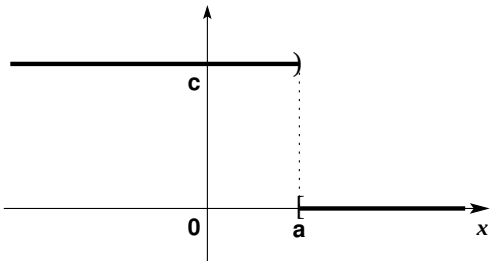
Răspuns:

a. Pentru convenienţă, vom reprezenta mai întâi graficele funcţiilor treaptă g_S şi u din enunţul problemei.

Graficul funcţiei $g_s(x)$:

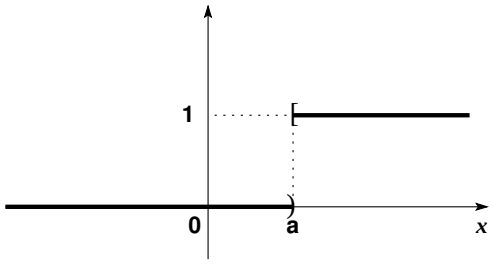


Graficul funcţiei $u(x)$:

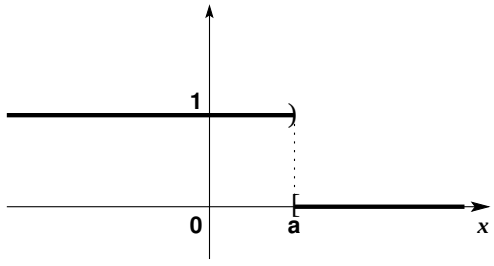


Pornind de la aceste reprezentări grafice, putem să obţinem în câţiva paşi funcţia treaptă $u(x)$ ca funcţie compusă, exprimată cu ajutorul lui g_S, după cum urmează:

Graficul funcţiei $g_S(x - a)$:



Graficul funcţiei $1 - g_S(x - a)$:

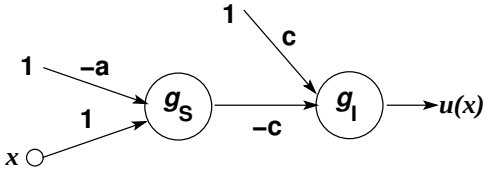


Comparând ultimul grafic de mai sus cu graficul funcţiei u, rezultă că $u(x) = c(1 - g_S(x - a)) \Rightarrow u(x) = c - c \cdot g_S(x - a)$.

Pe de altă parte, putem observa că o reţea neuronală care are o singură unitate pe nivelul ascuns având funcţia de activare de tip treaptă şi o singură unitate

pe nivelul de ieşire având funcţia de activare de tip identitate va avea ieşirea $out(x) = g_I(w_0 + w_1 \cdot g_S(w_0^{(1)} + w_1^{(1)} x))$**, adică** $out(x) = w_0 + w_1 \cdot g_S(w_0^{(1)} + w_1^{(1)} x)$**.**

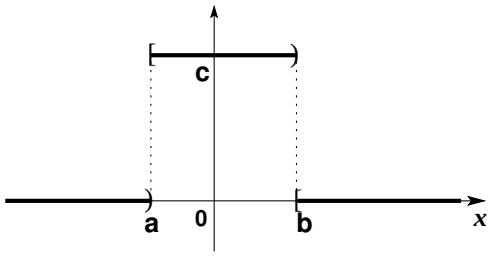
Prin urmare, ponderile reţelei neuronale care reprezintă funcţia u **vor fi:** $w_0 = c$**,** $w_1 = -c$**,** $w_0^{(1)} = -a$ **şi** $w_1^{(1)} = 1$**, iar reţeaua însăşi va arăta ca în figura alăturată.**



b. Funcţia

$$v(x) = \begin{cases} c & \text{dacă } x \in [a, b) \\ 0 & \text{în rest} \end{cases}$$

este reprezentată grafic în figura alăturată. Se poate observa din acest grafic că funcţia v **se obţine uşor din funcţia** u **de la punctul anterior.**



Într-adevăr, dacă notăm

$$u_a(x) = \begin{cases} c & \text{dacă } x < a \\ 0 & \text{în rest,} \end{cases}$$

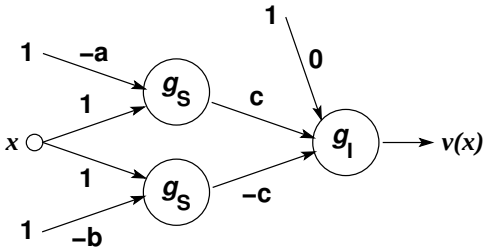
atunci

$$v(x) = u_b(x) - u_a(x) = c(1 - g_S(x - b)) - c(1 - g_S(x - a)).$$

Calculând, obţinem $v(x) = c - c \cdot g_S(x - b) - c + c \cdot g_S(x - a)$

$$\Rightarrow v(x) = c \cdot g_S(x - a) - c \cdot g_S(x - b)$$

Aşadar, putem construi o reţea neuronală cu două unităţi pe nivelul ascuns (fiecare având funcţia de activare de tip treaptă). Ieşirile acestor două unităţi vor constitui intrările unei unităţi [cu funcţie de activare] de tip liniar, aflată pe nivelul de ieşire. Setând corespunzător ponderile corespunzătoare acestui nivel, outputul reţelei va fi chiar $v(x)$**. Reprezentarea acestei reţele este cea din figura alăturată.**



7. **(Orice funcţie definită pe un interval mărginit din** $\mathbb{R}$**, care este continuă în sens Lipschitz, poate fi aproximată oricât de bine cu ajutorul unei reţele neuronale care are un singur nivel ascuns)**

■ ● ○ ***CMU, 2011 fall, T. Mitchell, A. Singh, HW5, pr. 2.3***

Considerăm o funcţie oarecare $f(x)$**, al cărei domeniu este de forma** $[C, D) \subset \mathbb{R}$**. Presupunem că** f ***este continuă în sens Lipschitz,***[771] **adică există o constantă**

[771] Continuitatea în sens Lipschitz este o formă [mai] tare de *uniform-continuitate*: se poate demonstra că orice funcţie continuă în sens Lipschitz este uniform continuă (şi, deci, continuă).

$L \geq 0$ **astfel încât**

$$|f(x') - f(x)| \leq L|x' - x|, \ \forall x, x' \in [C, D).$$

Folosiţi rezultatele obţinute la problema 6 pentru a construi o reţea neuronală cu un singur nivel ascuns, care aproximează funcţia f cu o eroare de cel mult $\varepsilon > 0$, adică

$$\forall x \in [C, D), |f(x) - \textbf{\textit{out}}(x)| \leq \varepsilon,$$

unde $out(x)$ desemnează ieşirea acestei reţele neuronale pentru intrarea x.

Va trebui ca reţeaua să folosească doar funcţiile de activare de tip identitate şi respectiv prag (notate cu g_I şi g_S în problema 6). Indicaţi

– numărul K de unităţi ascunse,

– funcţia de activare pentru fiecare unitate,

– o formulă pentru calcularea fiecărei ponderi de pe conexiunile input-to-hidden (notate cu $w_0^{(k)}$ şi $w_1^{(k)}$, unde $k \in \{1, 2, \ldots, K\}$), precum şi pentru conexiunile hidden-to-output (şi anume w_0, w_k, cu $k \in \{1, 2, \ldots, K\}$).

Aceste ponderi pot fi specificate în funcţie de C, D, L şi ε, precum şi în funcţie de valorile $f(x)$ obţinute pentru un număr finit de valori x, la libera alegere. (Va trebui să specificaţi în mod explicit care sunt valorile x pe care le veţi folosi.)

Nu vi se cere să scrieţi în mod explicit funcţia $out(x)$.

Cum justificaţi faptul că reţeaua concepută de dumneavostră atinge acurateţea indicată?

Răspuns:

Este imediat că din ipoteza $|f(x) - f(x')| \leq L|x - x'|$, în cazul în care $|x - x'| \leq \frac{\varepsilon}{L}$, rezultă

$$|f(x) - f(x')| \leq \varepsilon \tag{357}$$

Aşadar, este de dorit să „acoperim" intervalul $[C, D)$ cu intervale de lungime $\frac{2\varepsilon}{L}$ (sau mai mică):

$$[C, C + \frac{2\varepsilon}{L}), \ [C + \frac{2\varepsilon}{L}, C + 2\frac{2\varepsilon}{L}), \ \ldots, \ [C + (K-1)\frac{2\varepsilon}{L}, D))$$

şi să luăm punctele x' de forma $\xi_i = (\alpha_i + \beta_i)/2$, pentru cu $i = 1, \ldots, K$, unde

$$K \stackrel{not.}{=} \left\lceil (D - C)\frac{L}{2\varepsilon} \right\rceil,$$

$$\alpha_1 = C, \ \beta_1 = \alpha_2 = C + \frac{2\varepsilon}{L}, \ldots, \beta_{K-1} = \alpha_K = C + (K-1)\frac{2\varepsilon}{L}, \ \beta_K = D.$$

Aşadar, prin ξ_i am notat mijlocul intervalului i de mai sus, $[\alpha_i, \beta_i)$.

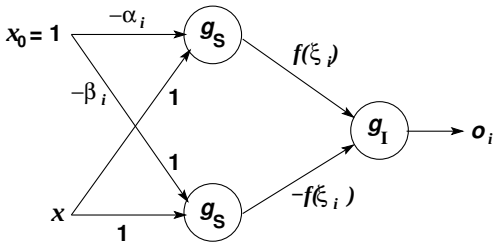
În continuare, vom arăta că putem construi o reţea neuronală cu un singur nivel ascuns şi care foloseşte doar unităţi de tip liniar sau de tip prag, astfel încât

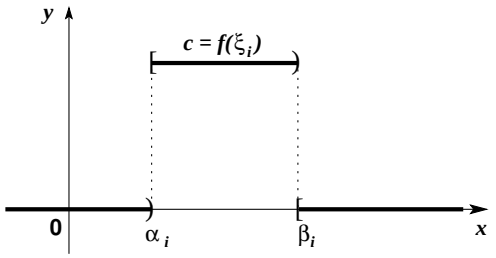
$$\textbf{pentru } \forall x \in [C, D), \ \textbf{dacă } x \in [\alpha_i, \beta_i), \ \textbf{atunci } \textbf{\textit{out}}(x) \stackrel{def.}{=} f(\xi_i). \tag{358}$$

În consecinţă, dacă în relaţia (357) vom înlocui x' cu ξ_i, va rezulta imediat că $|f(x) - \mathit{out}(x)| \leq \varepsilon$.

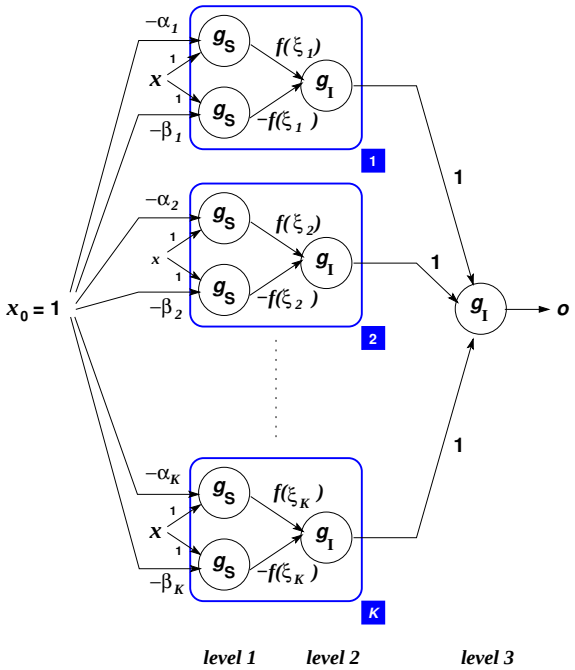
Revenind acum la proprietatea (358), vom arăta că există o reţea având K unităţi pe nivelul ascuns (şi o singură unitate de ieşire), astfel încât, dacă $x \in [\alpha_i, \beta_i)$, atunci unitatea i de pe nivelul ascuns se activează (producând valoarea $f(\xi_i)$, iar toate celelalte unităţi de pe nivelul ascuns rămân neactivate (adică produc output 0). Prin urmare, rezultatul va fi exact cel dorit.

Conform problemei 6 punctul b, există o reţea neuronală care produce valoarea $c = f(\xi_i)$ pentru orice input $x \in [\alpha_i, \beta_i)$ şi 0 în rest, şi care foloseşte funcţii de activare g_S pe nivelul ascuns şi g_I pe nivelul de ieşire:



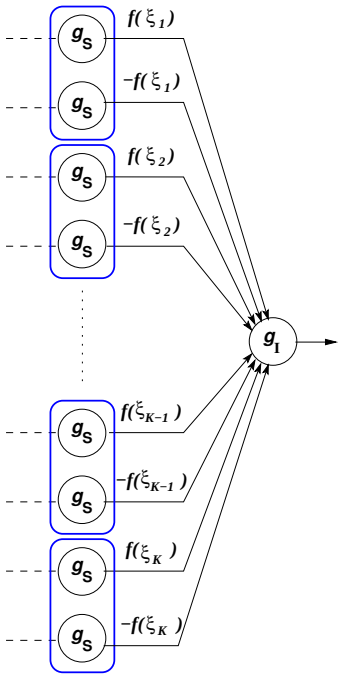


„Ansamblând“ K astfel de reţele, vom obţine reţeaua din figura alăturată.

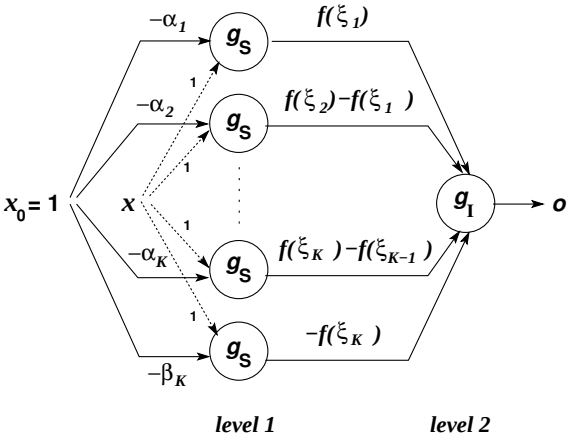


Outputul acestei reţele este exact cel dorit (deci satisface condiţia din enunţ):

$$\mathit{out}(x) = \begin{cases} f(\xi_1) & \textbf{pt.} \quad x \in [\alpha_1 = C, \beta_1) \\ f(\xi_2) & \textbf{pt.} \quad x \in [\alpha_2 = \beta_1, \beta_2) \\ \dots & \dots \quad \dots \\ f(\xi_K) & \textbf{pt.} \quad x \in [\alpha_K = \beta_{K-1}, \beta_K = D) \end{cases} \Rightarrow |f(x) - \mathit{out}(x)| \leq \varepsilon, \ \forall x \in [C, D).$$



Neajunsul **este că reţeaua aceasta are două niveluri ascunse, în loc de unul singur, aşa cum se cere în enunţ. Totuşi, el poate fi „corectat“ imediat, comasând nivelurile 2 şi 3 — formate doar din unităţi liniare — într-o singură unitate liniară (care va constitui nivelul de ieşire al noii reţele), ca în figura alăturată.**[772]



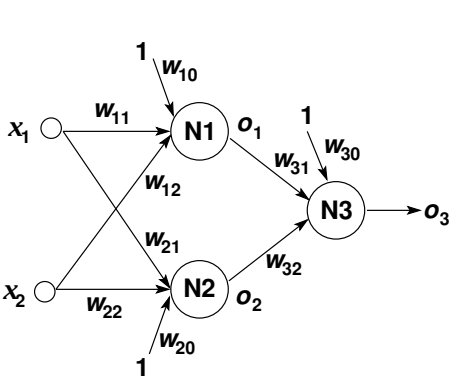
De asemenea, datorită faptului că $\beta_1 = \alpha_2, \beta_2 = \alpha_3, \ldots, \beta_{K-1} = \alpha_K$, nivelul ascuns (al acestei noi reţele), format din cele $2K$ unităţi de tip prag, poate fi echivalat cu un altul, compus din doar $K + 1$ unităţi de tip prag. La final, obţinem reţeaua din figura alăturată.

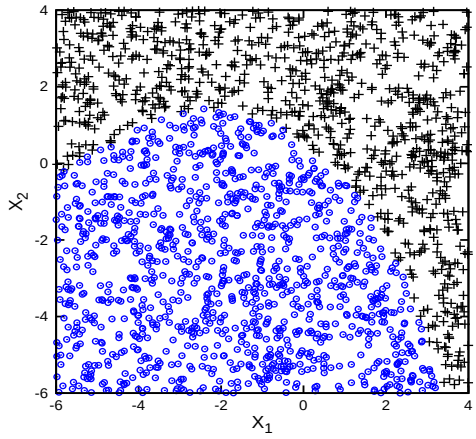
8. **(Reţele neuronale cu unităţi de tip sigmoidal: graniţe / suprafeţe de decizie)**

CMU, 2008 fall, Eric Xing, HW2, pr. 2.1
CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, HW3, pr. 2.1
CMU, 2011 fall, Eric Xing, HW1, pr. 3.2

Una dintre chestiunile importante pe care trebuie să ni le punem atunci când facem cunoştinţă cu un nou clasificator este *ce tip de suprafeţe de decizie* (sau: graniţe de decizie; engl., decision boundaries) *poate să înveţe respectivul clasificator*.

[772] Vedeţi proprietatea enunţată la problema 32.b.





Fie reţeaua neuronală din figura de mai sus, partea stângă. Cele trei unităţi din componenţa acestei reţele sunt de tip sigmoidal, adică au funcţia de activare $\sigma(x) = \dfrac{1}{1+e^{-x}}$. Imaginea din partea dreaptă a fost obţinută calculând valorile produse de reţea pentru o serie de puncte generate în mod aleatoriu în dreptunghiul $(-6,4)\times(-6,4)$ din planul euclidian, după ce ponderile au fost instanţiate cu următoarele valori: $w_{10} = -0.8, w_{11} = 0.8, w_{12} = 0.1, w_{20} = 0.3, w_{21} = 0.3, w_{22} = -0.4, w_{30} = 0.2, w_{31} = 1$ şi $w_{32} = -1$. Clasificarea instanţelor a fost desemnată folosind semnele $+$ şi respectiv $\circ$.

a. Exprimaţi ieşirile o_1 şi o_2 în funcţie de intrările x_1 şi x_2 şi ponderile $w_{10}, w_{11}, w_{12}, w_{20}, w_{21}$ şi w_{22}.

b. Scrieţi *regula de decizie* corespunzătoare acestei reţele neuronale.

c. Presupunem că eliminăm funcţia de activare sigmoidală din perceptronii de pe nivelul ascuns (aşadar, N1 şi N2 devin unităţi liniare), însă păstrăm aceleaşi valori pentru ponderi. Scrieţi regula de decizie pentru noua reţea.

d. Presupunem că lăsăm valorile ponderilor din reţeaua de la punctul precedent să varieze liber. Poate această reţea să înveţe conceptul reprezentat de reţeaua iniţială?

Răspuns:

a. $$o_1(x_1,x_2) = \sigma(w_{10}+w_{11}x_1+w_{12}x_2) = \frac{1}{1+\exp(-(w_{10}+w_{11}x_1+w_{12}x_2))}$$
$$o_2(x_1,x_2) = \sigma(w_{20}+w_{21}x_1+w_{22}x_2) = \frac{1}{1+\exp(-(w_{20}+w_{21}x_1+w_{22}x_2))},$$

unde simbolul exp **desemnează funcţia exponenţială cu baza e.**

b. $$o(x_1,x_2) = \sigma(w_{30}+w_{31}o_1(x_1,x_2)+w_{32}o_2(x_1,x_2))$$
$$= \frac{1}{1+\exp(-(w_{30}+w_{31}o_1(x_1,x_2)+w_{32}o_2(x_1,x_2)))}.$$

Regula de decizie corespunzătoare reţelei neuronale din enunţ este:

> **if** $o(x_1,x_2) \geq 1/2$ **then assign the instance** (x_1,x_2) **the label** $+$;
> **else assign it the label** $-$;

sau, echivalent:

if $w_{30} + w_{31}o_1(x_1, x_2) + w_{32}o_2(x_1, x_2) \geq 0$
then assign the instance (x_1, x_2) **the label** $+$;
else, assign it the label $-$.

Este de remarcat faptul că graniţa de decizie a reţelei neuronale, şi anume curba de ecuaţie $w_{30} + w_{31}o_1(x_1, x_2) + w_{32}o_2(x_1, x_2) = 0$**, este una *neliniară*, după cum se poate observa şi din figura din enunţ (partea dreaptă).**

c.
$$\begin{aligned} o'_1(x_1, x_2) &= w_{10} + w_{11}x_1 + w_{12}x_2 \\ o'_2(x_1, x_2) &= w_{20} + w_{21}x_1 + w_{22}x_2 \\ o'(x_1, x_2) &= \sigma(w_{30} + w_{31}o'_1(x_1, x_2) + w_{32}o'_2(x_1, x_2)) \\ &= \sigma(w_{30} + w_{31}(w_{10} + w_{11}x_1 + w_{12}x_2) + w_{32}(w_{20} + w_{21}x_1 + w_{22}x_2)) \\ &= \sigma(w_{30} + w_{31}w_{10} + w_{32}w_{20} + (w_{31}w_{11} + w_{32}w_{21})x_1 + \\ &\quad (w_{31}w_{12} + w_{32}w_{22})x_2). \end{aligned}$$

Regula de decizie este:

if $w_{30} + w_{31}w_{10} + w_{32}w_{20} + (w_{31}w_{11} + w_{32}w_{21})x_1 + (w_{31}w_{12} + w_{32}w_{22})x_2 \geq 0$
then assign the instance (x_1, x_2) **the label** $+$;
else, assign it the label $-$.

d. Se observă că la punctul c **graniţa (suprafaţa) de decizie este o dreaptă. Aşadar, noua reţea *nu* poate învăţa în mod convenabil conceptul reprezentat de reţeaua iniţială.**

***Observaţie*: Este util de remarcat faptul că o reţea exact de acelşi tip, având însă alte valori pentru ponderile** w_{ij}**, poate reprezenta funcţia** XOR**.**[773]

9. **(Unitatea sigmoidală: exemplificare)**

■ • ○ ***CMU, 2003 fall, T. Mitchell, A. Moore, midterm, pr. 6.a***

Considerăm o unitate neuronală sigmoidală care are intrările x_1, x_2 **şi** x_3**. Aşadar, ieşirea ei este de forma** $y = \sigma(w_0 + w_1x_1 + w_2x_2 + w_3x_3)$ **unde** $\sigma(z) = \dfrac{1}{1 + e^{-z}}$**. Valorile fiecăreia dintre aceste trei intrări sunt** 0 **sau** 1**.**

Asignaţi valori ponderilor w_0, w_1, w_2 **şi** w_3 **astfel încât outputul unităţii neuronale să fie strict mai mare decât 0.5 dacă şi numai dacă valoarea logică a expresiei** $(x_1 \wedge x_2) \vee x_3$ **este *adevărat*.**

Răspuns:

Vom scrie mai întâi tabela de valori a funcţiei / expresiei booleene date. Vom adăuga în această tabelă o coloană care va conţine forma particulară a sumei ponderate $w_0 + w_1x_1 + w_2x_2 + w_3x_3$ **— adică, *argumentul* pe care-l va lua funcţia sigmoidală** (σ) **— pentru fiecare combinaţie de valori ale variabilelor** x_1, x_2, x_3**. În plus, la fiecare linie vom preciza şi *condiţia* care trebuie să fie satisfăcută**

[773] Vedeţi CMU, 2008 fall, Eric Xing, midterm, pr. 4.2. Ar fi [un exerciţiu] instructiv să vizualizaţi cum arată zonele de decizie în acel caz.

de către suma ponderată de pe linia respectivă, în aşa fel încât perceptronul să realizeze codificarea cerută în enunţ.[774]

x_1	x_2	x_3	$(x_1 \wedge x_2) \vee x_3$	$w_0 + w_1x_1 + w_2x_2 + w_3x_3$	$\vdots\ 0$
0	0	0	0	w_0	< 0
0	0	1	1	$w_0 + w_3$	> 0
0	1	0	0	$w_0 + w_2$	< 0
0	1	1	1	$w_0 + w_2 + w_3$	> 0
1	0	0	0	$w_0 + w_1$	< 0
1	0	1	1	$w_0 + w_1 + w_3$	> 0
1	1	0	1	$w_0 + w_1 + w_2$	> 0
1	1	1	1	$w_0 + w_1 + w_2 + w_3$	> 0

(359)

Aşadar, rezumând, pentru ca unitatea sigmoidală să codifice expresia booleană dată, trebuie ca ponderile w_i să satisfacă sistemul de inecuaţii scrise în ultima coloană a tabelului de mai sus.

În vederea găsirii unei soluţii pentru acest sistem, putem observa că în expresia $(x_1 \wedge x_2) \vee x_3$ variabilele logice x_1 şi x_2 joacă rol simetric în raport cu operatorul $\wedge$. Deci este natural să considerăm că ponderile alocate lui x_1 şi x_2 [ca inputuri] în perceptronul sigmoidal pot fi egale. Introducând deci restricţia $w_1 = w_2$, sistemul (359) va deveni:

$$\begin{cases} w_0 < 0 \\ w_0 + w_3 > 0 \\ w_0 + w_1 < 0 \\ w_0 + w_1 + w_3 > 0 \\ w_0 + 2w_1 > 0 \\ w_0 + 2w_1 + w_3 > 0 \end{cases} \tag{360}$$

Mai departe, analizând din nou expresia $(x_1 \wedge x_2) \vee x_3$, este natural să gândim că, în raport cu operatorul $\vee$, ponderea variabilei x_3 ar trebui să fie aceeaşi cu ponderile „cumulate" ale variabilelor x_1 şi x_2. Prin urmare, „injectând" în sistemul de inecuaţii (360) restricţia $w_3 = w_1 + w_2 = 2w_1$, el va deveni:

$$\begin{cases} w_0 < 0 \\ w_0 + 2w_1 > 0 \\ w_0 + w_1 < 0 \\ w_0 + 3w_1 > 0 \\ w_0 + 4w_1 > 0 \end{cases} \tag{361}$$

Din forma acestui sistem, apare natural să explorăm ce anume se întâmplă dacă restricţionăm spaţiul de soluţii impunând condiţia $w_1 > 0$. În această ipoteză, sistemul (361) va avea aceeaşi soluţie ca şi inecuaţia dublă

$$w_0 + w_1 < 0 < w_0 + 2w_1,$$

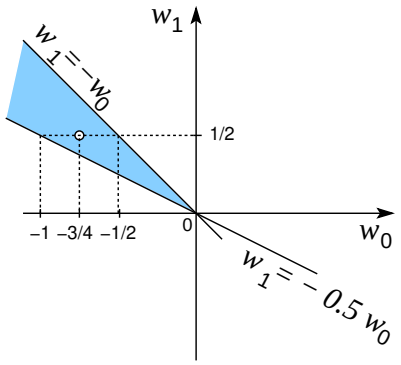
care este echivalentă cu

$$w_1 < -w_0 < 2w_1. \tag{362}$$

[774] Ştim că la clasificare cu unitatea sigmoidală se foloseşte echivalenţa $\sigma(z) > 1/2 \Leftrightarrow z > 0$ şi, complementar, $\sigma(z) < 1/2 \Leftrightarrow z < 0$.

Pentru această ultimă inecuaţie dublă este foarte uşor să indicăm soluţii. Iată una dintre ele: $w_1 = \frac{1}{2}$ şi $w_0 = -\frac{3}{4}$. Prin urmare, $w_2 = \frac{1}{2}$ şi $w_3 = 1$.

***Observaţie* (1): De fapt, dacă lucrăm într-un reper de coordonate w_0Ow_1, orice punct (w_0, w_1) din cadranul al doilea, şi care este situat între dreptele de ecuaţii $w_1 = -w_0$ şi respectiv $w_1 = -\frac{1}{2}w_0$ este o soluţie a inecuaţiei duble (362), deci şi a sistemului de inecuaţii (359), la care, aşa cum am văzut, au fost adăugate restricţiile $w_1 = w_2 = \frac{1}{2}w_3$. (Invers, adică a preciza care este întreg spaţiul de soluţii al sistemului (359) este mult în afara obiectivului acestui exerciţiu.)**



***Observaţie* (2): Rostul acestui [tip de] exerciţiu este să arate că pentru a găsi valori corespunzătoare ponderilor unui perceptron în aşa fel încât el să reprezinte / codifice o anumită funcţie, putem apela la [rezolvarea de] sisteme de restricţii / inecuaţii. Mai general, aşa cum precizează şi Tom Mitchell în cartea *Machine Learning* la pag. 95, putem folosi metode de programare liniară sau neliniară. Pe de altă parte, acest exerciţiu pune în evidenţă în mod indirect faptul că ar fi de dorit ca (alternativ) să dispunem de o procedură de calcul generală, cât mai simplă, prin care să atingem obiectivul menţionat anterior. Modul în care, pentru cazul perceptronului liniar, se fundamentează din punct de vedere teoretic o astfel de procedură — bazată pe metoda gradientului descendent — va face obiectul problemei propuse 36, iar aplicarea ei va fi exemplificată la problemele 10 şi 11. În raport cu programarea lininiară sau cea neliniară, această procedură are avantajul că se scalează în mod elegant / convenabil la nivel de reţea neuronală.**

6.1.2 Unităţi neuronale — algoritmi de antrenare

10. **(Unitatea liniară;
deducerea regulii de actualizare a ponderilor (în manieră "batch"),
în cazul particular al învăţării unei anumite funcţii booleane)**

Liviu Ciortuz, 2011

Vom considera din nou funcţia booleană $A \wedge \neg B$ (vedeţi problema 2.a), dar nu vom mai folosi perceptronul-prag ci unitatea liniară (perceptronul liniar). Ne propunem aici să *deducem* forma specifică a *regulilor* de actualizare a ponderilor acestui perceptron, cu *obiectivul* de a „învăţa" ulterior valorile concrete ale acestor poderi, astfel încât perceptronul rezultat să reprezinte funcţia booleană $A \wedge \neg B$ (aşa cum se va vedea la problema 11).

Considerând că antrenarea se va face în regim de lucru "batch" (folosind, desigur, ca instanţe de antrenament, liniile din tabela de adevăr a acestei

funcţii booleene, cu codificarea 1 pentru *true*, şi -1 pentru *false*), vă cerem să particularizaţi forma pe care o ia regula generală de actualizare a ponderilor ($\bar{w} \leftarrow \bar{w} + \Delta\bar{w}$) la învăţarea acestei funcţiei booleene, procedând în două moduri diferite:

a. calculând efectiv vectorul $\Delta\bar{w}$ (sau, pe componente, Δw_j) folosind formula dată la curs:

$$\Delta\bar{w} = \eta \sum_{i=1}^{4} (t_i - o_i)\bar{x}_i$$

unde $\bar{x}_i$ sunt instanţele din tabela de valori ale funcţiei $A \wedge \neg B$, $o_i = \bar{w} \cdot \bar{x}_i$ desemnează valoarea calculată de perceptron pentru $i = 1, \ldots, 4$, iar t_i este valoarea target a funcţiei booleene date, pentru intrarea $\bar{x}_i$, cu $i = 1, \ldots, 4$;[775]

b. calculând mai întâi expresia funcţiei $E(\bar{w})$ care desemnează *semisuma pătratelor erorilor* corespunzătoare instanţelor $\bar{x}_i$ ($i = 1, \ldots, 4$):

$$E(\bar{w}) = \frac{1}{2} \sum_{i=1}^{4} (t_i - o_i)^2 = \frac{1}{2} \sum_{i=1}^{4} (t_i - \bar{w} \cdot \bar{x}_i)^2 = \ldots,$$

precum şi cele trei derivate parţiale din expresia vectorului gradient

$$\nabla E(\bar{w}) = \left(\frac{\partial E(\bar{w})}{\partial w_0}, \frac{\partial E(\bar{w})}{\partial w_1}, \frac{\partial E(\bar{w})}{\partial w_2} \right),$$

folosind expresia funcţiei $E(\bar{w})$ care a fost calculată anterior, pentru ca la final, în spiritul *metodei gradientului descendent*, să obţinem $\Delta\bar{w} = -\eta \nabla E(\bar{w})$.

***Atenţie!* Dacă la final nu aţi obţinut acelaşi rezultat la punctele a şi b, înseamnă că aţi greşit la calcule.**

Răspuns:

a. Ştim că $o_i = \bar{w} \cdot \bar{x}_i$ şi $\Delta\bar{w} = \eta \sum_{i=1}^{4} (t_i - o_i)\bar{x}_i$. În cazul nostru, $o_i = (w_0, w_1, w_2) \cdot (1, x_{i,1}, x_{i,2})$, deci $o_1 = w_0 - w_1 - w_2$, $o_2 = w_0 - w_1 + w_2$, $o_3 = w_0 + w_1 - w_2$, $o_4 = w_0 + w_1 + w_2$ şi, ţinând cont că $t_1 = t_2 = t_4 = -1$ şi $t_3 = 1$, rezultă (în scriere matriceală):

$$\Delta\bar{w} = \eta \left[(-1 - o_1) \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix} + (-1 - o_2) \begin{pmatrix} 1 \\ -1 \\ 1 \end{pmatrix} + (1 - o_3) \begin{pmatrix} 1 \\ 1 \\ -1 \end{pmatrix} + (-1 - o_4) \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \right].$$

Scriind această egalitate pe componente, vom avea mai întâi:

$$\begin{aligned} \Delta w_0 &= \eta(-1 - o_1 - 1 - o_2 + 1 - o_3 - 1 - o_4) = \eta(-2 - (o_1 + o_2 + o_3 + o_4)) \\ &= \eta(-2 - (w_0 - w_1 - w_2 + w_0 - w_1 + w_2 + w_0 + w_1 - w_2 + w_0 + w_1 + w_2)) \\ &= \eta(-2 - 4w_0) = -2\eta(2w_0 + 1) \end{aligned}$$

Aşadar, regula de actualizare a ponderii w_0 va fi $w_0 \leftarrow w_0 - 2\eta(2w_0 + 1)$.

În mod similar, facând calculele, vom obţine $w_1 \leftarrow w_1 - 2\eta(2w_1 - 1)$ şi $w_2 \leftarrow w_2 - 2\eta(2w_2 + 1)$.

b. Conform definiţiei, $E(w_0, w_1, w_2) = \frac{1}{2} \sum_i (t_i - o_i)^2$, cu $o_i = w_0 + w_1 x_{i,1} + w_2 x_{i,2}$. Prin urmare,

[775] Atenţie! Aceste instanţe trebuie extinse cu componenta 1 corespunzătoare termenului liber din expresia separatorului liniar.

$$\begin{aligned}E(w_0,w_1,w_2) &= \frac{1}{2}[\ (-1-(w_0-w_1-w_2))^2+(-1-(w_0-w_1+w_2))^2+\\ &\quad (1-(w_0+w_1-w_2))^2+(-1-(w_0+w_1+w_2))^2]\\ &= \frac{1}{2}[\ (1+w_0-w_1-w_2)^2+(1+w_0-w_1+w_2)^2+\\ &\quad (1-w_0-w_1+w_2)^2+(1+w_0+w_1+w_2)^2]\\ &= \frac{1}{2}[\ 2(1+w_0-w_1)^2+2w_2^2+2(1+w_2)^2+2(w_0+w_1)^2]\\ &= (1+w_0-w_1)^2+w_2^2+(1+w_2)^2+(w_0+w_1)^2\\ &= 2(1+w_0^2+w_1^2+w_2^2+w_0-w_1+w_2)\end{aligned}$$

Derivatele parţiale ale lui E în raport cu variabilele w_0, w_1 şi w_2 sunt:

$$\frac{\partial E}{\partial w_0}=2(2w_0+1),\ \frac{\partial E}{\partial w_1}=2(2w_1-1),\ \frac{\partial E}{\partial w_2}=2(2w_2+1)$$

În concluzie, regulile de actualizare pentru ponderile unităţii liniare vor fi:

$$w_0\leftarrow w_0-2\eta(2w_0+1),\ w_1\leftarrow w_1-2\eta(2w_1-1),\ w_2\leftarrow w_2-2\eta(2w_2+1)$$

Se observă că rezultatele obţinute la punctele a şi b coincid, după cum era de aşteptat.

11. **(Perceptronul-prag;**
aplicarea algoritmului de actualizare a ponderilor
în manieră incrementală / stochastică,
pentru a învăţa o anumită funcţie booleană)

Liviu Ciortuz, 2011

La problema 2.a am arătat că funcţia booleană $A\wedge(\neg B)$ poate fi reprezentată cu ajutorul unui perceptron-prag — adică, un perceptron cu funcţie de activare de tip prag — ale cărui ponderi pe intrări sunt date de ecuaţia dreptei $y = x-1 \Leftrightarrow x-y-1=0 \Leftrightarrow w_0=-1, w_1=1, w_2=-1$. (A se vedea linia punctată din graficul de mai jos.) Evident, există o infinitate de drepte care separă în mod corect cele patru instanţe de antrenament.

În exerciţiul de faţă, spre deosebire de cum am procedat la problema 2.a, nu vom mai „ghici", ci vom *învăţa* un separator liniar care să reprezinte / codifice aceeaşi funcţie booleană de mai sus. Învăţarea se va face aplicând algoritmul de antrenare a perceptronului (adică de actualizare a ponderilor) din cartea *Machine Learning* a profesorului T. Mitchell, pag. 93.[776]

***Convenţie*: Vom considera că valorile logice Adevărat / Fals sunt codificate de numerele $+1/-1$. De asemenea, variabila numerică x (rescrisă uneori, pentru**

[776] *Atenţie*: La problema 10 s-a lucrat cu acelaşi concept — funcţia booleană $A\wedge(\neg B)$ — însă acolo am folosit perceptronul liniar în varianta "batch". Aici folosim perceptronul-prag în varianta incrementală / stochastică. Din punct de vedere analitic, forma regulii de actualizare este $\bar{w}\leftarrow\bar{w}+\Delta\bar{w}$, cu $\Delta w_i=-\eta\sum_n(t_n-o_n)\frac{\partial E}{\partial w_i}\cdot x_{n,i}$ pentru prima variantă şi, respectiv, $\Delta w_i=-\eta(t_n-o_n)\frac{\partial E}{\partial w_i}\cdot x_{n,i}$ pentru a doua variantă. Reţineţi că outputurile (o_n) produse de cei doi perceptroni pentru un acelaşi input x_n diferă în general, întrucât perceptronul-prag aplică funcţia de activare *sign* la combinaţia liniară $\bar{w}\cdot\bar{x}$, în vreme ce perceptronul liniar nu o aplică.

convenienţă, ca x_1) corespunde variabilei A, iar y (rescrisă ca x_2) corespunde variabilei B.

Vom considera că valorile iniţiale ale ponderilor perceptronului sunt cele corespunzătoare dreptei $y = \frac{1}{3}x - \frac{1}{3} \Leftrightarrow \frac{1}{3}x - y - \frac{1}{3} = 0 \Leftrightarrow w_0 = -\frac{1}{3}, w_1 = \frac{1}{3}, w_2 = -1$. Se poate constata imediat că funcţia $\frac{1}{3}x - y - \frac{1}{3}$ clasifică în mod eronat punctul $(-1, -1)$. Prin urmare, separatorul liniar reprezentat de ecuaţia $\frac{1}{3}x - y - \frac{1}{3} = 0$ — vedeţi dreapta continuă din graficul de mai jos — nu este consistent cu conceptul logic $A \wedge \neg B$.

a. Folosind rata de învăţare $\eta = 0.1$, să se aplice atâtea iteraţii câte sunt necesare pentru actualizarea ponderilor perceptronului-prag — în manieră incrementală / stochastică — astfel încât, la final, toate instanţele din tabloul funcţiei booleene $A \wedge \neg B$ (vedeţi tabelul alăturat) să fie clasificate corect. Scrieţi ecuaţia separatorului liniar rezultat.

A	B	$A \wedge \neg B$
-1	-1	-1
-1	$+1$	-1
$+1$	-1	$+1$
$+1$	$+1$	-1

Vă reamintim că regula de actualizare a ponderilor pentru perceptronul-prag, în varianta stochastică / incrementală este de forma:

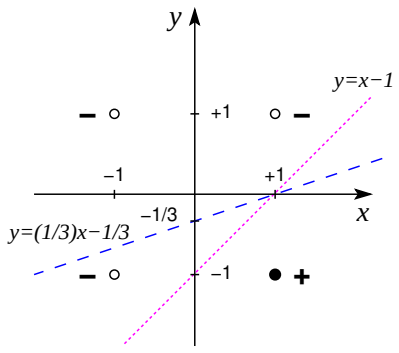
$$\bar{w} \leftarrow \bar{w} + \Delta\bar{w}, \text{ cu } \Delta\bar{w} = \eta(t - o)\bar{x}$$

unde $\bar{x} = (1, x_1, x_2)$ desemnează o instanţă de antrenament oarecare, o este valoarea produsă de către perceptron pentru intrarea $\bar{x}$, iar t este valoarea ţintă (engl., target) pentru aceeaşi intrare $\bar{x}$.

b. Ce s-ar fi întâmplat dacă am fi folosit o rată de învăţare mult mai mică, de exemplu $\eta = 0.01$?

Răspuns:

Reprezentarea grafică a datelor de antrenament şi a poziţiei iniţiale a separatorului liniar ($\frac{1}{3}x - y - \frac{1}{3} = 0$) asociat perceptronului din enunţ este cea din figura alăturată. Acest separator liniar greşeşte la clasificarea instanţei logice $A = F, B = F$ (reprezentată numeric de către $x = -1$ şi $y = -1$), fiindcă $\frac{1}{3}(-1) - (-1) - \frac{1}{3} = 1 - \frac{2}{3} = \frac{1}{3} > 0$. În schimb, toate celelalte instanţe de antrenament sunt clasificate corect (verificarea este imediată).



În vederea aplicării regulii de actualizare a ponderilor, vom calcula mai întâi $\Delta\bar{w} = \eta(t_1 - o_1) \cdot \bar{x}_1 = \eta(-1 - 1) \cdot \bar{x}_1$, unde $\bar{x}_1 = (1, -1, -1)$ este punctul incorect clasificat. Aşadar,

$$\Delta\bar{w} = -2\eta(1, -1, -1) = -0.2 \cdot (1, -1, -1) = (-0.2,\ 0.2,\ 0.2)$$

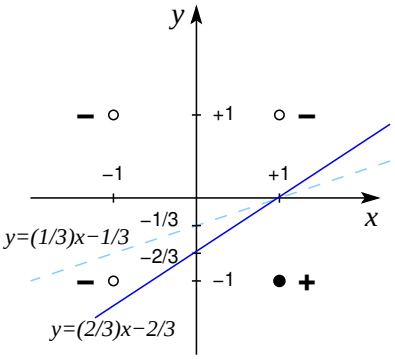
sau, rescris pe componente, $\Delta w_0 = -0.2,\ \Delta w_1 = 0.2,\ \Delta w_2 = 0.2$.[777] **Aplicând regula de actualizare a ponderilor, după efectuarea calculelor vom obţine:**

$$w_0 = -\frac{1}{3} - 0.2 = -\frac{8}{15},\ w_1 = \frac{1}{3} + 0.2 = \frac{8}{15},\ w_2 = -1 + 0.2 = -\frac{4}{5}$$

În consecinţă, perceptronul învaţă o nouă poziţie pentru separatorul liniar:

$$\frac{8}{15}x - \frac{4}{5}y - \frac{8}{15} = 0 \Leftrightarrow \frac{2}{3}x - y - \frac{2}{3} = 0 \Leftrightarrow y = \frac{2}{3}x - \frac{2}{3}$$

Valoarea funcţiei $\frac{2}{3}x - y - \frac{2}{3}$ **pentru** $x = -1$ **şi** $y = -1$ **este** $-\frac{2}{3} + 1 - \frac{2}{3} = -\frac{1}{3} < 0$. **Aşadar, această instanţă de antrenament este acum corect clasificată. Mai mult, se verifică imediat că toate instanţele de antrenament sunt clasificate corect de către noul separator liniar (vedeţi graficul alăturat), deci algoritmul de actualizare a ponderilor perceptronului-prag se opreşte după ce s-a efectuat o singură iteraţie.**



***Observaţie*: Aplicarea variantei "batch" pentru algoritmul de învăţare a ponderilor perceptronului-prag (bazată pe regulile deduse prin folosirea metodei gradientului descendent pentru perceptronul liniar) nu ar fi diferit de modul de lucru de mai sus (iar soluţia obţinută ar fi fost aceeaşi), fiindcă** $t_n = o_n$ **pentru orice** $n \neq 1$. **Doar scrierea / redactarea calculelor ar fi diferit uşor.**

b. Dacă s-ar fi lucrat folosind o valoare mai mică pentru rata de învăţare η, **este foarte posibil să fi fost necesar să efectuăm mai multe iteraţii. Cele patru instanţe de antrenament fiind separabile liniar, convergenţa algoritmului (la valoarea minimă, adică** 0, **pentru eroarea totală) este asigurată pentru valori mici ale lui** η. **(Vedeţi T. Mitchell, *Machine Learning*, pag. 89.)**

[777] Dacă în locul perceptronului-prag am fi lucrat cu perceptronul liniar în varianta stochastică, aici am fi obţinut

$$\begin{aligned}\Delta\bar{w} &= \eta(-1 - o(-1,-1))\bar{x}_1 = \eta(-1 - (-\frac{1}{3} + 1 - \frac{1}{3}))\bar{x}_1 = \eta(-1 - \frac{1}{3})\bar{x}_1 = -\eta\frac{4}{3}\bar{x}_1 = -\frac{1}{10}\frac{4}{3}\bar{x}_1 \\ &= -\frac{2}{15}(1,-1,-1) = \left(-\frac{2}{15}, \frac{2}{15}, \frac{2}{15}\right).\end{aligned}$$

După actualizare, aceasta ar fi dus la valorile

$$w_0 = -\frac{1}{3} - \frac{2}{15} = -\frac{7}{15},\ w_1 = \frac{1}{3} + \frac{2}{15} = \frac{7}{15},\ w_2 = -1 + \frac{2}{15} = -\frac{13}{15}.$$

Dreapta determinată de aceste ponderi are ecuaţia $\frac{17}{15}x_1 - \frac{13}{15}x_2 - \frac{7}{15} = 0 \Leftrightarrow x_2 = \frac{7}{13}x_1 - \frac{7}{13} = 0$. Pentru instanţa $\bar{x}_1$, expresia $\frac{17}{15}x_1 - \frac{13}{15}x_2 - \frac{7}{15}$ are valoarea $-\frac{17}{15} + \frac{13}{15} - \frac{17}{15} = -\frac{1}{15} < 0$.

12. **(Deducerea regulii de antrenare pentru un tip particular de unitate liniară)**

• ***CMU, 1995 fall, Tom Mitchell, HW4, pr. 5***

Considerăm un nou tip de unitate neuronală liniară care are două intrări x_1 şi x_2 şi o ieşire definită astfel:

$$output = w_0 + w_1 w_0 x_1 + w_2 w_0 x_2.$$

Derivaţi regula de actualizare a ponderilor cu metoda gradientului descendent, folosind următoarea definiţie pentru *funcţia de cost / pierdere* (engl., loss function):

$$E = \frac{1}{2} \sum_{d \in D} [target_d - output_d]^2,$$

unde D este setul de instanţe de antrenament.

Răspuns:

Conform metodei gradientului descendent, modificarea ponderilor se va face după formula $w_i \leftarrow w_i - \eta \frac{\partial E}{\partial w_i}$. Calculăm deci derivata funcţiei de eroare:

$$\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i} \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 = \frac{1}{2} \sum_{d \in D} \frac{\partial}{\partial w_i} (t_d - o_d)^2 \\
&= \frac{1}{2} \sum_{d \in D} 2(t_d - o_d) \frac{\partial}{\partial w_i} (t_d - o_d) = \sum_{d \in D} -(t_d - o_d) \frac{\partial o_d}{\partial w_i} \\
&= -\sum_{d \in D} (t_d - o_d) \frac{\partial}{\partial w_i} (w_0 + w_1 w_0 x_{1,d} + w_2 w_0 x_{2,d}) \\
&= \begin{cases} -\sum\limits_{d \in D} (t_d - o_d)(1 + w_1 x_{1,d} + w_2 x_{2,d}) & \text{dacă } i = 0 \\ -\sum\limits_{d \in D} (t_d - o_d) w_0 x_{1,d} & \text{dacă } i = 1 \\ -\sum\limits_{d \in D} (t_d - o_d) w_0 x_{2,d} & \text{dacă } i = 2. \end{cases}
\end{aligned}$$

Aşadar, actualizarea ponderererilor o vom face folosind formulele:

$$\begin{aligned}
w_0 &\leftarrow w_0 + \eta \sum_{d \in D} (t_d - o_d)(1 + w_1 x_{1,d} + w_2 x_{2,d}) \\
w_1 &\leftarrow w_1 + \eta \sum_{d \in D} (t_d - o_d) w_0 x_{1,d} \\
w_2 &\leftarrow w_2 + \eta \sum_{d \in D} (t_d - o_d) w_0 x_{2,d}
\end{aligned}$$

unde η este o constantă pozitivă (rata de învăţare).

13. **(Aplicarea metodei gradientului descendent pentru modelarea unui dataset particular, cu „zgomot“ de tip gaussian)**

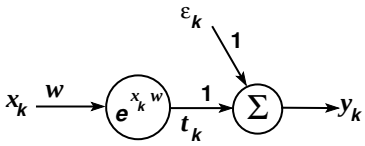
prelucrare de Liviu Ciortuz, după CMU, 2001 fall, Andrew Moore, final exam, pr. 10.2

Presupunem că avem la dispoziţie un set de date, în care fiecare instanţă are un atribut de intrare $x \in \mathbb{R}$ şi un atribut de ieşire y. Bănuim că relaţia dintre valorile celor două atribute este de forma

$$y_k = e^{wx_k} + \varepsilon_k,$$

unde w este o constantă reală, iar $\varepsilon_k \in \mathbb{R}$ este un „zgomot“, care este independent de x_k.

Putem să ne imaginăm y_k ca fiind produs de o reţea neuronală [având pe nivelul ascuns un perceptron de un tip particular], care are o singură pondere nespecificată w, ca în figura alăturată. Se va considera că $t_k \overset{not.}{=} e^{wx_k}$, este targetul de învăţat.



a. Considerând că instanţele $x_1, \ldots, x_n$ (fixate) au fost generate în mod independent şi că „zgomotul“ ε urmează o distribuţie de tip gaussian cu media 0 şi varianţa σ^2 (pentru orice $k \in \{1, \ldots, n\}$),[778] demonstraţi că valoarea lui w pentru care se maximizează verosimilitatea condiţională a datelor de antrenament, adică

$$\operatorname*{argmax}_w P(y_1, \ldots, y_n \mid x_1, \ldots, x_n, w) \overset{indep.}{=} \operatorname*{argmax}_w \prod_{k=1}^{n} P(y_k \mid x_k, w),$$

este exact aceeaşi cu valoarea (lui w) pentru care se minimizează suma pătratelor erorilor:

$$\operatorname*{argmin}_w \sum_{k=1}^{n} (e^{wx_k} - y_k)^2.$$

Acest rezultat oferă o explicaţie suplimentară, probabilistă pentru utilizarea criteriului sumei pătratelor erorilor — a cărui justificare ar rămâne altminteri doar intuitivă — drept criteriu de optimizat la aplicarea metodei gradientului descendent pentru perceptroni şi reţele neuronale.

***Indicaţie*: Puteţi face demonstraţia nu doar pentru cazul de faţă, ci mai general, şi anume pentru toate situaţiile în care [ca target] vrem să învăţăm o funcţie oarecare f (în cazul nostru e^{wx}), datele de antrenament sunt de forma (x_k, y_k), iar diferenţa dintre y_k şi $f(x_k)$ (în cazul nostru ε_k) umează o distribuţie gaussiană de medie 0.**

b. Aplicaţi metoda gradientului descendent — adică, scrieţi regula de actualizare a ponderii w — pentru a afla valoarea ponderii w care minimizează suma pătratelor erorilor.[779]

[778] Pentru definiţia distribuţiei gaussiene unidimensionale, vedeţi problema 32 de la capitolul de *Fundamente*.

[779] În enunţul original, în locul *sumei* este specificată *media* (engl., mean) *pătratelor erorilor*. Este vorba de medie în sensul de medie aritmetică. Cele două formulări sunt echivalente din punctul de vedere al optimizării (fiindcă diferă doar printr-un factor pozitiv), deci ambele sunt corecte. Am preferant să folosim termenul *sumă*, întrucât la punctul *a* se lucrează probabilist, deci în contextul respectiv termenul *medie* (ambiguu în româneşte!) ar avea [şi] o altă semnificaţie.

Răspuns:

a. Considerând că deviaţia standard a distribuţiei urmate de ε_k este σ şi ţinând cont că $P(\varepsilon_k) \stackrel{def.}{=} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(\varepsilon_k-\mu)^2}{2\sigma^2}}$, cu $\mu = 0$, vom avea:

$$\begin{aligned}
&\operatorname*{argmax}_w \prod_{k=1}^n P(y_k \mid x_k, w) = \operatorname*{argmax}_w \prod_{k=1}^n P(y_k - f(x_k) \mid x_k, w) \\
= \;& \operatorname*{argmax}_w \prod_{k=1}^n P(\varepsilon_k \mid x_k, w) = \operatorname*{argmax}_w \prod_{k=1}^n P(\varepsilon_k \mid w) \\
= \;& \operatorname*{argmax}_w \prod_{k=1}^n P(y_k - t_k \mid w) = \operatorname*{argmax}_w \prod_{k=1}^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_k-t_k)^2}{2\sigma^2}} \\
= \;& \operatorname*{argmax}_w \ln \prod_{k=1}^n \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_k-t_k)^2}{2\sigma^2}} = \operatorname*{argmax}_w \sum_{k=1}^n \ln \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(y_k-t_k)^2}{2\sigma^2}} \\
= \;& \operatorname*{argmax}_w \sum_{k=1}^n \left(\ln \frac{1}{\sqrt{2\pi}\sigma} - \frac{(y_k-t_k)^2}{2\sigma^2} \right) = \operatorname*{argmax}_w \sum_{k=1}^n \left(-\frac{(y_k-t_k)^2}{2\sigma^2} \right) \\
= \;& \operatorname*{argmax}_w \left(-\sum_{k=1}^n \frac{(y_k-t_k)^2}{2\sigma^2} \right) = \operatorname*{argmin}_w \sum_{k=1}^n \left(\frac{(y_k-t_k)^2}{2\sigma^2} \right) \\
= \;& \operatorname*{argmin}_w \sum_{k=1}^n (y_k - t_k)^2.
\end{aligned}$$

b. Ca şi la exerciţiile precedente, în locul sumei pătratelor erorilor vom considera drept criteriu de optimizat pentru metoda gradientului descendent semisuma pătratelor erorilor, $E(w) = \frac{1}{2}\sum_k (y_k - t_k)^2$, unde $t_k = e^{wx_k}$, întrucât aceasta este uşor mai convenabilă la calcule şi nu modifică rezultatul problemei. (A se vedea echivalenţa $(cf)'(x) = 0 \Leftrightarrow cf'(x) = 0$, unde c este constantă iar f este funcţie derivabilă.)

Conform metodei gradientului descendent, regula de actualizare a ponderii w va fi (în varianta "batch") de forma

$$w \leftarrow w - \eta \frac{\partial E(w)}{\partial w}.$$

unde $\eta > 0$ este rata de învăţare.

Calculăm derivata lui E în raport cu w:

$$\begin{aligned}
\frac{\partial E(w)}{\partial w} &= \frac{\partial}{\partial w} \frac{1}{2} \sum_k (y_k - t_k)^2 = -\frac{1}{2} \sum_k 2(y_k - t_k) \frac{\partial t_k}{\partial w} \\
&= -\sum_k (y_k - t_k) \frac{\partial}{\partial w}(e^{wx_k}) = -\sum_k (y_k - t_k) e^{wx_k} x_k = \sum_k (e^{wx_k} k - y_k) e^{wx_k} x_k.
\end{aligned}$$

Aşadar, regula de actualizare a ponderii w va fi:

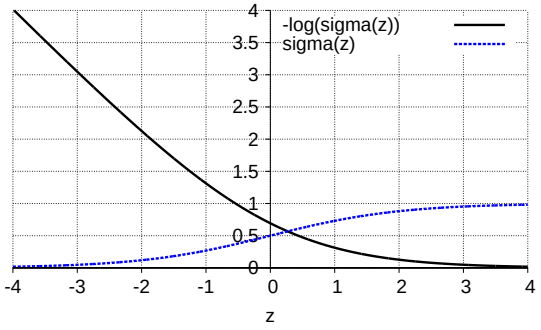
$$w \leftarrow w + \eta \sum_k (y_k - o_k) e^{wx_k} x_k.$$

14. **(O variantă a regulii de actualizare a ponderilor pentru perceptronul liniar, obţinută prin aplicarea metodei gradientului descendent pe o altă funcţie de pierdere / cost (engl., loss function) în locul sumei pătratelor erorilor: funcţia log-sigmoidală)**

prelucrare de Liviu Ciortuz, după
■ • ***University of Utah, 2008 fall, Hal Daumé III, HW4, pr. 2-4***

În acest exerciţiu, vom construi o nouă variantă de unitate liniară (perceptron fără prag) care va optimiza în locul semisumei pătratelor erorilor, o *funcţie de cost* de tip log-sigmoidal:[780]

$$l(y, h(x)) = -\ln \sigma(\underbrace{yh(x)}_{z}),$$



unde $\sigma(z) = 1/(1 + \exp(-z))$.

***Convenţie*: Vom considera că o ipoteză oarecare** h **produce *valori reale* şi că valorile pozitive desemnează clasa** $+1$**, iar cele negative clasa** -1**.**

a. Verificaţi că funcţia log-sigmoidală de mai sus este într-adevăr o *funcţie de cost*, adică: atunci când clasa desemnată de ipoteza $h(x)$ **este corectă (**$y = h(x)$**), valoarea** $l(y, h(x))$ **este mică şi, invers, atunci când clasa desemnată de** $h(x)$ **este incorectă (**$y \neq h(x)$**), valoarea** $l(y, h(x))$ **este mare.**

b. Folosind o unitate liniară, dorim să învăţăm o funcţie liniară minimizând în locul semisumei pătratelor erorilor — cum am procedat la problemele 10, 11, 12 şi 13 —, o funcţie de cost de tip log-sigmoidal, şi anume $f(b, \bar{w}) = \sum_n -\ln \sigma(y_n(\bar{w} \cdot \bar{x}_n + b))$**, unde** n **este indicele folosit pentru indexarea instanţelor de antrenament** $(\bar{x}_1, y_1), (\bar{x}_2, y_2), \ldots$**.**[781] **Calculaţi gradientul acestei funcţii în raport cu** $\bar{w}$ **şi în raport cu** b**.**

c. Arătaţi că funcţia log-sigmoidală de la punctul precedent este convexă atât în $\bar{w}$ **cât şi în** b**. Aşadar, se justifică următoarea cerinţă: formulaţi atât pentru** w_i **cât şi pentru** b ***regulile de actualizare*** **(engl., update rules) care sunt necesare pentru antrenarea perceptronului liniar folosind metoda gradientului descendent în conjuncţie cu funcţia de cost log-sigmoidală.**

***Indicaţie*: Pentru uşurinţa demonstraţiei, puteţi folosi următoarele *propoziţii*:**[782]

i. **Dacă** f **şi** g **sunt funcţii convexe, iar** g **este o funcţie crescătoare (nu neapărat strict crescătoare) de o variabilă reală, atunci funcţia** h **definită prin relaţia**

[780] La capitolul *Metode de regresie* (vedeţi problema 13) am numit această funcţie *funcţia de pierdere logistică*.
[781] Pentru clarificare, facem următoarea *precizare*: La problemele 8 şi 9, funcţia σ se aplica la calcularea outputului perceptronului sigmoidal: $o_n = \sigma(\bar{w} \cdot \bar{x}_n)$. Mai departe, o_n contribuia la calcularea funcţiei de eroare $E(\bar{w}) = 1/2 \sum_n (t_n - o_n)^2$ folosită la aplicarea metodei gradientului descendent. În schimb, la prezenta problemă funcţia σ se aplică ieşirii produse de perceptronul liniar, sub forma $-\ln \sigma(t_n\, o_n)$, ceea ce corespunde termenului $(t_n - o_n)^2$ din $E(\bar{w})$.
[782] Vedeţi https://en.wikipedia.org/wiki/Convex_function.

$h(x) = g(f(x))$ **este convexă.**[783]
ii. **Dacă** f **este funcţie concavă, iar** g **este funcţie convexă şi descrescătoare (nu neapărat strict descrescătoare) de o variabilă reală, atunci funcţia** h **definită prin relaţia** $h(x) = g(f(x))$ **este convexă.**[784]

Răspuns:

a. Funcţia l **se poate scrie sub forma:**

$$l(y, h(x)) = -\ln \sigma(y\, h(x)) = -\ln \frac{1}{1 + e^{-yh(x)}} = \ln(1 + e^{-yh(x)}).$$

Din proprietăţile logaritmului ştim că $\ln 1 = 0$, $\ln a < 0$ **dacă** $0 < a < 1$, **şi** $\ln a > 0$ **dacă** $a > 1$.

Dacă ipoteza h **este corectă pentru** x, **adică** $h(x)$ **are acelaşi semn cu** y, **atunci** $e^{-y\, h(x)}$ **este o valoare pozitivă mică (şi cu atât mai mică cu cât** $|h(x)|$ **este mai mare), deci şi valoarea funcţiei** $l(y, h(x))$ **este mică.**

Dacă ipoteza h **este incorectă pentru** x, **adică** $h(x)$ **are semn contrar semnului lui** y, **atunci** $e^{-y\, h(x)}$ **este o valoare mare (şi cu atât mai mare cu cât** $|h(x)|$ **este mai mare), deci şi valoarea funcţiei** $l(y, h(x))$ **este mare.**

De *exemplu*, dacă $y = 1$ **şi** $h(x) = -1$ **(sau** $y = -1$ **şi** $h(x) = 1$**) atunci pierderea / costul este** $-\ln \frac{1}{1+e} = \ln(1 + e) \approx 1.31$, **iar dacă** $y = h(x) = 1$ **(sau, ambele,** -1**) atunci costul este** $-\ln \frac{1}{1+e^{-1}} = \ln(1 + e^{-1}) \approx 0.31$.

b. Calculăm derivatele parţiale ale funcţiei $f(b, \bar{w}) = \sum_n -\ln \sigma(y_n(\bar{w} \cdot \bar{x}_n + b))$ **în raport cu** w_i **şi respectiv** b. **Vom folosi următoarele *proprietăţi*:**

- **derivata sumei de funcţii derivabile este suma derivatelor, adică** $(f + g)' = f' + g'$;
- **derivata unei compuneri de funcţii derivabile este** $(f \circ g)' = (f' \circ g)g'$;
- **derivata logaritmului este** $\ln'(x) = \frac{1}{x}$;
- **derivata funcţiei sigmoidale este** $\sigma'(x) = \sigma(x)(1 - \sigma(x))$.[785]

Vom obţine:

$$\begin{aligned}
\frac{\partial f}{\partial w_i} &= \frac{\partial}{\partial w_i} \sum_n -\ln \sigma(y_n(\bar{w} \cdot \bar{x}_n + b)) = -\sum_n \frac{\partial}{\partial w_i} \ln \sigma(y_n(\bar{w} \cdot \bar{x}_n + b)) \\
&= -\sum_n \frac{1}{\sigma(y_n(\bar{w} \cdot \bar{x}_n + b))} \frac{\partial}{\partial w_i} \sigma(y_n(\bar{w} \cdot \bar{x}_n + b)) \\
&= -\sum_n \frac{1}{\cancel{\sigma(y_n(\bar{w} \cdot \bar{x}_n + b))}} \cancel{\sigma(y_n(\bar{w} \cdot \bar{x}_n + b))} (1 - \sigma(y_n(\bar{w} \cdot \bar{x}_n + b)))\, y_n x_{n,i} \\
&= -\sum_n (1 - \sigma(y_n(\bar{w} \cdot \bar{x}_n + b)))\, y_n x_{n,i}.
\end{aligned}$$

[783] De *exemplu*, dacă f este funcţie convexă, tot aşa va fi şi $e^{f(x)}$, întrucât e^x este funcţie convexă şi monoton crescătoare.
[784] Ca *exemplu* puteţi considera $f(x) = \ln x$ şi $g(x) = e^{-x}$. Va rezulta funcţia $h(x) = g(f(x)) = 1/x$ pentru $x > 0$, care este, într-adevăr, convexă.
[785] Vedeţi ex. 33 de la capitolul *Metode de regresie*.

Similar, vom obţine

$$\frac{\partial f}{\partial b} = -\sum_n (1 - \sigma(y_n(\bar{w} \cdot \bar{x}_n + b)))\, y_n.$$

***Observaţie*: La curs, b a fost asimilat cu w_0, iar $x_{n,0} = 1$ prin definiţie. Dacă s-ar fi procedat la fel şi aici, atunci nu am fi avut de calculat decât o derivată la punctul a şi una la punctul b.**

c. Faptul că funcţia $f(b, \bar{w})$ definită la punctul b este convexă în raport cu variabilele $\bar{w}$ şi b se demonstează uşor, folosind cea de-a doua *propoziţie* din *Indicaţia* din enunţ. Într-adevăr, funcţia univariată $l(z) \stackrel{def.}{=} -\ln \sigma(z)$ este convexă şi strict descrescătoare (vedeţi graficul din enunţ),[786] iar funcţia $y(\bar{w} \cdot \bar{x} + b)$ este liniară în raport cu $\bar{w}$ şi cu b, deci este concavă (şi convexă în acelaşi timp). Prin urmare, funcţia $-\ln \sigma(y(\bar{w} \cdot \bar{x} + b))$ este convexă. În fine, se ţine cont de faptul că o sumă de funcţii convexe având acelaşi domeniu de definiţie este de asemenea funcţie convexă.

***Observaţie*: Pentru o demonstraţie alternativă, şi anume folosind matricea hessiană, vedeţi problema 14 de la capitolul *Metode de regresie*, unde se lucrează cu regresia logistică.**

În *concluzie*, valoarea optimă a funcţiei f este un minim, iar algoritmul de învăţare automată bazat pe metoda gradientului descendent va găsi (la limită, eventual) acest minim. Antrenarea acestui perceptron se face similar cu antreanarea unităţii liniare, folosind reguli de actualizare de forma:

$$w_i \leftarrow w_i + \Delta w_i \text{ şi } b \leftarrow w_i + \Delta b.$$

În cazul de faţă, Δw_i şi Δb sunt definite de expresiile

$$\Delta w_i = -\eta \frac{\partial f}{\partial w_i} = \eta \sum_n (1 - \sigma(y_n(\bar{w} \cdot \bar{x}_n + b)))\, y_n \bar{x}_{n,i}$$

şi respectiv

$$\Delta b = -\eta \frac{\partial f}{\partial b} = \eta \sum_n (1 - \sigma(y_n(\bar{w} \cdot \bar{x}_n + b)))\, y_n.$$

15. (Unităţi neuronale sigmoidale [şi reţele neuronale de unităţi sigmoidale]: deducerea expresiei funcţiei de eroare de tip cross-entropie; două variante: cu şi, respectiv, fără „zgomot“)

prelucrare de Liviu Ciortuz, după
■ • ○ ***CMU, 2011 fall, Eric Xing, HW1, pr. 3.3***

Ştim că pentru a antrena o unitate neuronală — şi vom vedea mai târziu că este la fel şi în cazul reţelelor neuronale — avem nevoie să definim o funcţie de eroare / pierdere / cost, care poate fi minimizată cu ajutorul algoritmului de „învăţare“ a ponderilor unităţii / reţelei. Până acum,[787] am folosit [ca funcţie de eroare] suma pătratelor erorilor. Se

[786] Faptul că funcţia $l(z)$ este descrescătoare şi convexă se poate demonstra în mod direct, cu mijloace clasice de analiză matematică.

[787] Adică: la toate problemele de până acum, cu excepţia problemei 14.

poate arăta că obiectivul de a minimiza această funcţie este implicat de principiul verosimilităţii maxime (engl., MLE), în contextul „procesării“ unui „semnal“ însoţit de un „zgomot“ care urmează o distribuţie de tip Gaussian.[788] În acest exerciţiu vom arăta că în cazul antrenării unităţilor sigmoidale se foloseşte un tip de funcţie de cost care poate fi interpretată ca o cross-entropie (vedeţi ex 63 de la capitolul de *Fundamente*).[789]

Fie datele de antrenament $D = (X, t) = \{(x_i, t_i)\}, i = 1, 2, \ldots, N$. De exemplu, x_i (element dintr-un spaţiu $\mathbb{R}^d$) poate fi imaginea unei feţe, în vreme ce t_i este o etichetă binară care are valoarea 0 dacă faţa respectivă este a unui băiat/bărbat, respectiv 1 în cazul unei fete/femei.

Vom considera o unitate neuronală de tip sigmoidal.[790] Notând cu y valoarea reală produsă de această unitate, rezultă că $y \in (0, 1)$. Este natural să interpretăm acest output ca fiind probabilitatea ca eticheta booleană t să fie 1. Formal, putem scrie $y \stackrel{def.}{=} P(t = 1 \mid x; w)$. În consecinţă, este natural să căutăm valori convenabile pentru ponderile w folosind principiul verosimilităţii maxime (MLE), sub forma următoare:

$$w_{MLE} = \underset{w}{\operatorname{argmax}} \prod_{i=1}^{N} P(t_i \mid x_i; w).$$

a.[791] Arătaţi că a maximiza expresia $\prod_{i=1}^{N} P(t_i \mid x_i; w)$ revine la a minimiza expresia următoare, despre care putem spune, după o observare atentă, că este o cross-entropie:[792]

$$E(w) = -\sum_{i=1}^{N} [t_i \ln y_i + (1 - t_i) \ln(1 - y_i)],$$

unde y_i este outputul produs de reţeaua neuronală pentru exemplul x_i.

b. Să presupunem că este posibil ca etichetele datelor de antrenament să fi fost puse eronat, şi anume cu probabilitatea ε. Considerând că datele de antrenament sunt distribuite în mod identic şi independent unele de altele, scrieţi expresia funcţiei de eroare / cost $E^*(w, \varepsilon)$ care corespunde log-verosimilităţii cu semn schimbat (engl., the negative log likelihood).

c. Verificaţi că funcţia de eroare $E(w)$ se obţine din $E^*(w, \varepsilon)$ pentru cazul particular $\varepsilon = 0$.

d. Explicaţi de ce anume funcţia de eroare $E^*(w, \varepsilon)$ va face ca modelul obţinut să fie [mai] robust în raport cu date incorect etichetate, spre deosebire de funcţia uzuală $E(w)$.

Răspuns:

[788] Pentru a vedea cum se face deducerea expresiei funcţiei de eroare ca sumă a pătratelor erorilor într-un context particular (însă sugestiv), cititorul poate consulta problema 13.

[789] De fapt, tipul *sigmoidal* al funcţiei de activare *nu* este important în acest context. Singurul fapt determinant este acela că output-ul este un număr din intervalul $(0, 1)$, deci poate fi interpretat ca o probabilitate.

[790] Mai general, se poate considera o reţea formată din unităţi sigmoidale. (De fapt, aşa a fost formulată problema originală de la CMU.) Într-un astfel de caz este însă suficient să presupunem că tipul unităţii / unităţilor de pe nivelul de ieşire din reţea este cel sigmoidal.

[791] Acest punct corespunde problemei 13.a de la capitolul *Metode de regresie*.

[792] A se vedea problema 63 de la capitolul de *Fundamente*.

a. Conform enunţului, $y_i \stackrel{def.}{=} P(t_i = 1|x_i; w)$**. Aşadar, putem scrie:**

$$P(t_i|x_i; w) = \begin{cases} y_i & \textbf{dacă} \quad t_i = 1, \\ 1 - y_i & \textbf{dacă} \quad t_i = 0. \end{cases}$$

Se verifică imediat că, din punct de vedere matematic, relaţia de mai sus se poate exprima în mod echivalent sub forma

$$P(t_i|x_i; w) = (y_i)^{t_i}(1 - y_i)^{1-t_i}.$$

Această expresie, în ciuda faptului că este mai puţin intuitivă, este mult mai convenabilă pentru calculele pe care trebuie să le facem în vederea aplicării principiului verosimilităţii maxime:[793]

$$\begin{aligned} w_{MLE} &\stackrel{def.}{=} \operatorname*{argmax}_w P(t_1, \ldots, t_n|x_1, \ldots, x_n; w) \stackrel{i.i.d.}{=} \operatorname*{argmax}_w \prod_{i=1}^{N} P(t_i \mid x_i; w) \\ &= \operatorname*{argmax}_w \sum_{i=1}^{N} \ln P(t_i \mid x_i; w) = \operatorname*{argmax}_w \sum_{i=1}^{N} \ln (y_i)^{t_i}(1 - y_i)^{1-t_i} \\ &= \operatorname*{argmax}_w \sum_{i=1}^{N} (t_i \ln y_i + (1 - t_i) \ln(1 - y_i)) = \operatorname*{argmax}_w (-E(w)) \\ &= \operatorname*{argmin}_w E(w). \end{aligned}$$

Pentru penultima egalitate de mai sus, vedeţi formula de definiţie a funcţiei E **din enunţ.**

***Observaţie*: Pentru demonstrarea faptului că** $E(w)$ **este funcţie convexă (echivalent:** $-E(w)$ **este funcţie concavă), vedeţi problema 14 de la capitolul *Metode de regresie*.**

b. În cazul în care etichetarea datelor de antrenament s-a făcut în mod eronat, cu probabilitatea ε**, rezultă imediat că**

$$x_i \textbf{ a fost etichetat cu } \begin{cases} 1 & \textbf{cu probabilitatea } 1 - \varepsilon\textbf{, dacă } t_i = 1, \\ 0 & \textbf{cu probabilitatea } \varepsilon\textbf{, dacă } t_i = 1, \\ 0 & \textbf{cu probabilitatea } 1 - \varepsilon\textbf{, dacă } t_i = 0, \\ 1 & \textbf{cu probabilitatea } \varepsilon\textbf{, dacă } t_i = 0. \end{cases}$$

Prin urmare, putem scrie în manieră condensată

$$P(t_i|x_i; w) = \begin{cases} y_i(1 - \varepsilon) + (1 - y_i)\varepsilon & \textbf{dacă} \quad t_i = 1, \\ (1 - y_i)(1 - \varepsilon) + y_i\varepsilon & \textbf{dacă} \quad t_i = 0. \end{cases}$$

Ba chiar şi mai mult, această expresie este echivalentă cu următoarea:

$$P(t_i|x_i; w) = [y_i(1 - \varepsilon) + (1 - y_i)\varepsilon]^{t_i}[(1 - y_i)(1 - \varepsilon) + y_i\varepsilon]^{1-t_i}.$$

Făcând un calcul similar cu cel de la punctul a**, va rezulta:**[794]

$$E^*(w, \varepsilon) = -\sum_{i=1}^{N} \{t_i \ln[y_i(1 - \varepsilon) + (1 - y_i)\varepsilon] + (1 - t_i) \ln[(1 - y_i)(1 - \varepsilon) + y_i\varepsilon]\}.$$

[793] De fapt, în cazul de faţă vom lucra cu verosimilitate condiţională.
[794] Se verifică imediat că $[y_i(1 - \varepsilon) + (1 - y_i)\varepsilon] + [(1 - y_i)(1 - \varepsilon) + y_i\varepsilon] = 1$.

c. Înlocuind parametrul ε cu valoarea 0 în expresia pe care tocmai am obţinut-o mai sus, vom avea:

$$\begin{aligned} E^*(w,0) &= -\sum_{i=1}^{N}\{[t_i \ln y_i + (1-y_i)\times 0] + (1-t_i)\ln[(1-y_i) + y_i \times 0]\} \\ &= -\sum_{i=1}^{N}[t_i \ln y_i + (1-t_i)\ln(1-y_i)] = E(w). \end{aligned}$$

d. Vom considera ca *exemplu* cazul extrem (însă instructiv) în care toate etichetele au fost puse incorect. Dacă am folosi funcţia de eroare $E(w)$, atunci modelul produs de algoritmul de retro-propagare ar fi complet eronat. Dacă în schimb vom folosi funcţia de eroare $E^*(w,\varepsilon)$ cu $\varepsilon = 1$, atunci algoritmul de retro-propagare va încerca să înveţe un model care clasifică datele exact invers faţă de etichetele originale, ceea ce corespunde obiectivului nostru.

***Notă*: Problema 50 cere să se adapteze algoritmul de retro-propagare folosit pentru antrenarea reţelelor neuronale de tip "feed-forward" (vedeţi pr. 20) pentru cazul în care funcţia de optimizat este de tip cross-entropie, aşa cum a fost definită în acest exerciţiu.**

16. (Algoritmul de antrenare a perceptronului-prag, versiunea Rosenblatt: aplicare)

• ***prelucrare de Liviu Ciortuz, după***
CMU, 2013 fall, W. Cohen, E. Xing, Sample questions, pr. 2
CMU, 2013 fall, A. Smola, G. Gordon, midterm ex. practice, pr. 10
CMU, 2013 spring, A. Smola, B. Poczos, HW2, pr. 2

Se consideră o secvenţă de exemple $\bar{x}_i \in \mathbb{R}^d$, pentru $i = 1, \ldots, n$, împreună cu etichetele corespunzătoare $y_i \in \{-1, 1\}$. Vom antrena perceptronul-prag folosind următorul algoritm, cunoscut în literatura de specialitate sub numele de *perceptronul Rosenblatt*:

initialize $\bar{w} \leftarrow \bar{0}$
for $i = 1, \ldots, n$
  if $y_i\,(\bar{w} \cdot \bar{x}_i) \leq 0$ then
    $\bar{w} \leftarrow \bar{w} + y_i \bar{x}_i$
  end if
end for

unde operatorul $\cdot$ reprezintă produsul scalar, vectorii $\bar{x}_i \in \mathbb{R}^d$ desemnează instanţele de antrenament, iar etichetele asignate lor sunt notate (ca de obicei) cu y_i.

***Observaţie*: În pseudo-codul folosit mai sus se consideră că termenul liber w_0 fie este absent fie este inclus în $\bar{w}$ şi, în acest ultim caz, prima componentă a vectorului $\bar{x}_i$ (notată cu $x_{0,i}$) este prin convenţie 1.**

a. Calculaţi actualizările vectorului de parametri $\bar{w}$ folosind algoritmul de mai sus pe următorul set de exemple (considerând că nu se foloseşte termenul liber w_0):

$$\bar{x}_1 = (0,0,0,1,0,0,1),\ y_1 = 1$$
$$\bar{x}_2 = (1,1,0,0,0,1,0),\ y_2 = -1$$
$$\bar{x}_3 = (0,0,1,1,0,0,0),\ y_3 = 1$$
$$\bar{x}_4 = (1,0,0,0,1,1,0),\ y_4 = -1$$
$$\bar{x}_5 = (1,0,0,0,0,1,0),\ y_5 = -1$$

b. Verificaţi dacă separatorul obţinut la finalul execuţiei algoritmului va clasifica perfect toate datele de antrenament.

c. Care sunt diferenţele ce caracterizează acest algoritm în raport cu algoritmul prezentat în cartea *Machine Learning* a lui Tom Mitchell (pag. 88-94), care este bazat pe aşa-numita *regulă delta*?[795] **Vă puteţi referi de exemplu la modul de lucru "batch" vs. modul incremental / stochastic, iniţializarea ponderilor, rata de învăţare, şi criteriul de oprire.**

Răspuns:

a. Conform algoritmului dat, avem:

$$\begin{aligned}
i=1: &\quad \bar{w} = \bar{0} \overset{not.}{=} (0,0,0,0,0,0,0) \Rightarrow y_1\,(\bar{w}\cdot\bar{x}_1) = 0 \leq 0\\
&\quad \Rightarrow \bar{w} \leftarrow y_1\bar{x}_1 = \bar{x}_1 = (0,0,0,1,0,0,1)\\
i=2: &\quad y_2\,(\bar{w}\cdot\bar{x}_2) = -(0,0,0,1,0,0,1)\cdot(1,1,0,0,0,1,0) = 0 \leq 0\\
&\quad \Rightarrow \bar{w} \leftarrow \bar{w} + y_2\bar{x}_2 = \bar{x}_1 - \bar{x}_2 = (-1,-1,0,1,0,-1,1)\\
i=3: &\quad y_3\,(\bar{w}\cdot\bar{x}_3) = (-1,-1,0,1,0,-1,1)\cdot(0,0,1,1,0,0,0) = 1 > 0\\
i=4: &\quad y_4\,(\bar{w}\cdot\bar{x}_4) = -(-1,-1,0,1,0,-1,1)\cdot(1,0,0,0,1,1,0) = 2 > 0\\
i=5: &\quad y_5\,(\bar{w}\cdot\bar{x}_5) = -(-1,-1,0,1,0,-1,1)\cdot(1,0,0,0,0,1,0) = 2 > 0.
\end{aligned}$$

b. Ştim deja de la punctul precedent că $y_i(\bar{w}\cdot\bar{x}_i) > 0$ pentru $i \in \{3,4,5\}$. Aşadar, va trebui să mai verficăm doar dacă are loc această relaţie şi pentru $i \in \{1,2\}$:

$$\begin{aligned}
i=1: &\quad y_1(\bar{w}\cdot\bar{x}_1) = (-1,-1,0,1,0,-1,1)\cdot(0,0,0,1,0,0,1) = 2 > 0\\
i=2: &\quad y_2(\bar{w}\cdot\bar{x}_2) = -(-1,-1,0,1,0,-1,1)\cdot(1,1,0,0,0,1,0) = 3 > 0.
\end{aligned}$$

În concluzie, separatorul $\bar{w} = (-1,-1,0,1,0,-1,1)$ clasifică perfect datele de antrenament.

c. Este imediat că perceptronul Rosenblatt corespunde modului de lucru incremental: el face actualizarea ponderilor după fiecare instanţă de antrenament $\bar{x}_i$ pentru care $sign(\bar{w}\cdot\bar{x}_i)$, i.e., clasificarea produsă de actualul $\bar{w}$, nu coincide cu eticheta y_i.

Iniţializarea ponderilor se face în acest algoritm cu 0. În algoritmul clasic (cel din cartea lui Tom Mitchell), iniţializarea ponderilor w_j, pentru $j = 1,\ldots,d$, se face cu numere mici (în modul), alese în mod aleatoriu.

Forma regulii de actualizare a ponderilor perceptronului Rosenblatt nu coincide cu *regula delta*:

$$\bar{w} \leftarrow \bar{w} + \eta(y_i - o_i)\bar{x}_i \tag{363}$$

[795] Vedeţi formula (4.10) din cartea citată, pag. 93.

unde $\eta > 0$ este rata de învăţare. Legătura dintre cele două forme se poate face în felul următor:

Este evident că regula (363) are efect doar în cazul în care $y_i \neq o_i$, ceea ce echivalează cu $y_i(\bar{w} \cdot \bar{x}_i) \leq 0$ din algoritmul Rosenblatt. Perceptronul-prag foloseşte funcţia de activare *sign*, aşadar faptul că y_i este diferit de o_i implică fie $y_i = 1$ şi $o_i = -1$, fie $y_i = -1$ şi $o_i = 1$. În ambele situaţii rezultă $y_i - o_i = 2y_i$. Prin urmare, folosind pre-condiţia $y_i(\bar{w} \cdot \bar{x}_i) \leq 0$, regula (363) devine

$$\bar{w} \leftarrow \bar{w} + 2\eta \, y_i x_i$$

Luând $\eta = 1/2$, obţinem forma regulii din algoritmul Rosenblatt.

Algoritmul Rosenblatt face o singură trecere prin setul de instanţe $\{x_i\}_i$. Evident, se poate extinde în mod natural acest algoritm pentru a proceda ca în cazul clasic), permiţând trecerea de mai multe ori prin setul de instanţe, eventual până la convergenţă, adică până când în cursul unei aceleiaşi iteraţii nu se mai fac deloc ajustări ale ponderilor w (presupunând că că setul de instanţe de antrenament este liniar separabil).

17. **(Perceptronul Rosenblatt, câteva proprietăţi simple: atât numărul de greşeli cât şi poziţia finală a separatorului depind de ordinea de furnizare a exemplelor)**

■ ● · ***CMU, 2015 spring, T. Mitchell, N. Balcan, HW6, pr. 3.ab***
CMU, 2017 spring, Barnabas Poczos, HW3, pr. 1.bc

Considerăm că rulăm algoritmul *Perceptron* al lui Rosenblatt,[796] folosind un anumit şir de exemple S. (Vă reamintim că un *exemplu* este o *instanţă* — punct din $\mathbb{R}^d$ — împreună cu *eticheta* sa). Fie S' acelaşi set de exemple ca şi S, însă prezentate într-o altă ordine.

a. Face oare algoritmul *Perceptron* acelaşi număr de greşeli pe şirul / succesiunea de exemple S ca şi pe şirul S'? Dacă da, de ce? Dacă nu, indicaţi două astfel de şiruri (S şi S') pe care algoritmul *Perceptron* face un număr diferit de greşeli.

b. Din punct de vedere geometric, algoritmul *Perceptron* clasifică instanţele date conform poziţiei lor relativ la un hiperplan de separare. Vectorul de ponderi ($\bar{w}$) învăţat de către *Perceptron* va fi normal [adică, perpendicular] pe acest hiperplan? Răspundeţi prin *Adevărat* sau *Fals* şi justificaţi în mod riguros.

Răspuns:

a. Considerăm următorul set de exemple (S), în ordinea indicată:

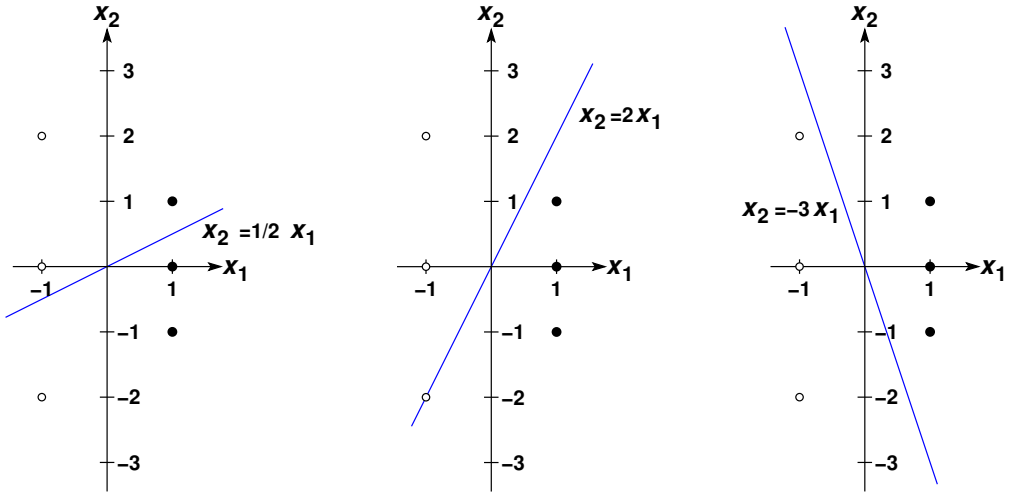
exemplul	1	2	3	4	5	6
instanţa (x_1, x_2)	$(-1, 2)$	$(1, 0)$	$(1, 1)$	$(-1, 0)$	$(-1, -2)$	$(1, -1)$
eticheta y	-1	$+1$	$+1$	-1	-1	$+1$

[796] Pentru pseudo-cod, vedeţi enunţul problemei 16.

Sintetizăm execuţia algoritmului *Perceptron* pe aceste date, fără a folosi termenul "bias" (w_0), alcătuind tabelul următor:

iteraţia(i)	$\bar{x}_i$	y_i	$y_i\bar{w}\cdot\bar{x}_i$	$\bar{w}$	greşeală	ec. separatorului
iniţializare	–	–	$(0,0)$	–	–	
1	$(-1,2)$	-1	$0\le 0$	$(1,-2)$	da	$x_2=0.5x_1$
2	$(1,0)$	$+1$	$1>0$	$(1,-2)$	nu	$x_2=0.5x_1$
3	$(1,1)$	$+1$	$-1\le 0$	$(2,-1)$	da	$x_2=2x_1$
4	$(-1,0)$	-1	$2>0$	$(2,-1)$	nu	$x_2=2x_1$
5	$(-1,-2)$	-1	$0\le 0$	$(3,1)$	da	$x_2=-3x_1$
6	$(1,-1)$	$+1$	$2>0$	$(3,1)$	nu	$x_2=-3x_1$

Aşadar, algoritmul a comis trei greşeli. Cei trei separatori obţinuţi în cursul aplicării algoritmului *Perceptron* pe şirul de exemple S sunt prezentaţi în figura următoare.



Este imediat că modul în care se construieşte separatorul — a cărui expresie analitică este o combinaţie liniară de instanţe greşit catalogate — este dependent de exemplele prezentate. Întrebarea care a fost pusă în enunţ este dacă pentru două succesiuni / ordini diferite, rezultatele finale (i.e., expresiile separatorilor obţinuţi) coincid (întotdeuna) sau nu. În continuare vom arăta că în mod obişnuit rezultatele finale (i.e., poziţiile finale ale separatorilor obţinuţi) *nu* coincid.

Considerăm S' secvenţa de exemple obţinută din S prin inversarea [ordinii] exemplelor $\bar{x}_1$ şi $\bar{x}_2$. Este imediat că la prima iteraţie perceptronul greşeşte şi apoi calculează $\bar{w}=(1,0)$, ceea ce conduce la ecuaţia separatorului $x_1=0$, adică axa Ox_2. Evident, aceasta dreaptă este un separator perfect pentru întregul set de exemple, deci algoritmul nu va mai comite până la final nicio [altă] greşeală. *Concluzia* este că, pe un acelaşi set de exemple de antrenament, algoritmul *Perceptron* poate comite un număr diferit de greşeli (şi, de asemenea, un alt separator) dacă exemplele îi sunt prezentate în două seccesiuni diferite.

***Observaţie importantă*: Deşi în ambele cazuri de mai sus (S şi S') algoritmul *Perceptron* găseşte un separator liniar pentru datele de intrare, acest fapt nu este garantat în gazul general, chiar dacă datele sunt liniar separabile. Vedeţi**

de exemplu problema 38. (Motivele sunt: rata de învăţare (implicită) prea mare şi / sau neparcurgerea setului de date decât o singură dată.)

b. Cunoaştem din geometria analitică faptul că orice doi vectori $\bar{w}$ şi $\bar{x}$ pentru care are loc relaţia $\bar{w} \cdot \bar{x} = 0$ sunt ortogonali (adică, perpendiculari unul pe celălalt).

În cazul perceptronilor (de tip liniar, prag sau sigmoidal), ecuaţia separatorului este tocmai de forma $\bar{w} \cdot \bar{x} = 0$, unde $\bar{x}$ este un punct arbitrar de pe dreapta [sau, în general, hiperplanul] separator. Considerând $\bar{x}_1$ şi $\bar{x}_2$ două astfel de puncte, diferenţa lor, $\bar{x}_1 - \bar{x}_2$, va fi un vector care are direcţia dreptei-separator. Din relaţiile $\bar{w} \cdot \bar{x}_1 = 0$ şi $\bar{w} \cdot \bar{x}_2 = 0$ rezultă $\bar{w} \cdot (\bar{x}_1 - \bar{x}_2) = 0$. Prin urmare, vectorul de ponderi $\bar{w}$ este perpendicular pe direcţia dreptei-separator.

Proprietatea aceasta se poate verifica în mod particular pe datele de la punctele precedente. De exemplu, la punctul a, pe şirul de exemple S, la iteraţia 1 avem $\bar{w} = (1, -2)$ şi se poate observa direct pe grafic că acest vector este perpendicular pe direcţia dreptei $y = \frac{1}{2}x$.

Aşadar, răspunsul la întrebarea din enunţ este *Adevărat*.

18. **(Convergenţa algoritmului de antrenare a perceptronului-prag, varianta Rosenblatt)**

prelucrare de Liviu Ciortuz, 2014, după
■ □ • ***Tommy Jaakkola, MIT, ML course, 2009 fall, lecture notes 2,***
cf. Block and Novikoff's results, 1962
(see also CMU, 2016 fall, E. Xing, Z. Bar-Joseph, HW3, pr. 3
CMU, 2014 fall, E. Xing, B. Poczos, HW1, pr. 4.3)

Presupunem că folosim perceptronul de tip prag în varianta Rosenblatt (vedeţi problema 16) şi vrem să învăţăm un concept, folosind instanţele de antrenament x_1, ..., x_n, ... $\in \mathbb{R}^d$ împreună cu etichetele corespunzătoare $y_1, \ldots, y_n, \ldots \in \{-1, 1\}$.

Demonstraţi că în cazul în care sunt îndeplinite condiţiile *i-iv* de mai jos, algoritmul de actualizare a ponderilor perceptronului „converge", adică termină într-un număr finit de paşi. Formal, exprimăm acest fapt astfel: $\exists m \in \mathbb{N}$ astfel încât $y_t w^{(m)} \cdot x_t > 0$ pentru orice $t \in \{1, \ldots, n, \ldots\}$, unde $w^{(m)}$ este vectorul de ponderi obţinut de perceptron la iteraţia m.

Iată acum *condiţiile* menţionate mai sus:

i. **Instanţele $x_1, \ldots, x_n, \ldots$ sunt separabile liniar prin originea sistemului de coordonate, cu o margine finită $\gamma > 0$. Din punct de vedere formal, aceasta înseamnă că există $w^* \in \mathbb{R}^d$ astfel încât $y_t w^* \cdot x_t \geq \gamma$ pentru $t = 1, \ldots, n, \ldots$.**

ii. **Toate instanţele $x_1, \ldots, x_n, \ldots$ sunt conţinute într-o sferă din $\mathbb{R}^d$ cu centrul în origine, adică $\exists R > 0$ astfel încât $||x_t|| \stackrel{def.}{=} \sqrt{x_t \cdot x_t} \leq R$ pentru orice t.**

iii. **Învăţarea se face în manieră *incrementală*, folosind următoarea regulă de actualizare a ponderilor:**

$$w^{(k+1)} = w^{(k)} + y_{t_k} x_{t_k} \text{ pentru un } t_k \in \{1, \ldots, n, \ldots\} \text{ a.î. } y_{t_k} w^{(k)} \cdot x_{t_k} \leq 0, \quad (364)$$

ceea ce înseamnă că instanţa x_{t_k} este clasificată eronat de către perceptron la iteraţia k.

iv. **Startarea procesului de învăţare se face cu** $w^{(0)} = 0 \in \mathbb{R}^d$**.**

***Indicaţie*: Arătaţi că la fiecare iteraţie** (k) **a algoritmului, sunt satisfăcute următoarele proprietăţi:**

a. $w^* \cdot w^{(k)} \geq k\gamma$**;**

b. $||w^{(k)}||^2 \leq kR^2$**;**

c. $k \leq \left(\frac{||w^*||}{\gamma} R\right)^2$**.**

Ultima inegalitate indică [o margine superioară pentru] numărul maxim de iteraţii executate de către perceptron.

***Observaţia 1*:** Notând cu θ_t unghiul format de vectorii x_t şi w^* în $\mathbb{R}^d$ şi ţinând cont de faptul că

$$cos\,(\theta_t) = cos\,(x_t, w^*) = \frac{x_t \cdot w^*}{||x_t||\,||w^*||},$$

deci

$$y_t w^* \cdot x_t = y_t\,||w^*||\,||x_t||\,cos\,(\theta_t),$$

ceea ce, coroborat cu condiţia i, implică $y_t\,||x_t||\,cos\,(\theta_t)\,||w^*|| \geq \gamma$, adică $y_t\,||x_t||\,cos\,(\theta_t) \geq \frac{\gamma}{||w^*||}$. Din punct de vedere geometric, aceasta înseamnă că în spaţiul $\mathbb{R}^d$ distanţa de la orice instanţă x_t la hiperplanul de separare care are ecuaţia $w^* \cdot x = 0$ (şi care trece prin originea sistemului de coordonate) este mai mare sau egală cu $\frac{\gamma}{||w^*||}$.

***Observaţia 2*:** Separatorul $w^{(k)}$ este o combinaţie liniară de instanţele x_i, întrucât regula de actualizare este $w^{(k+1)} = w^{(k)} + y_i x_i$. Observaţia aceasta este valabilă şi la antrenarea perceptronului-prag clasic, bazat pe regula delta.

Răspuns:

Algoritmul de actualizare a ponderilor perceptronului determină schimbarea poziţiei separatorului la fiecare iteraţie (k) **la care avem de a face cu un exemplu clasificat greşit. Intuitiv, ne aşteptăm ca poziţia separatorului la iteraţia** $k+1$ **(adică hiperplanul de ecuaţie** $w^{(k+1)} \cdot x = 0$**) să se apropie de poziţia separatorului** w^* **care defineşte conceptul de învăţat. În consecinţă, cosinusul unghiului dintre** $w^{(k)}$ **şi** w^* **ar trebui să crească de la o iteraţie la alta. Pentru a dovedi / verifica în mod riguros aceasta, vom ţine cont că, prin definiţie,**

$$\cos(w^{(k)}, w^*) = \frac{w^{(k)} \cdot w^*}{||w^{(k)}||\,||w^*||}.$$

Mai întâi vom compara valorile produsului scalar de la numărătorul fracţiei de mai sus, la două iteraţii succesive. Folosind regula (364), putem scrie:

$$w^{(k+1)} \cdot w^* = (w^{(k)} + y_{t_k} x_{t_k}) \cdot w^* = w^{(k)} \cdot w^* + y_{t_k} x_{t_k} \cdot w^*.$$

Întrucât $y_{t_k} x_{t_k} \cdot w^* \geq \gamma$ **(vedeţi condiţia** i **din enunţ), rezultă că valoarea produsului scalar** $w^{(k)} \cdot w^*$ **creşte la fiecare iteraţie cu o cantitate cel puţin egală cu** γ**. Cum** $w^{(0)} = 0$ **conform restricţiei *iv*, este imediat că** $w^{(k)} \cdot w^* \geq k\gamma$ **la iteraţia** k**. Cu aceasta, tocmai am demonstrat pe de o parte că relaţia** a **este adevărată,**

iar pe de altă parte că numărătorul fracţiei care defineşte $\cos(w^{(k)}, w^*)$ **creşte cel puţin liniar în raport cu** k**.**

A analiza evoluţia numitorului fracţiei care defineşte $\cos(w^{(k)}, w^*)$ **revine la a compara** $||w^{(k+1)}||$ **cu** $||w^{(k)}||$**, întrucât** $||w^*||$ **este constant în raport cu** k**.**

$$\begin{aligned} ||w^{(k+1)}||^2 &\overset{def.}{=} w^{(k+1)} \cdot w^{(k+1)} \overset{(364)}{=} (w^{(k)} + y_{t_k} x_{t_k}) \cdot (w^{(k)} + y_{t_k} x_{t_k}) \\ &= w^{(k)} \cdot w^{(k)} + 2 y_{t_k} w^{(k)} \cdot x_{t_k} + y_{t_k}^2 x_{t_k} \cdot x_{t_k} \\ &= ||w^{(k)}||^2 + 2 y_{t_k} w^{(k)} \cdot x_{t_k} + y_{t_k}^2 ||x_{t_k}||^2 \\ &= ||w^{(k)}||^2 + 2 y_{t_k} w^{(k)} \cdot x_{t_k} + ||x_{t_k}||^2 \\ &\leq ||w^{(k)}||^2 + R^2. \end{aligned}$$

Inegalitatea de mai sus derivă din faptul că $y_{t_k} w^{(k)} \cdot x_{t_k} \leq 0$ **(i.e., exemplul** t_k **este clasificat greşit la iteraţia** k**, conform condiţiei** iii **din enunţ) şi din ipoteza că orice instanţă de antrenament este conţinută în sfera de rază** R **şi având centrul în originea spaţiului** $\mathbb{R}^d$**, conform condiţiei** ii**. Cum** $w^{(0)} = 0$**, este imediat că** $||w^{(k)}||^2 \leq k R^2$**, deci** $||w^{(k)}|| \leq \sqrt{k}\, R$ **la iteraţia** k**. Cu aceasta, am demonstrat relaţia** b**.**

Acum, combinând rezultatele a **şi** b**, obţinem următoarea inegalitate:**

$$\cos(w^{(k)}, w^*) = \frac{w^{(k)} \cdot w^*}{||w^{(k)}||\, ||w^*||} \geq \frac{k\gamma}{\sqrt{k}\, R\, ||w^*||} = \sqrt{k} \frac{\gamma}{R\, ||w^*||}.$$

Ştim că valoarea funcţiei cos **este întotdeauna cel mult egală cu** 1**. În consecinţă,**

$$1 \geq \cos(w^{(k)}, w^*) \geq \sqrt{k} \frac{\gamma}{R\, ||w^*||} \Rightarrow k \leq \left(R \frac{||w^*||}{\gamma} \right)^2,$$

deci am demonstrat relaţia c**. Aceasta înseamnă că în condiţiile specificate în enunţ, antrenarea perceptronului-prag se va face în cel mult** $\left\lfloor \left(R \frac{||w^*||}{\gamma} \right)^2 \right\rfloor$ **iteraţii, unde perechea de simboluri** $\lfloor\ \rfloor$ **desemnează funcţia parte întreagă inferioară.**

***Observaţia 3*:** Marginea superioară calculată mai sus este remarcabilă, întrucât ea nu depinde nici de instanţele x_i şi nici de dimensiunea (d) a spaţiului din care sunt selectate aceste instanţe.

***Observaţia 4*:** Restricţia referitoare la *separabilitatea prin origine* a instanţelor de antrenament (vedeţi condiţia i, prima parte) — şi anume, termenul „liber“ w_0 asociat separatorului w^* trebuie să fie 0 — nu este de fapt limitativă.[797] Cazul general al separabilităţii liniare în $\mathbb{R}^d$, adică $y_t(w_0^* + w^* \cdot x_t) \geq 0$ pentru orice $t \in \{1, 2, \ldots, n \ldots\}$, poate fi redus la cazul particular al separabilităţii prin origine în $\mathbb{R}^{d+1}$ dacă pe de o parte se consideră $w' = (w^0, w^1, \ldots, w^d)$, cu $w^* = (w^1, \ldots, w^d)$, iar pe de altă parte se mapează toate instanţele de antrenament $x_t = (x_t^1, \ldots, x_t^d)$ din $\mathbb{R}^d$ în $\mathbb{R}^{d+1}$ astfel: $x'_k = (x_t^0, x_t^1, \ldots, x_t^d)$, cu $x_t^0 = 1$ pentru orice t. Cu această mapare, rezultă $y_t w' \cdot x'_t \geq 0$ pentru orice t (respectiv $y_t w' \cdot x'_t \geq \gamma$ când se lucrează cu margine finită, ca în enunţul acestei probleme). Nici *restricţia* iv ($w^{(0)} = 0$) nu este cu adevărat limitativă; în general algoritmii de antrenare a unităţilor / reţelelor neuronale iniţializează ponderile w la valori mici (în modul).

[797] În esenţă, ea ne uşurează calculele pe parcursul demonstraţiei, fiindcă ne plasează într-o situaţie particulară.

Similar, este imediat că *restricţia* potrivit căreia sfera în care sunt conţinute instanţele $x_1, \ldots, x_n, \ldots$ trebuie să aibă centrul în originea lui $\mathbb{R}^d$ (vedeţi condiţia ii, partea a doua) poate fi eliminată fără ca rezultatul de convergenţă să fie afectat.

***Observaţia 5*:** Deducerea marginii superioare de la punctul c de mai sus în cazul în care se foloseşte o rată de învăţare oarecare $\eta > 0$ se face extinzând în mod natural demonstraţia de mai sus (vedeţi pr. 40). În concluzie, mai rămân de examinat două condiţii: separabilitatea liniară cu margine $\gamma > 0$ şi conţinerea instanţelor de antrenament într-o sferă de rază finită. Problema 39.B va cere să se demonstreze că ambele condiţii sunt esenţiale pentru convergenţa perceptronului Rosenblatt în regim de învăţare online.

19. (Perceptronul Rosenblatt [dual] kernel-izat)

prelucrare de Liviu Ciortuz, după
■ ***CMU, 2013 spring, A. Smola, B. Poczos, HW2, pr. 2.d***
Stanford, 2016 fall, A. Ng, J. Duchi, HW2, pr. 2
MIT, 2006 fall, Tommy Jaakkola, HW2, pr. 3

Majoritatea clasificatorilor liniari pot fi kernel-izaţi, în vederea clasificării de date care sunt neseparabile liniar. În acest exerciţiu vă cerem să elaboraţi varianta kernel-izată a algoritmului *Perceptron* [dual]. (Pentru pseudo-codul variantei simple, nekernel-izate a algoritmului *Perceptron*, vedeţi pr. 16.)

Ca input, acest algoritm va lua secvenţa de instanţe etichetate $\{(x_i, y_i)\}_{i=1}^n$ cu $x_i \in \mathbb{R}^d$ şi $y_i \in \{-1, 1\}$, precum şi funcţia-nucleu $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, cu proprietatea că există $m \in \mathbb{N}^*$ (în general, $m > d$) şi o funcţie („mapare") $\phi : \mathbb{R}^d \to \mathbb{R}^m$ astfel încât $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ pentru orice x_i, x_j.[798] Perceptronul kernel-izat [dual] va lucra nu cu instanţele x_i, ci cu imaginile lor, $\phi(x_i)$, şi va încerca să găsească pentru acestea un separator liniar în spaţiul $\mathbb{R}^m$.

***Comentariu*:** Analizând algoritmul *Perceptron*, veţi constata că vectorul de ponderi $w \in \mathbb{R}^d$ este o combinaţie liniară de instanţe x_i, desemnată prin expresia $w = \sum_{i=1}^n \alpha_i x_i$. Din acest motiv este suficient să găsim valorile coeficienţilor α_i care asigură separabilitatea datelor. Această *proprietate* se dovedeşte deosebit de utilă în cazul funcţiior de mapare (ϕ) cu vectori de „trăsături" ($\phi(x)$) de dimensiuni foarte mari sau chiar infiniţi, aşa cum este cazul funcţiilor-nucleu gaussiene, numite şi *funcţii cu bază radială* (engl., Radial Basis Functions, RBF). Aşadar, în loc să actualizeze vectorul de ponderi w (cum face *Perceptron*-ul simplu), Perceptronul kernel-izat [dual] actualizează direct coeficienţii $\alpha_1, \ldots, \alpha_n$. Se va vedea că în timpul actualizărilor, operaţiile care se fac asupra instanţelor de antrenament date (x_i) sunt *doar* produse scalare de forma $\phi(x_i) \cdot \phi(x_j)$, care, aşa cum am precizat mai sus, pot fi văzute ca valori ale funcţiei-nucleu, $K(x_i, x_j)$. Această [a doua] *proprietate* asigură eficienţa calculelor.

Arătaţi cum anume se scrie regula de actualizare a coeficienţilor $\alpha_1, \ldots, \alpha_n$ la procesarea unui nou exemplu (x_i, y_i) de către algoritmul *Perceptron* kernel-izat [dual], şi cum se face predicţia ($y = 1$ sau -1) pentru o instanţă nouă x.

Răspuns:

În spaţiul $\mathbb{R}^m$ (care se mai numeşte, în contextul kernel-izării, *spaţiul de trăsături*), regula de actualizare a ponderilor w se scrie astfel:

[798] Din punct de vedere practic, este necesar ca funcţia K să poată fi calculată în mod eficient.

$$w^{(i)} \leftarrow w^{(i-1)} + y_i \phi(x_i) \text{ dacă } y_i w^{(i-1)} \cdot \phi(x_i) \leq 0,$$

unde operatorul $\cdot$ desemnează produsul scalar.

În consecinţă, întrucât facem iniţializarea $w^{(0)} = \bar{0}$, vectorul $w \in \mathbb{R}^m$ va fi întotdeauna o combinaţie liniară de vectori de trăsături, $\phi(x_i)$. Aceasta înseamnă că există coeficienţii $\alpha_l \in \mathbb{R}$ astfel încât $w^{(i)} = \sum_{l=1}^{i} \alpha_l \phi(x_l)$ după procesarea primelor i instanţe de antrenament. Aşadar, vectorul $w^{(i)}$ poate fi reprezentat în mod compact prin coeficienţii α_l (cu $l = 1, \ldots, i$) din această combinaţie liniară. În particular, valoarea iniţială $w^{(0)}$ corespunde cazului când suma nu conţine niciun termen (adică avem o listă vidă de coeficienţi α_l).

Arătăm acum că *putem calcula în mod eficient coeficienţii* α_i.
La iteraţia i trebuie să calculăm produsul scalar $w^{(i-1)} \cdot \phi(x_i)$. Întrucât produsul scalar în spaţiul de trăsături $\mathbb{R}^m$ este o operaţie costisitoare atunci când m este mare, vom ţine cont că

$$w^{(i-1)} \cdot \phi(x_i) = \left(\sum_{l=1}^{i-1} \alpha_l \phi(x_l) \right) \cdot \phi(x_i) = \sum_{l=1}^{i-1} \alpha_l (\phi(x_l) \cdot \phi(x_i)) = \sum_{l=1}^{i-1} \alpha_l K(x_l, x_i),$$

ceea ce înseamnă că aceşti coeficienţi se pot calcula într-adevăr în mod eficient.

În mod similar, se poate face în mod eficient predicţia clasei / etichetei pentru o instanţă nouă (de test) $x \in \mathbb{R}^d$:

$$w^{(i)} \cdot \phi(x) = \sum_{l=1}^{i} \alpha_l \phi(x_l) \cdot \phi(x) = \sum_{l=1}^{i} \alpha_l K(x_l, x).$$

Sumarizând, algoritmul pentru antrenarea Perceptronului kernel-izat [dual] se poate scrie în pseudo-cod astfel:

```
initialize α_i = 0 for i = 1, ..., n;
for i = 1, ..., n do
    if y_i Σ_{l=1}^{i-1} α_l K(x_l, x_i) ≤ 0 then
        α_i ← y_i;
    end if
end for
```

***Observaţie importantă*: Se poate arăta relativ uşor că raţionamentele (şi rezultatele) din acest exerciţiu se pot extinde şi la cazul perceptronului care foloseşte o altă codificare a ieşirilor y_i (de exemplu, 0 şi 1 în loc de -1 şi $+1$) şi, corespunzător, o altă funcţie de activare h (recte funcţia prag $0-1$ în locul funcţiei *sign*), o rată de învăţare oarecare $\eta > 0$ — dar face iniţializarea vectorului de ponderi $w \in \mathbb{R}^d$ cu $\bar{0}$, ceea ce este echivalent cu $\alpha_i = 0$, pentru $i = 1, \ldots, n$ — şi care eventual parcurge de mai multe ori setul de date de antrenament. În acest caz, regula de actualizare a perceptronului kernel-izat [dual] este de forma**

$$\alpha_i \leftarrow \alpha_i + \eta(y_i - h(w^{(i-1)} \cdot \phi(x_i))) \text{ dacă } y_i \neq h(w^{(i-1)} \cdot \phi(x_i)),$$

unde, exact ca mai sus,

$$w^{(i-1)} \cdot \phi(x_i) = \left(\sum_{l=1}^{i-1} \alpha_l \phi(x_l) \right) \cdot \phi(x_i) = \sum_{l=1}^{i-1} \alpha_l K(x_l, x_i).$$

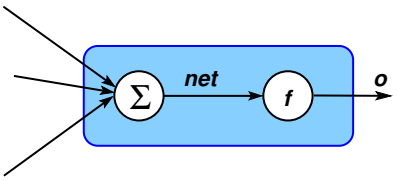
6.1.3 Reţele "feed-forward" — algoritmul de retropropagare

20. **(Reţele feed-forward, algoritmul de retro-propagare: o generalizare simplă în raport cu *Machine Learning* de Tom Mitchell, pag. 101-103)**

prelucrare de Liviu Ciortuz, după
■ • ***CMU, 2008(?) spring, HW2, pr. 2.2-4***

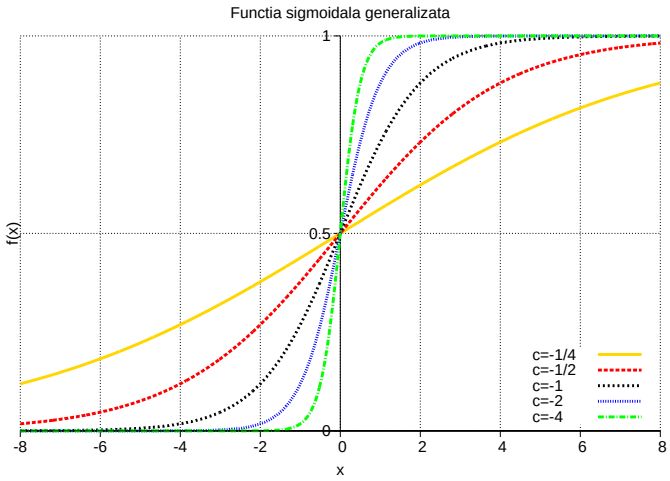
La acest exerciţiu veţi deriva regula de actualizare pentru algoritmul de retro-propagare (varianta stochastică) pentru o reţea neuronală artificială de tip "feed-forward", cu două niveluri de unităţi sigmoidale. Se consideră d unităţi de intrare, H unităţi ascunse şi K unităţi de ieşire.
Funcţia de eroare cu care se lucrează este

$$E(\bar{w}) = \frac{1}{2}\sum_k (t_k - o_k)^2,$$



unde $o_k = f(net_k) = \dfrac{1}{1+e^{c\ net_k}}$**, iar** c **este o constantă.**

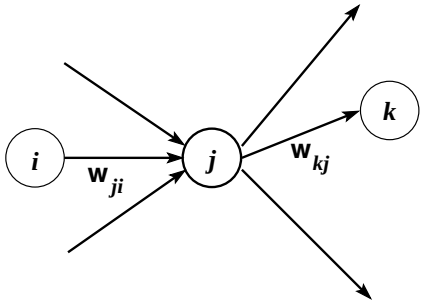
Observaţie: Este de remarcat faptul că se poate folosi constanta c pentru a „controla" panta cu care se face smoothing-ul (netezirea) funcţiei treaptă 0/1. Valori negative mici (adică, mari în modul) ale lui c implică o trecere foarte rapidă dinspre valori apropiate de 0 înspre valori apropiate de 1. Similar, valori negative mari (adică, apropiate de 0) pentru c implică o trecere lentă de la valori ≈ 0 la valori ≈ 1.



adaptare după: http://en.wikipedia.org/wiki/File:Logistic-curve.svg

a. Calculaţi derivata funcţiei sigmoidale generalizate $f(z) = \dfrac{1}{1+e^{cz}}$ **şi exprimaţi rezultatul în funcţie de însuşi** $f(z)$**, adică fără a-l implica direct pe** z**.**

b. Fie w_{kj} **ponderea conexiunii de la unitatea ascunsă** j **către unitatea de ieşire** k**. Arătaţi că regula de actualizare pentru** w_{kj} **este de forma** $w_{kj} \leftarrow w_{kj} + \eta\delta_k y_j$**, unde** y_j **este ieşirea unităţii ascunse** j**, iar** $\delta_k = f'(net_k)(t_k - o_k)$**, cu** $net_k = \sum_{j'} w_{kj'}\, y_{j'}$**.**



c. Arătaţi că regulile de actualizare pentru ponderile care corespund conexiunilor input-to-hidden sunt de forma $w_{ji} \leftarrow w_{ji} + \eta\tilde{\delta}_j x_i$**, unde** x_i **este intrarea** i**, iar** $\tilde{\delta}_j = f'(net_j)[\sum_{k=1}^{K} w_{kj}\delta_k]$**, cu** $net_j = \sum_{i'} w_{ji'}\, x_{i'}$**.**

Răspuns:

Demonstraţia de mai jos urmează, în esenţă, aceeaşi linie de gândire ca şi cea din cartea lui Tom Mitchell, *Machine Learning* (1997), pag. 101–103, unde s-a lucrat cu $c = -1$. Noi am preferat însă o redactare care să urmeze mai îndeaproape aplicarea formulelor de analiză matematică.

a. Mai întâi calculăm derivata lui f în raport cu argumentul z:

$$\frac{\partial f(z)}{\partial z} = \frac{\partial}{\partial z}\left(\frac{1}{1+e^{cz}}\right) = -\frac{1}{(1+e^{cz})^2}\cdot\frac{\partial}{\partial z}(1+e^{cz}) = -\frac{1}{(1+e^{cz})^2}\cdot c\cdot e^{cz}.$$

Apoi, în expresia pe care tocmai am obţinut-o, forţăm apariţia lui $f(z)$:

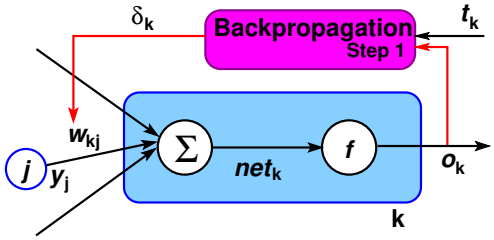
$$\begin{aligned}\frac{\partial f(z)}{\partial z} = -\frac{1}{(1+e^{cz})^2}\cdot c\cdot e^{cz} &= -c\cdot\frac{1}{1+e^{cz}}\cdot\frac{e^{cz}}{1+e^{cz}} = -c\cdot\frac{1}{1+e^{cz}}\cdot\frac{1+e^{cz}-1}{1+e^{cz}}\\ &= -c\cdot\frac{1}{1+e^{cz}}\cdot\left(1-\frac{1}{1+e^{cz}}\right) = -cf(z)(1-f(z)).\end{aligned}$$

***Observaţie importantă*: Punctele b şi c vor fi rezolvate de fapt în raport cu o funcţie oarecare f derivabilă, lăsată nespecificată. Forma particulară dată în enunţ pentru funcţia f (folosită la punctul a) nu are nicio consecinţă particulară asupra raţionamentului dezvoltat în continuare.**

b. Conform metodei gradientului descendent, actualizarea ponderilor w_{kj} de pe conexiunile care intră într-o unitate de ieşire k se va face conform formulei

$$w_{kj} \leftarrow w_{kj} - \eta\frac{\partial E}{\partial w_{kj}},$$

unde η este o constantă pozitivă (rata de învăţare).

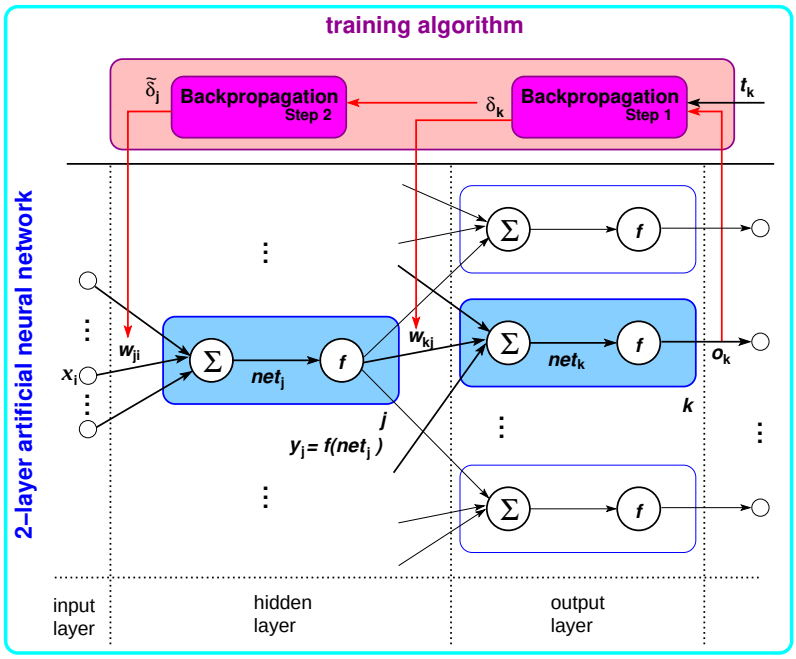


Intuitiv, ponderea w_{kj} influenţează valoarea funcţiei de eroare E (doar) prin intermediul lui net_k, valoarea care este transmisă componentei de activare a unităţii k de pe nivelul de ieşire. (S-a notat cu net_k suma $\sum_{j'} w_{kj'}y_{j'}$.)

Aşadar, este necesară aplicarea formulei pentru derivarea unei compuneri de funcţii derivabile:

$$\begin{aligned}\frac{\partial E}{\partial w_{kj}} &= \frac{\partial E}{\partial net_k}\frac{\partial net_k}{\partial w_{kj}} = \left(\frac{\partial}{\partial net_k}\frac{1}{2}\sum_{k'=1}^{K}(t_{k'}-o_{k'})^2\right)\left(\frac{\partial}{\partial w_{kj}}\sum_{j'=0}^{n_k}w_{kj'}y_{j'}\right)\\ &= \left(\frac{\partial}{\partial net_k}\frac{1}{2}(t_k - f(net_k))^2\right)y_j\\ &= y_j\left(\frac{1}{2}2(t_k-f(net_k))\frac{\partial}{\partial net_k}(t_k-f(net_k))\right) = -y_j(t_k-f(net_k))f'(net_k),\end{aligned}$$

unde n_k este numărul de intrări ale unităţii k. În particular, $n_k = H$ dacă se consideră toate conexiunile posibile (hidden-to-output).



Dacă notăm $\delta_k = (t_k - f(net_k))f'(net_k)$**, atunci avem** $\dfrac{\partial E}{\partial w_{kj}} = -\delta_k y_j$**, iar regula de actualizare a ponderilor unităţii ascunse** j **devine:**

$$w_{kj} \leftarrow w_{kj} - \eta \frac{\partial E}{\partial w_{kj}} = w_{kj} + \eta \delta_k y_j$$

c. Ca şi la punctul precedent, actualizarea ponderilor w_{ji} **de pe conexiunile care intră într-o unitate ascunsă** j **se va face după formula** $w_{ji} \leftarrow w_{ji} - \eta \dfrac{\partial E}{\partial w_{ji}}$**. Ponderea** w_{ji} **influenţează valoarea funcţiei de eroare** E **(doar) prin intermediul lui** net_j**, valoarea care intră în componenta de activare a unităţii ascunse** j**. Folosind din nou formula pentru derivarea unei compuneri de funcţii derivabile şi ţinând cont că** $net_j = \sum_{i'} w_{ji'} x_{i'}$**, vom avea:**

$$\begin{aligned}
\frac{\partial E}{\partial w_{ji}} &= \frac{\partial E}{\partial net_j}\frac{\partial net_j}{\partial w_{ji}} = \Big(\frac{\partial}{\partial net_j}\frac{1}{2}\sum_{k=1}^{K}(t_k - o_k)^2\Big)\Big(\frac{\partial}{\partial w_{ji}}\sum_{i'=0}^{d} w_{ji'}x_{i'}\Big) \\
&= \Big(\frac{1}{2}\sum_{k=1}^{K} 2(t_k - o_k)\frac{\partial}{\partial net_j}(t_k - o_k)\Big)x_i = -x_i\sum_{k=1}^{K}(t_k - o_k)\frac{\partial o_k}{\partial net_j}
\end{aligned}$$

Pe de altă parte, valoarea net_j **influenţează valoarea** o_k **(doar) prin intermediul lui** net_k**. Aşadar, urmează că**

$$\begin{aligned}
\frac{\partial o_k}{\partial net_j} &= \frac{\partial o_k}{\partial net_k}\frac{\partial net_k}{\partial net_j} = f'(net_k)\frac{\partial}{\partial net_j}\sum_{j'=0}^{n_k} w_{kj'}y_{j'} \\
&= f'(net_k)\frac{\partial}{\partial net_j}\sum_{j'=0}^{n_k} w_{kj'}f(net_{j'}) = f'(net_k)f'(net_j)w_{kj}
\end{aligned}$$

Înlocuind acest rezultat în egalitatea precedentă, vom avea:

$$\begin{aligned}\frac{\partial E}{\partial w_{ji}} &= -x_i \sum_{k=1}^{K}(t_k - o_k)f'(net_k)f'(net_j)w_{kj} \\ &= -x_i f'(net_j)\sum_{k=1}^{K}(t_k - o_k)f'(net_k)w_{kj} = -x_i f'(net_j)\sum_{k=1}^{K}\delta_k w_{kj}\end{aligned}$$

Folosind notaţia $\widetilde{\delta_j} = f'(net_j)\sum_{k=1}^{K}\delta_k w_{kj}$**, egalitatea de mai sus devine** $\frac{\partial E}{\partial w_{ji}} = -x_i\widetilde{\delta_j}$**, iar regula de actualizare a ponderii se transformă în:**

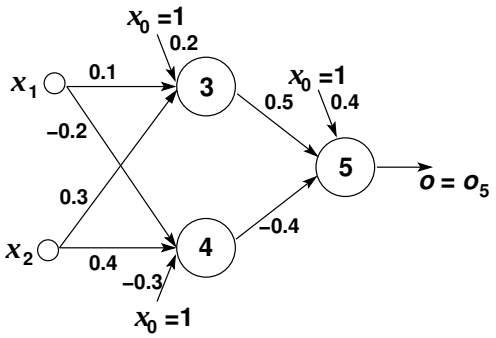
$$w_{ji} \leftarrow w_{ji} + \eta\widetilde{\delta_j}x_i$$

***Observaţie*: În mod similar cu demonstraţia din cartea *Machine Learning* de Tom Mitchell, pag. 101-103, demonstraţia de aici poate fi extinsă în mod facil la reţele neuronale "feed-forward" cu un număr oarecare de niveluri ascunse. De asemenea, mai este posibilă *încă o generalizare*: funcţia de activare f poate fi diferită de la o unitate la alta (sigmoidală, generalizat-sigmoidală (cu o anumită constantă c fixată), liniară etc). În demonstraţie va fi suficient să ataşăm simbolului f indicele unităţii neuronale respective.**

21. **(Aplicarea algoritmului de retro-propagare a erorilor pe o reţea feed-forward formată din unităţi sigmoidale dispuse pe 2 niveluri)**

prelucrare de Liviu Ciortuz,
după un exemplu din „Apprentissage artificiel",
A. Cornuéjols, L. Miclet, 2010, pag. 332, 341

Reţeaua neuronală din figura următoare este formată din unităţi sigmoidale.



a. Care este rezultatul produs de această reţea pentru intrarea $x \overset{not.}{=} (x_1, x_2) = (1, 1)$**?**

***Recomandare*: Pentru uşurinţa urmăririi calculelor, vă cerem să completaţi un tabel de forma următoare:**

i	$net_i = \sum_j w_{ij}x_j$	$o_i = \sigma(net_i)$
3		
4		
5		

b. La acest punct, veţi executa manual prima iteraţie a algoritmului de retro-propagare pe reţeaua dată. Presupunem că în cazul intrării $x = (1, 1)$ ieşirea produsă de reţea ar trebui să fie $t = 0$. Luând rata de învăţare $\eta = 1$, precizaţi care vor fi

- **valorile ponderilor $w_{30}, w_{31}, w_{32}, w_{40}, w_{41}, w_{42}, w_{50}, w_{51}, w_{52}$, după aplicarea acestei prime iteraţii în antrenarea reţelei;**[799]
- **noul output produs de reţea (după actualizarea ponderilor) pe aceeaşi intrare $x = (1, 1)$.**

c. Comparaţi rezultatele de la punctele a şi b. Ce constataţi?

Răspuns:

a. Completăm tabelul dat, ţinând cont că ieşirile neuronilor de pe nivelul ascuns (unităţile sigmoidale 3 şi 4) constituie intrările neuronului de pe nivelul exterior (unitatea sigmoidală 5).

i	$net_i = \sum_j w_{ij} x_j$	$o_i = \sigma(net_i)$
3	$0.2 + 0.1 + 0.3 = 0.6$	$1/(1 + e^{-0.6}) \simeq 0.646$
4	$-0.3 - 0.2 + 0.4 = -0.1$	$1/(1 + e^{0.1}) \simeq 0.475$
5	$0.4 + 0.5 \cdot 0.646 - 0.4 \cdot 0.475 = 0.533$	$1/(1 + e^{-0.533}) \simeq 0.630$

Aşadar, outputul produs de reţea este 0.63.

b. Pentru a vedea detaliile algoritmului de retro-propagare a erorii, cititorul poate consulta cartea *Machine Learning* de Tom Mitchell, pag. 98 sau problema 20 din prezentul capitol, punctele b şi c (unde parametrul c va fi considerat ca având valoarea 1).

***Precizare*: Ca să facilităm înţelegerea modului în care se aplică acest algoritm, în schema de mai jos am arătat cum anume vor fi calculate noile valori ale ponderilor w, precum şi cantităţile δ implicate în execuţia unei iteraţii a algoritmului. Ilustrarea se rezumă la parcurgerea (în ordinea output-to-input) unuia dintre drumurile din această reţea, şi anume drumul 5, 4, 2. Extensia la celelalte căi este facilă. În această imagine (dar numai aici) am notat cu w' noile valori pentru ponderi (calculate în ultimul pas al iteraţiei), pentru a nu fi confundate cu vechile valori, care intervin în calculul cantităţilor δ.**

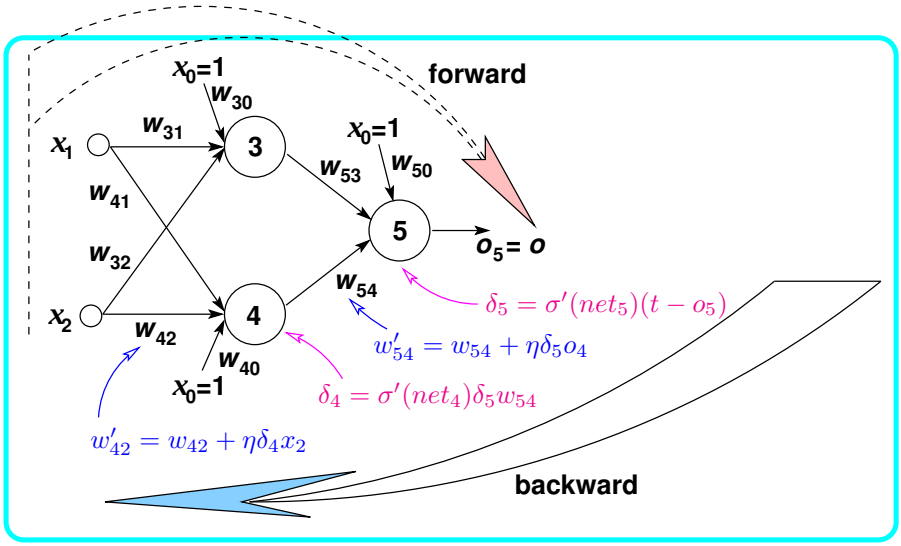
Conform algoritmului de retro-propagare,

$$\begin{aligned} \delta_5 &\overset{def.}{=} -\frac{\partial E}{\partial net_5} = \sigma'(net_5)(t - o) = \sigma(net_5)(1 - \sigma(net_5))(t - o) \\ &= o(1 - o)(t - o) = 0.63(1 - 0.63)(0 - 0.63) = -0.147 \end{aligned}$$

Outputul neuronului 3 constituie unul din inputurile neuronului 5. În notaţia folosită de Tom Mitchell, vom scrie $Downstream(3) = \{5\}$. Aşadar,

$$\delta_3 \overset{def.}{=} -\frac{\partial E}{\partial net_3} = o_3 \cdot (1 - o_3) \cdot \delta_5 \cdot w_{53} = 0.646 \cdot (1 - 0.646) \cdot (-0.147) \cdot 0.5 \simeq -0.017$$

[799] Semnificaţia pentru w_{ji} este cea din cartea *Machine Learning*a lui Tom Mitchell, la pag. 98: w_{ji} este ponderea de pe arcul / legătura de la unitatea (sau intrarea) i către unitatea j. A se vedea şi reprezentarea grafică a reţelei din rezolvarea de mai jos.



În mod similar, $Downstream(4) = \{5\}$**, şi**

$$\delta_4 \overset{def.}{=} -\frac{\partial E}{\partial net_4} = o_4 \cdot (1 - o_4) \cdot \delta_5 \cdot w_{54} = 0.475 \cdot (1 - 0.475) \cdot (-0.147) \cdot (-0.4) \simeq 0.015$$

Întrucât am terminat de calculat toate cantităţile de tip δ_j**, vom trece acum la actualizarea ponderilor reţelei. Mai întâi vom calcula noile valori pentru ponderile de pe conexiunile hidden-to-output, deci vom avea:**

$$w_{5j} \leftarrow w_{5j} + \Delta w_{5j} \text{ cu } \Delta w_{5j} = \eta\delta_5 o_j = \delta_5 o_j \text{ pentru } j \in \{0, 3, 4\}.$$

Aşadar,

$$\begin{cases} \Delta w_{50} = -0.147 \cdot 1 = -0.147 \\ \Delta w_{53} = -0.147 \cdot 0.646 \simeq -0.095 \\ \Delta w_{54} = -0.147 \cdot 0.475 \simeq -0.070 \end{cases} \Rightarrow \begin{cases} w_{50} \leftarrow 0.4 - 0.147 \simeq 0.253 \\ w_{53} \leftarrow 0.5 - 0.095 = 0.405 \\ w_{54} \leftarrow -0.4 - 0.070 = -0.470 \end{cases}$$

Apoi vom actualiza ponderile care corespund conexiunilor de tip input-to-hidden. Ştim că $\Delta w_{3i} = \eta\delta_3 x_i$ **pentru** $i \in \{0, 1, 2\}$**, aşadar inputul** $x_0 = x_1 = x_2 = 1$ **va implica** $\Delta w_{30} = \Delta w_{31} = \Delta w_{32} = 1 \cdot (-0.017) \cdot 1 = -0.017$**.**

În consecinţă,

$$\begin{cases} w_{30} \leftarrow 0.2 - 0.017 = 0.183 \\ w_{31} \leftarrow 0.1 - 0.017 = 0.083 \\ w_{32} \leftarrow 0.3 - 0.017 = 0.283 \end{cases}$$

Similar, fiindcă $\Delta w_{4i} = \eta\delta_4 x_i$ **pentru** $i \in \{0, 1, 2\}$**, rezultă că** $\Delta w_{40} = \Delta w_{41} = \Delta w_{42} = 1 \cdot 0.015 \cdot 1 = 0.015$ **şi, prin urmare:**

$$\begin{cases} w_{40} \leftarrow -0.3 + 0.015 = -0.285 \\ w_{41} \leftarrow -0.2 + 0.015 = -0.185 \\ w_{42} \leftarrow 0.4 + 0.015 = 0.415 \end{cases}$$

Acum putem calcula noua valoare produsă de reţeaua neuronală:

i	$net_i = \sum_j w_{ij} x_j$	$o_i = \sigma(net_i)$
3	$0.183 + 0.083 + 0.283 = 0.549$	$1/(1+e^{-0.549}) \simeq 0.634$
4	$-0.285 - 0.185 + 0.415 = -0.055$	$1/(1+e^{0.055}) \simeq 0.486$
5	$0.253 + 0.405 \cdot 0.634 - 0.47 \cdot 0.486 = 0.281$	$1/(1+e^{-0.281}) \simeq 0.569$

Aşadar, outputul produs de reţea după efectuarea primei iteraţii din cadrul algoritmului de retro-propagare a erorii este 0.569**.**

c. Valoarea produsă de reţea după ce a fost aplicată o iteraţie din algoritmul de retro-propagare (0.569) **este mai aproape de valoarea-target** (0) **decât fusese outputul produs de reţea înainte de aplicarea acestei prime iteraţii** (0.63)**. Este de aşteptat ca rezultatul să se îmbunătăţească la execuţia următoarelor iteraţii ale algoritmului.**

22. **(Regularizare: prevenirea overfitting-ului; cazul perceptronului liniar şi al reţelelor feed-forward formate din astfel de perceptroni)**

prelucrare de Liviu Ciortuz, după
■ • ***Tom Mitchell, "Machine Learning", 1997, pr. 4.10***

În vederea reducerii riscului de overfitting, pentru o unitate liniară (engl., un-thresholded perceptron) se poate considera funcţia de eroare

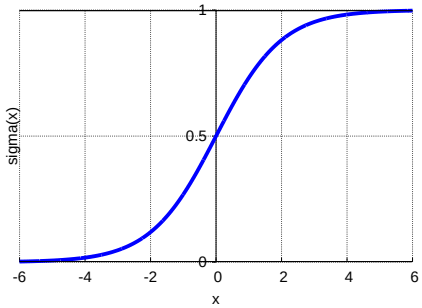
$$E(\bar{w}) = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2 + \gamma \sum_i w_i^2,$$

unde:

d **indexează instanţele de antrenament din mulţimea** $D \subset \mathbb{R}^n$**,**
$\bar{w} = (w_0, w_1, \ldots, w_n)$ **sunt ponderile pentru intrările perceptronului,**
t_d **este target-ul de învăţat pentru instanţa** d**,**
o_d **este outputul obţinut de perceptron pentru aceeaşi instanţă** d**,**
γ **este o constantă pozitivă.**

***Observaţie*:**
Suma $\sum_i w_i^2$ se poate scrie ca $||w||^2$, unde $||w||$ notează norma vectorului w. Intuitiv, minimizarea funcţiei obiectiv $E(\bar{w})$ va implica menţinerea lui $||w||^2$ la o valoare destul de redusă.

Efectul practic, în cazul folosirii de unităţi sigmoidale este următorul: graniţa (suprafaţa) de decizie calculată de către reţea pentru ponderi w mici (în modul) este aproape liniară — a se vedea graficul funcţiei σ în jurul originii —, ceea ce o împiedică să se „muleze" în jurul neregularităţilor din datele de antrenament.



a. Folosind metoda gradientului descendent, derivaţi regula de actualizare a ponderilor w_i pentru unitatea liniară care utilizează funcţia de eroare dată mai sus.

b. La acest punct vom considera că avem o reţea de tip feed-forward cu două niveluri de unităţi liniare şi că forma funcţiei de eroare pentru întreaga reţea este similară cu cea indicată la punctul precedent. Pentru acest tip de reţea, vă cerem să deduceţi regulile de actualizare a ponderilor. Pentru conveninenţă, veţi considera varianta stochastică / incrementală, adică veţi lucra cu o singură instanţă de antrenament pe ciclu / „epocă“ de antrenare.

***Observaţie*:** Deşi orice reţea neuronală formată doar din unităţi liniare este echivalentă cu o unitate liniară, am propus punctul de mai sus pentru a servi drept model pentru rezolvarea problemei propuse 49.

c. Arătaţi că minimizarea funcţiei $E(\bar{w})$ într-adevăr va conduce la găsirea unor valori mai apropiate de 0 pentru w_i decât în varianta clasică.

Răspuns:

Se poate arăta imediat că expresia

$$E(\bar{w}) = \frac{1}{2}\sum_{d\in D}(t_d - o_d)^2 + \gamma\sum_i w_i^2 = \frac{1}{2}\sum_{d\in D}(t_d - \sum_i w_i\, x_{i,d})^2 + \gamma\sum_i w_i^2$$

este o funcţie convexă în raport cu toate argumentele w_i. Aşadar, în vederea minimizării erorii, se justifică folosirea metodei gradientului descendent.

a. Avem:

$$\frac{\partial E}{\partial w_i} = \frac{\partial E}{\partial w_i}\Big(\frac{1}{2}\sum_{d\in D}(t_d - o_d)^2 + \gamma\sum_{i'} w_{i'}^2\Big) = 2\cdot\frac{1}{2}\sum_d (t_d - o_d)\frac{\partial}{\partial w_i}(t_d - o_d) + 2\gamma w_i$$

$$= -\sum_d (t_k - o_k)\frac{\partial}{\partial w_i}\sum_{i'} w_{i'}x_{i',d} + 2\gamma w_i = -\sum_d (t_d - o_d)x_{i,d} + 2\gamma w_i$$

Aşadar,

$$w_i \leftarrow w_i - \eta\frac{\partial E}{\partial w_i} = w_i + \eta[\sum_d (t_d - o_d)x_{i,d} - 2\gamma w_i] = (1 - 2\eta\gamma)w_i + \eta\sum_d (t_d - o_d)x_{i,d}$$

b. Înainte de a proceda la rezolvarea propriu-zisă a acestui punct al problemei, vom face următoarea *observaţie importantă*:

Cititorul atent va remarca, desigur, analogia cu problema 20 (şi problema 47). Această analogie este justificată, însă doar într-o anumită măsură. Va trebui să ţinem cont că:

i. Aici f corespunde funcţiei identitate, fiindcă lucrăm cu unitatea liniară. Aşadar, $f'(net) = 1$ pentru orice valoare a expresiei net.

ii. Nu vom mai putea lucra cu expresiile

$$\delta_k \overset{def.}{=} \frac{\partial E}{\partial net_k} \text{ şi } \tilde{\delta}_j \overset{def.}{=} \frac{\partial E}{\partial net_j}$$

unde k este indicele unei unităţi neuronale de pe nivelul de ieşire iar j este indicele unei unităţi de pe nivelul ascuns. Explicaţia ţine de faptul că expresia E nu se poate scrie ca o compunere de funcţii $E(net_-(w_-))$, ci doar ca $E(net_-(w_-), w_-)$, deci pur şi simplu $E(\bar{w})$.[800] Aşadar, nu vom putea scrie, ca în rezolvarea problemei 20 (şi a problemei 47), că

[800] În această notaţie, caracterul '_' ţine loc de indice, lăsându-i valoarea nespecificată.

$\frac{\partial E}{\partial w_-} = \frac{\partial E}{\partial net_-}\frac{\partial net_-}{\partial w_-}$. În consecinţă, va trebui să calculăm direct derivatele parţiale $\frac{\partial E}{\partial w_-}$. Similar cu demonstraţia din cartea *Machine Learning* de Tom Mitchell de la pag. 101-103 şi cu rezolvarea problemei 20 punctele b şi c, vom calcula aceste derivate parţiale mai întâi pentru variabilele care reprezintă ponderile de pe conexiunile hidden-to-output ale reţelei şi apoi pentru variabilele care corespund ponderilor de pe conexiunile input-to-hidden.

Pentru convenienţă, la acest punct al problemei vom urma convenţia de scriere a variabilelor w folosind doi indici: w_{kj} va corespunde unei conexiuni hidden-to-output (unde j indică o unitate de pe nivelul ascuns, iar k o unitate de pe nivelul de ieşire), iar w_{ji} va corespunde unei conexiuni input-to-hidden (i – input, j – hidden). Notaţiile t_k şi o_k vor corespunde valorii-target şi respectiv outputului unităţii liniare k de pe nivelul de ieşire al reţelei.

Avem:

$$E(\bar{w}) = \frac{1}{2}\sum_{k\in Outputs}(t_k - o_k)^2 + \gamma||\bar{w}||^2$$

şi, în consecinţă:

$$\begin{aligned}
\frac{\partial E}{\partial w_{kj}} &= \frac{\partial}{\partial w_{kj}}\left[\frac{1}{2}(t_k - o_k)^2 + \gamma w_{kj}^2\right] + \sum_{k'\neq k}\frac{\partial}{\partial w_{kj}}\left[\frac{1}{2}(t_{k'} - o_{k'})^2 + \gamma w_{k'j}^2\right] \\
&= -\frac{1}{2}2(t_k - o_k)\frac{\partial}{\partial w_{kj}}o_k + 2\gamma w_{kj} \\
&= -(t_k - o_k)\frac{\partial}{\partial w_{kj}}\sum_{j'} w_{kj'}y_{j'} + 2\gamma w_{kj} = -(t_k - o_k)y_j + 2\gamma w_{kj}
\end{aligned}$$

Aşadar,

$$w_{kj} \leftarrow w_{kj} - \eta\frac{\partial E}{\partial w_{kj}} = w_{kj} + \eta[(t_k - o_k)y_j - 2\gamma w_{kj}] = (1 - 2\eta\gamma)w_{kj} + \eta(t_k - o_k)y_j$$

Fie acum indicii j şi i fixaţi ca mai sus, corespunzător unei conexiuni input-to-hidden. Notând ca şi Tom Mitchell $Downstream(j)$ mulţimea tuturor indicilor k cu proprietatea că există conexiune de la unitatea j către unitatea k, vom avea:

$$\begin{aligned}
\frac{\partial E}{\partial w_{ji}} &= \frac{\partial}{\partial w_{ji}}\Big\{\Big[\sum_{k\in Downstream(j)}\frac{1}{2}(t_k - o_k)^2\Big] + \gamma w_{ji}^2\Big\} \\
&= 2\gamma w_{ji} - \sum_{k\in Downstream(j)}(t_k - o_k)\frac{\partial}{\partial w_{ji}}o_k \\
&= 2\gamma w_{ji} - \sum_{k\in Downstream(j)}\Big[(t_k - o_k)\frac{\partial}{\partial w_{ji}}\Big(\sum_{j'} w_{kj'}y_{j'}\Big)\Big] \\
&= 2\gamma w_{ji} - \sum_{k\in Downstream(j)}\Big\{(t_k - o_k)\frac{\partial}{\partial w_{ji}}\Big[\sum_{j'} w_{kj'}\Big(\sum_{i'} w_{j'i'}x_{i'}\Big)\Big]\Big\} \\
&= 2\gamma w_{ji} - \sum_{k\in Downstream(j)}\Big[(t_k - o_k)\frac{\partial}{\partial w_{ji}}\Big(\sum_{j'}\sum_{i'} w_{kj'}\,w_{j'i'}\,x_{i'}\Big)\Big]
\end{aligned}$$

$$= \quad 2\gamma w_{ji} - x_i\Big(\sum_{k\in Downstream(j)} (t_k - o_k)w_{kj}\Big)$$

Aşadar,

$$w_{ji} \leftarrow w_{ji} - \eta\frac{\partial E}{\partial w_{ji}} = w_{ji} + \eta\Big[x_i\Big(\sum_{k\in Downstream(j)} w_{kj}(t_k - o_k)\Big) - 2\gamma w_{ji}\Big]$$
$$= (1 - 2\eta\gamma)w_{ji} + \eta x_i\Big(\sum_{k\in Downstream(j)} w_{kj}(t_k - o_k)\Big)$$

c. Analizând regulile de actualizare a ponderilor deduse la punctul b, se observă că [în ambele cazuri, input-to-hidden şi hidden-to-output] ele sunt de forma $w \leftarrow (1 - 2\eta\gamma)w + \Delta w$. Deosebirea faţă de cazul clasic ($w \leftarrow w + \Delta w$), este evidentă. Întrucât algoritmul de retro-propagare iniţializează ponderile w cu valori apropiate de 0 — iar în practică se folosesc valori mici pentru constantele η şi γ —, lucrând cu varianta funcţiei de eroare propusă în această problemă vom obţine valori ale ponderilor w mai aproape de 0 (decât în varianta clasică).

23. **(De ce nu sunt convenabile funcţiile-prag pentru antrenarea perceptronilor şi a reţelelor neuronale artificiale?)**

• ○ ***CMU, 2011 spring, Tom Mitchell, HW5, pr. 3.4***

Care sunt motivele pentru care în general funcţiile-prag nu convin pentru antrenare perceptronilor şi a reţelelor neuronale artificiale [folosind metoda gradientului]?

Răspuns:

Răspunsul cel mai des citat, şi anume că funcţia-prag nu este derivabilă nu este în totalitate corect. Funcţia modul ($|x|$) este folosită adeseori la aplicarea metodei gradientului, deşi este derivabilă peste tot în afară de un singur punct (şi anume, în 0), ca şi funcţia-prag. Un avantaj important al funcţiei modul este faptul că este continuă. Din contră, cel mai mare dezavantaj al funcţiei-treaptă este faptul că în orice punct în care este derivabilă, derivata sa este 0. Se ştie că la aplicarea metodei gradientului descendent actualizarea parametrilor w se face după o regulă de forma $w \leftarrow w \pm \eta\nabla_w E(w)$. Prin urmare, în cazul în care derivatele parţiale ale funcţiei $E(w)$ sunt constant nule, atunci regula gradientului va lăsa parametrul w neschimbat. În consecinţă, funcţia-prag nu poate servi la actualizarea valorilor parametrilor atunci când se foloseşte metoda gradientului.

24. **(Reţele cu unităţi neuronale având funcţii de activare liniare şi / sau sigmoidale: corespondenţa cu suprafeţe de decizie liniare / neliniare)**

□ • ○ ***CMU, 2008 fall, Eric Xing, midterm, pr. 4.1***

Fie o reţea neuronală având două niveluri, cu două unităţi pe nivelul ascuns şi o singură unitate pe nivelul de ieşire (output). Vom lucra cu *funcţia de*

***activare* liniară $y = C \cdot a \stackrel{not.}{=} C \cdot \sum_i w_i x_i$, unde C este o constantă, iar a este suma ponderată a intrărilor şi, de asemenea, cu funcţia sigmoidală (sau, logistică): $y = \sigma(a) \stackrel{not.}{=} \frac{1}{1+e^{-a}}$.**

a. Presupunem că toate unităţile sunt liniare. Poate oare reţeaua aceasta să genereze / determine graniţe de decizie (engl., decision boundaries) pe care un *model de regresie* standard de forma $y = b_0 + b_1 x_1 + b_2 x_2 + \varepsilon$ nu le poate genera?

b. Presupunem că unităţile ascunse ale reţelei noastre sunt de tip *sigmoidal*, iar unitatea de [pe nivelul de] ieşire este de tip *liniar*. Este oare posibil ca în acest caz reţeaua să genereze graniţe de decizie neliniare?

c. Folosind funcţia de activare *sigmoidală* pentru toate unităţile (atât cele de pe nivelul ascuns cât şi cea de pe nivelul de ieşire), este oare posibil să aproximăm suprafeţe de decizie oricât de complicate prin combinarea multor [porţiuni de] graniţe de decizie neliniare? Ce schimbări ar trebui să operaţi asupra reţelei de mai sus pentru a putea aproxima orice graniţă de decizie?

Răspuns:

a. Nu. Orice reţea formată doar din unităţi liniare poate fi redusă la un model liniar simplu.

b. Da. Răspunsul se justifică imediat din punct de vedere matematic.

c. Da; avem nevoie de unităţi ascunse adiţionale. Atunci când folosim mai multe unităţi ascunse putem obţine suprafeţe de decizie mai complicate.

25. **(Adevărat sau Fals?)**

a. ***CMU, 2003 fall, T. Mitchell, A. Moore, midterm, pr. 6.b.5***

Regula gradientului descendent aplicată în regim "batch" pentru o unitate neuronală care are intrările x_1 şi x_2 şi ieşirea $w_0 + w_1(x_1 + 1) + w_2(x_2^2)$ produce:

$$\begin{aligned}
\Delta w_0 &= \eta \sum_d (t_i - o_i) \\
\Delta w_1 &= \eta \sum_i [(t_i - o_i) x_{i,1} + (t_i - o_i)] \\
\Delta w_2 &= \eta \sum_i [(t_i - o_i) 2 x_{i,2}]
\end{aligned}$$

unde:

– t_i este valoarea dorită (engl., target) pentru outputul unităţii neuronale pentru exemplul i;

– o_i este outputul unităţii neuronale pentru exemplul i;

– $x_{i,1}$ este valoarea intrării / atributului x_1 din exemplul i;

– $x_{i,2}$ este valoarea intrării / atributului x_2 din exemplul i.

b. ***CMU, 2003 fall, T. Mitchell, A. Moore, midterm, pr. 6.b.3***

Varianta incrementală (sau, stochastică) a metodei gradientului descendent se comportă întotdeauna mai bine decât varianta "batch".

c. ***CMU, 2003 fall, T. Mitchell, A. Moore, midterm, pr. 6.b.2***

Suprafaţa de eroare urmată (conform metodei gradientului descendent) de către algoritmul de retro-propagare se modifică dacă modificăm datele de antrenament.

d. ***CMU, 2003 fall, T. Mitchell, A. Moore, final exam, pr. 7.e***

Perceptronul poate fi (dar nu neapărat este) capabil să obţină o mai bună performanţă de clasificare dacă (anterior antrenării) intrările sale sunt mapate într-un „spaţiu de trăsături" folosind o funcţie-nucleu cu baza radială.

Răspuns:

a. Fals.
Valoarea funcţiei de ieşire pentru exemplul i este $o_i(w_0, w_1, w_2) = w_0 + w_1(x_{i,1} + 1) + w_2(x_{i,2}^2)$, iar funcţia de eroare este $E(w_0, w_1, w_2) = \frac{1}{2}\sum_i (t_i - o_i)^2$. Ştim că $w_j = w_j + \Delta w_j$, unde $\Delta w_j = -\eta \frac{\partial E}{\partial w_j}$. Este imediat că $\frac{\partial E}{\partial w_j} = \sum_i (t_i - o_i) \frac{\partial (t_i - o_i)}{\partial w_j}$. Calculând derivatele parţiale din această expresie, vom obţine:

$$\frac{\partial (t_i - o_i)}{\partial w_0} = -1 \Rightarrow \frac{\partial E}{\partial w_0} = -\sum_i (t_i - o_i) \Rightarrow \Delta w_0 = \eta \sum_i (t_i - o_i);$$

$$\frac{\partial (t_i - o_i)}{\partial w_1} = -(x_{i,1} + 1) \Rightarrow \frac{\partial E}{\partial w_1} = -\sum_i (t_i - o_i)(x_{i,1} + 1)$$
$$\Rightarrow \Delta w_1 = \eta \sum_i (t_i - o_i)(x_{i,1} + 1);$$

$$\frac{\partial (t_i - o_i)}{\partial w_2} = -x_{i,2}^2 \Rightarrow \frac{\partial E}{\partial w_2} = -\sum_i (t_i - o_i) x_{i,2}^2 \Rightarrow \Delta w_2 = \eta \sum_i (t_i - o_i) x_{i,2}^2.$$

Se observă că Δw_2 pe care tocmai l-am calculat mai sus diferă de cel dat în enunţ, care este aşadar greşit.

b. Fals. Comparativ cu varianta "batch", varianta incrementală a metodei gradientului descendent are *avantajul* de a evita unele optime (în speţă, minime) locale. Însă varianta incrementală este doar o aproximare a metodei "batch" (care, pentru perceptronul liniar, obţine în mod garantat optimul (global)). În plus, atunci când cele două variante folosesc rate ale învăţării egale, varianta incrementală este mai lentă decât varianta "batch".

c. Adevărat. Algoritmul de retro-propagare aplică metoda gradientului descendent funcţiei de eroare, care poate fi de exemplu $\frac{1}{2}\sum_d (t_i - o_i)^2$. În calculul funcţiei de eroare intervin atât target-ul (eticheta) t_i cât şi atributele instanţelor de antrenament (acestea din urmă prin intermediul outputului o_i). În consecinţă, orice schimbare în datele de antrenament ca atare — cât şi extinderea sau reducerea setului de date de antrenament — poate afecta funcţia de eroare şi, în consecinţă, rezultatul algoritmului de retro-propagare.

d. Adevărat. Chestiunea aceasta se poate pune pentru orice clasificator care învaţă un separator liniar. Maparea mulţimii de instanţe antrenament $S = \{x_1, x_2, \ldots, x_l\} \subset \mathbb{R}^n$ folosind o funcţie-nucleu cu baza radială sau, mai general, o funcţie oarecare $\Phi : \mathbb{R}^n \to \mathbb{R}^m$ cu $m > n$ poate conduce la găsirea unui separator liniar pentru mulţimea $\{\Phi(x_1), \Phi(x_2), \ldots, \Phi(x_l)\}$ în $\mathbb{R}^m$, chiar dacă mulţimea S nu este liniar separabilă. Totuşi, găsirea unui astfel de separator (în urma aplicării funcţiei Φ) nu este garantată.

În general, chiar dacă nu se obţine separabilitate liniară în urma mapării cu funcţia Φ, este posibil (dar nu obligatoriu) ca în $\mathbb{R}^m$ să se obţină un optim mai convenabil pentru funcţia de optimizat. În particular, pentru perceptron, ca funcţie de optimizat se ia în mod obişnuit semisuma pătratelor erorilor. „Un optim mai convenabil" se va traduce în practică în faptul că funcţia învăţată în $\mathbb{R}^m$ se comportă mai bine pe date de test (prin comparaţie, eventual, cu funcţia învăţată în $\mathbb{R}^n$).

***Observaţie*:** Funcţiile cu baza radială sunt un caz particular de funcţii-nucleu. (A se vedea capitolul *Maşini cu vectori-suport*.) Funcţiile-nucleu prezintă avantajul că produsele scalare $\Phi(x) \cdot \Phi(y)$ pot fi calculate în mod eficient, chiar dacă m este mult mai mare decât n. Această proprietate este foarte convenabilă pentru implementarea maşinilor cu vectori-suport.

26. **(Reţele neuronale profunde: dispariţia – ori, dimpotrivă, „explozia" – vectorului gradient)**

■ □ • ○ ***CMU, 2015 fall, E. Xing, Z. Bar-Joseph, HW3, pr. 1.3***

În acest exerciţiu veţi studia anumite dificultăţi care apar la aplicarea algoritmului de retro-propagare pentru antrenarea reţelelor neuronale profunde (engl., deep neural networks).[801] Pentru convenienţă, vom considera cea mai simplă reţea neuronală profundă, şi anume una în care există câte un singur neuron pe fiecare nivel, iar outputul produs de neuronul de pe nivelul j este

$$z_j = f(net_j) \text{ cu } net_j \overset{not.}{=} \begin{cases} b_1 + w_{11}x_1 + \ldots + w_{1d}x_d & \text{pentru } j = 1; \\ b_j + w_j z_{j-1} & \text{pentru } j = 2, \ldots, m, \end{cases} \tag{365}$$

unde

– f este o anumită funcţie de activare, a cărei derivată în punctul x este $f'(x)$,

– m este numărul de niveluri din această reţea neuronală,

– b_j este bias-ul (adică, termenul liber) al unităţii neuronale de pe nivelul j.

Vom nota cu L funcţia de eroare / cost / pierdere (engl., loss function) care este folosită ca funcţie obiectiv la antrenare.

a. Calculaţi derivata parţială a funcţiei L în raport cu b_1, bias-ul (adică, termenul liber) pentru neuronul de pe primul nivel.

***Indicaţie*: Pentru acest scop, nu este necesar să precizăm cine anume este funcţia L şi, la fel, cine este funcţia f. Presupunând doar că L se exprimă în**

[801] Se spune că o reţea neuronală este profundă dacă are cel puţin două niveluri ascunse.

funcţie de f şi este derivabilă, veţi exprima $\frac{\partial L}{\partial b_1}$ în funcţie de $net_1, \ldots, net_m$ şi $w_2, \ldots, w_m$. Veţi folosi regula pentru derivarea compunerilor de funcţii: $(f_1(f_2(x)))' = f_1'(f_2(x)) \cdot f_2'(x)$.

b. Presupunem că funcţia de activare f este obişnuita funcţie sigmoidală, $\sigma(x) = 1/(1 + \exp(-x))$ şi că ponderile $\bar{w}$ sunt iniţializate astfel încât $|w_j| < 1$ pentru $j = 1, \ldots, m$.

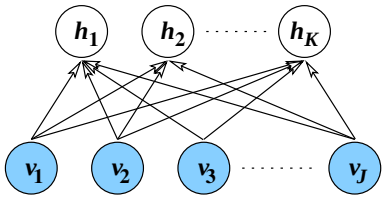
Explicaţi de ce derivata parţială de la punctul precedent, $\frac{\partial L}{\partial b_1}$, tinde la 0 atunci când m are valori mari. *Comentariu*: Din acest motiv, se spune că — în astfel de condiţii — avem de a face cu „dispariţia“ gradientului (engl., *gradient vanishing*).

Explicaţi de ce chiar şi atunci când valoarea lui $|w|$ este mare, derivata parţială menţionată mai sus tinde la 0 (deci, tinde să dispară), nu tinde la infinit (adică, să „explodeze“).

c. Una dintre modalităţile propuse pentru rezolvarea (parţială) a problemei „dispariţiei“ gradientului este ca în locul funcţiei sigmoidale să se folosească pentru activare *funcţia liniar-rectificată* (engl., rectified linear function), notată cu ReL. Funcţia de activare ReL este definită prin $\max\{0, x\}$. Explicaţi de ce funcţia ReL ne poate ajuta să evităm fenomenul de „dispariţie“ a gradientului.

d. O altă modalitate de rezolvare (parţială) a problemei „dispariţiei“ ori „exploziei“ gradientului este *pre-antrenarea nivel-cu-nivel* (engl., layer-wise pretraining).

Modelul *maşinii Boltzmann restricţionate* (engl., Restricted Boltzmann machine), abreviat RBM, este unul dintre modelele cele mai des folosite pentru pre-antrenarea [unei reţele neuronale] nivel-cu-nivel. Figura alăturată prezintă un exemplu de RBM conţinând K unităţi ascunse $(h_1, \ldots, h_K)$ şi J intrări $(v_1, \ldots, v_J)$.



În vederea antrenării unei RBM, definim [ca funcţie obiectiv] *distribuţia probabilistă comună* — peste vectorii de valori posibile $\bar{v} \overset{not.}{=} (v_1, \ldots, v_J)$ şi $\bar{h} \overset{not.}{=} (h_1, \ldots, h_K)$ — având forma [generală] următoare:

$$P(\bar{v}, \bar{h}) = \frac{1}{Z} \exp\left(\sum_i \theta_i \phi_i(\bar{v}, \bar{h})\right),$$

unde

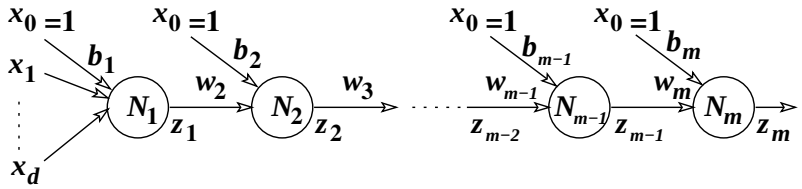
- $Z = \sum_{\bar{v}', \bar{h}'} \exp\left(\sum_i \theta_i \phi_i(\bar{v}', \bar{h}')\right)$ **este constanta de normalizare;**
- $\phi_i(\bar{v}, \bar{h})$ **sunt anumite trăsături / atribute;**
- θ_i **sunt parametrii care corespund ponderilor din RBM.**

Considerând că pentru antrenarea unei RBM se foloseşte metoda gradientului descent, arătaţi că expresia derivatei paţiale a logaritmului funcţiei de probabilitate marginală $P(\bar{v})$ în raport cu parametrul / ponderea θ_i este:

$$\frac{\partial \ln P(\bar{v})}{\partial \theta_i} = \sum_{\bar{h}} \phi_i(\bar{v}, \bar{h}) P(\bar{h}|\bar{v}) - \sum_{\bar{v}', \bar{h}'} \phi_i(\bar{v}', \bar{h}') P(\bar{v}', \bar{h}').$$

Răspuns:

a. Ilustrăm mai jos reţeaua neuronală profundă definită în enunţ.



Apoi vom scrie în mod desfăşurat relaţiile (365):

$$\begin{aligned} z_1 &= f(net_1) = f(b_1 + w_{11}x_1 + \ldots + w_{1d}x_d) \\ z_2 &= f(net_2) = f(b_2 + w_2 z_1) \\ &\ldots \\ o = z_m &= f(net_m) = f(b_m + w_m z_{m-1}) \end{aligned}$$

Funcţia de pierdere L se va exprima astfel:

$$\begin{aligned} & L(w_{11}, \ldots, w_{1d}, w_2, \ldots, w_m, b_1, \ldots, b_m) \\ & \overset{not.}{=} L(f(\underbrace{net_m}_{b_m + w_m z_{m-1}})) \\ & = L(f(b_m + w_m f(\underbrace{net_{m-1}}_{b_{m-1} + w_{m-1} z_{m-2}}))) \\ & \ldots \\ & = L(f(b_m + w_m f(b_{m-1} + w_{m-1} f(b_{m-2} + \ldots + w_3 f(\underbrace{net_2}_{b_2 + w_2 z_1}) \ldots))) \\ & = L(f(b_m + w_m f(b_{m-1} + w_{m-1} f(b_{m-2} + \ldots + w_3 f(b_2 + w_2 z_1) \ldots))) \\ & = L(f(b_m + w_m f(b_{m-1} + w_{m-1} f(b_{m-2} + \ldots + w_3 f(b_2 + w_2 f(\underbrace{net_1}_{b_1 + w_{11}x_1 + \ldots + w_{1d}x_d}) \ldots))) \end{aligned}$$

Pentru a deriva funcţia L în raport cu b_1, ne vom aminti mai întâi regula de derivare a unei compuneri multiple de funcţii $f_1(f_2(\ldots f_n(x)\ldots))$. Pe lângă forma clasică pe care o ştim din liceu, $f_1'(f_2(\ldots f_n(x)\ldots)) \cdot f_2'(\ldots f_n(x)\ldots) \cdot f_n'(x)$, ea se poate scrie sub forma aşa-numitei *reguli de înlănţuire*: $\frac{\partial f_1}{\partial f_2} \cdot \frac{\partial f_2}{\partial f_3} \cdot \ldots \cdot \frac{\partial f_2}{\partial f_1} \cdot \frac{\partial f_1}{\partial x}$.

Adoptând / aplicând această regulă pentru a calcula derivata parţială cerută în enunţ, vom avea:

$$\frac{\partial L}{\partial b_1} = \frac{\partial L}{\partial net_m} \cdot \frac{\partial net_m}{\partial net_{m-1}} \cdot \ldots \cdot \frac{\partial net_2}{\partial net_1} \cdot \frac{\partial net_1}{\partial b_1}$$

$$\begin{aligned} &= L'(f(net_m)) \cdot f'(net_m) \cdot \\ &\quad w_m \cdot f'(net_{m-1}) \cdot \\ &\quad w_{m-1} \cdot f'(net_{m-2}) \cdot \\ &\quad \dots \\ &\quad w_2 \cdot f'(net_1) \cdot \\ &\quad 1 \\ &= L'(f(net_m)) \cdot f'(net_1) \cdot \prod_{k=2}^{m} (f'(net_k) \cdot w_k) \end{aligned}$$

b. Ştim că $\sigma(x) \in (0,\, 1)$ şi $\sigma'(x) = \sigma(x)(1-\sigma(x))$ pentru orice $x \in \mathbb{R}$, iar maximul funcţiei $z(1-z)$ este $1/4$ şi se atinge în punctul $z = 1/2$. Prin urmare, $|\prod_{k=2}^{m} \sigma'(net_k)| \leq \left(\frac{1}{4}\right)^{m-1}$. Întrucât $|w_k| < 1$ pentru orice k (conform ipotezei din enunţ), rezultă imediat că $\left|\frac{\partial L}{\partial b_1}\right| = |L'(\sigma(net_m)) \cdot \sigma'(net_1)| \cdot \prod_{k=2}^{m} |\sigma'(net_k) \cdot w_k|$ tinde la 0 pentru $k \to +\infty$ şi pentru orice input $\bar{x}$ fixat.

Pentru a contracara acest fenomen de „dispariţie" a gradientului, ar trebui să impunem condiţii de forma $|w_k \sigma'(net_k)| > 1$. Trebuie însă să remarcăm faptul că $\sigma(net_k)$ depinde de w_k: $\sigma(net_k) = \sigma(b_k + w_k z_{k-1})$. În consecinţă, dacă îi vom da lui w_k posibilitatea să ia valori mari în modul, atunci $z_k \overset{not.}{=} b_k + w_k z_{k-1}$ va lua de asemenea valori mari în modul, iar $\sigma(z_k)$ va tinde fie la 0 fie la 1, deci $\sigma'(z_k) = \sigma(z_k)(1-\sigma(z_k))$ va tinde la 0. În sfârşit, se poate verifica în mod riguros că $\lim_{z \to \pm\infty} z\sigma'(z) = 0$ (lăsăm această demonstraţie ca exerciţiu cititorului).

c. Dacă f este funcţia de activare ReL, atunci $f'(x) = 0$ pentru $x < 0$ şi $f'(x) = 1$ pentru $x > 0$. Prin urmare, folosind această funcţie putem împiedica „dispariţia" gradientului.

d. Pornind de la expresia distribuţiei probabiliste $P(\bar{v}, \bar{h})$ care a fost dată în enunţ, explicitând mai întâi constanta de normalizare Z, vom putea scrie apoi expresia distribuţiei marginale $P(\bar{v})$:

$$P(\bar{v}, \bar{h}) = \frac{1}{\sum_{\bar{v}', \bar{h}'} \exp(\sum_k \theta_k \phi_k(\bar{v}', \bar{h}'))} \exp\Big(\sum_k \theta_k \phi_k(\bar{v}, \bar{h})\Big)$$

$$\Rightarrow P(\bar{v}) = \frac{1}{\sum_{\bar{v}', \bar{h}'} \exp(\sum_k \theta_k \phi_k(\bar{v}', \bar{h}'))} \sum_{\bar{h}} \exp\Big(\sum_k \theta_k \phi_k(\bar{v}, \bar{h})\Big)$$

Prin urmare,

$$\ln P(\bar{v}) = \ln \sum_{\bar{h}} \exp\Big(\sum_k \theta_k \phi_k(\bar{v}, \bar{h})\Big) - \ln \sum_{\bar{v}', \bar{h}'} \exp(\sum_k \theta_k \phi_k(\bar{v}', \bar{h}')).$$

În consecinţă, derivata parţială a funcţiei $\ln P(\bar{v})$ în raport cu parametrul θ_i se poate calcula astfel:

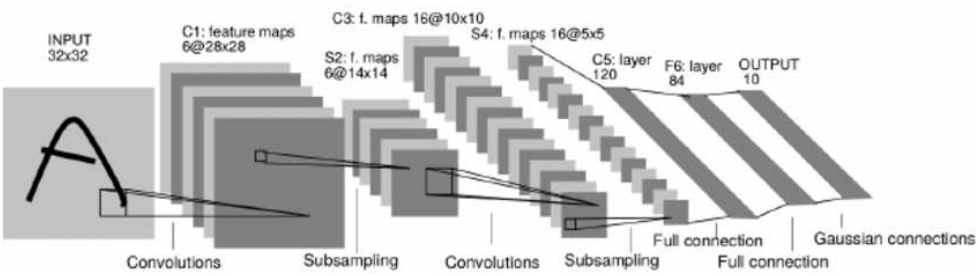
$$\begin{aligned} &\frac{\partial \ln P(\bar{v})}{\partial \theta_i} = \\ &= \frac{\sum_{\bar{h}} \exp\big(\sum_k \theta_k \phi_k(\bar{v}, \bar{h})\big) \cdot \phi_i(\bar{v}, \bar{h})}{\sum_{\bar{h}} \exp\big(\sum_k \theta_k \phi_k(\bar{v}, \bar{h})\big)} - \frac{\sum_{\bar{v}', \bar{h}'} \exp\big(\sum_k \theta_k \phi_k(\bar{v}', \bar{h}')\big) \cdot \phi_i(\bar{v}, \bar{h})}{\underbrace{\sum_{\bar{v}', \bar{h}'} \exp(\sum_k \theta_k \phi_k(\bar{v}', \bar{h}'))}_{Z}} \end{aligned}$$

$$
\begin{aligned}
&= \frac{\sum_{\bar{h}} \frac{1}{Z} \exp\left(\sum_k \theta_k \phi_k(\bar{v}, \bar{h})\right) \cdot \phi_i(\bar{v}, \bar{h})}{\sum_{\bar{h}} \frac{1}{Z} \exp\left(\sum_k \theta_k \phi_k(\bar{v}, \bar{h})\right)} - \sum_{\bar{v}', \bar{h}'} \frac{1}{Z} \exp\left(\sum_k \theta_k \phi_k(\bar{v}', \bar{h}')\right) \cdot \phi_i(\bar{v}, \bar{h}) \\
&= \frac{\sum_{\bar{h}} P(\bar{v}, \bar{h}) \cdot \phi_i(\bar{v}, \bar{h})}{P(\bar{v})} - \sum_{\bar{v}', \bar{h}'} P(\bar{v}', \bar{h}') \phi_i(\bar{v}', \bar{h}') \\
&= \sum_{\bar{h}} \frac{P(\bar{v}, \bar{h})}{P(\bar{v})} \cdot \phi_i(\bar{v}, \bar{h}) - \sum_{\bar{v}', \bar{h}'} P(\bar{v}', \bar{h}') \phi_i(\bar{v}', \bar{h}') \\
&= \sum_{\bar{h}} P(\bar{h} | \bar{v}) \cdot \phi_i(\bar{v}, \bar{h}) - \sum_{\bar{v}', \bar{h}'} P(\bar{v}', \bar{h}') \phi_i(\bar{v}', \bar{h}')
\end{aligned}
$$

27. **(Determinarea numărului de parametri şi de conexiuni din reţeaua neuronală convolutivă LeNet)**

prelucrare de Liviu Ciortuz, după
□ • ○ ***CMU, 2015 fall, E. Xing, Z. Bar-Joseph, HW3, pr. 1.2.1***

Ca să puteţi rezolva acest exerciţiu, trebuie să fiţi deja familiari cu modelul de reţea neuronală convolutivă LeNet, a cărei reprezentare grafică o reproducem mai jos.[802]



Vă cerem să determinaţi numărul total de parametri şi numărul total de conexiuni din această reţea. Câţi parametri şi câte conexiuni sunt în fiecare dintre nivelurile convolutive şi de sub-selecţie, C1, S2, C3 şi S4? Câţi parametri sunt în în fiecare dintre nivelurile total-conectate (engl., fully-connected layers), C5, F6 şi OUTPUT?

***Precizări*:**

i. **Mărimea *filtrului* pentru fiecare dintre nivelurile convolutive şi nivelurile de sub-selecţie (engl., sub-sampling layers):**

– **C1:** 5×5**, adică, fiecare unitate neuronală din C1 are un *spaţiu de recepţie* (engl., receptive field) de** 5×5 **în nivelul precedent lui;**

– **S2:** 2×2**;**

– **C3:** 5×5**;**

– **S4:** 2×2**.**

[802] Vă recomandăm să citiţi secţiunile A şi B din articolul *Gradient-based learning applied to document recognition*, de Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Proceedings of the IEEE, November 1998, 86 (11), p. 2278-2324. De asemenea, puteţi vedea următorul exerciţiu de tip implementare: CMU, 2016 spring, W. Cohen, N. Balcan, HW7 (sau CMU, 2016 fall, N. Balcan, M. Gormley, HW6).

ii. **Spre deosebire de nivelul C1, care este total conectat cu nivelul precedent (INPUT), nivelul C3 nu este total conectat cu nivelul S2. Tabelul de mai jos arată care dintre hărţile de trăsături (sau, *planele*) de pe nivelul S2 sunt conectate cu (toate!) planele de pe nivelul C3.**

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X		X	X
1	X	X				X	X	X			X	X	X	X		X
2	X	X	X				X	X	X			X		X	X	X
3		X	X	X			X	X	X	X			X		X	X
4			X	X	X			X	X	X	X		X	X		X
5				X	X	X			X	X	X	X		X	X	X

Răspuns:

Vom prezenta o sinteză a calculelor, precum şi răspunsurile finale, în tabelul următor:

	nr. parametri	**nr. conexiuni**
ConvLayers:		
C1	$(5*5+1)*6 = 156$	$28*28*26*6 = 122304$
S2	$2*6 = 12$	$14*14*6*5 = 5880$
C3	$(5*5*3+1)*6+$ $(5*5*4+1)*9+$ $(5*5*6+1) = 1516$	$1516*10*10 = 151600$
S4	$2*16 = 32$	$5*5*16*16 = 2000$
FCLayers:		
C5	$(5*5*16+1)*120 = 48120$	48120
F6	$(120+1)*84 = 10164$	10164
OUTPUT	$(84+1)*10 = 850$	850
Total	60850	340918

6.2 Reţele neuronale artificiale — Probleme propuse

6.2.1 Chestiuni introductive

28. **(Perceptronul-prag: funcţia calculată; zone de decizie)**

• ***CMU, 1997 fall, T. Mitchell, A. Moore, midterm exam, pr. 3.1***

Considerăm un perceptron-prag cu intrările x_1 şi x_2 şi având ponderile $w_0 = 0, w_1 = 1$ şi $w_2 = -3$. Desenaţi graniţa de decizie a acestui perceptron şi indicaţi zonele din planul bidimensional în care perceptronul produce rezultatele $+1$ şi -1.

29. **(Perceptronul-prag: exemplificare pentru disjuncţii / conjuncţii generalizate)**

prelucrare de Liviu Ciortuz, după

• ***CMU, 2010 fall, Ziv Bar-Joseph, HW3, pr. 1.1***

Imaginează-ţi că eşti responsabil cu design-ul unor preceptroni de tip prag şi că trebuie să implementezi câteva funcţii logice.

a. Concepe un perceptron-prag care implementează operatorul logic AND. Se vor considera două intrări, x_1 şi $x_2 \in \{-1, +1\}$, şi ieşirea $y \in \{-1, +1\}$.

Scrie mai întâi *tabela de adevăr* a operatorului AND.

În planul bidimensional al intrărilor x_1 şi x_2, desenează *separatorul* (adică graniţa de decizie) a perceptronului. Determină *ecuaţia* $f(x_1, x_2) = 0$ corespunzătoare acestei separator. Verifică semenele funcţiei f pe *zonele de decizie* determinate de separator.

Desenează perceptronul-prag AND specificând ponderile w_1, w_2 şi w_0.

Extinde rezultatul precedent la

- **conjuncţii de n variabile (A_1 AND A_2 AND ... AND A_n);**
- **conjuncţii de n literali (l_1 AND l_2 AND ... AND l_n) unde l_i este fie A_i fie $\neg A_i$.**

b. Procedează similar pentru funcţia logică OR.

30. **(Perceptroni-prag şi reţele de astfel de perceptroni: exemplificare)**

• ***TU Dresden, 2006 summer, S. Hölldobler, A. Grossmann, HW3, pr. 2***

Fie setul de instanţe de antrenament din $\mathbb{R}^2$ prezentate în tabelul de mai jos (partea dreaptă).

a. Indicaţi fie un perceptron cu funcţie de activare de tip prag fie o reţea formată din astfel de perceptroni — alegeţi varianta cea mai simplă! — care să fie consistent(ă) cu aceste exemple. (Nu trebuie să apelaţi la niciun algoritm de învăţare automată!)

Exemplu	X	Y	*Clasa*
1	1	6	$-$
2	1	2	$+$
3	2	4	$+$
4	5	2	$-$
5	6	5	$-$
6	7	8	$-$

b. Îndepliniţi aceleaşi cerinţe ca mai sus după ce în prealabil aţi adăugat la tabelul dat şi exemplul din tabelul alăturat.

Exemplu	X	Y	*Clasa*
7	7	1	$+$

31. **(O proprietate: extensia funcţiei XOR pe $\mathbb{R}^2$ (vedeţi pr. 1.d) nu poate fi calculată de reţele de perceptroni-prag care au un singur nivel ascuns)**

■ □ • ○ ***prelucrare de Liviu Ciortuz, după CMU, 2010 fall, Aarti Singh, HW5, pr. 4.1.2***

În rezolvarea problemei 1.d s-a arătat că funcţia $f : \mathbb{R}^2 \to \mathbb{R}$ definită prin $f(x_1, x_2) = 1$ pentru $x_1, x_2 \geq 0$ sau $x_1, x_2 < 0$, şi -1 în rest este calculabilă de către o reţea neuronală cu *două niveluri ascunse*, folosind doar unităţi de tip prag.

Demonstraţi că nu există nicio reţea cu *un singur nivel ascuns*, constituită doar din astfel de unităţi, care să calculeze funcţia f. Veţi presupune că numărul de unităţi de pe nivelul ascuns poate fi oricât de mare, însă finit.

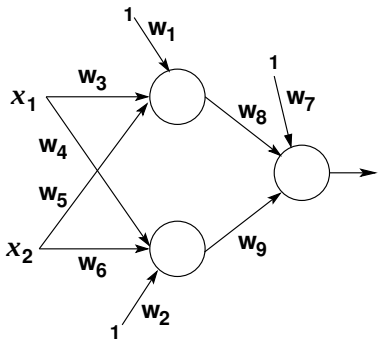
***Indicaţie*: Întrucât reţeua are un singur nivel ascuns, fiecare unitate de pe acest nivel poate fi pusă în corespondenţă cu o dreaptă din planul bidimensional, iar outputul calculat de către respectiva unitate este $+1$ pentru punctele (x_1, x_2) situate de o parte a dreptei şi -1 pentru punctele de cealaltă parte. Considerăm o vecinătate [de formă circulară] a originii reperului de coordonate, astfel încât dacă dreapta (adică, separatorul, sau graniţa de separare) asociată unei unităţi ascunse (oricare dintre acestea!) intersectează această vecinătate atunci ea trece (în mod necesar) prin origine. Este oare posibil ca o reţea neuronală [alcătuită doar din unităţi-prag şi] având un singur nivel ascuns să reprezinte funcţia f în această vecinătate?**

32. **(Reţele neuronale; o proprietate de bază: unităţile liniare de pe nivelurile ascunse pot fi „absorbite“ pe nivelul următor)**

• ○ ***CMU, ? spring, ML course 10-701, midterm, pr. 4***

Considerăm reţeaua neuronală pentru clasificare binară din figura de mai jos (partea dreaptă).

Pentru unităţile ascunse folosim o funcţie liniară de activare $h(z) = cz$, iar pentru unitatea de pe nivelul de ieşire funcţia de activare sigmoidală $\sigma(z) = \frac{1}{1+e^{-z}}$.



a. Desenaţi o reţea neuronală care este echivalentă cu reţeaua dată, dar care nu are niciun nivel ascuns. Scrieţi ponderile $\tilde{w}$ ale acestei noi reţele în funcţie de c şi de w_i.

b. Este adevărat că orice reţea neuronală de tip feed-forward, cu mai multe niveluri şi având toţi neuronii situaţi pe nivelurile ascunse de tip unităţi liniare poate fi reprezentată ca o reţea neuronală fără niciun nivel ascuns? Explicaţi succint răspunsul dat.

33. **(Alegerea ponderilor pentru o reţea cu structură / compoziţie specificată, în aşa fel încât să codifice funcţia XOR)**

● ○ ⋆ ***CMU, 2011 spring, Tom Mitchell, HW5, pr. 3.3***

Fie o reţea neuronală, constând dintr-un nivel ascuns format din două unităţi sigmoidale, şi un nivel de ieşire pe care se află o singură unitate, de tip prag. La niciuna dintre aceste trei unităţi nu se foloseşte termenul liber $x_0 = 1$. Găsiţi valori pentru ponderile din reţea astfel încât ea să codifice conceptul logic XOR.

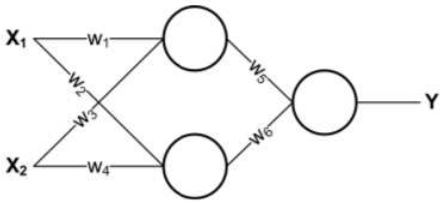
***Indicaţie*: Întrucât unităţile de pe nivelul ascuns sunt de tip sigmoidal, se va considera că intrările x_1 şi x_2 iau valorile 0 şi / sau 1.**

***Observaţie*: În raport cu problema 2.b, aici la toţi neuronii s-a eliminat termenul liber $x_0 = 1$ (ca şi la problema 4), iar tipul funcţiei de activare, şi anume prag, a fost înlocuit cu cel sigmoidal. Din cele trei probleme se observă că astfel de modificări pot afecta capacitatea de reprezentare a unei reţele neuronale (chiar dacă topologia de ansamblu este aceeaşi în toate cele trei cazuri).**

34. **(Exemplificarea legăturii dintre reţele neuronale, regresia liniară, regresia logistică şi boosting)**

□ ● ○ ***CMU, 2015 fall, E. Xing, Z. Bar-Joseph, HW3, pr. 1.1***

În figura alăturată este reprezentată o reţea neuronală cu două straturi, care învaţă o funcţie $f : \mathbb{R}^2 \to \mathbb{R}$. Ponderile $w = (w_1, \ldots, w_6)$ pot fi numere reale oarecare. Vom nota cu $X = (X_1, X_2) \in \mathbb{R}^2$ inputul reţelei, iar cu $Y \in \mathbb{R}$ outputul ei.



În ce priveşte funcţia de activare implementată de fiecare dintre unităţile neuronale care compun această reţea, vom presupune că sunt posibile două variante:

- S: **funcţia compusă „semn-sigmoidală“ (engl., signed sigmoid),**

$$S(a) \stackrel{def.}{=} sign\,(\sigma(a) - 0.5) = sign\left(\frac{1}{1+\exp(-a)} - 0.5\right);$$

- L: **funcţia liniară** $L(a) \stackrel{def.}{=} ca$**, unde** c **este o constantă reală, fixată.**

În ambele cazuri, $a \stackrel{not.}{=} \sum_i w_i X_i$**.**

a. Asignaţi fiecărei unităţi neuronale din figură câte o funcţie de activare (S **sau** L**) la în aşa fel încât această reţea să simuleze o regresie liniară:** $Y = \beta_1 X_1 + \beta_2 X_2$**.**

b. Asignaţi fiecărei unităţi neuronale din figură câte o funcţie de activare (S **sau** L**) la în aşa fel încât această reţea să simuleze o regresie logistică binară:** $Y = \arg\max_y P(Y = y|X)$**, unde** $P(Y = 1|X) = \frac{\exp(\beta_1 X_1 + \beta_2 X_2)}{1+\exp(\beta_1 X_1 + \beta_2 X_2)}$**, iar** $P(Y = -1|X) = \frac{1}{1+\exp(\beta_1 X_1 + \beta_2 X_2)}$**. Exprimaţi** β_1 **şi** β_2 **în funcţie de** $w_1, \ldots, w_6$**.**

c. Asignaţi fiecărei unităţi neuronale din figură câte o funcţie de activare (S **sau** L**) la în aşa fel încât această reţea să simuleze un clasificator de tip boosting care combină doi clasificatori de tip regresie logistică** $f_1 : X \to Y_1$ **şi** $f_2 : X \to Y_2$**, în aşa fel încât să se obţină outputul final** $Y = sign\,(\alpha_1 Y_1 + \alpha_2 Y_2)$**. Folosiţi pentru** f_1 **şi** f_2 **aceleaşi „distribuţii“ ca la punctul** b**. Exprimaţi** α_1 **şi** α_2 **în funcţie de** $w_1, \ldots, w_6$**.**

35. **(Expresivitate: reţele neuronale vs. arbori de decizie)**

• · ***CMU, 2009 spring, Ziv Bar-Joseph, HW2, pr. 3***

La acest exerciţiu veţi compara puterea de reprezentare a arborilor de decizie pe de o parte şi a reţelelor neuronale pe de altă parte, privitor la capacitatea de a codifica două funcţii, şi anume *funcţia majoritate* (engl., majority function) şi *funcţia paritate* (engl., parity function). Ambele funcţii primesc ca input n**-uple** $x_1, \ldots, x_n \in \{0,1\}^n$**, iar outputul lor este în mulţimea** $\{0,1\}$**. Funcţia *majoritate* returnează valoarea 1 dacă mai mult de jumătate dintre componentele din intrare sunt 1. De *exemplu*, tuplul** $(1,1,1,1,0)$ **conţine mai multe cifre de 1 decât de 0, deci va produce outputul 1. Funcţia *paritate* returnează 1 dacă şi numai dacă un număr par de componentele din intrare sunt 1. De *exemplu*, tuplul** $(1,0,1,0,1,0)$ **conţine 3 cifre de 1, deci produce outputul 0.**

a. Poate oare perceptronul[-prag] să fie folosit ca să codifice

- **funcţia majoritate?**
- **funcţia paritate?**

Pentru fiecare dintre aceste două funcţii, în cazul în care aţi răspuns afirmativ, prezentaţi perceptronul corespunzător, iar dacă aţi răspuns negativ, justificaţi.

b. Poate oare un arbore de decizie să codifice

- **funcţia majoritate?**
- **funcţia paritate?**

Pentru fiecare dintre aceste două funcţii, în cazul în care aţi răspuns afirmativ, prezentaţi arborele de decizie corespunzător, iar dacă aţi răspuns negativ, justificaţi.

c. Pentru fiecare dintre cele două funcţii de la punctul a la care aţi răspuns negativ, este oare posibil să obţinem codificarea dorită folosind o reţea neuronală cu două niveluri (un nivel ascuns şi un nivel de ieşire)? Dacă este aşa, desenaţi reţeaua care rezolvă problema. Altminteri, precizaţi câte niveluri ascunse sunt necesare pentru a putea rezolva problema.

6.2.2 Unităţi neuronale — algoritmi de antrenare

36. **(Antrenarea pereceptronului liniar: deducerea regulii de actualizare)**

Liviu Ciortuz, 2017, urmând
■ ***Tom Mitchell, Machine Learning, 1997, p. 89-93***

În acest exerciţiu vi se cere mai întâi să elaboraţi / redaţi partea de fundamentare teoretică pentru antrenarea perceptronului liniar, date fiind instanţele $(\bar{x}_1, t_1), \ldots, (\bar{x}_n, t_n)$, cu $\bar{x}_i \in \mathbb{R}^d$ şi $t_i \in \mathbb{R}$ pentru $i = 1, \ldots, n$.

Aşadar, veţi considera perceptronul liniar având intrările $x_0 = 1, x_1, \ldots, x_d \in \mathbb{R}$ şi ponderile $w_0, w_1, \ldots, w_d \in \mathbb{R}$.

a. În linie cu *metoda gradientului descendent*, deduceţi care este forma *regulii de actualizare* a vectorului de ponderi $\bar{w} \in \mathbb{R}^{d+1}$ (sau, echivalent, forma regulilor de actualizare a ponderilor w_i, cu $i = 0, \ldots, d$) pentru găsirea acelei valori care minimizează semisuma pătratelor erorilor comise de perceptron,[803] adică

$$E(\bar{w}) \stackrel{def.}{=} \frac{1}{2} \sum_{i=1}^{n} (t_i - o_i)^2.$$

În această ultimă expresie, o_i este outputul perceptronului liniar pentru intrarea $\bar{x}_i \stackrel{renot.}{=} (x_0 = 1, x_{i,1}, \ldots, x_{i,d})$, adică $o_i = w_0 + \sum_{j=1}^{d} w_j x_{i,j}$.

Vă reamintim că la aplicarea metodei gradientului descendent pentru găsirea unui punct de minim al unei funcţii derivabile $f : \mathbb{R}^{d+1} \to \mathbb{R}$, regula de actualizare a parametrilor $\bar{w}$ ai funcţiei f are forma $\bar{w} \leftarrow \bar{w} + \Delta\bar{w}$, cu $\Delta\bar{w} \stackrel{def.}{=} -\eta \nabla_{\bar{w}} f(\bar{w})$, unde $\eta > 0$ este un număr real mic, numit *rata de învăţare*, iar $\nabla_{\bar{w}} f(\bar{w})$ este vectorul *gradient*, adică $\left(\frac{\partial}{\partial w_0} f(\bar{w}), \frac{\partial}{\partial w_1} f(\bar{w}), \ldots, \frac{\partial}{\partial w_d} f(\bar{w}) \right)$.

[803] Conform problemei 13.b, am putea scrie — în ipoteza că datele au fost generate probabilist, cu o componentă „zgomot“ urmând o distribuţie gaussiană unidimensională de medie 0, aşa cum s-a precizat acolo —,

$$\bar{w}_{\text{ML}} = \arg\min_{\bar{w}} E(\bar{w}) = \arg\min_{\bar{w}} \frac{1}{2} \sum_{i=1}^{n} (t_i - \bar{w} \cdot \bar{x}_i)^2.$$

b. Dacă în locul perceptronului liniar vom considera perceptronul-prag, respectiv perceptronul sigmoidal, prin ce va diferi forma regulii de actualizare a ponderilor faţă de rezultatul de la punctul a? (Pentru perceptronul-prag, se va presupune că $t_i \in \{-1, +1\}$, în vreme ce pentru perceptronul sigmoidal vom considera $t_i \in \{0, 1\}$, pentru $i = 1, \ldots, n$.)

c. Elaboraţi algoritmul de antrenare a perceptronului liniar (şi apoi, dacă doriţi, şi a celui sigmoidal, dar veţi preciza doar diferenţele!). La *iniţializare*, ponderile w_i vor primi ca valori numere reale mici. În ce priveşte *condiţia de oprire* a algoritmului, se poate opta pentru una din următoarele variante (eventual combinate):

– toate instanţele de antrenament sunt corect clasificate (în ipoteza că datele $\bar{x}_i$, cu $i = 1, \ldots, n$, sunt liniar separabile);

– efectuarea unui număr prestabilit de iteraţii;

– verificarea condiţiei $|E(\bar{w}^{(t+1)}) - E(\bar{w}^{(t)})| < \varepsilon$, unde pragul numeric $\varepsilon > 0$ este stabilit iniţial, iar $E(\bar{w}^{(t)})$ este valoarea *funcţiei de eroare* E (i.e., ceea ce mai sus am numit semisuma pătratelor erorilor) pe setul de date de antrenament, la finalul iteraţiei t.

d. Ce cunoaşteţi (de la curs) despre convergenţa algoritmului de la punctul c?

37. **(Deducerea regulii de actualizare a ponderilor pentru un tip particular de perceptron)**

• ⋆ *CMU, 1997 fall, Tom Mitchell, midterm, pr. 3.d*

Se consideră o variantă particulară a perceptronului, [fără funcţie de activare,] la care ieşirea o depinde de intrările x_i astfel:

$$o = w_0 + w_1 x_1 + w_1 x_1^3 + w_2 x_2 + w_2 x_2^3 + \ldots + w_n x_n + w_n x_n^3$$

Derivaţi un algoritm de antrenare bazat pe metoda descreşterii gradientului, care să minimizeze suma pătratelor erorilor pentru acest tip de perceptron. Daţi răspunsul sub forma

$$w_i \leftarrow w_i + \ldots \quad \textbf{pentru } 1 \leq i \leq n.$$

38. **(Algoritmul *Perceptron* (al lui Rosenblatt): aplicare pe date din $\mathbb{R}^2$, folosind termen liber / bias)**

□ • *CMU, 2015 spring, Alex Smola, midterm, pr. 4*

Considerăm următoarele exemple de antrenament $(x, y) \in \mathbb{R}^2 \times \{\pm 1\}$, in ordinea indicată:

exemplul	1	2	3	4	5	6	7	8
instanţa (x_1, x_2)	$(10, 10)$	$(0, 0)$	$(8, 4)$	$(3, 3)$	$(4, 8)$	$(0.5, 0.5)$	$(4, 3)$	$(2, 5)$
eticheta y	$+1$	-1	$+1$	-1	$+1$	-1	$+1$	$+1$

Aplicaţi algoritmul *Perceptron* (al lui Rosenblatt) pe această succesiune de exemple. La iniţializare, vectorul de ponderi va primi valoarea $w = (1, 1)$, iar bias-ul (adică, termenul liber) va fi $b = 0$.[804]

39. **(Învăţare online cu perceptronul Rosenblatt)**

prelucrare de Liviu Ciortuz, după
• ∗ CMU, 2008 spring, T. Mitchell, W. Cohen, HW2, pr. 4
CMU, 2008 spring, T. Mitchell, W. Cohen, midterm exam, pr. 3.2

Problema 18 poate fi uşor reformulată în termenii învăţării automate online.[805] **În această situaţie, instanţele de antrenament nu sunt furnizate în avans perceptronului-prag. În schimb — după ce vectorul de ponderi w a fost iniţializat cu valori arbitrare, sau cu 0 ca în problema 18 —, la fiecare iteraţie (k) a algoritmului de antrenare, supervizorul prezintă sistemului de învăţare automată câte o instanţă de antrenament x_{t_k}, fără a divulga şi eticheta y_{t_k}. După ce a primit vectorul x_{t_k}, perceptronul calculează ieşirea o_{t_k} (în funcţie de $w^{(k)}$), iar supervizorul îi comunică eticheta y_{t_k}. Dacă $y_{t_k} \neq o_{t_k}$, perceptronul îşi actualizează / modifică vectorul de ponderi w.**

Dacă presupunem că instanţele de antrenament respectă condiţiile specificate în problema 18 şi ne punem întrebarea *care este numărul maxim de greşeli pe care le poate face perceptronul în regim online înainte de a „converge“*, răspunsul va fi: $\left\lfloor \left(R\frac{||w^*||}{\gamma} \right)^2 \right\rfloor$. Demonstraţia este practic aceeaşi cu cea de la problema 18.[806]

A. Considerăm un set de date de antrenament notat cu $D = \{(x_1, y_1), \ldots, (x_T, y_T)\}$, unde

i. **fiecare variabilă x_i reprezintă un document (d_i) în limba engleză, în care nu există mai mult de 100 de cuvinte (pentru această problemă, vom considera că nu există mai mult de $100\,000$ de cuvinte englezeşti);**

ii. **fiecărui document x_i îi este asociat un vector „rar“ (engl., sparse) $b_1, \ldots, b_{100\,000}$, unde b_w este 1 în cazul în care cuvântul w — mai exact, cuvântul cu indicele w din dicţionarul / lista cuvintelor din limba engleză — este prezent în documentul d_i, şi 0 în caz contrar;**

iii. **$y_i \in \{+1, -1\}$ este eticheta care desemnează clasa documentului d_i;**

iv. **există un vector w^* astfel încât $y_i w^* \cdot x_i > 1$ pentru orice document d_i.**

Indicaţi câte o *margine superioară* pentru

a. R – distanţa maximă dintre originea sistemului de coordonate din $\mathbb{R}^{100\,000}$ şi orice exemplu x_i;

[804] [LC:] Iniţializarea cu 0 a bias-ului nu înseamnă că el va păstra această valoare pe parcursul întregii execuţii a algoritmului. Atenţie: faptul că se foloseşte bias înseamnă că va trebui să consideraţi în mod implicit (sau, să adăugaţi explicit) componenta $x_0 = 1$ la fiecare instanţă de antrenament!

[805] Toate metodele de învăţare prezentate până acum sunt modele off-line, adică nu permit / prevăd nicio interacţiune a supervizorului cu algoritmul de antrenare.

[806] Acesta este motivul pentru care am specificat acolo un număr posibil infinit de instanţe de antrenament $x_1, \ldots, x_n, \ldots$.

b. m – numărul de erori făcute de perceptronul Rosenblatt înainte de a converge la o ipoteză corectă atunci când se face antrenarea pe datele din mulţimea D în regim de învăţare online.

***Observaţie*: Se va considera că $||w^*|| = 1$.**

B. Acum vă cerem să concepeţi un algoritm care produce o secvenţă de exemple care forţează perceptronul să producă un şir de lungime arbitrară (m) de greşeli, dacă oricare dintre cele două presupuneri de mai jos este eliminată, dar datele de antrenament rămân în continuare separabile:[807]

- **instanţele de antrenament sunt liniar-separabile (de către un vector w^*), cu marginea $\gamma > 0$;**[808]
- **toate instanţele de antrenament sunt situate într-o sferă cu raza R, cu centrul în originea sistemului de coordonate din $\mathbb{R}^d$.**[809]

40. **(De la convergenţa antrenării perceptronului Rosenblatt la convergenţa antrenării perceptronului-prag. Alte două proprietăţi legate de convergenţa antrenării perceptronului Rosenblatt)**

• ***CMU, 2013 spring, A. Smola, B. Poczos, HW2, pr. 2.a-c***
CMU, 2015 spring, A. Smola, HW6, pr. 2

Fie o secvenţă formată din n instanţe de antrenament $x_i \in \mathbb{R}^d$, împreună cu etichetele corespunzătoare $y_i \in \{-1, 1\}$. Vă readucem aminte[810] **că algoritmul lui Rosenblatt pentru antrenarea unui perceptron-prag — în cele ce urmează vom folosi pentru acest algoritm numele de *Perceptron* — procedează astfel:**

```
initialize w ← 0
for i = 1, ..., n do
    if y_i w · x_i ≤ 0 then
        w ← w + y_i x_i
    end if
end for
```

Pentru simplitate, am inclus termenul liber (sau, bias-ul, de la engl., "bias") w_0 în vectorul w, adică am renotat cu w vectorul $[w_0, w]$ şi, similar, cu x_i vectorul $[1, x_i]$. Aşadar, în această problemă nu este necesar să tratăm în mod explicit termenul liber.

Presupunem că $||x_i|| \leq R$ pentru orice i. La problema 18 am demonstrat că atunci când există un vector de ponderi w^* astfel încât $||w^*|| = 1$ şi pentru orice i are loc inegalitatea

$$y_i \, w^* \cdot x_i \geq \gamma,$$

numărul de actualizări ale lui w este mărginit superior de

$$R^2/\gamma^2. \tag{366}$$

[807] Adică, există w^* astfel încât $y_i \, w^* \cdot x_i > 0$.
[808] Negarea acestei condiţii înseamnă că pentru orice $\gamma > 0$ există o valoare a lui i astfel încât $0 < y_i \, w^* \cdot x_i < \gamma$.
[809] Negarea acestei condiţii înseamnă că pentru orice $R > 0$ există o valoare a lui i astfel încât $||x_i|| > R$.
[810] Vedeţi problema 16.

a. Ştim că regula [mai] generală pentru actualizarea perceptronului-prag este $w \leftarrow w + 2\eta y_i x_i$, unde $\eta > 0$ este rata de învăţare.[811] **Aşadar, algoritmul *Perceptron* al lui Rosenblatt corespunde cazului special $\eta = 1/2$.**

Găsiţi o margine superioară (engl., upper bound) pentru numărul de actualizări ale perceptronului-prag, similar cu modul în care a fost obţinută marginea (366). Cum variază această margine în funcţie de rata η?

b. Din relaţia (366) ştim că pentru valori mici ale lui γ problema antrenării *Perceptronului* poate fi dificil de rezolvat. De fapt, complexitatea ei poate fi chiar exponenţială în raport cu numărul de instanţe de antrenament. Ca să ilustrăm acest fapt, vom folosi următorul *exemplu*:

$$y_i = (-1)^{i+1} \text{ şi } x_i = (\underbrace{(-1)^i, \ldots, (-1)^i, (-1)^{i+1}}_{i \text{ elemente}}, 0, \ldots, 0) \text{ pentru } i = 1, \ldots, m.$$

Demonstraţi că sunt necesare $O(2^m)$ actualizări pentru ca algoritmul *Perceptron* să găsească o soluţie optimă w^*, satisfăcând inegalitatea $y_i\, x_i \cdot w^* > 0$ pentru orice i, indiferent de modul în care sunt selectate instanţele la fiecare iteraţie.

c. Acum vom renunţa la presupunerea că instanţele (x_i, y_i) sunt liniar separabile.[812] **(Viaţa nu este aşa de simplă! Însă, din fericire, nu este nici din cale afară de complicată.) Fie u un vector oarecare din $\mathbb{R}^d$, cu $||u|| = 1$, şi $\gamma > 0$. Definim *deviaţia* lui x_i prin expresia $d_i = \max(0, \gamma - y_i\, u \cdot x_i)$, iar $\delta = \left(\sum_{i=1}^n d_i^2\right)^{-1/2}$. Arătaţi că numărul de actualizări făcute de către algoritmul *Perceptron* — într-n „spaţiu de trăsături“ ales în mod convenabil — este mărginit superior de $(R + \delta)^2/\gamma^2$.**[813]

Sugestie: Aţi putea încerca ca, pornind de la instanţele x_i să construiţi în mod convenabil instanţe separabile x_i' şi apoi să refolosiţi / adaptaţi demonstraţia de la punctul precedent. De exemplu, puteţi defini $x_i' = [x_i, 0, \ldots, 0, c, 0, \ldots, 0]$, vector în spaţiul $\mathbb{R}^{d+n}$, având pe poziţia $d + i$ constanta $c \in \mathbb{R}$. Cum se poate oare construi în mod corespunzător un vector de ponderi u' astfel încât u' să separe instanţele x_i'? Apoi — odată ce aţi obţinut o margine superioară similară cu (366) —, cum ar trebui să fie aleasă constanta c astfel încât să minimizăm această margine superioară? Este oare această margine valabilă / bună [şi] pentru instanţele originale x_i?[814]

[811] În enunţul original se dă pentru regula de actualizare forma $w \leftarrow w + \eta y_i x_i$. Noi însă am folosit definiţia clasică: $w \leftarrow w - \eta \Delta w$, cu $\Delta w = \dfrac{\partial E_i}{\partial w} = -(y_i - o_i)x_i$, unde o_i este outputul perceptronului pentru inputul x_i. De remarcat că $y_i \neq o_i$ implică $y_i = 1$ şi $o_i = -1$ sau $y_i = -1$ şi $o_i = 1$, deci $y_i - o_i = 2y_i$ în ambele situaţii. Prin urmare, $\Delta w = -2y_i x_i$, iar forma regulii de actualizare este, în această abordare, $w \leftarrow w + 2\eta y_i x_i$.

[812] Deci nu există w^* astfel încât $y_i w^* \cdot x_i > 0$ pentru $i = 1, \ldots, n$.

[813] *Observaţie*: *Ideea principală* din spatele demonstraţiei de la acest punct este maparea / „proiectarea“ instanţelor x_i într-un spaţiu de dimensiune mai mare, în care separarea liniară să fie posibilă Aceasta coincide cu ideea folosirii *funcţiilor-nucleu* (engl., kernel functions), care mapează / transformă x_i în $\phi(x_i)$. Pentru varianta kernel-izată [dual] a algoritmului *Perceptron*, vedeţi problema 19.

[814] Pentru o variantă de mapare mai simplă decât cea indicată aici, vedeţi problema 73 de la capitolul de *Fundamente*.

41. **(Proprietăţi ale unei variante a algoritmului *Perceptron* kernel-izat, în cazul când nucleul este de tip RBF)**

prelucrare de L. Ciortuz, după
□ • ○ ***MIT, 2009 fall, Tommi Jaakkola, midterm, pr. 2***

Algoritmul *Perceptron* — cu care am făcut cunoştinţă la problema 16 — constituie probabil una dintre cele mai simple modalităţi de a rezolva probleme de clasificare. La problema 19 am prezentat varianta kernel-izată a acestui algoritm şi am menţionat funcţia-nucleu de tip Gaussian (care se mai numeşte şi *funcţie cu bază radială*, RBF).[815] În pseudo-codul următor redăm algoritmul *Perceptron* kernel-izat într-o variantă uşor extinsă în raport cu pseudo-codul din problema 19, în sensul că aici se permite parcurgerea de mai multe ori a setului de date de antrenament.

Date de intrare: $x_1, \ldots x_n \in \mathbb{R}^d$ şi $y_1, \ldots y_n \in \{-1, +1\}$;
Iniţializare: $\alpha_1 = \ldots = \alpha_n = 0$;
Procedură:
  Parcurge ciclic $i = 1, \ldots, n$ până când nu se mai produce nicio greşeală
    dacă $y_i(\sum_{j=1}^{n} \alpha_j y_j K(x_j, x_i)) \leq 0$, atunci $\alpha_i \leftarrow \alpha_i + 1$;

În cele de mai jos, funcţia K va fi considerată funcţie-nucleu de tip RBF, deci $K(x, x') = \exp\left(-\frac{1}{2\sigma^2}\|x - x'\|^2\right)$ şi $\sigma > 0$.

a. Converge oare întotdeauna acest algoritm? (În acest context, *a coverge* înseamnă că la un moment dat algoritmul nu mai actualizează variabilele α_i.) Trebuie oare să adăugăm vreo condiţie suplimentară pentru a ne asigura că algoritmul nostru se opreşte?

b. Indicaţi — şi apoi justificaţi în mod riguros — care dintre afirmaţiile următoare sunt *adevărate*.

□ *i*. **Dacă algoritmul converge, atunci el găseşte o soluţie[816] pentru care putem calcula „marginea“ de separare.[817] Această „margine“ depinde de ordinea în care sunt parcurse exemplele de antrenament.**

□ *ii*. **Presupunem că toate instanţele de antrenament x_i, cu $i = 1, \ldots, n$, sunt distincte. În acest caz, dacă parametrul nucleului RBF (şi anume, σ) are o valoare suficient de mică, algoritmul converge în mod cert după ce va fi făcut maximum n clasificări eronate (engl., mistakes).**

□ *iii*. **Numărul de clasificări eronate făcute de acest algoritm (în cazul în care converge) depinde de valoarea parametrului σ.**

***Indicaţie*: Pentru a putea răspunde corect la întrebările de mai sus, vă recomandăm să faceţi referire la rezultatele teoretice demonstrate la problema 29 de la capitolul *Maşini cu vectori-support* din prezenta culegere,[818] precum şi la problema 18 din capitolul de faţă (*Reţele neuronale artificiale*).**

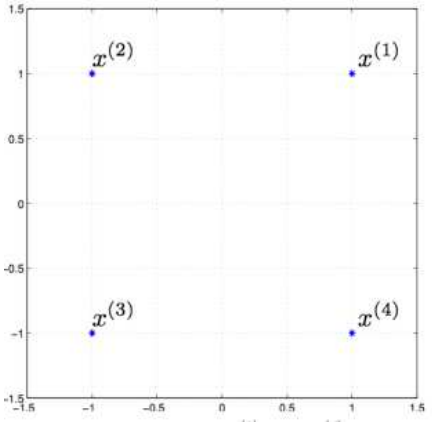
[815] Vedeţi problema 74 de la capitolul de *Fundamente*.
[816] Adică un *separator liniar* în *spaţiul de trăsături* determinat de „mapărea“ ϕ specifică nucleului RBF.
[817] Adică distanţa de la [hiperplanul] separator până le cele mai apropiate „imagini“ $\phi(x_i)$.
[818] Ar fi util să vedeţi şi problema 10 de la capitolul *Metode de regresie*.

42. **(Perceptronul Rosenblatt, cu termen liber (engl., offset): varianta simplă vs. varianta kernel-izată; aplicare pe un set simplu de date din $\mathbb{R}^2$)**

prelucrare de Liviu Ciortuz, după

□ • · ***MIT, 2014 spring, T. Jaakkola, R. Barzilay, final exam, pr. 1***

În acest exerciţiu vom considera patru instanţe de antrenament, $x^{(1)}, \ldots, x^{(4)} \in \mathbb{R}^2$, după cum se observă în figura alăturată. Aceste instanţe au respectiv etichetele $y^{(1)}, \ldots, y^{(4)} \in \{-1, +1\}$. Veţi explora diferite variante de a eticheta aceste instanţe şi [mai precis] efectul pe care îl au aceste etichetări asupra algoritmului *perceptron cu termen liber* (engl., offset).



Acest algoritm este caracterizat de următorul pseudo-cod:

Initialize: $w = 0$ (vector), $w_0 = 0$
  Cycle through $i = 1, \ldots, n$
    If $y^{(i)}(w \cdot x^{(i)} + w_0) \leq 0$ then
      update $w \leftarrow w + y^{(i)} x^{(i)}$ and $w_0 \leftarrow w_0 + y^{(i)}$

***Observaţie*:** Pseudo-codul acesta se obţine în mod natural din pseudo-codul perceptronului Rosenblatt (vedeţi problema 16) dacă se consideră instanţele de forma $x \overset{not.}{=} (1, x_1, \ldots, x_d)$, cu $d \in \mathbb{N}^*$.

Pentru o *instanţă de test* x, algoritmul va prezice eticheta

$$sign\,(w \cdot x + w_0).$$

a. Vă cerem să elaboraţi varianta kernel-izată a acestui tip de *perceptron cu termen liber*. Pentru aceasta, ca de obicei, se va considera mai întâi că fiecare instanţă x este „mapată“ / transformată sub forma unui vector de trăsături $\phi(x)$, iar apoi algoritmul va fi reformulat astfel încât să folosească instanţele x doar prin intermediul funcţiei-nucleu $K(x, x') \overset{def.}{=} \phi(x) \cdot \phi(x')$.

Odată antrenat, algoritmul va prezice pentru instanţa x eticheta y, care este exact valoarea expresiei

$$sign\left(\left(\sum_{i=1}^{n} \alpha_i y^{(i)} \phi(x^{(i)})\right) \cdot \phi(x) + w_0\right) = sign\left(\sum_{i=1}^{n} \alpha_i y^{(i)} K(x^{(i)}, x) + w_0\right),$$

unde $n = 4$ este numărul de exemple de antrenament, iar existenţa şi rolul coeficienţilor $\alpha_i \in \mathbb{N}$ sunt justificate în mod similar cu cazul Perceptronului Rosenblatt kernel-izat (vedeţi problema 19).[819]

[819] Veţi observa că atât la problema 19 cât şi la problema de faţă se parcurge o singură dată setul de date de antrenament. Însă la ambele probleme este posibilă extinderea algoritmului în aşa fel încât să se poată face mai multe astfel de parcurgeri. De asemenea, în problema de faţă rolul coeficientului α_i va fi disociat de eticheta $y^{(i)}$, el indicând pur şi simplu de câte ori instanţa $x^{(i)}$ a cauzat erori / greşeli în timpul antrenamentului, provocând deci „actualizarea“ perceptronului.

Pentru o justificare cu caracter mai general a existenţei şi rolului coeficienţilor $\alpha_i \in \mathbb{R}$ în ce priveşte exprimarea [formei] soluţiilor unor probleme de clasificare sau de regresie care fac optimizarea unor funcţii de cost / pierdere, vedeţi *Teorema de reprezentare* care a fost demonstrată la problema 88 de la capitolul *Fundamente*.

Vă cerem să completaţi în pseudo-codul de mai jos detaliile algoritmului *perceptron cu termen liber* kernel-izat.

Initialize: ...

Cycle through $i = 1, \ldots, n$

If ... then

update ...

b. Presupunem că rulăm algoritmul acesta kernel-izat folosind nucleul RBF

$$K(x, x') = \exp\big(-\frac{1}{2}\|x - x'\|^2\big).$$

Considerăm acum exemplele de antrenament din figura de mai sus. Are oare importanţă cum sunt etichetate aceste patru puncte în ce priveşte convergenţa algoritmului? Justificaţi pe scurt răspunsul dumneavoastră.

***Indicaţie*: Vedeţi proprietăţile prezentate la problema 18 din prezentul capitol şi problema 29.A de la capitolul *Maşini cu vectori-suport*.**

c. Găsiţi o etichetare pentru cele patru instanţe de antrenament

$$y^{(1)} = \ldots,\ y^{(2)} = \ldots,\ y^{(3)} = \ldots,\ y^{(4)} = \ldots$$

care să satisfacă următoarele două criterii:

i. **Varianta kernel-izată a algoritmului perceptron cu offset, folosind nucleul RBF menţionat mai sus, converge după ce face o singură actualizare a coeficienţilor (α, α_0).**
ii. **Varianta simplă (deci nekernel-izată) a algoritmului perceptron cu offset converge, însă necesită mai multe actualizări ale ponderilor (w, w_0).**

Faceţi toate calculele necesare.

43. **(Varianta perceptronului Rosenblatt pentru clasificare ternară: kernel-izare)**

prelucrare de Liviu Ciortuz, după
□ • · ***MIT, 2008 fall, Tommi Jaakkola, midterm, pr. 3.4***

În acest exerciţiu vom considera varianta algoritmului Perceptron (al lui Rosenblatt) pentru rezolvarea unei probleme de clasificare ternară (adică, cu 3 clase). Setul de date de antrenament constă din n exemple de forma (x_i, y_i), pentru $i = 1, \ldots, n$, cu $x_i \in \mathbb{R}^d$ şi $y_i \in \{1, 2, 3\}$. Pseudo-codul acestui algoritm este următorul:[820]

[820] Veţi observa în acest pseudo-cod — analizaţi instrucţiunea if — că, spre deosebire de perceptronul kernelizat [dual] de la problema 19, unde se lucrează cu un singur vector de ponderi (w), aici se lucrează cu trei vectori de ponderi (w_1, w_2 şi w_3), câte un vector pentru fiecare clasă. (S-a procedat similar şi în cazul regresiei liniare multivaluate şi în cel al regresiei logistice n-are (regresia *softmax*). Vedeţi problemele 31 şi respectiv 19 de la capitolul *Metode de regresie*.)

De asemenea, veţi observa că spre deosebire de perceptronul kernelizat [dual], unde la fiecare greşeală de clasificare produsă de către perceptron se actualizează vectorul de ponderi w, aici la fiecare greşeală de clasificare se actualizează doi vectori: w_{y_i} — unde y_i este eticheta asignată instanţei curente, x_i — şi w_z —unde z este clasa indicată (în mode eronat) de către perceptronul ternar. Tot aşa se procedează şi în cazul perceptronului n-ar. Vedeţi http://proceedings.mlr.press/v97/beygelzimer19a/beygelzimer19a-supp.pdf. (LC: Mulţumesc studentului Răzvan Ciocoiu pentru aceste două precizări.)

Iniţializare:
for $y \in \{1, 2, 3\}$
$w_y = 0 \in \mathbb{R}^d$
Algoritm:
Repeat until convergence:
$z = \arg\max_{y \in \{1,2,3\}} w_y \cdot x_i$
for $i = 1, \ldots, n$
if $z \neq y_i$
$w_{y_i} = w_{y_i} + x_i$
$w_z = w_z - x_i$
Clasificarea unei instanţe de test x:
$f(x) = \arg\max_{y \in \{1,2,3\}} w_y \cdot x$

Vă cerem să elaboraţi o versiune kernel-izată a acestui algoritm de tip perceptron. Veţi considera că nucleul pe care îl veţi folosi este $K(x, z)$. Completaţi pseudo-codul următor în aşa fel încât să rezulte algoritmul kernel-izat.

Iniţializare:
for $y \in \{1, 2, 3\}$, for $i = 1, \ldots, n$
$\alpha_{i,y} = 0$
Algoritm:
Repeat until convergence:

Clasificarea unei instanţe de test x:
$\arg\max_{y \in \{1,2,3\}} \sum_{i=1}^{n} \alpha_{i,y} K(x, x_i)$

44. **(Antrenarea perceptronului: elaborare cod C / pseudo-cod)**

Liviu Ciortuz, 2012

Elaboraţi fie în pseudo-cod fie în limbajul de programare C funcţii pentru

a. construirea (adică reprezentarea ca structură de date) a unui perceptron. Veţi lucra cu n intrări (numere reale), iar funcţia de activare va fi de *tip* prag, liniar sau sigmoidal. Într-o versiune ulterioară veţi putea adăuga tipurile: generalizat-sigmoidal (a se vedea problema 20) sau tangentă hiperbolică (*tanh*, ca în problema 47).

b. calculul outputului unui perceptron (construit la punctul precedent), pentru un input dat;

c. algoritmul de antrenare (adică algoritmul de actualizare a ponderilor corespunzătoare intrărilor perceptronului), conform tipului de funcţie de activare indicat mai sus. Veţi elabora atât varianta "batch" cât şi varianta stochastică / incrementală (a se vedea cartea lui Tom Mitchell, *Machine Learning*, pag. 92-94).

Indicaţie: Acest algoritm se va opri atunci când una din următoarele condiţii sunt satisfăcute:

– toate instanţele de antrenament sunt corect clasificate, *sau*

– s-au executat deja un număr *maxim* de iteraţii dat ca parametru. Alternativ, se poate testa dacă funcţia de optimizat (de exemplu, semisuma pătratelor erorilor, sau suma costurilor / pierderilor de tip log-sigmoidal (vedeţi problema 14)) a scăzut sub un anumit prag ε (dat ca parametru) la ultima iteraţie executată, în comparaţie cu iteraţia precedentă.

Vă sugerăm să testaţi implementarea dumneavoastră pe datele de la problemele 2.a, 29, 30.a şi 9 (pentru punctele *a* şi *b*), şi respectiv problema 11 (pentru punctul *c*). Ulterior, veţi putea implementa varianta în care funcţia de activare este de tip log-sigmoidal (ca la problema 14), sau de tip tangentă hiperbolică (ca la problema 47).

6.2.3 Reţele "feed-forward" — algoritmul de retropropagare

45. **(Reţele "feed-forward": algoritmul de retro-propagare, aplicare)**

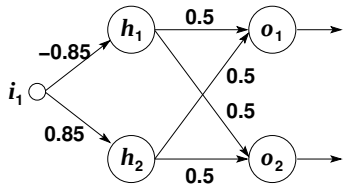
• ⋆ ***CMU, 2014 spring, Seyoung Kim, HW2, pr. 1.3***

Considerăm că toate unităţile reţelei neuronale din figura de mai jos sunt de tip sigmoidal.

Vă readucem aminte că acest fapt înseamnă că pentru inputul $x_1, \ldots, x_d$, o astfel de unitate neuronală calculează outputul

$$\frac{1}{1 + e^{-(w_0 + \sum_i w_i x_i)}},$$

unde w_i este ponderea corespunzătoare intrării x_i, pentru fiecare $i = 1, \ldots, d$.



În cele ce urmează vom considera că $w_0 = 0$ pentru fiecare unitate din această reţea.

a. Presupunem că inputul reţelei este $i_1 = 1$.

Cât va fi valoarea outputului calculat de către nodul o_1?
Dar valoarea outputului calculat de către nodul o_2?

b. De data aceasta presupunem că outputurile corecte pentru inputul $i_1 = 1$ sunt $t_1 = t_2 = 1$. Folosind formulele date pentru cantităţile δ_i în algoritmul de retro-propagare — a se vedea pr. 20 şi / sau cartea *Machine Learning* de Tom Mitchell, pag. 98 —, calculaţi:

– valoarea lui δ_{o_1};
– valoarea lui δ_{o_2}.

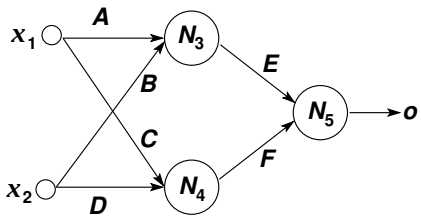
c. În final, folosind aceste valori pentru δ şi considerând că rata de învăţare este $\eta = 0.1$, calculaţi:

– valoarea lui δ_{h_1};
– noua valoare a ponderii de pe conexiunea care leagă unitatea h_1 de unitatea o_1;
– noua valoare a ponderii de pe conexiunea care leagă unitatea h_1 de unitatea o_2;
– noua valoare a ponderii de pe conexiunea care leagă intrarea i_1 de unitatea h_1.

46. **(Algoritmul de retro-propagare: Adevărat sau Fals?)**

• ⋆ ***CMU, 1994 fall, A. Blum, T. Mitchell, HW4, pr. 1***

Reţeaua neuronală din figura alăturată este formată din unităţi de acelaşi tip, de exemplu (pentru fixarea ideilor) sigmoidal. Presupunem că executăm algoritmul de retro-propagare pe această reţea, folosind un set oarecare de date de antrenament şi dând iniţial valori egale tuturor ponderilor din reţea.



a. Precizaţi care dintre afirmaţiile de mai jos vor fi întotdeauna adevarate:

i. **Ponderile A şi B nu vor diferi niciodată.**
ii. **Ponderile A şi C nu vor diferi niciodată.**
iii. **Ponderile E şi F nu vor diferi niciodată.**

b. Justificaţi de ce NU este bine să se iniţializeze toate ponderile dintr-o reţea neuronală cu o aceeaşi valoare atunci când se rulează algoritmul de retro-propagare.

47. **(Reţele neuronale — algoritmul de retro-propagare pentru reţele de unităţi neuronale având funcţia de activare $tanh$)**

⋆ ***prelucrare de Liviu Ciortuz, după T. Mitchell, "Machine Learning", 1997, pr. 4.8 şi CMU, 2011 spring, Roni Rosenfeld, HW4, pr. 2.a***

Rescrieţi algoritmul de retro-propagare pentru reţele neuronale aciclice în care unităţile neuronale folosesc funcţia de activare tangentă hiperbolică ($tanh$) în locul funcţiei sigmoidale.

***Recomandare*: Pentru a reduce din generalitatea problemei, vă veţi limita la a elabora varianta stochastică / incrementală a acestui algoritm pentru reţele de tip feed-forward cu două niveluri.**

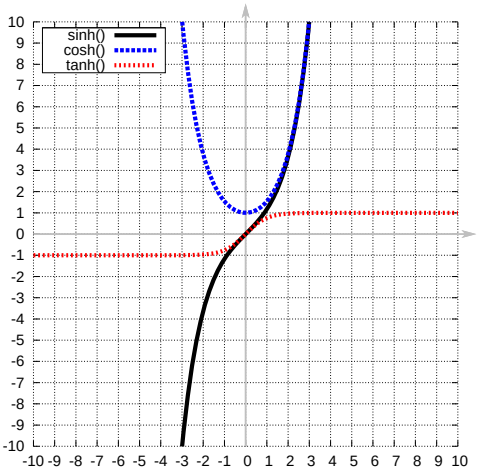
Vă reamintim că

$$tanh(y) = \frac{e^y - e^{-y}}{e^y + e^{-y}} = \frac{e^{2y} - 1}{e^{2y} + 1} = \frac{\boldsymbol{sinh}(y)}{\boldsymbol{cosh}(y)},$$

iar

$$\boldsymbol{sinh}(y) = \frac{e^y - e^{-y}}{2} \text{ şi } \boldsymbol{cosh}(y) = \frac{e^y + e^{-y}}{2}$$

Din definiţia de mai sus rezultă imediat că funcţia $tanh$ este derivabilă şi deci continuă pe întreg domeniul de definiţie (axa reală), iar din imaginea alăturată se observă că funcţia $tanh$ este o formă „netezită" (engl., smoothed) a funcţiei treaptă $sign$.



Adaptare după
http://en.wikipedia.org/wiki/File:Sinh_cosh_tanh.svg

Aşadar, vom considera că outputul unităţilor neuronale din reţea este de forma $o(\bar{x}) = tanh(\bar{w} \cdot \bar{x})$**.**

a. Demonstraţi că $tanh'(y) = 1 - tanh^2(y)$ **pentru orice** $y \in \mathbb{R}$**.**

b. Arătaţi că pentru regula de actualizare a ponderilor unităţilor neuronale de pe nivelul de ieşire, $w_{kj} \leftarrow w_{kj} + \Delta w_{kj}$**, avem**

$$\Delta w_{kj} \stackrel{def.}{=} -\eta \frac{\partial E}{\partial w_{kj}} = \eta(t_k - o_k)(1 - tanh^2(net_k))y_j$$

unde notaţiile sunt similare cu cele din cartea *Machine Learning* de Tom Mitchell.[821]

c. Similar, pentru ponderile unităţilor neuronale de pe nivelul ascuns:

$$\Delta w_{ji} \stackrel{def.}{=} -\eta \frac{\partial E}{\partial w_{ji}} = \eta(1 - tanh^2(net_j))x_i \left(\sum_{k \in Downstream(j)} \delta_k w_{kj} \right),$$

unde

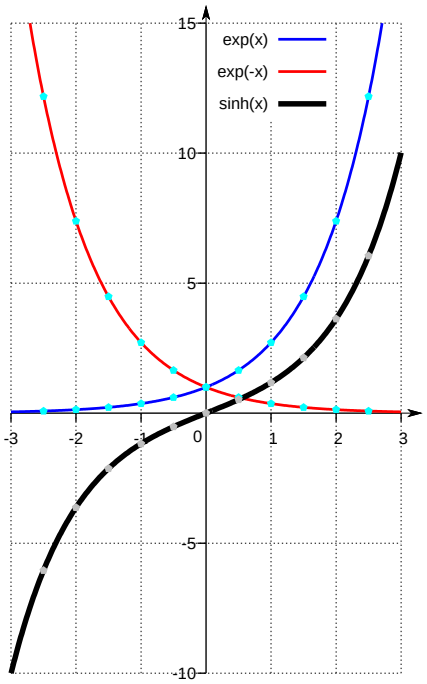
$$\delta_k = -\frac{\partial E_d(\bar{w})}{\partial net_k} \textbf{ iar } E_d(\bar{w}) = \sum_{k \in outputs} \frac{1}{2}(t_{k,d} - o_{k,d})^2,$$

d **fiind o instanţă de antrenament oarecare sau (echivalent) indicele asociat ei.**

Adăugăm câteva ***observaţii*** cu caracter instructiv / informativ în legătură cu funcţia *tanh*:

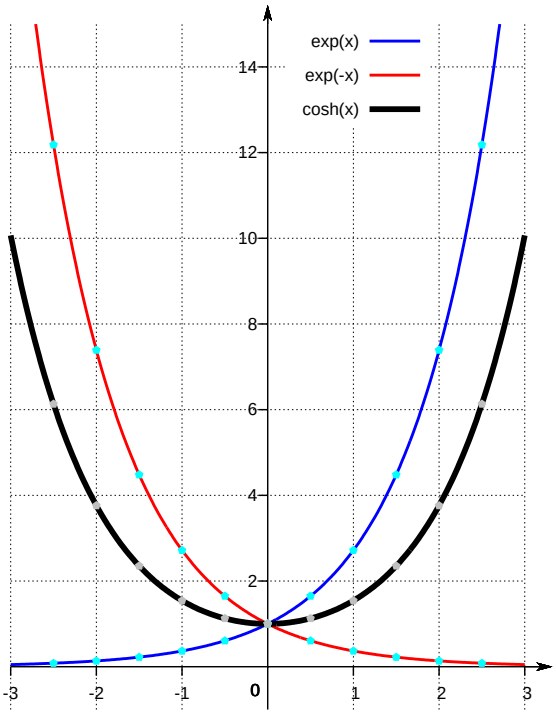
(1) Pentru a vedea relaţia dintre funcţia *sinh* (şi respectiv *cosh*) pe de o parte şi funcţiile exponenţială (e^x) şi invers-exponenţială (e^{-x}) pe de altă parte, cititorul poate observa graficele de mai jos.

[821] Aceleaşi notaţii au fost folosite şi în problema 20.



Adaptat după: http://en.wikipedia.org/wiki/File:Hyperbolic_and_exponential;_sinh.svg

Observaţie: Funcţia $sinh$ este semidiferenţa funcţiilor e^x şi e^{-x}.



Adaptat după: http://en.wikipedia.org/wiki/File:Hyperbolic_and_exponential;_cosh.svg

Observaţie: Funcţia $cosh$ este media funcţiilor e^x şi e^{-x}.

(2) Legătura dintre funcţia sigmoidală $\sigma(x) = \dfrac{1}{1+e^{-x}}$ (care mai este numită şi *funcţia logistică*) şi funcţia *tanh* este dată de relaţia:

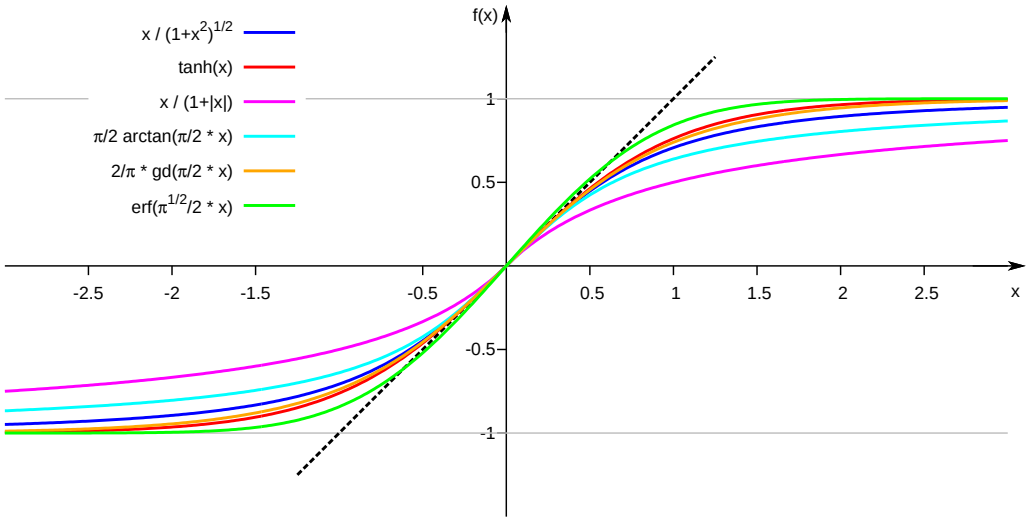
$$\sigma(x) = \frac{1}{2}\left(1 + tanh\,\frac{x}{2}\right)$$

(3) Spre deosebire de funcţia sigmoidală (care este o variantă „netezită“ a funcţiei treaptă 0/1), funcţia *tanh* este simetrică faţă de originea sistemului de coordonate. Din această cauză, în literatura de specialitate se apreciază că funcţia *tanh* favorizează o convergenţă mai rapidă a algoritmului de antrenare a reţelei neuronale.

(4) Există şi alte funcţii care „netezesc“ funcţiile-treaptă, de exemplu: $\dfrac{x}{1+\mid x \mid}$, $\dfrac{x}{\sqrt{1+x^2}}$, funcţia arctg (inversa funcţiei tangentă trigonometrică), funcţia de eroare gaussiană (notată *erf*, prin abreviere de la engl. error function) şi funcţia Gudermannian (*gd*), unde

$$erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt, \text{ iar } gd(x) = \int_0^x \frac{dt}{cosh}(t) = 2\,arctg(e^x) - \frac{\pi}{2}$$

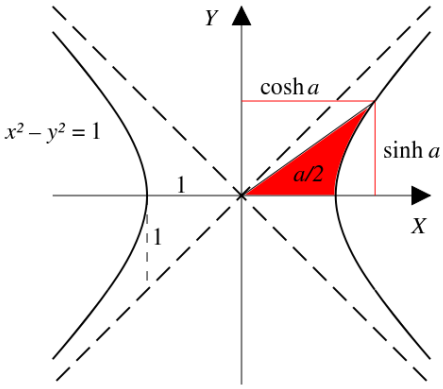
Pentru aceste funcţii se pot defini versiuni „normalizate“, în aşa fel încât panta tangentei lor pentru $x = 0$ să fie 1. A se vedea graficul următor.



Adaptat după: http://en.wikipedia.org/wiki/File:Gjl-t(x).svg

(5) Legătura dintre funcţiile trigonometrice hiperbolice şi funcţiile trigonometrice clasice („circulare“) poate fi ilustrată astfel:

Pentru funcţiile trigonometrice circulare avem proprietatea fundamentală: $sin^2\alpha + cos^2\alpha = 1$. În consecinţă, dacă notăm $x = cos\alpha$ şi $y = sin\alpha$, rezultă că punctul (x, y) este situat pe cercul de ecuaţie $x^2 + y^2 = 1$ (având centrul O şi raza 1). Pentru funcţiile trigonometrice hiperbolice avem o altă proprietate, care se verifică imediat: $cosh^2 a - sinh^2 a = 1$. Aşadar, dacă notăm $x' = cosh\, a$ şi $y' = sinh\, a$, rezultă că punctul (x', y') este situat pe hiperbola de ecuaţie $(x')^2 - (y')^2 = 1$, ale cărei asimptote la $+\infty$ şi $-\infty$ sunt prima şi cea de-a doua bisectoare a sistemului de coordonate în plan. (De remarcat proprietatea: pentru $cosh\, a$ şi $sinh\, a$, aria zonei haşurate din imagine este $a/2$.)



Sursa: http://en.wikipedia.org/wiki/File: Hyperbolic_functions-2.svg

48. **(Aplicarea algoritmului de retro-propagare pe o reţea feed-forward formată din unităţi sigmoidale dispuse pe 2 niveluri, utilizând şi componenta „moment“)**

adaptare făcută de către Liviu Ciortuz, după
• ⋆ T. Mitchell, "Machine Learning", 1997, pr. 4.7

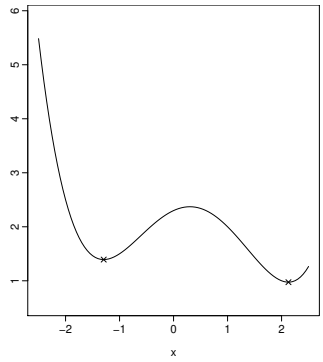
Se consideră o reţea neuronală de tip feed-forward cu două niveluri, având două intrări a şi b, o unitate ascunsă c şi o unitate de ieşire d. Ambele unităţi sunt de tip sigmoidal. Această reţea are 5 ponderi: w_{ca}, w_{cb}, w_{c0}, w_{dc}, w_{d0}, unde w_{c0} reprezintă ponderea asociată termenului liber ($x_0 = 1$) pentru unitatea c şi similar pentru unitatea d.

Iniţializaţi ponderile cu valorile $(0.1, 0.1, 0.1, 0.1, 0.1)$, apoi calculaţi valorile lor după fiecare dintre primele două iteraţii ale algoritmului de retro-propagare în varianta "batch". Considerăm rata de învăţare $\eta = 0.3$, momentul $\alpha = 0.9$ şi datele de antrenament din tabelul alăturat.[822]

a	b	t
1	0	1
0	1	0

***Comentariu*:**

Vă readucem aminte următoarele:



1. La aplicarea metodei gradientului descendent, pentru evitarea „căderii" în unele minime locale, regula de actualizare a parametrilor w_{ji}, şi anume $w_{ji}^{(n)} \leftarrow w_{ji}^{(n-1)} + \Delta w_{ji}^{(n)}$, se poate extinde cu încă un termen, $\alpha \Delta w_{ji}^{(n-1)}$, numit *termenul moment*, datorită analogiei cu mişcarea unui punct material din fizică. (Constanta $\alpha > 0$ desemnează cât anume dorim să păstrăm din „inerţia" de mişcare a particulei considerate.) Veţi lua $\Delta w_{(ji)}^{(0)} = 0$ pentru orice j şi orice i.

2. Folosind notaţiile de la pag. 101 din cartea *Machine Learning* de Tom Mitchell, avem $\Delta w_{ji}^{(n)} = -\eta \dfrac{\partial E}{\partial w_{ji}}(w^{(n-1)})$. Făcând abstracţie de n, această cantitate Δw_{ji} este $\eta \left(\sum_l \delta_{j,l} x_{ji,l}\right)$, cu $\delta_{j,l} = o_{j,l}(1 - o_{j,l})(t_{j,l} - o_{j,l})$ atunci când j este indicele unei unităţi de pe nivelul de ieşire şi, respectiv, $\delta_{j,l} = o_{j,l}(1 - o_{j,l}) \left(\sum_{k \in Downstream(j)} \delta_{k,l} w_{kj}\right)$ atunci când j este indicele unei unităţi de pe un nivel ascuns. Indicele l parcurge setul de instanţe de antrenament.

49. **(Regularizare: prevenirea overfitting-ului pentru o reţea feed-forward cu unităţi de tip (generalizat-)sigmoidale; extensie pentru problemele 20 şi 47)**

prelucrare de Liviu Ciortuz, după Tom Mitchell, "Machine Learning", 1997, pr. 4.10 CMU, 2011 spring, Roni Rosenfeld, HW4, pr. 2.b

Considerăm o reţea neuronală de tip feed-forward care foloseşte o funcţie de eroare adecvată pentru prevenirea overfitting-ului (a se vedea problema 22):

$$E_D(w) = \sum_{d \in D} \sum_{k \in outputs} \frac{1}{2}(t_{kd} - o_{kd})^2 + \gamma \sum_{i,j} w_{ji}^2,$$

unde D este mulţimea instanţelor de antrenament.

Deduceţi regulile de actualizare a ponderilor de pe conexiunile din reţea, corespunzător acestei definiţii a erorii presupunând că reţeaua foloseşte

[822] Dacă limităm exerciţiul acesta la aplicarea manuală a primei iteraţii, atunci nu avem de-a face cu componenta *moment*, însă el rămâne un exerciţiu instructiv prin faptul că se solicită aplicarea algoritmului de retro-propagare în regim "batch", adică non-incremental (ceea ce nu mai apare, deocamdată, în niciun alt exerciţiu din acest capitol). Aceasta implică folosirea formulelor de actualizare a ponderilor în forma din *Comentariul* al doilea de mai jos.

a. doar unităţi (generalizat-)sigmoidale (ca la problema 20);

b. doar unităţi cu funcţia de activare de tip $tanh$ (ca la problema 47).

50. **(Algoritmul de retro-propagare pentru reţele feed-forward cu funcţia obiectiv de tip cross-entropie)**

• · ***CMU, 2008 fall, E. Xing, HW2, pr. 2.2***
CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, HW3, pr. 2.2

Fie o reţea neuronală de tip feed-forward având d intrări, H unităţi pe nivelul ascuns şi K unităţi pe nivelul de ieşire. Toate unităţile sunt de tip sigmoidal. Vom considera N exemple de antrenament, iar funcţia de eroare cu care vom lucra este de tip cross-entropie:

$$E(w) = -\sum_{n=1}^{N}\sum_{k=1}^{K}[t_{nk}\ln y_{nk} + (1-t_{nk})\ln(1-y_{nk})]$$

unde

t_{nk} **este target-ul pentru al n-lea exemplu şi pentru ieşirea k a reţelei,**

y_{nk} **este răspunsul reţelei pentru ieşirea k atunci când cel de-al n-lea exemplu este furnizat reţelei.**

În această problemă veţi calcula regula de actualizare pentru algoritmul de tip retro-propagare (folosind metoda gradientului descendent) în varianta stochastică. După cum ştiţi, regula de actualizare a ponderii de pe conexiunea care leagă intrarea i de unitatea j este:

$$w_{ji}^{(t)} = w_{ji}^{(t-1)} - \eta\frac{\partial E_n}{\partial w_{ji}^{(t-1)}},$$

unde E_n este funcţia de eroare pentru exemplul n.

Vom nota cu net_j intrarea în componenta de activare a unităţii j:

$$net_j = \sum_i w_{ji}z_i,$$

unde z_i este outputul unităţii ascunse i (dacă unitatea j este pe nivelul de ieşire) sau intrarea i (dacă unitatea j este pe nivelul ascuns).

***Observaţii*: (i) Întrucât veţi calcula regula de actualizare pentru un singur exemplu, puteţi elimina indicele n din notaţiile de mai sus. (ii) În mod *implicit* vom considera că nu se folosesc termeni liberi (engl., bias) pentru neuronii reţelei.**

a. Câţi parametri (i.e., câte ponderi) are reţeaua aceasta? (Veţi presupune, la acest punct, că se folosesc termeni liberi.)

b. Calculaţi derivata funcţiei de activare, $\sigma'(x)$, în funcţie de însuşi $\sigma(x)$.

c. Definim $\delta_j = \dfrac{\partial E_n}{\partial net_j}$. Arătaţi că $\dfrac{\partial E_n}{\partial w_{ji}} = \delta_j z_i$.[823]

d. Arătaţi că pentru orice unitate de ieşire k, avem $\delta_k = y_k - t_k$.

e. Considerăm intrarea $(x_1, x_2, \ldots, x_d)$. Deduceţi regula de actualizare pentru $w_{kj}^{(t)}$, valoarea ponderii de pe conexiunea dintre unitatea ascunsă j şi unitatea de ieşire k la iteraţia t, în funcţie de x_i şi de valoarea ponderilor din reţea la iteraţia $t-1$.

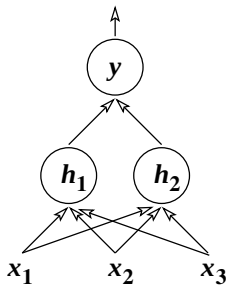
f. Pentru o unitate ascunsă j oarecare, avem $\delta_j = \dfrac{\partial E_n}{\partial net_j} = \sum_k \dfrac{\partial E_n}{\partial net_k} \dfrac{\partial net_k}{\partial net_j}$. Arătaţi că $\delta_j = \sigma(net_j)(1 - \sigma(net_j)) \sum_k \delta_k w_{kj}$.

g. Considerăm intrarea $(x_1, x_2, \ldots, x_d)$. Deduceţi regula de actualizare pentru $w_{ji}^{(t)}$, valoarea ponderii de pe conexiunea dintre intrarea i şi unitatea ascunsă j la iteraţia t, în funcţie de x_i şi de valoarea ponderilor din reţea la iteraţia $t-1$.

51. **(Aplicarea metodei gradientului descendent pentru o reţea formată din unităţi cu funcţia de activare ReL)**

□ • ○ ***CMU, 2015 spring, Alex Smola, final, pr. 2.2***

În figura de mai jos vi se dă structura unei reţele neuronale simple, având un singur nivel ascuns.



Inputul este 3-dimensional: $x = (x_1, x_2, x_3)$. Nivelul ascuns este constituit din două unităţi, deci vom putea scrie outputul acestui nivel sub forma $h = (h_1, h_2)$. Nivelul de ieşire este constituit dintr-o singură unitate, y. Pentru convenienţă, nu vom folosi termeni liber (adică, bias-uri). Pentru toate unităţile neuronale din această reţea vom folosi funcţia de activare liniar-rectificată (engl., linear rectified activation function, ReL), definită prin expresia $f(z) = \max(0, z)$ pentru orice $z \in \mathbb{R}$.

Considerăm funcţia de eroare / cost / pierdere (engl., loss function) $l(y, t) = \dfrac{1}{2}(y - t)^2$, unde t este valoarea target pentru inputul x, iar y este valoarea produsă de reţea pentru acelaşi input. De asemenea, vom nota cu W şi V matricele de ponderi care corespund conexiunilor dintre nivelul de intrare şi nivelul ascuns, şi respectiv dintre nivelul ascuns şi nivelul de ieşire. Aceste matrice sunt iniţializate astfel:

$$W = \begin{bmatrix} 1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} \text{ şi } V = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

[823] Egalitatea $\dfrac{\partial E_n}{\partial w_{ji}} = \delta_j z_i$ are loc nu doar pentru funcţia E din enunţ (o cross-entropie), ci şi, de exemplu, pentru semisuma pătratelor erorilor (vedeţi pr. 20). Mai general, ea are loc pentru acele funcţii de cost E în care ponderile w sunt folosite doar în interiorul termenilor de forma $net_j \overset{def.}{=} \sum_i w_{ii} x_i$. În rest — vedeţi, de exemplu, cazul folosirii termenilor pentru regularizare, pr. 22 —, această manieră de lucru (centrată pe calculul cantităţilor δ) nu mai este adecvată / utilizabilă.

a. Folosind simbolii f, W and V, scrieţi [în manieră matriceală] expresia funcţiei $x \to y$ calculate de către reţeaua neuronală dată. (Nu este nevoie să folosiţi aici valorile care au fost specificate mai sus pentru matricele W şi V.)

b. Presupunem că inputul reţelei este $x = (1, 2, 1)$, iar valoarea target este $t = 1$. Calculaţi valoarea *numerică* pentru outputul y, elaborând în mod clar toţi paşii intermediari. Puteţi refolosi rezultatele de la punctul precedent. Folosirea formei matriceale este recomandată, dar nu este obligatorie.

c. Calculaţi vectorul gradient pentru funcţia de eroare l, în raport cu ponderile reţelei. În mod specific, calculaţi mai întâi [în manieră *simbolică*] expresiile gradientului

– relativ la matricea de ponderi V, adică $\frac{\partial l}{\partial V}$;

– relativ la matricea de ponderi W, adică $\frac{\partial l}{\partial W}$.

Calculaţi apoi valoarea *numerică* a gradientului funcţiei l, pentru valorile specificate mai sus pentru W, V, x şi y.

52. **(Adevărat / Fals?)**

◦ ***CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, midterm, pr. 1.5***
CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, midterm, pr. 1.a

a. Graniţa de decizie învăţată de către o reţea neuronală este întotdeauna neliniară.

b. Este posibil ca la antrenarea perceptronului liniar — aşadar, urmărind să minimizăm suma pătratelor erorilor, folosind metoda gradientului descendent — să obţinem puncte [multiple] de optim local.

c. Indiferent de mărimea reţelei neuronale, algoritmul de retro-propagare poate întotdeauna să găsească valorile optime globale pentru ponderile reţelei.

53. **(Antrenarea reţelelor feed-forward în diverse variante: elaborare cod C / pseudo-cod)**

Liviu Ciortuz, 2012

Elaboraţi fie în pseudo-cod fie în limbajul de programare C funcţii pentru

a. configurarea (adică reprezentarea ca structură de date) a unei reţele neuronale artificiale de tip feed-forward;

Indicaţie: Într-o primă versiune, veţi lucra cu n intrări (numere reale), un singur nivel ascuns şi o singură unitate pe nivelul de ieşire, iar toate unităţile neuronale vor fi de acelaşi tip (de exemplu, sigmoidal). Ulterior, veţi elabora o variantă extinsă a acestei funcţii, în aşa fel încât să se permită lucrul cu K niveluri ascunse, iar pentru fiecare unitate din reţea să se poată alege unul din tipurile de perceptroni indicate la problema 44.

b. calculul outputului unei reţele de tipul indicat la punctul precedent, pentru un input dat.

Indicaţii:
1. La punctele *a* şi *b* veţi putea folosi funcţiile corespunzătoare de la implementarea perceptronului (vedeţi problema 44).
2. Vă sugerăm să testaţi implementarea dumneavoastră (la acest stadiu) pe datele de la problemele 1.ab, 2.b, 3, 6, 8, [7,] 30.b şi 33.

c. algoritmul de antrenare (adică algoritmul de retro-propagare) pentru reţele neuronale, de asemenea de tipul indicat mai sus. Veţi elabora atât varianta "batch" cât şi varianta stochastică / incrementală (a se vedea cartea lui Tom Mitchell, *Machine Learning*, pag. 92-94).

Indicaţie: Vă sugerăm să elaboraţi mai întâi o versiune de bază a acestei funcţii, pe care să o testaţi pe datele de la problema rezolvată 21 (verificând la final rezultatele) şi de la problemele propuse 45 şi 46. Ulterior puteţi extinde această versiune (fie în manieră parametrizată, fie orientat pe obiecte) în aşa fel încât să poată opera cu diverse variante ale *funcţiei obiectiv* de optimizat: fie semisuma pătratelor erorilor, fie varianta cu *regularizare*, pentru prevenirea overfitting-ului (ca în problemele 22 şi 49), fie o funcţie de tip cross-entropie (ca în problema 50). În sfârşit, încă o variantă posibilă este cea în care regulile de actualizare a ponderilor conţin şi o componentă de tip „moment", pentru evitarea unor optime locale ale funcţiei de eroare (a se vedea problema 48).

54. **(Reţele neuronale convolutive: determinarea mărimii hărţii de trăsături de pe un anumit nivel)**

□ • ○ ***CMU, 2015 fall, E. Xing, Z. Bar-Joseph, midterm, pr. 4.1***

Considerăm un nivel convolutiv C urmat de un nivel de tip selecţie a maximului (engl., max pooling layer) P. Inputul nivelului C are 50 de *canale* / planuri, fiecare dintre ele având dimensiunea 12×12. Nivelul C are 20 de *filtre*, fiecare dintre acestea fiind de mărime 4×4. Adausul / *bordarea* convolutivă (engl., the convolution padding) este de mărime 1, iar lungimea *pasului*[824] (engl., the stride) este 2.

Nivelul P efectuează o selecţie a maximului din fiecare *hartă de trăsături* (engl., feature maps) din outputul nivelului C, cu *ferestre* (sau, câmpuri locale receptive; engl., local receptive fields) de dimensiune 3×3 şi lungimea *pasului* este 1.

Cât este mărimea fiecărei *hărţi de trăsături* (engl., feature map) din outputul nivelului P?

[824] Pasul se referă la deplasarea *ferestrei* care corespunde filtrului de pe nivelul respectiv.



© *M. Romanică*

7 Clusterizare

Sumar

7.0 Noţiuni de bază

– **instanţă neetichetată vs. instanţă etichetată (exemplu de antrenament);**

– **învăţare nesupervizată (de exemplu: clusterizare, estimarea parametrilor distribuţiilor probabiliste şi reducerea dimensionalităţii) vs. învăţare supervizată (de exemplu: clasificare şi regresie);**

– **[funcţie / măsură de] *distanţă* definită pe $\mathbb{R}^d \times \mathbb{R}^d$: ex. 2 de la capitolul *Învăţare bazată pe memorare*;**

– **cluster / grup / grupare / bin (engl.) vs. clasă;**

– **tipuri de clusterizare: ierarhică vs. neierarhică;**

– **tipuri de ierarhii: ierarhii (arbori de clusterizare, dendrograme) obişnuite vs. ierarhii plate (engl., flat hierarchies);**
exemple: ex. 1.a şi respectiv ex. 1.b, ex. 6.a;

– **tipuri de apartenenţă a unei instanţe la un cluster: "hard" vs. "soft" (ultima numai pt. clusterizare neierarhică).**

7.1. Clusterizare ierarhică

7.1.1. Noţiuni specifice

– **[funcţie de] *similaritate* între clustere, definită pe baza [extinderii] noţiunii de distanţă la $\mathcal{P}(X) \times \mathcal{P}(X)$, unde $X \subset \mathbb{R}^d$ este mulţimea de instanţe, iar $\mathcal{P}(X)$ este mulţimea părţilor lui X;**

***tipuri* de [funcţii de] similaritate:**

"single-linkage":[825] $d(A,B) = \min\{d(x,y) | x \in A, y \in B\}$

"complete-linkage":[826] $d(A,B) = \max\{d(x,y) | x \in A, y \in B\}$

"average-linkage": $d(A,B) = \frac{1}{|A|\ |B|} \sum_{x \in A, y \in B} d(x,y)$

***metrica lui Ward* (bazată pe centroizi): ex. 33, 34, 35.**

În general, putem considera $sim(A,B) = 1/(1 + d(A,B))$ **sau chiar** $sim(A,B) = 1/d(A,B)$ **dacă lucrăm doar cu clustere non-singleton;**

proprietate / restricţie: $sim(A \cup B, C) \leq \min\{sim(A,C), sim(B,C)\}$ **pentru orice clustere** A, B **selectate de algoritmul de clusterizare ierarhică la execuţia unei iteraţii oarecare [din bucla principală a algoritmului] şi orice alt cluster** C**;**

– **[funcţie de] *coeziune* internă a unui cluster (sau: între elementele / instanţele dintr-un cluster);**
exemplu (pentru clustere non-singleton): inversul mediei aritmetice a distanţelor dintre perechi de puncte din clusterul respectiv, adică
$\text{coh}(A) = \left(\frac{1}{C^2_{|A|}} \sum_{x,y \in A} d(x,y) \right)^{-1} = \frac{C^2_{|A|}}{\sum_{x,y \in A} d(x,y)}$.

[825] Sau: nearest-neighbour.
[826] Sau: furthest-neighbour.

7.1.2. Algoritmi de clusterizare ierarhică

– **tipuri de algoritmi de clusterizare ierarhică:**
bottom-up (*clusterizare aglomerativă*) vs. top-down (*clusterizare divizivă*);

– **pseudo-coduri (cf. Manning & Schütze, *Foundations of Statistical Natural Language Processing*, 2002, pag. 502):**

bottom-up:

Given: **a set** $X = \{x_1, \ldots, x_n\}$ **of objects**
a function sim: $\mathcal{P}(X) \times \mathcal{P}(X) \to R$
for $i = 1, n$ **do**
$c_i = \{x_i\}$ **end**
$C = \{c_1, \ldots, c_n\}$
$j = n + 1$
while $\mid C \mid > 1$
$(c_{n_1}, c_{n_2}) = \text{argmax}_{(c_u, c_v) \in C \times C}$ **sim**(c_u, c_v)
$c_j = c_{n_1} \cup c_{n_2}$
$C = C \backslash \{c_{n_1}, c_{n_2}\} \cup \{c_j\}$
$j = j + 1$

top-down:

Given: **a set** $X = \{x_1, \ldots, x_n\}$ **of objects**
a function coh: $\mathcal{P}(X) \to R$
a function split: $\mathcal{P}(X) \to \mathcal{P}(X) \times \mathcal{P}(X)$
$C = \{X\} (= \{c_1\})$
$j = 1$
while $\exists c_i \in C$ **such that** $\mid c_i \mid > 1$
$c_u = \arg\min_{c_v \in C}$ **coh**(c_v)
$c_{j+1} \cup c_{j+2} =$ **split** (c_u)
$C = C \backslash \{c_u\} \cup \{c_{j+1}, c_{j+2}\}$
$j = j + 2$

– **analiza (ca algoritmi *per se*): ambii algoritmi sunt iterativi şi "greedy";**

– **complexitate:** $\mathcal{O}(n^2 \log n)$ **(cf. CMU, 2021 fall, Aarti Singh, Lecture Notes 21);**

– **exemple de aplicare: ex. 1-5, ex. 27-33 pentru bottom-up, respectiv ex. 6 pentru top-down;**

– **o legătură cu alţi algoritmi de învăţare automată: ex. 37;**

– **implementări: ex. 35, 38, 39.**

7.1.3 Proprietăţi

– **(P0) rezultatul aplicării algoritmilor de clusterizare ierarhică depinde de ce măsură de distanţă şi ce funcţie de similaritate sunt folosite;**

– **(P1) rezultatul aplicării celor doi algoritmi de clusterizare ierarhică pe un set de date oarecare nu este în mod neapărat unic determinat: ex. 3.b;**

– **(P2) clusterizarea folosind similaritate de tip "single-linkage" are tendinţa să creeze clustere alungite; invers, folosind similaritate "complete-linkage" sau "average-linkage", se formează clustere de formă mai degrabă sferică: ex. 2, 5 şi 31;**

– **(P3) dacă atunci când folosim "single-linkage" şi "complete-linkage" se obţin dendrograme identice, *nu* rezultă în mod neapărat că folosind "average-linkage" vom obţine aceeaşi dendrogramă: ex. 3.b;**

– **(P4) numărul maxim de niveluri dintr-o dendrogramă (văzută ca arbore în sensul teoriei grafurilor) este $n-1$, unde n este numărul de instanţe de clusterizat: ex. 4.a; numărul minim de niveluri: $\lceil \log_2 n \rceil$; ex. 4.b;**

– **(P5) există o anumită corespondenţă între clusterizare ierarhică cu similaritate de tip**

  • **"single-linkage" şi aflarea *arborelui* [*de acoperire*] *de cost minim* dintr-un graf: ex. 6;**

  • **"complete-linkage" şi aflarea unei *clici* (subgraf maximal complet) dintr-un graf (vedeţi Manning & Schütze, *op. cit.*, pag. 506-507);**

– **(P6) algoritmul de clusterizare aglomerativă la al cărui pseudo-cod am făcut referire mai sus are complexitate $\mathcal{O}(n^3)$: ex. 27; atunci când se foloseşte single-linkage sau complete-linkage, există însă versiuni / algoritmi de complexitate $\mathcal{O}(n^2)$: SLINK (1973) şi respectiv CLINK (1976);**

– **la clusterizare ierarhică aglomerativă cu similaritate "average-linkage":**
(P7) dacă se foloseşte ca măsură de similaritate între 2 instanţe cosinusul unghiului dintre vectorii care reprezintă instanţele şi se „normalizează" aceşti vectori (i.e., se lucrează cu 2 vectori coliniari cu ei, dar de normă egală cu 1), atunci calculul coeziunii [interne a] unui cluster nou format, precum şi calculul „distanţei" dintre două clustere se pot face în timp constant: ex. 36.

7.2. Clusterizare partiţională

7.2.1 Noţiuni specifice

– **centroid (centru de greutate) al unui cluster: ex. 33,**
K-partiţie, K-configuraţie [iniţială] a centroizilor: ex. 11;

– **o funcţie de evaluare a „calităţii" clusterelor (sau: funcţie de „coeziune" / „distorsiune" / „eroare" totală):**
„suma celor mai mici pătrate": $J_K(C,\mu) = \sum \|x_i - \mu_{C(x_i)}\|^2$, unde C este K-partiţie, μ este K-configuraţie de centroizi, iar $\mu_{C(x_i)}$ este centroidul cel mai apropiat de x_i: ex. 12.

7.2.2 Algoritmul K-means

– **pseudo-cod (cf. Manning & Schütze, *op. cit.*, pag. 516):**

Given: a set $X = \{x_1, \ldots, x_n\} \subseteq \mathcal{R}^m$,
a distance measure d on $\mathcal{R}^m$,
a function for computing the mean $\mu : \mathcal{P}(\mathcal{R}^m) \to \mathcal{R}^m$
Select (arbitrarily) k initial centers $f_1, \ldots, f_k$ in $\mathcal{R}^m$;
while the *stopping criterion* is not satisfied
for all clusters c_j do $c_j = \{x_i \mid \forall f_l \; d(x_i, f_j) \leq d(x_i, f_l)\}$ end
for all means f_j do $f_j \leftarrow \mu(c_j)$ end

alternativ, vedeţi enunţul ex. 12 (sau, echivalent, folosind variabile-indicator: ex. 45);

- **complexitate: $\mathcal{O}(lKn)$, unde l este numărul de iteraţii executate (cf. CMU, 2021 fall, Aarti Singh, Lecture Notes 21);**
- **exemple de aplicare: ex. 7-11, ex. 17.a, ex.21.a, ex. 22.a, ex. 40, ex. 41.**
- **exemple de *euristici pentru iniţializarea centroizilor*:**
  iniţializare arbitrară / random în $\mathbb{R}^d$ sau în $X = \{x_1, x_2, \ldots, x_n\} \subseteq \mathbb{R}^d$ (setul de date de clusterizat);
  aplicarea, în prealabil, a unui algoritm de clusterizare ierarhică;
  folosind o anumită distribuţie probabilistă definită pe X: K-means++ (David Arthur, Sergei Vassilvitskii, 2007): ex. 50.
- **exemple de *criterii de oprire*:**
  după efectuarea unui număr maxim de iteraţii (fixat iniţial);
  când componenţa clusterelor nu se mai modifică de la o iteraţie la alta;
  când poziţiile centroizilor nu se mai modifică de la o iteraţie la alta;
  când descreşterea valorii criteriului J_K de la o iteraţie la alta nu mai este strictă sau nu mai este peste un anumit prag ε fixat în prealabil.
- **ca algoritm *per se*:**

  K-means este un algoritm de *căutare*:
  ***spaţiul de căutare* este mulţimea tuturor K-partiţiilor care se pot forma pe dataset-ul de intrare;**
  (P0) întrucât acest spaţiu de căutare (deşi este finit) este exponenţial (K^n), K-means explorează doar parţial spaţiul de căutare, procedând *iterativ*: el pleacă de la o „soluţie“ (K-partiţie) aleasă eventual în mod arbitrar / aleatoriu şi o „îmbunătăţeşte“ la fiecare iteraţie;

  (P1) soluţia găsită este dependentă de iniţializarea centroizilor: ex. 10;
  (P1′) mai mult, chiar la o aceeaşi iniţializare, rezultatele pot diferi(!) dacă avem instanţe multiple / redundante, situate la egală distanţă de 2 centroizi la o iteraţie oarecare: ex. 11.b (cazul 2), ex. 12.b;
  (P1″) rezultatele lui K-means sunt dependente [şi] de măsura de distanţă folosită: ex. 49;

  K-means poate fi văzut şi ca *algoritm de optimizare* — vedeţi criteriul J_K de mai sus;
  (P2) strategia de căutare / optimizare folosită de K-means este de tipul *descreştere pe coordonate* (engl., coordinate descent), i.e. descreştere iterativă, mergând alternativ pe fiecare din cele două coordonate ale criteriului $J_K(C^t, \mu^t)$: ex. 12.a;
  (P2′) algoritmul K-means nu garantează atingerea optimului global (i.e., minimul) pentru criteriul J_K:[827] ex. 12.b, ex. 47.b.
- **ca algoritm de *învăţare automată*:**
  [urmat de] „generalizare“: o instanţă nouă x se asociază clusterului având centroidul cel mai apropiat de x;
  (P3) „graniţele“ de separare dintre [perechile de] clustere produse de K-means sunt liniare atunci când se foloseşte distanţa euclidiană: ex. 11.b (cazul 1);
  (P3′) este însă posibil să se obţină separatori neliniari dacă se foloseşte o versiune „kernelizată“ a algoritmului K-means: ex. 51;
  (P4) rezultatele lui K-means pot fi influenţate de prezenţa outlier-elor: ex. 10; în astfel de cazuri este de preferat să se folosească distanţa Manhattan în locul distanţei euclidiene: ex. 49.B.

[827] Acest minim există; vedeţi proprietatea (P6) de mai jos.

– **o euristică pentru alegerea unei valori convenabile / „naturale” pentru K (pentru un dataset dat) — criteriul “elbow”: ex. 44 (şi CMU, 2012f, E. Xing, A. Singh, HW3, ex. 1.de);**

– **adaptarea algoritmului K-means pentru cazul în care în locul distanţei euclidiene se foloseşte distanţa Manhattan: ex. 49.B;**

– **recapitulare: ex. 54;**

– **implementare: ex. 55.**

7.2.3 Alte proprietăţi ale algoritmului K-means

– **în legătură cu criteriul definit mai sus, $J_K : \mathcal{P}_K \times (\mathbb{R}^d)^K \to [0, +\infty)$, unde $\mathcal{P}_K$ este mulţimea tuturor K-partiţiilor peste mulţimea de instanţe, $X = \{x_1, x_2, \ldots, x_n\} \subseteq \mathbb{R}^d$:**
  ∘ (P5) $J_K(C^{t-1}, \mu^{t-1}) \geq J_K(C^t, \mu^t)$ la orice iteraţie ($t > 0$) a algoritmului K-means: ex. 12.a (exemplificare: ex. 7.a, ex. 42);
  ∘ (P5′) în consecinţă, dacă se impune restricţia ca la fiecare iteraţie inegalitatea de mai sus să fie satisfăcută în varianta strictă ($J_K(C^{t-1}, \mu^{t-1}) > J_K(C^t, \mu^t)$), atunci algoritmul K-means termină într-un număr finit de paşi: ex. 12.b.

– **(P6) pentru $K > 0$ fixat, $|\mathcal{P}_K| = K^n$, deci este finit, şi există $\underline{J_K} \overset{not.}{=} \min_C J_K(C, \mu_C)$: ex. 12.b;**
  acest minim ($\underline{J_K}$) se poate obţine prin explorarea exhaustivă a spaţiului $\mathcal{P}_K$, însă consumul de timp este prohibitiv în practică;
  ∘ (P6′) valoarea 0 pentru $\underline{J}$ este atinsă, şi anume atunci când $K = n$, C este K-partiţia de clustere singleton $C_i = \{x_i\}$, iar $\mu_i = x_i$, pentru $i = 1, \ldots, n$ (ex. 44);
  ∘ (P6″) $\underline{J}_1 \geq \underline{J}_2 \geq \ldots \geq \underline{J}_{n-1} \geq \underline{J}_n = 0$: ex. 13, ex. 42.c.

– **(P7) în vreme ce maximizează *coeziunea intra-clustere*, i.e. minimizează criteriul J, algoritmul K-means maximizează (în mod aproximativ!) o sumă ponderată a pătratelor distanţelor dintre centroizii clusterelor şi centrul de greutate al întregului set de instanţe:**

$$\sum_{k=1}^{K} \left(\frac{\sum_{i=1}^{n} \gamma_{ik}}{n} \right) \|\mu_k - \bar{x}\|^2,$$

  unde $\bar{x}$ este media instanţelor $x_1, x_2, \ldots, x_n$ (ex. 45, ex. 46).

– **(P8) dacă $d = 1$, deci $x_1, x_2, \ldots, x_n \in \mathbb{R}$,**
  ∘ orice K-partiţie $(C_1, \ldots, C_K)$ pentru care se atinge $\underline{J_K}$ este de forma unei colecţii de „intervale“: $C_1 = \{x_1, \ldots, x_{i_1}\}$, $C_2 = \{x_{i_1+1}, \ldots, x_{i_2}\}$, …, $C_K = \{x_{i_{K-1}+1}$, …, $x_n\}$, cu $i_1 < i_2 < \ldots < i_{K-1} < i_K = n$;
  ∘ există un algoritm [de programare dinamică] de complexitate $\mathcal{O}(Kn^2)$ care calculează $\underline{J_K}$: ex. 47.

7.3. Clusterizare prin modelare probabilistă

7.3.1 Noţiuni preliminare

– **variabile aleatoare (discrete, resp. continue);**
  media, varianţa şi covarianţa variabilelor aleatoare;

- **vector de variabile aleatoare; matrice de covarianţă pentru un astfel de vector; proprietăţi: matricea de covarianţă trebuie să fie în mod necesar simetrică şi pozitiv definită: ex. 20 de la capitolul de *Fundamente*;**
- **distribuţie (funcţie de densitate) de probabilitate (p.d.f.);**
  parametri ai unei distribuţii de probabilitate;
  distribuţia gaussiană, cazurile uni- şi multidimensional: vedeţi secţiunea corespunzătoare din *Sumarul* capitolului de *Fundamente*;
- **mixtură de distribuţii probabiliste:**
  văzută ca o formă particulară de *combinaţie liniară* de distribuţii de probabilitate $\pi_1\Psi_1 + \pi_2\Psi_2 + \ldots + \pi_k\Psi_k$ (cu $\pi_i \geq 0$ şi $\sum_{i=1}^{k} \pi_i = 1$),
  definită [şi mai specific] scriind distribuţia $P(X)$ ca o sumă ponderată de probabilităţi condiţionate: $\sum_z P(X|Z)P(Z)$, unde X sunt variabilele „observabile“, iar variabila Z (eventual multiplă) poate fi „neobservabilă“ / „latentă“ / „ascunsă“;
  exemple:
  o mixtură de distribuţii categoriale, respectiv o mixtură de distribuţii Bernoulli: ex. 29 şi ex. 114 şi ex. 113 de la capitolul de *Fundamente*;
  o mixtură de distribuţii gaussiene multidimensionale: ex. 119 de la capitolul de *Fundamente*;
  o mixtură de distribuţii oarecare: ex. 120 de la capitolul de *Fundamente*;
- **funcţie de *verosimiliate* a unui set de date (D), în raport cu o distribuţie probabilistă dată: $L(\theta) = P(D|\theta)$, unde prin θ se notează parametrii respectivei distribuţii. Exemplificare: ex. 43.abd, ex. 42 de la capitolul de *Fundamente*;**
- **MLE (Maximum Likelihood Estimation): estimarea [valorilor] parametrilor unei distribuţii probabiliste în sensul maximizării verosimilităţii datelor disponibile. Exemplificare: capitolul de *Fundamente*, ex. 43-54, ex. 125-136. Aplicare în cazul distribuţiei gaussiene unidimensionale: ex. 15.ab de la capitolul *Clasificare bayesiană*;**
- **regula de decizie pentru algoritmul Bayes [Naiv] Gaussian:**[828]
  pentru cazul unidimensional: ex. 15, ex. 45, ex. 16, şi ex. 46 de la capitolul *Clasificare bayesiană*;
- **Observaţie: Algoritmul EM este [sau, mai degrabă, poate fi folosit ca] o metodă de estimare a parametrilor unei mixturi de distribuţii probabiliste. *Alternativ*, pentru acelaşi obiectiv pot fi folosite alte metode, de exemplu *metoda gradientului ascendent*: ex. 70.**

7.3.2 Algoritmul EM pentru clusterizare prin estimarea parametrilor unui model de mixturi de distribuţii gaussiene (EM/GMM)

- **pseudo-cod:**[829]
  cazul unidimensional, varianta când doar parametrul μ este lăsat liber: ex. 15 (cf. *Machine Learning*, Tom Mitchell, 1997, pag. 193); aplicare: ex. 16, ex. 17.b, ex. 56, ex. 57;

[828] În cazul separării linare, în literatura de specialitate se foloseşte termenul de *analiză discriminativă gaussiană*.

[829] Nu doar pentru pseudo-cod, ci şi (sau, mai ales) pentru o privire de ansamblu unitară, atât pentru cazul unidimensional cât şi pentru cazul multidimensional (ambele urmând a fi sistematizate mai jos), puteţi consulta documentul *An Introduction to Expectation-Maximization*, de Dahua Lin, MIT, ML course 6768, 2012 fall.

cazul unidimensional, varianta când toţi parametrii (π, μ şi σ) sunt lăsaţi liberi: ex. 18; aplicare: ex. 17.c, ex. 58, ex. 59;
cazul unidimensional, alte variante: [ex. 19] ex. 60, ex. 61, ex. 62, ex. 63;
cazul multidimensional, varianta când toţi parametrii (π, μ şi Σ) sunt lăsaţi liberi: ex. 24;
cazul multidimensional, alte variante: ex. 20, ex. 64;
aplicarea algoritmului EM/GMM, cazul bidimensional: ex. 21.b, ex. 22.b, ex. 23, ex. 25, ex. 66, ex. 67, ex. 68;

- **schema algoritmică EM: vedeţi Tom Mitchell, *Machine Learning* book, 1997, pag. 194-195; ex. 19;**
- **pentru *proprietăţi generale* ale schemei algoritmice EM, vedeţi secţiunea corespunzătoare din *Sumarul* capitolului *Schema algoritmică EM*;**
- **câteva *proprietăţi* ale algoritmului EM/GMM:**

  (P0) rezultatele algoritmului EM/GMM depind (ca şi la K-means) de valorile atribuite parametrilor la iniţializare (ex. 17.c);

  (P1) anumite valori atribuite iniţial parametrilor algoritmului EM/GMM pot provoca rularea la infinit a algoritmului, fără ca [la pasul M] valorile parametrilor să se modifice de la o iteraţie la alta: ex. 19.c;

  (P2) spre deosebire de cazul algoritmului K-means, suprafeţele / graniţele de separare create de algoritmul EM/GMM nu sunt în mod neapărat liniare. (Vedeţi de exemplu situaţiile întâlnite la rezolvarea ex. 17.c, pag. 897, sau la ex. 68.c şi ex. 69.c);

- **comparativ cu algoritmul K-means,**
  (P3) algoritmul EM/GMM este în general mai lent — mişcarea centroizilor poate explora într-o manieră mai fină spaţiul (vedeţi de exemplu ex. 21) —, dar din acest motiv el poate să obţină uneori rezultate mai bune / convenabile. (Vedeţi spre exemplu ex. 22);

  (P4) apare un fenomen de "atracţie" reciprocă a mediilor gaussienelor (aceste medii fiind echivalentul centroizilor din algoritmul K-means), datorită faptului că fiecare instanţă aparţine (cu o anumită probabilitate) la fiecare cluster. Atracţia mediilor este cu atât mai puternică cu cât varianţele sunt mai mari. (Vedeţi spre exemplu ex. 17.b);

  (P5) EM/GMM este mai robust la influenţa outlier-elor.

  (P6) atunci când $\Sigma = \sigma^2 I$, iar $\sigma^2 \to 0$ (şi sunt satisfăcute încă două restricţii), algoritmul EM/GMM tinde să se comporte asemenea algoritmului K-means:[830] ex. 65;

- **o legătură interesantă între algoritmul EM/GMM şi metoda gradientului ascendent, în cazul în care matricele de covarianţă sunt de forma $\sigma_k^2 I$: ex. 70;**
- **algoritmul EM/GMM *semi-supervizat*: ex. 71;**
  o variantă a algoritmului EM/GMM semi-supervizată pentru cazul când matricele de covarianţă sunt de forma $\sigma_k^2 I$ şi este satisfăcută presupoziţia de independenţă condiţională de tip Bayes Naiv: ex. 72 (are loc o legătură interesantă cu clasificatorul Bayes Naiv gaussian).

[830] Formularea se referă la cazul multidimensional, dar propritatea este valabilă şi în cazul unidimensional.

7.1 Clusterizare — Probleme rezolvate

7.1.1 Clusterizare ierarhică

1. **(Clusterizare ierarhică aglomerativă: exemplificare pe date din $\mathbb{R}$, folosind similaritate de tip "single-linkage")**

CMU, 2006 spring, Carlos Guestrin, HW5, pr. 1

a. Desenaţi *dendrograma* (adică arborele de clusterizare ierarhică) pentru următoarea mulţime de 10 puncte pe axa reală:

$$S = \{-2.2,\ -2.0,\ -0.3,\ 0.1,\ 0.2,\ 0.4,\ 1.6,\ 1.7,\ 1.9,\ 2.0\}$$

Folosiţi *similaritate* de tip "single-linkage", adică $d(C_i, C_j) = \min_{x \in C_i, x' \in C_j} |x - x'|$**.**

***Observaţie*: În dendrogramă, înălţimea (h) corespunzătoare nodului rădăcină al unui cluster va fi considerată direct proporţională cu media aritmetică a distanţelor dintre punctele din clusterul respectiv,[831] iar *coeziunea* (c) clusterului va fi definită aici ca inversul acestei medii.[832]**

b. Bazat pe arborele obţinut, justificaţi — în manieră informală — faptul că 3 este numărul natural de clustere care se formează în acest set de date.

Răspuns:

a. Numerotăm cele 10 puncte date în ordinea crescătoare a valorilor lor: $x_1 = -2.2, x_2 = -2, x_3 = -0.3, x_4 = 0.1, x_5 = 0.2, x_6 = 0.4, x_7 = 1.6, x_8 = 1.7, x_9 = 1.9, x_{10} = 2.0$**.**

Dendrograma mulţimii de instanţe date se construieşte în manieră *bottom-up* astfel:

- $d(x_4, x_5) = d(x_7, x_8) = d(x_9, x_{10}) = 0.1 = \min_{1 \le i < j \le 10} d(x_j, x_i)$**, prin urmare primele 3 clustere care se vor forma sunt** $C_1 = \{x_4, x_5\}, C_2 = \{x_7, x_8\}, C_3 = \{x_9, x_{10}\}$**. În mod evident,** $h(C_1) = h(C_2) = h(C_3) = 0.1$ **iar** $c(C_1) = c(C_2) = c(C_3) = 1/0.1 = 10$**.**
- **Avem:**

  $$d(x_1, x_2) = 0.2$$
  $$d(C_1, x_6) = d(x_5, x_6) = 0.2$$
  $$d(C_2, C_3) = d(x_8, x_9) = 0.2$$

  Celelalte distanţe sunt toate mai mari decât 0.2**, aşadar următorul nivel al arborelui va fi alcătuit din clusterele** $C_4 = \{x_1, x_2\}, C_5 = \{x_4, x_5, x_6\}$ **şi** $C_6 = \{x_7, x_8, x_9, x_{10}\}$**. Făcând calculele, obţinem:** $h(C_4) = h(C_5) = 0.2$ **şi** $h(C_6) = \frac{1.4}{6} = 0.2(3)$**, iar** $c(C_4) = c(C_5) = 1/0.2 = 5$ **şi** $c(C_6) = \frac{6}{1.4} \approx 4.285$**.**

[831] În rezolvarea acestei probleme, factorul de proporţionalitate va fi considerat 1.
[832] Evident, această definiţie pentru coeziune are sens doar în cazul clusterelor non-singleton.

- **Avem:**

$$d(C_4, x_3) = d(x_2, x_3) = 1.7 \qquad d(C_4, C_5) = d(x_2, x_4) = 2.1$$
$$d(C_5, x_3) = d(x_4, x_3) = 0.4 \qquad d(C_4, C_6) = d(x_2, x_7) = 3.6$$
$$d(C_6, x_3) = d(x_7, x_3) = 1.9 \qquad d(C_5, C_6) = d(x_6, x_7) = 1.2$$

Aşadar, pe următorul nivel al ierarhiei se va afla clusterul $C_7 = C_5 \cup \{x_3\} = \bigcup_{3 \le i \le 6}\{x_i\}$**. Înălţimea şi respectiv coeziunea acestui cluster sunt:** $h(C_7) = \dfrac{2.2}{6} = 0.3(6)$ **şi** $c(C_7) = \dfrac{6}{2.2} = 2.(72)$**.**

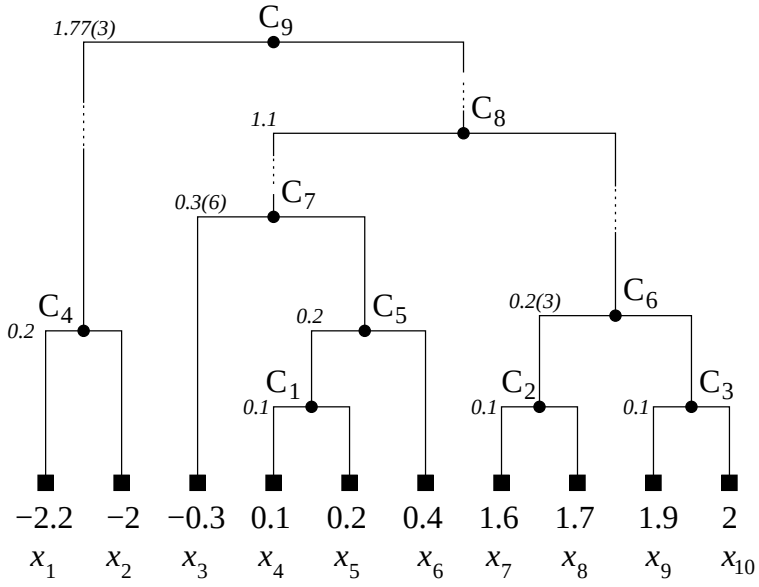
- **Avem:**

$$d(C_4, C_6) = d(x_2, x_7) = 3.6$$
$$d(C_4, C_7) = d(x_2, x_3) = 1.7$$
$$d(C_6, C_7) = d(x_6, x_7) = 1.2$$

În consecinţă, pe următorul nivel al ierarhiei se va afla clusterul $C_8 = C_6 \cup C_7 = \bigcup_{3 \le i \le 10}\{x_i\}$**. Făcând calculele, obţinem** $h(C_8) = \dfrac{30.8}{28} = 1.1$ **şi** $c(C_8) = 1/1.1 = 0.9(09)$**.**

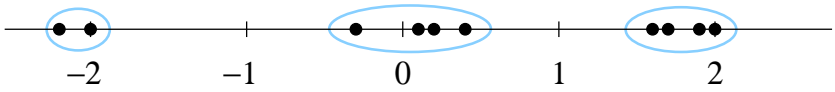
- **Pentru clusterul** $C_9 = C_4 \cup C_8$**, avem** $h(C_9) = \dfrac{79.8}{45} = 1.77(3)$ **şi** $c(C_9) \approx 0.564$**.**

Dendrograma mulţimii date va arăta deci astfel:



b. După alcătuirea dendrogramei, împărţirea mulţimii de instanţe date în clustere cu coeziune comparabilă se face „tăind“ dendrograma cu o linie paralelă cu baza. Împărţirea în clustere fine corespunde liniilor paralele apropiate de bază, iar împărţirea în clustere ample corespunde liniilor paralele apropiate de vârful / rădăcina dendrogramei.

În cazul nostru, se observă că pentru o mare parte a înălţimii dendrogramei (şi anume între 0.3(6) şi 1.1), rezultatul operaţiei descrise mai sus rămâne neschimbat (mulţimea dată descompunându-se corespunzător în clusterele C_4, C_7 **şi** C_6**), ceea ce sugerează că „numărul natural“ de clustere este 3. Acest lucru se poate vedea şi dacă reprezentăm cele 10 puncte pe axa reală:**



2. **(Clusterizare ierarhică aglomerativă: exemplificare pe date din $\mathbb{R}^2$, folosind tipurile de similaritate "single-linkage" şi "complete-linkage")**

Edinburgh, 2009 fall, C. Williams, V. Lavrenko, HW4, pr. 1

Considerăm următorul set de date, în care fiecare instanţă (x, y) este un punct în plan. Pentru convenienţă, în cele ce urmează, vom identifica aceste instanţe prin literele asociate punctelor respective.

$$A : (-4, -2), B : (-3, -2), C : (-2, -2), D : (-1, -2), E : (+1, -1)$$
$$F : (+1, +1), G : (+2, +3), H : (+3, +2), I : (+3, +4), J : (+4, +3)$$

a. Reprezentaţi grafic datele din enunţ.

b. Realizaţi clusterizarea ierarhică a datelor în maniera bottom-up, folosind similaritate de tip "single-linkage" şi distanţa euclidiană între puncte.

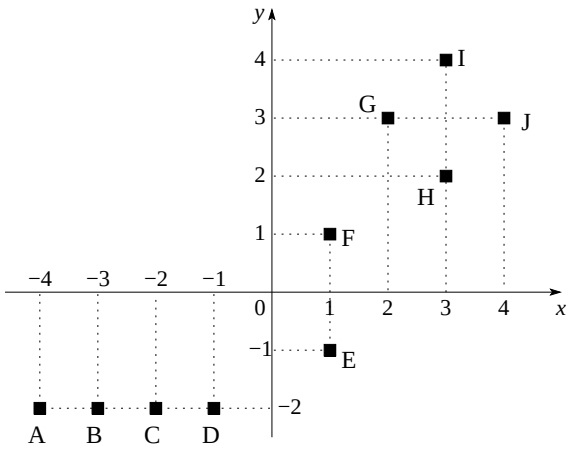
***Observaţie*: Dacă la o iteraţie a algoritmului de clusterizare distanţele (adică similarităţile) dintre două perechi de clustere au aceeaşi valoare, prioritatea la alcătuirea noului cluster este dictată de ordinea alfabetică.**

c. Realizaţi clusterizarea datelor, folosind de această dată similaritate de tip "complete-linkage".

d. Discutaţi diferenţa dintre clusterizările obţinute în urma folosirii celor două tipuri de [măsuri de] similaritate.

Răspuns:

a. Reprezentarea în plan a celor 8 puncte date în enunţ este cea din figura alăturată.



b. Clusterizarea folosind similaritate de tip "single-linkage" (pentru care distanţa dintre două clustere este definită ca fiind distanţa dintre cele mai apropiate două puncte, câte unul din fiecare cluster) se realizează astfel:

- **Minimul distanţelor mutuale dintre punctele date este 1. Acest minim este obţinut pentru mai multe perechi de puncte, deci se va recurge la ordinea alfabetică pentru a stabili prioritatea. Prin urmare, primul cluster format va fi $C_1 = \{A, B\}$. Evident, $h(C_1) = c(C_1) = 1$, unde h şi c sunt înălţimea asociată clusterului C_1, respectiv coeziunea lui, calculate după aceeaşi metodă ca la problema 1.**

- **Acum, distanţa minimă este obţinută (ţinând cont şi de regula alfabetică) între clusterul C_1 şi punctul C, deci $C_2 = C_1 \cup \{C\} = \{A, B, C\}$. Prin urmare, $h(C_2) = 4/3 = 1.(3)$, iar $c(C_2) = 3/4 = 0.75$.**

- **În mod similar, tot la distanţa 1, vom avea $C_3 = C_2 \cup \{D\} = \{A, B, C, D\}$, iar $h(C_3) = 10/6 = 1.(6)$ şi $c(C_3) = 6/10 = 0.6$.**

- **La acest nou pas se consideră clusterul C_3 şi punctele $E, F, \ldots, J$. Distanţa minimă este $\sqrt{2}$, şi anume — având din nou în vedere ordinea alfabetică — între punctele G şi H, deci $C_4 = \{G, H\}$, cu $h(C_4) = \sqrt{2} \approx 1.414$ şi $c(C_4) = 1/\sqrt{2} \approx 0.707$.**

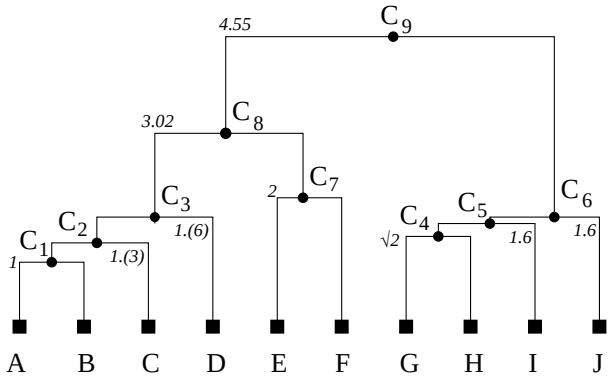
- **Apoi $C_5 = C_4 \cup \{I\} = \{G, H, I\}$ şi $C_6 = C_5 \cup \{J\} = \{G, H, I, J\}$, clustere care au înălţimile (egale!) $h(C_5) = h(C_6) = \frac{2}{3}(\sqrt{2} + 1) \approx 1.609$ şi, deci, şi coeziunile egale: $c(C_5) = c(C_6) = \frac{3}{2} \frac{1}{\sqrt{2} + 1} \approx 0.621$.**

- **Acum se consideră clusterele C_3 şi C_6 şi punctele E şi F. Distanţa minimă este 2, şi anume cea dintre punctele E şi F, deci $C_7 = \{E, F\}$, cu $h(C_7) = 2$ şi $c(C_7) = 0.5$.**

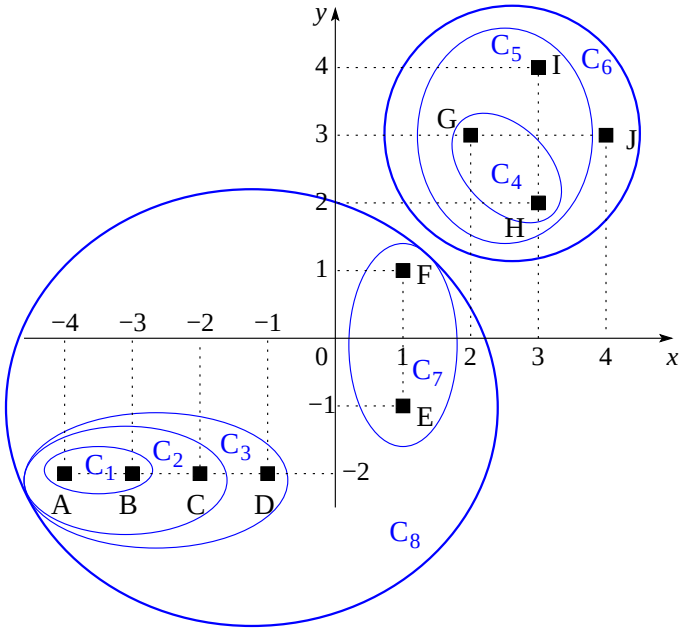
- **Între clusterele C_3, C_6 şi C_7, distanţa minimă este $\sqrt{5}$, deci $C_8 = C_3 \cup C_7 = \{A, B, C, D, E, F\}$, cu $h(C_8) \approx 3.02$ şi $c(C_8) \approx 0.331$.**

- **Rămâne în final $C_9 = C_8 \cup C_6$, cu $h(C_9) \approx 4.552$ şi $c(C_9) \approx 0.219$.**

Dendrograma obţinută este redată în figura alăturată.



Varianta „aplatizată“ a acestei ierarhii (engl., flat hierarchy) este prezentată în desenul alăturat.



c. Clusterizarea ierarhică de tip “complete-linkage” (pentru care similaritatea dintre două clustere este dată de distanţa dintre cele mai depărtate două puncte, câte unul din fiecare cluster) se realizează astfel:

- **Minimul distanţelor dintre oricare două puncte este 1, iar acest minim este obţinut pentru mai multe perechi de puncte. Luând în considerare ordinea alfabetică, primul cluster format va fi $C_1 = \{A, B\}$, cu $h(C_1) = c(C_1) = 1$.**
- **La acest nou pas, distanţa minimă este $d(C, D) = 1$. Deci, spre deosebire de cazul “single-linkage”, vom avea $C_2 = \{C, D\}$, cu $h(C_2) = c(C_2) = 1$.**
- **În continuare, cea mai mică distanţă este $\sqrt{2}$, între G şi H, deci $C_3 = \{G, H\}$, cu $h(C_3) = \sqrt{2} \approx 1.414$ şi $c(C_3) = 1/\sqrt{2} \approx 0.707$.**
- **Tot $\sqrt{2}$ este şi distanţa dintre I şi J, deci $C_4 = \{I, J\}$, cu $h(C_4) = \sqrt{2}$ şi $c(C_4) = 1/\sqrt{2}$.**
- **Apoi, $C_5 = \{E, F\}$, între aceste două puncte distanţa fiind 2, deci $h(C_5) = 2$ şi $c(C_5) = 0.5$.**
- **În acest moment se consideră clusterele C_1, C_2, C_3, C_4 şi C_5. Distanţele dintre ele sunt:**

$$
\begin{array}{ll}
d(C_1, C_2) = d(A, D) = 3 & d(C_2, C_4) = d(C, I) = d(C, J) = 7.81 \\
d(C_1, C_3) = d(A, H) = 8.06 & d(C_2, C_5) = d(C, F) = 4.24 \\
d(C_1, C_4) = d(A, J) = 9.43 & d(C_3, C_4) = d(G, J) = d(H, I) = 2 \\
d(C_1, C_5) = d(A, F) = 5.83 & d(C_3, C_5) = d(E, G) = 4.12 \\
d(C_2, C_3) = d(C, G) = d(C, H) = 6.4 & d(C_4, C_5) = d(E, I) = 5.39
\end{array}
$$

  Luând minimul acestor distanţe, si anume $d(C_3, C_4) = 2$, se obţine $C_6 = C_3 \cup C_4 = \{G, H, I, J\}$, cu $h(C_6) = \frac{2}{3}(\sqrt{2} + 1) \approx 1.609$ şi $c(C_6) \approx 0.621$.
- **Urmează $C_7 = C_1 \cup C_2 = \{A, B, C, D\}$, cu $h(C_7) = 10/6 = 1.(6)$ şi $c(C_7) = 0.6$.**

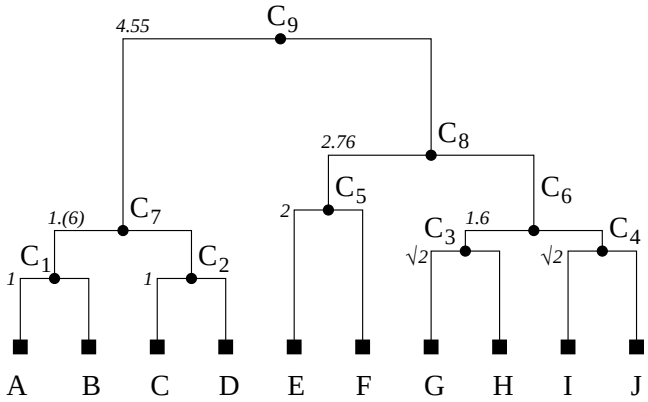
- **Între C_5, C_6 şi C_7, distanţele sunt:**

$$d(C_5, C_6) = d(E, I) = 5.39$$
$$d(C_5, C_7) = d(A, F) = 5.83$$
$$d(C_6, C_7) = d(A, J) = 9.43$$

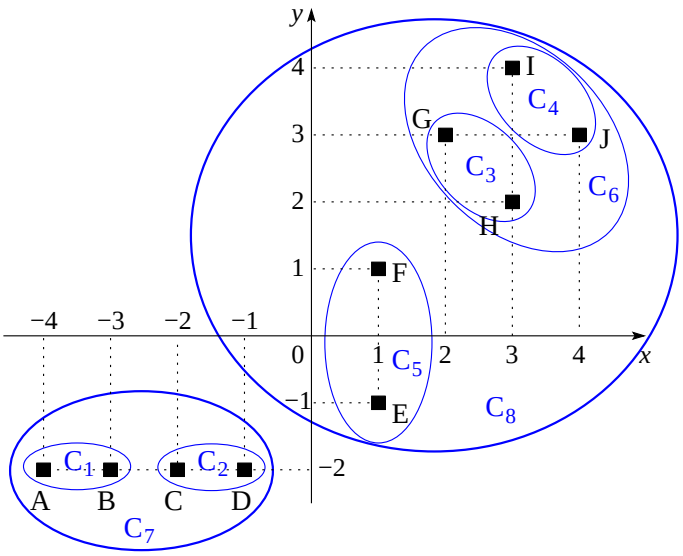
Întrucât cea mai mică dintre aceste distanţe este 5.39**, rezultă** $C_8 = C_5 \cup C_6 = \{E, F, G, H, I, J\}$**, cu** $h(C_8) \approx 2.76$ **şi** $c(C_8) \approx 0.36$**.**

- **Rămâne în final** $C_9 = C_7 \cup C_8$**, distanţa dintre cele două clustere constituente fiind de** 9.43**. Înălţimea şi coeziunea clusterului** C_9 **sunt respectiv** 4.55 **şi** 0.21**.**

Dendrograma rezultată este cea din figura alăturată.



Varianta „aplatizată“ a acestei ierarhii (engl., flat hierarchy) este prezentată în desenul alăturat.



d. Clusterizarea ierarhică folosind similaritate de tip “single-linkage” are tendinţa de a înlănţui / alungi clusterele, spre deosebire de cea “complete-linkage” care grupează clusterele mai degrabă sub formă sferică.

3. **(Clusterizare ierarhică aglomerativă: exemplificare în $\mathbb{R}$, folosind tipurile de similaritate "single-linkage" şi "complete-linkage", comparativ cu "average-linkage")**

prelucrare făcută de Liviu Ciortuz, după CMU, 2009 spring, Ziv Bar-Joseph, final exam, pr. 9.1

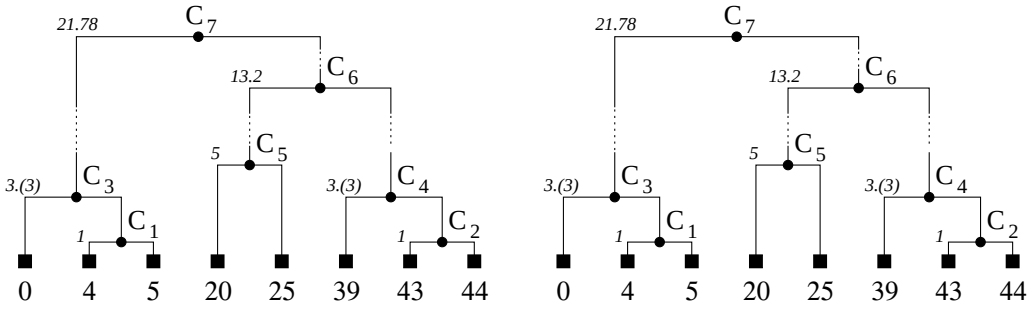
Considerăm următorul set de date: $\{0, 4, 5, 20, 25, 39, 43, 44\}$. Presupunem că vrem să obţinem cele mai importante 2 clustere din dendrogramă (cele situate imediat sub nodul rădăcină), numite în cele ce urmează *top-clustere*.

a. Care sunt cele două top-clustere atunci când se foloseşte similaritate *single-linkage* şi respectiv *complete-linkage*?

b. Dacă *single-linkage* şi *complete-linkage* produc rezultate identice (relativ la componenţa [top-]clusterelor), rezultă că funcţia de similaritate *average-linkage* va conduce şi ea la acelaşi rezultat? Pentru a elabora răspunsul dumneavoastră, vedeţi ce se întâmplă lucrând pe datele de mai sus.

Răspuns:

a. Dendrogramele corespunzătoare similarităţilor *single-linkage* şi *complete-linkage* sunt următoarele:[833]



Single-linkage *Complete-linkage*

Se observă că atât în cazul *single-linkage* cât şi în cazul *complete-linkage*, cele două top-clustere sunt $\{0, 4, 5\}$ şi $\{20, 25, 39, 43, 44\}$. Mai mult, cele două dendrograme sunt identice.

b. Clusterizarea ierarhică cu similaritate *average-linkage* foloseşte ca „distanţă" între două clustere oarecare media aritmetică a distanţelor calculate pentru toate perechile de puncte ce se pot forma luând un punct dintr-un cluster şi un punct din celălalt cluster. Aplicarea acestui algoritm se realizează astfel:

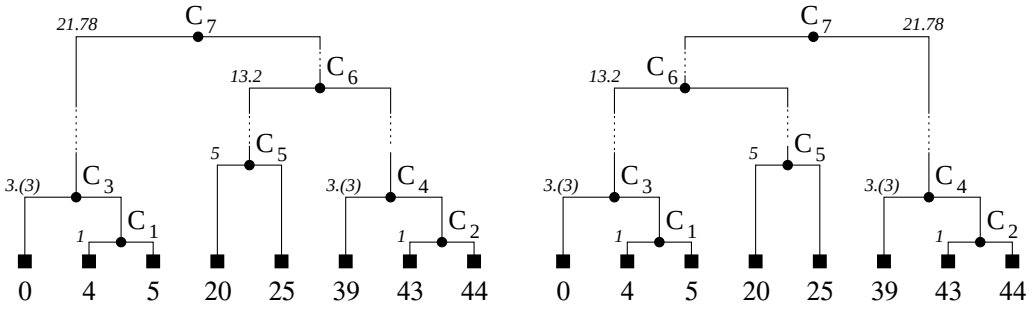
- **$C_1 = \{4, 5\}$ şi $C_2 = \{43, 44\}$, ambele constituite din câte o pereche de clustere "singleton" aflate la distanţa 1. Înălţimile clusterelor rezultate sunt $h(C_1) = h(C_2) = 1$.**[834]
- **$C_3 = \{0\} \cup C_1 = \{0, 4, 5\}$, distanţa dintre clusterul "singleton" $\{0\}$ şi clusterul C_1 fiind $\frac{4+5}{2} = 4.5$. Rezultă $h(C_3) = \frac{4+5+1}{3} = 3.(3)$.**

[833] Calculele sunt absolut similare celor de la rezolvarea problemei 2.
[834] Pentru calculul înălţimilor am procedat ca la problemele 1 şi 2.

- $C_4 = \{39\} \cup C_2 = \{39, 43, 44\}$. **Distanţa dintre clusterul** $\{39\}$ **şi clusterul** C_2 **este** $\frac{(43-39)+(44-39)}{2} = \frac{4+5}{2} = 4.5$. **Înălţimea noului cluster este** $h(C_4) = 3.(3)$.
- $C_5 = \{20, 25\}$**, cu distanţa dintre clusterele "singleton" constitutive** 5 **şi înălţimea rezultantă tot** 5.
- **În acest moment există** 3 **clustere:** $C_3 = \{0, 4, 5\}$**,** $C_4 = \{39, 43, 44\}$ **şi** $C_5 = \{20, 25\}$. **Sunt relevante două dintre cele trei distanţe mutuale dintre aceste clustere, şi anume:**

$$d(C_3, C_5) = \frac{20+25+16+21+15+20}{6} = \frac{117}{6} = 19.5$$
$$d(C_4, C_5) = \frac{19+14+23+18+24+19}{6} = \frac{117}{6} = 19.5$$

Se observă că aceste două distanţe sunt egale. Deoarece în enunţ nu s-a specificat — pentru un astfel de caz — vreo regulă de prioritate de care să se ţină cont atunci când se combină două clustere,[835] **cele două top-clustere care se pot obţine în cazul nostru sunt fie** $\{0, 4, 5\}$ **şi** $\{20, 25, 39, 43, 44\}$ **— adică exact ca la punctul** a **—, fie** $\{0, 4, 5, 20, 25\}$ **şi** $\{39, 43, 44\}$. **Acest fapt este ilustrat în cele două dendrograme de mai jos:**



***Concluzie*: Chiar dacă, lucrând pe un acelaşi set de date, la clusterizare ierarhică folosind similaritate *single-linkage* şi respectiv *complete-linkage* se obţin rezultate identice, este posibil ca atunci când se foloseşte *average-linkage* să obţinem un alt rezultat.**

4. **(Clusterizare ierarhică aglomerativă, folosind similaritate "single-linkage": exemplificare pe date din $\mathbb{R}$; corespondenţa cu numărul maxim / minim de niveluri din arborele clasic corespunzător dendrogramei)**

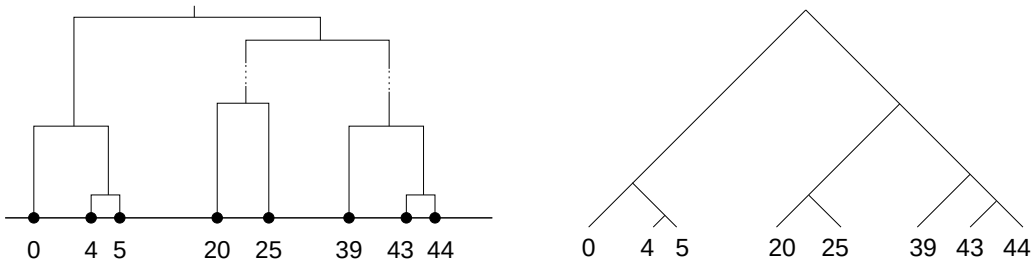
CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW3, pr. 5.2

În acest exerciţiu veţi folosi metoda clusterizării ierarhice aglomerative cu similaritate de tip "single-linkage" la compararea a două clustere oarecare.[836]

[835] Vedeţi regulile folosite la problemele 1 şi 2.

[836] În acest caz, distanţa dintre două clustere este definită ca minimul distanţelor $d(x, y)$, unde x aparţine primului cluster, iar y aparţine celui de-al doilea cluster.

Vom defini *înălţimea* unei ierarhii ca fiind $l-1$, unde l este numărul de niveluri din arborele (în sens clasic) corespunzător acestei ierarhii. De *exemplu*, pentru numerele $0, 4, 5, 20, 25, 39, 43, 44$, ierarhia "single-linkage" (arătată în desenul de mai jos, partea stângă) are înălţimea 4. Pentru convenienţă (şi pentru a elimina posibilele confuzii datorate înălţimilor diferitelor noduri din dendrogramă), am desenat şi arborele clasic care corespunde acestei ierarhii (vedeţi desenul din partea dreaptă); evident, el are 5 niveluri.



a. Care este înălţimea maximă a unei ierarhii care se poate construi cu N puncte? Dar înălţimea minimă?

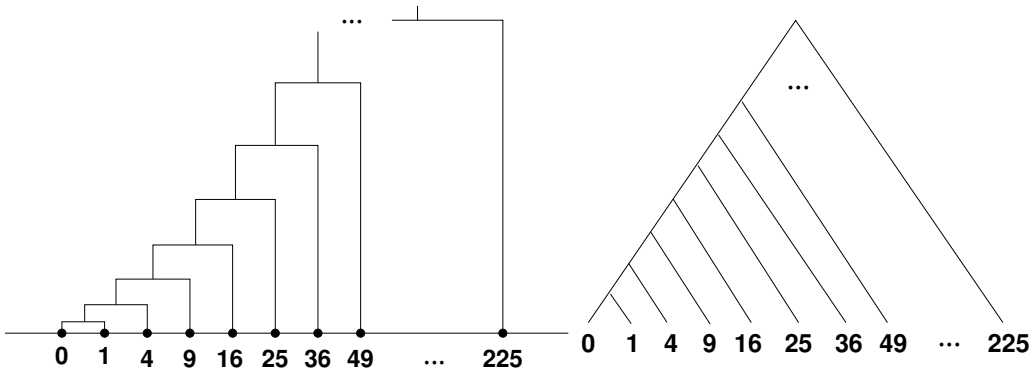
b. Daţi un *exemplu* de 16 puncte din mulţimea numerelor întregi care la acest tip de clusterizare produce ($i.$) o ierarhie de înălţime maximă, ($ii.$) o ierarhie de înălţime minimă.

Răspuns:

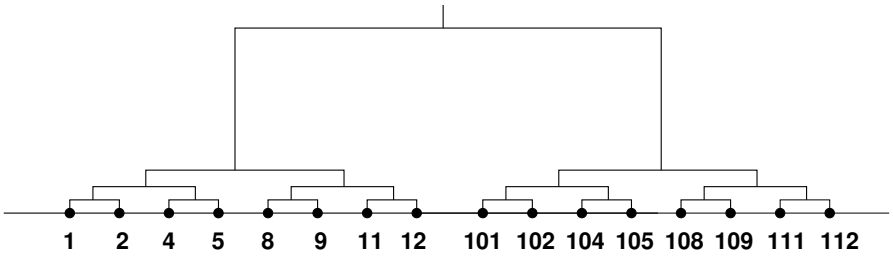
a. Pentru a avea o ierarhie de înălţime maximă, putem considera spre exemplu cazul în care distanţele dintre punctele care se clusterizează succesiv sunt în ordine strict crescătoare, obţinând un arbore cu N niveluri. Deci înălţimea maximă a unei ierarhii este $N-1$.

Pentru a avea înălţime minimă, trebuie ca arborele corespuzător să fie pe cât posibil un arbore binar echilibrat. Deci înălţimea minimă este partea întreagă superioară din $\log_2(N)$.

b. O mulţime de 16 puncte care conduce la o ierarhie de tip "single-linkage" de înălţime maximă este: $0, 1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225$. Ierarhia este următoarea (am folosit ambele variante de reprezentare grafică):



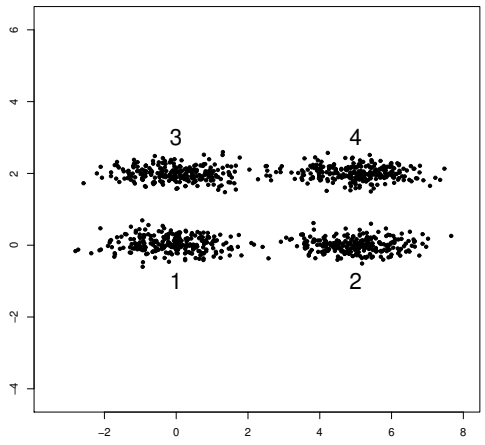
Pentru o ierarhie de tip "single-linkage" de înălţime minimă putem considera următoarele 16 puncte: $1, 2, 4, 5, 8, 9, 11, 12, 101, 102, 104, 105, 108, 109, 111, 112$. Ierarhia este următoarea:



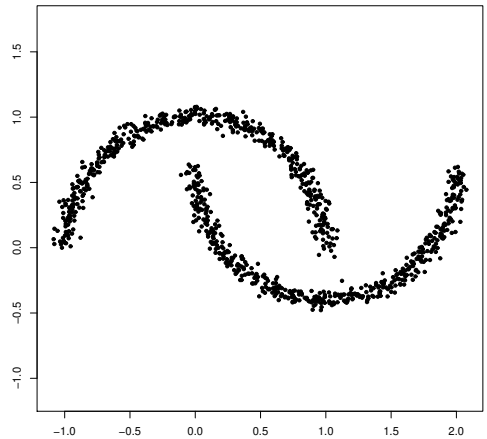
5. **(Clusterizare ierarhică: aplicare în manieră intuitivă pe date din $\mathbb{R}^2$)**

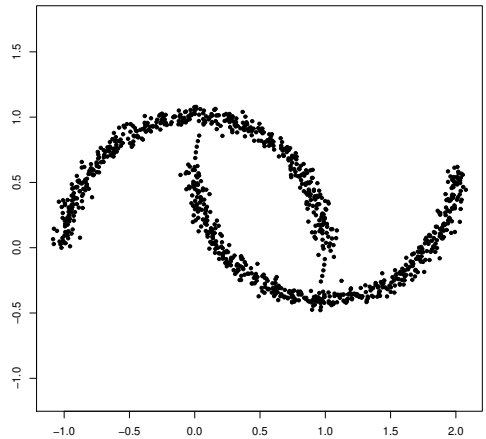
• ○ ***CMU, 2012 fall, E. Xing, A. Singh, HW3, pr. 1.2.bc***

a. Considerăm datele din figura alăturată.
Ce rezultat vom obţine dacă extragem cele două clustere de "top" din arborele obţinut prin clusterizare ierarhică pe acest set de date folosind măsura de similaritate "single-linkage"? (Formulaţi răspunsul raportându-vă la etichetele 1–4 care identifică cele patru grupuri din date.) Procedaţi similar pentru "complete-linkage" şi "average-linkage".



b. Poate vreuna dintre cele trei măsuri de similaritate să separe cu succes cele două „semiluni" din figura de mai jos, partea stângă? Dar în ce priveşte figura din partea dreaptă? Justificaţi pe scurt.





Răspuns:

a. "Single-linkage" va asigna aglomerările de puncte notate cu 1 şi 2 la un cluster, iar pe cele notate cu 3 şi 4 la celălalt cluster. "Complete-linkage" şi "average-linkage" vor asigna aglomerările de puncte notate cu 1 şi 3 la un cluster, iar pe cele notate cu 2 şi 4 la alt cluster.

b. "Single-linkage" va reuşi să separe cu succes cele două semiluni din figura din partea stângă (din enunţ), în vreme ce "complete-linkage" şi "average-linkage" nu vor reuşi. Niciuna dintre cele trei metode nu va avea succes pe datele din figura din partea dreaptă.

6. (Un algoritm de clusterizare ierarhică divizivă: aplicare; comparaţie cu variantele algoritmului de clusterizare aglomerativă; echivalenţa cu algoritmii de aflare a MST din teoria grafurilor)

prelucrare făcută de Liviu Ciortuz, după
■ • ***CMU, 2009 spring, Ziv Bar-Joseph, final exam, pr. 9.3***

Clusterizarea ierarhică poate fi de două tipuri: *divizivă* (top-down) sau *aglomerativă* (bottom-up). Întrebarea la care am dori să răspundem prin această problemă este dacă un algoritm din categoria top-down poate să fie echivalent (relativ la rezultatul obţinut ca atare, sau chiar la modul efectiv de elaborare a clusterizării) cu un algoritm din categoria bottom-up.

Să considerăm următorul *algoritm de clusterizare de tip top-down*:

• *Intrare*: o mulţime de instanţe $S = \{x_1, x_2, \ldots, x_n\}$ şi o măsură de distanţă definită pe $S \times S$;

• *Ieşire*: o dendrogramă, adică un arbore de clusterizare binară în ale cărui noduri frunză sunt elementele din S, satisfăcând proprietăţile dezirabile pentru clusterizare ierarhică (i.e., coeziune cât mai mare în interiorul fiecărui cluster şi distanţe maxime între clustere, pe fiecare nivel din dendrogramă);

• *Procedură*:

***Pasul* 1: Se construieşte graful neorientat, complet[837] şi ponderat, având ca mulţime de noduri chiar mulţimea S (aşadar, fiecare nod din graf este reprezentat de o instanţă $x_i \in S$), iar ponderea / costul fiecărei muchii din graf este exact distanţa dintre punctele reprezentate de nodurile adiacente acestei muchii.**

***Pasul* 2: Se calculează un arbore de acoperire de cost minim (engl., Minimum Spanning Tree) corespunzător grafului obţinut la *Pasul* 1. Această operaţie revine la alegerea unei submulţimi de muchii din graful complet care *i.* constituie un arbore T ce foloseşte / conexează toate nodurile din graf, iar *ii.* suma costurilor / lungimilor muchiilor sale este minimă (în raport cu toţi arborii care satisfac condiţia precedentă). Acest arbore se poate obţine folosind *algoritmul lui Kruskal*[838] sau *algoritmul lui Prim*[839].**

[837] Un graf neorientat este complet dacă între oricare două noduri ale sale există o [singură] muchie.

[838] Publicat în 1956 de americanul Joseph Kruskal (1928-2010) în *Proceedings of American Mathematical Society*, pag. 48-50.

[839] Algoritm descoperit de Vojtech Jarnik în 1930, redescoperit de Robert Prim în 1957 şi din nou de către Edsger Dijkstra în 1959. Din această cauză, el este numit uneori în literatura de specialitate *algoritmul DJP*. Atât acest algoritm cât şi algoritmul lui Kruskal au complexitatea $O(n \log n)$.

***Pasul* 3: Se elimină din arborele de cost minim obţinut la *Pasul* 2 muchia cu costul maxim, obţinându-se astfel doi arbori. Aceştia vor corespunde celor două clustere de pe nivelul cel mai de sus în dendrograma pe care o construim în manieră top-down.**

***Pasul* 4: Se repetă recursiv *Pasul* 3 atât timp cât este posibil, obţinând astfel o clusterizare de tip top-down pe mulţimea de instanţe date.**

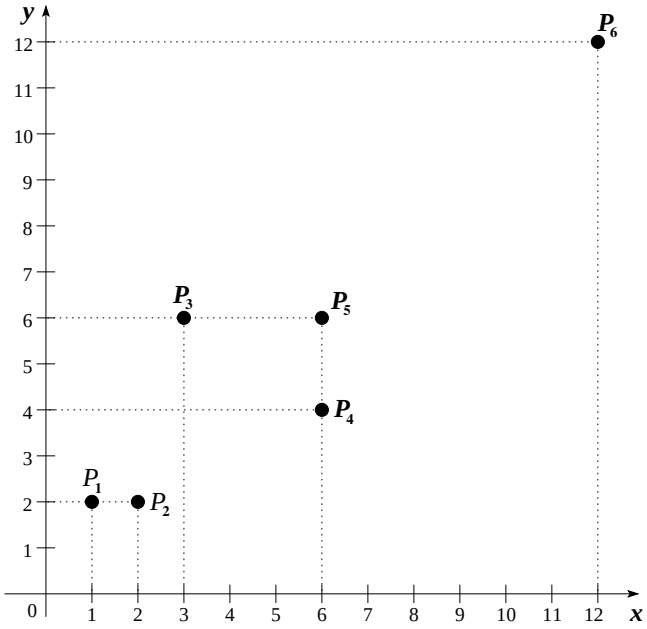
a. Aplicaţi acest algoritm pe setul de date din $\mathbb{R}^2$ din tabelul alăturat, folosind distanţa euclidiană. Desenaţi rezultatul sub forma unei ierarhii „aplatizate“ sau ca dendrogramă (fără a face calculul exact al înălţimilor).

Punctul	x	y
P_1	1	2
P_2	2	2
P_3	3	6
P_4	6	4
P_5	6	6
P_6	12	12

b. Este oare acest algoritm diviziv echivalent[840] cu unul dintre algoritmii de clusterizare aglomerativă studiaţi? Justificaţi atât pe datele acestea cât şi în cazul general. Atenţie la situaţiile în care se poate alege între mai mulţi candidaţi.

Răspuns:

a. Reprezentarea celor şase puncte în plan este dată în figura alăturată.



Vom aplica paşii algoritmului descris în enunţ:

***Pasul* 1: La calcularea distanţelor dintre perechile de puncte din setul de instanţe date, se poate completa un tabel, aşa cum apare mai jos. Deoarece aceste distanţe se pot calcula folosind teorema lui Pitagora, iar ulterior ele vor fi folosite doar pentru a face diverse comparaţii între ele, este suficient să le exprimăm cu ajutorul radicalului. Notăm faptul că acest tabel al distanţelor**

[840] Adică: produc exact acelaşi rezultat şi eventual exact în aceeaşi ordine (vedeţi pasul 2 şi respectiv paşii 3-4).

poate fi considerat *matricea de adiacenţă* a grafului complet pe care vom lucra în continuare.

		$(1,2)$ P_1	$(2,2)$ P_2	$(3,6)$ P_3	$(6,4)$ P_4	$(6,6)$ P_5	$(12,12)$ P_6
$(1,2)$	P_1	0	$\sqrt{1}$	$\sqrt{20}$	$\sqrt{29}$	$\sqrt{41}$	$\sqrt{221}$
$(2,2)$	P_2	$\sqrt{1}$	0	$\sqrt{17}$	$\sqrt{20}$	$\sqrt{32}$	$\sqrt{200}$
$(3,6)$	P_3	$\sqrt{20}$	$\sqrt{17}$	0	$\sqrt{13}$	$\sqrt{9}$	$\sqrt{117}$
$(6,4)$	P_4	$\sqrt{29}$	$\sqrt{20}$	$\sqrt{13}$	0	$\sqrt{4}$	$\sqrt{100}$
$(6,6)$	P_5	$\sqrt{41}$	$\sqrt{32}$	$\sqrt{9}$	$\sqrt{4}$	0	$\sqrt{72}$
$(12,12)$	P_6	$\sqrt{221}$	$\sqrt{200}$	$\sqrt{117}$	$\sqrt{100}$	$\sqrt{72}$	0

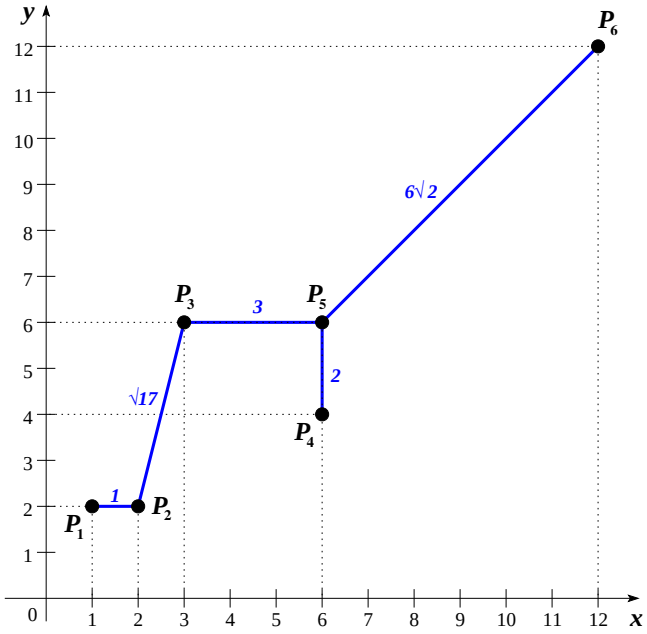
***Pasul* 2: Vom determina un arbore de cost minim folosind *algoritmul lui Kruskal*. Pentru un graf conex,[841] acest algoritm poate fi descris pe scurt astfel:**

Se porneşte de la graful parţial de cost minim [care are toate nodurile grafului conex dat, dar] care nu are nicio muchie. Apoi, în mod iterativ se alege câte o muchie de cost minim din graful iniţial, care n-a fost încă folosită şi care uneşte două componente conexe ale grafului parţial de la iteraţia anterioară. Algoritmul se încheie după $n-1$ iteraţii, unde n este numărul de noduri din graf, adică atunci când graful parţial construit este conex. Se poate demonstra că acesta este chiar un arbore de cost minim.

Aplicat pe setul de date din enunţ, algoritmul va alege în ordine următoarele muchii:

- (P_1, P_2) **cu costul** 1**,**
- (P_4, P_5) **cu costul** 2**,**
- (P_3, P_5) **cu costul** 3**,**
- (P_2, P_3) **cu costul** $\sqrt{17}$
- (P_5, P_6) **cu costul** $6\sqrt{2}$**.**

Arborele de cost minim rezultat este reprezentat în figura alăturată.



***Pasul* 3: Eliminând muchia de cost maxim, adică muchia (P_5, P_6), care are costul $\sqrt{72} = 6\sqrt{2}$, rezultă clusterele: $C_1 = \{P_1, P_2, P_3, P_4, P_5\}$ şi $C_2 = \{P_6\}$.**

La iteraţia următoare se elimină muchia (P_2, P_3) cu costul $\sqrt{17}$, rezultând clusterele: $C_3 = \{P_1, P_2\}$ şi $C_4 = \{P_3, P_4, P_5\}$.

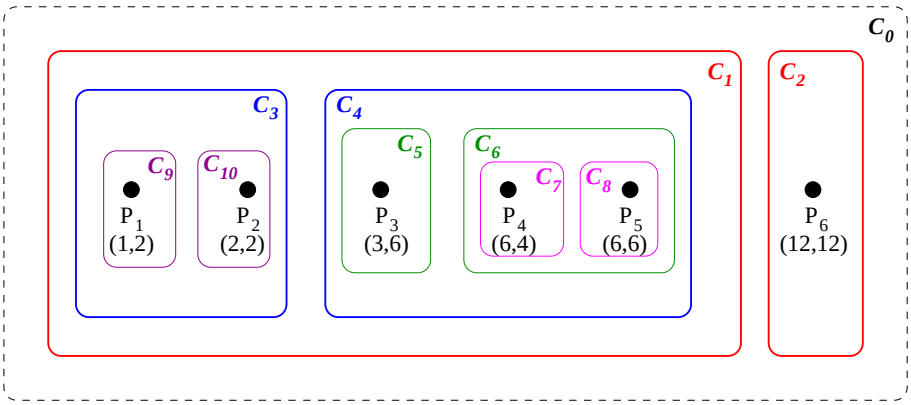
Apoi se elimină muchia (P_3, P_5) cu costul 3, rezultând clusterele: $C_5 = \{P_3\}$ şi $C_6 = \{P_4, P_5\}$.

[841] Un graf neorientat este conex dacă între oricare două noduri ale sale există cel puţin un drum.

Au mai rămas două muchii, care se vor elimina pe rând, şi anume: mai întâi muchia (P_4, P_5), producând clusterele $C_7 = \{P_4\}$ şi $C_8 = \{P_5\}$, şi în final muchia (P_1, P_2), rezultând clusterele $C_9 = \{P_1\}$ şi $C_{10} = \{P_2\}$.

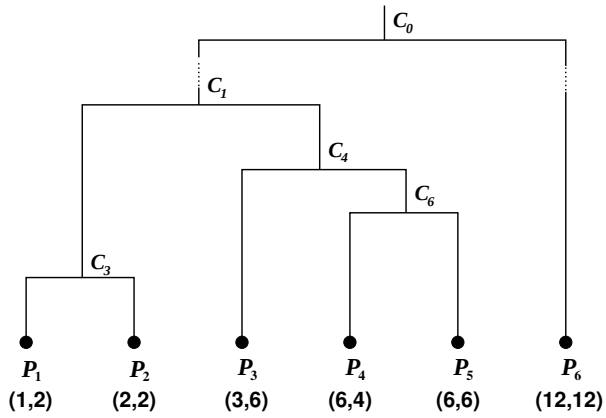
***Observaţie*: Eliminarea muchiilor din arbore se realizează exact în ordine inversă faţă de ordinea în care au fost alese iniţial de către algoritmul lui Kruskal.**

Putem reprezenta acum într-o formă aplatizată efectul aplicării algoritmului de clusterizare ierarhică top-down pe setul de instanţe date:



b. Dacă se foloseşte algoritmul lui Kruskal pentru determinarea arborelui de cost minim, atunci algoritmul de clusterizare top-down din enunţ este echivalent (în privinţa rezultatelor obţinute) cu algoritmul de clusterizare aglomerativă care foloseşte similaritate de tip "single-linkage". Chiar mai mult, alegerea muchiilor arborelui de cost minim corespunde (în ordine inversă, pas cu pas) calculului distanţelor de tip "single-linkage" dintre clustere.

Dendrograma obţinută — fără a include calculul înălţimilor — de către algoritmul de clusterizare aglomerativă folosind similaritate de tip "single-linkage" pe datele din enunţ este cea din figura alăturată.



***Observaţia* 1: Este posibil ca atunci când se foloseşte *algoritmul lui Prim* pentru determinarea arborelui de cost minim să se obţină rezultate diferite faţă de cazul când se foloseşte algoritmul lui Kruskal. Pentru un graf conex, algoritmul lui Prim poate fi descris pe scurt astfel:**

Iniţial, arborele este alcătuit dintr-un singur nod, ales în mod aleatoriu dintre nodurile grafului. La fiecare iteraţie se adaugă la acest arbore muchia de cost minim (şi nodul corespunzător) care uneşte unul dintre nodurile nefolosite cu arborele deja existent. Algoritmul se încheie după $n - 1$ iteraţii, unde n este numărul de noduri din graf, adică

atunci când graful parţial construit conectează toate nodurile grafului iniţial. Acest graf parţial este chiar un arbore de cost minim.

***Observaţia* 2: Dacă graful considerat admite un singur arbore de cost minim — se poate verifica imediat că aşa se întâmplă şi în cazul de faţă —, atunci rezultatul aplicării algoritmilor lui Kruskal şi respectiv al lui Prim este, evident, acelaşi. În caz contrar, arborii obţinuţi de fiecare dintre cei doi algoritmi depind de alegerile care pot fi făcute atunci când la un pas oarecare există mai multe muchii de cost minim. În plus, arborele obţinut de către algoritmul lui Prim depinde de alegerea primului nod. Chiar dacă prin aplicarea celor doi algoritmi se obţine acelaşi rezultat (adică acelaşi arbore de cost minim), în general ordinea de alegere a muchiilor diferă. Algoritmul lui Prim extinde pas cu pas un arbore care în final devine arbore(le) de cost minim, în vreme ce algoritmul lui Kruskal construieşte o *pădure de arbori* pe care-i uneşte până la final într-un arbore de cost minim.**

Aşadar, în cazul în care la *Pasul* 2 al algoritmului de clusterizare top-down prezentat în enunţ se foloseşte algoritmul lui Prim, în general nu se poate stabili o echivalenţă directă între modul de lucru al algoritmului de clusterizare top-down pe de o parte (Pasul 2) şi modul de lucru al algoritmul de clusterizare aglomerativă cu similaritate de tip "single-linkage" pe de altă parte,[842] chiar dacă rezultatul final (adică, dendrograma) este exact acelaşi ca în cazul folosirii algoritmului lui Kruskal.

Indiferent dacă se foloseşte algoritmul lui Kruskal sau algoritmul lui Prim, la paşii 3-4 din algoritmul de clusterizare top-down se procedează exact în ordine inversă faţă de construirea dendrogramei prin clusterizare bottom-up folosind similaritate single-linkage (făcând alegerile corespunzătoare, în cazul situaţiilor care comportă posibilităţi de alegere multiple).

7.1.2 Algoritmul K-means

7. **(Algoritmul K-means:[843] un exemplu simplu de aplicare pe date din $\mathbb{R}^2$)**

prelucrare de Liviu Ciortuz, după
• Univ. of Utah, 2008 spring, HW3A, pr. 2

Considerăm în planul euclidian bidimensional un set de date format din următoarele puncte: $A(-1,0)$, $B(1,0)$, $C(0,1)$, $D(3,0)$ şi $E(3,1)$.

a. Rulaţi manual două iteraţii ale algoritmului 2-means pe acest set de date,[844] pornind de la următoarele poziţii iniţiale ale centroizilor: $(-1,0)$ şi $(3,1)$. Veţi folosi distanţa euclidiană. La fiecare iteraţie indicaţi cum sunt asignate punctele la clustere şi care sunt poziţiile actualizate ale centroizilor. A convers algoritmul după ce au fost efectuate cele două iteraţii?

[842] Şi cu atât mai puţin pentru similaritate "complete-linkage" sau "average-linkage".
[843] La problema aceasta, precum şi la fiecare dintre problemele următoare, dacă nu se specifică altfel, se va folosi forma algoritmului K-means din cartea *Foundations of Statistical Natural Language Processing* (capitolul 14) de C. Manning and H. Schütze, editura MIT Press, 2002.
[844] Pentru pseudo-codul algoritmului K-means, puteţi vedea enunţul problemei 12.

b. Definim

$$J(\mu^{(t)}) = \sum_{x^{(i)} \in \{A,B,C,D,E\}} \|x^{(i)} - \mu^{(t)}(x^{(i)})\|^2$$

unde $\mu^{(t)}$ desemnează ansamblul centroizilor la momentul / iteraţia t, iar $\mu^{(t)}(x^{(i)})$ este centroidul clusterului la care este asignată instanţa $x^{(i)}$ la iteraţia t. Simbolul $\| \ \|$ reprezintă norma euclidiană. Pentru un vector oarecare $x \overset{not.}{=} (x_1, \ldots, x_d) \in \mathbb{R}^d$, avem $\|x\|^2 \overset{def.}{=} x \cdot x \overset{def.}{=} x_1^2 + \ldots + x_d^2 \in \mathbb{R}^+$. Arătaţi pur şi simplu prin calcul numeric că pentru $t = 1$ avem

$$J(\mu^{(t)}) \leq J(\mu^{(t-1)}).$$

(Am desemnat cu $t = 0$ iteraţia iniţială.)

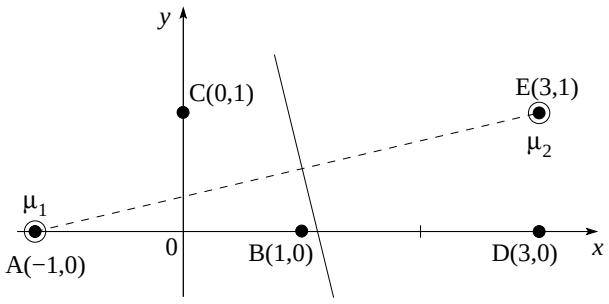
Răspuns:

a. Conform enunţului, centroizii iniţiali sunt chiar două dintre punctele considerate: $\mu_1 \equiv A(-1,0)$ şi $\mu_2 \equiv E(3,1)$. Asignarea celorlalte puncte la clustere se poate face în manieră *analitică*, calculând distanţele dintre puncte şi centroizi:

$$\begin{aligned} d(B,\mu_1) = 2 \text{ şi } d(B,\mu_2) = \sqrt{5} &\Rightarrow B \text{ se asignează clusterului cu centroidul } \mu_1 \\ d(C,\mu_1) = \sqrt{2} \text{ şi } d(C,\mu_2) = 3 &\Rightarrow C \text{ se asignează clusterului cu centroidul } \mu_1 \\ d(D,\mu_1) = 4 \text{ şi } d(D,\mu_2) = 1 &\Rightarrow D \text{ se asignează clusterului cu centroidul } \mu_2 \end{aligned}$$

Alternativ, adică *geometric*, se reprezintă punctele date şi centroizii iniţiali în planul euclidian, după care se trasează mediatoarea segmentului $\mu_1\mu_2$. În figura de mai jos am reprezentat instanţele cu un cerculeţ umplut (•), iar centroizii cu un cerculeţ simplu. Cele două clustere sunt separate de către mediatoarea segmentului care uneşte cei doi centroizi.

Se observă că punctele B şi C sunt situate de aceeaşi parte a mediatoarei ca şi centroidul μ_1, deci rezultă că B şi C vor aparţine clusterului cu centroidul μ_1. Similar, punctul D va aparţine clusterului cu centroidul μ_2.

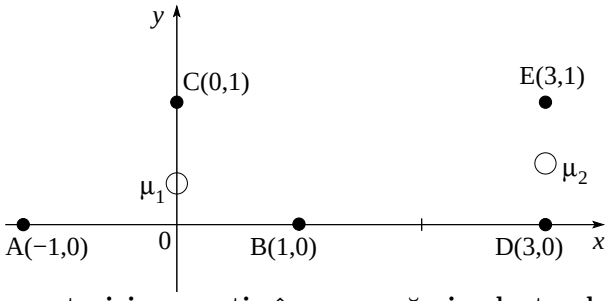


Având aşadar componenţa iniţială a clusterelor $\{A(-1,0), B(1,0), C(0,1)\}$ şi, respectiv, $\{D(3,0), E(3,1)\}$, la următoarea iteraţie a algoritmului se calculează noile poziţii ale centroizilor (vedeţi calculul alăturat).

$$\left.\begin{aligned} x_{\mu_1} &= \frac{x_A + x_B + x_C}{3} = 0 \\ y_{\mu_1} &= \frac{y_A + y_B + y_C}{3} = \frac{1}{3} \end{aligned}\right\} \Rightarrow \mu_1(0, \frac{1}{3})$$

$$\left.\begin{aligned} x_{\mu_2} &= \frac{x_D + x_E}{2} = 3 \\ y_{\mu_2} &= \frac{y_D + y_E}{2} = \frac{1}{2} \end{aligned}\right\} \Rightarrow \mu_2(3, \frac{1}{2})$$

Acum, reprezentarea grafică este cea din figura alăturată.



Apoi se re-asignează punctele la centroizi, operaţie în urma căreia clusterele se pot modifica:

$$d(A,\mu_1)=\frac{\sqrt{10}}{3} \text{ şi } d(A,\mu_2)=\frac{\sqrt{65}}{2} \Rightarrow A \text{ se asignează la centroidul } \mu_1$$

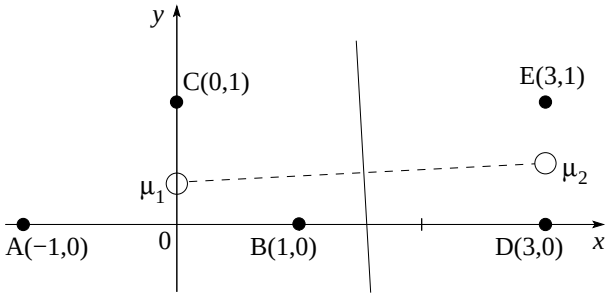
$$d(B,\mu_1)=\frac{\sqrt{10}}{3} \text{ şi } d(B,\mu_2)=\frac{\sqrt{17}}{2} \Rightarrow B \text{ se asignează la centroidul } \mu_1$$

$$d(C,\mu_1)=\frac{2}{3} \text{ şi } d(C,\mu_2)=\frac{\sqrt{37}}{2} \Rightarrow C \text{ se asignează la centroidul } \mu_1$$

$$d(D,\mu_1)=\frac{\sqrt{82}}{3} \text{ şi } d(D,\mu_2)=\frac{1}{2} \Rightarrow D \text{ se asignează la centroidul } \mu_2$$

$$d(E,\mu_1)=\frac{\sqrt{85}}{3} \text{ şi } d(E,\mu_2)=\frac{1}{2} \Rightarrow E \text{ se asignează la centroidul } \mu_2$$

Se observă deci analitic (dar şi grafic, vedeţi figura alăturată) că după această (a doua) iteraţie avem aceeaşi componenţă a clusterelor (ca şi la finalul precedentei iteraţii), şi anume: $\{$A(-1,0)$, B(1,0), C(0,1)\}$ şi $\{D(3,0), E(3,1)\}$. Prin urmare, centroizii rămân $\mu_1(0,1/3)$ şi $\mu_2(3,1/2)$.



Putem concluziona că după aceste două iteraţii algoritmul K-means a convers.

***Observaţie importantă*:**

Atât în această problemă cât şi în cele următoare, ne-am străduit să elaborăm soluţia algoritmului K-means atât bazat pe calculul distanţelor (am numit-o deci soluţia *analitică*), cât şi bazat pe reprezentarea datelor în planul euclidian (am numit-o soluţia *geometrică*). Aceasta din urmă are avantajul că oferă un suport vizual care ajută studentul să înţeleagă lucrurile într-o manieră mai intuitivă.

La implementare, cele două tipuri de soluţii apelează la procedee de calcul care diferă parţial.[845]

În condiţii de seminar sau de examen, cele două procedee pot fi combinate — iar acolo unde detaliile sunt evidente, calculele pot fi lăsate de o parte dacă nu se specifică altfel —, datorită restricţiilor de timp impuse.

[845] Este mai simplu de realizat implementarea bazată pe metoda analitică. Însă pentru vizualizarea rezultatelor intermediare sau finale — aşa cum se procedează la rezolvarea majorităţii problemelor din această secţiune a prezentului capitol — se utilizează şi componente ale variantei geometrice.

b. Vom reprezenta punctele / instanţele de clusterizat sub formă de vector-coloană, cu două componente (una pentru abscisa x şi cealaltă pentru ordonata y).

$$\begin{aligned}
J(\mu^{(0)}) &= \left\{\left[\begin{pmatrix}-1\\0\end{pmatrix}-\begin{pmatrix}-1\\0\end{pmatrix}\right]^2+\left[\begin{pmatrix}1\\0\end{pmatrix}-\begin{pmatrix}-1\\0\end{pmatrix}\right]^2+\left[\begin{pmatrix}0\\1\end{pmatrix}-\begin{pmatrix}-1\\0\end{pmatrix}\right]^2\right\} \\
&\quad+\left\{\left[\begin{pmatrix}3\\0\end{pmatrix}-\begin{pmatrix}3\\1\end{pmatrix}\right]^2+\left[\begin{pmatrix}3\\1\end{pmatrix}-\begin{pmatrix}3\\1\end{pmatrix}\right]^2\right\} \\
&= \left\{\begin{pmatrix}0\\0\end{pmatrix}^2+\begin{pmatrix}2\\0\end{pmatrix}^2+\begin{pmatrix}1\\1\end{pmatrix}^2\right\}+\left\{\begin{pmatrix}0\\-1\end{pmatrix}^2+\begin{pmatrix}0\\0\end{pmatrix}^2\right\} \\
&= 0+4+2+1+0=7. \\
J(\mu^{(1)}) &= \left\{\left[\begin{pmatrix}-1\\0\end{pmatrix}-\begin{pmatrix}0\\1/3\end{pmatrix}\right]^2+\left[\begin{pmatrix}1\\0\end{pmatrix}-\begin{pmatrix}0\\1/3\end{pmatrix}\right]^2+\left[\begin{pmatrix}0\\1\end{pmatrix}-\begin{pmatrix}0\\1/3\end{pmatrix}\right]^2\right\} \\
&\quad+\left\{\left[\begin{pmatrix}3\\0\end{pmatrix}-\begin{pmatrix}3\\1/2\end{pmatrix}\right]^2+\left[\begin{pmatrix}3\\1\end{pmatrix}-\begin{pmatrix}3\\1/2\end{pmatrix}\right]^2\right\} \\
&= \left\{\begin{pmatrix}-1\\-1/3\end{pmatrix}^2+\begin{pmatrix}1\\-1/3\end{pmatrix}^2+\begin{pmatrix}0\\2/3\end{pmatrix}^2\right\}+\left\{\begin{pmatrix}0\\-1/2\end{pmatrix}^2+\begin{pmatrix}0\\1/2\end{pmatrix}^2\right\} \\
&= \frac{10}{9}+\frac{10}{9}+\frac{4}{9}+\frac{1}{4}+\frac{1}{4}=\frac{24}{9}+\frac{1}{2}=\frac{8}{3}+\frac{1}{2}=\frac{19}{6}.
\end{aligned}$$

Prin urmare, se verifică inegalitatea $J(\mu^{(1)}) \leq J(\mu^{(0)})$.

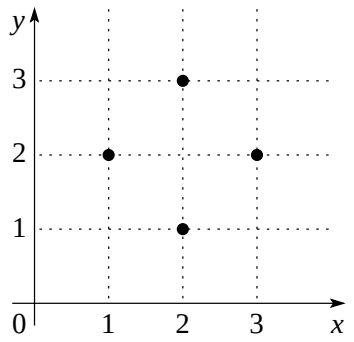
***Observaţie*:**

Proprietatea de descreştere [nu neapărat strictă] a valorilor funcţiei J în cursul execuţiei algoritmului K-means (la orice iteraţie şi pe orice set de date de clusterizat) va face obiectul problemei 12.

8. **(Algoritmul K-means: exemplificare pentru alegerea centroizilor iniţiali astfel încât clusterele obţinute să satisfacă anumite restricţii))**

CMU, 2009 spring, Ziv Bar-Joseph, HW5, pr. 2.2

Considerăm 4 puncte dispuse în spaţiu aşa cum se arată în figura alăturată.



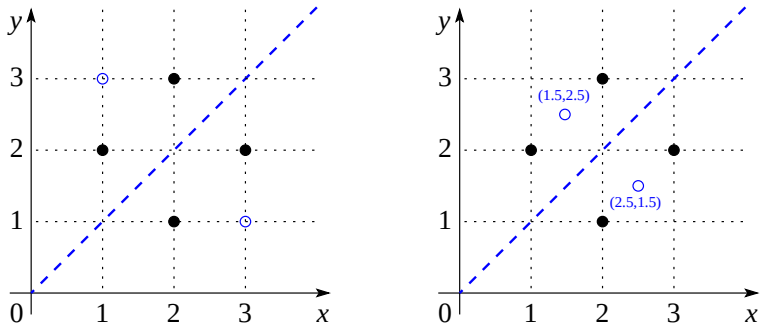
a. Aplicând algoritmul K-means pentru $K = 2$, alegeţi centroizii iniţiali în aşa fel încât să obţineţi o clusterizare în care fiecare cluster să conţină câte 2 elemente.

b. Aceeaşi cerinţă, de data aceasta pentru a obţine 2 clustere dintre care unul să conţină un element, iar celălalt 3 elemente.

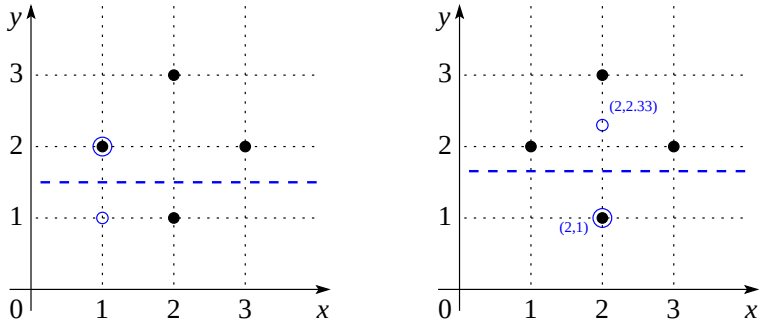
***Observaţie*. Se va folosi distanţa euclidiană.**

Răspuns:

a. Pentru a se obţine două clustere de câte două elemente, se pot alege centroizii iniţiali $\mu_1 = (1, 3)$ şi $\mu_2 = (3, 1)$. După prima iteraţie (care se dovedeşte a fi şi ultima), se vor obţine centroizii $\mu_1' = (1.5, 2.5)$ şi $\mu_2' = (2.5, 1.5)$, ca în figura de mai jos. Instanţele au fost reprezentate prin cerculeţe umplute (•), iar simbolurile ∘ marchează poziţiilor centroizilor. Liniile punctate constituie separatorii celor două clustere.



b. Pentru a se obţine două clustere cu 1 şi respectiv 3 elemente, se pot alege centroizii iniţiali $\mu_1 = (1, 1)$ şi $\mu_2 = (1, 2)$. După o primă iteraţie, se vor obţine centroizii $\mu_1' = (2, 1)$ şi $\mu_2' = (2, 2.(3))$, ca în figura de mai jos, după care algoritmul converge.



9. **(Algoritmul K-means: aplicare pe date din $\mathbb{R}^2$)**

CMU, 2009 spring, Ziv Bar-Joseph, final exam, pr. 1.8

Punctul	x	y
P_1	1.90	0.97
P_2	1.76	0.84
P_3	2.32	1.63
P_4	2.31	2.09
P_5	1.14	2.11
P_6	5.02	3.02
P_7	5.74	3.84
P_8	2.25	3.47
P_9	4.71	3.60
P_{10}	3.17	4.96

Se dă un set de puncte din planul euclidian, conform tabelului alăturat.

Vă cerem să rulaţi manual algoritmul K-means pentru a identifica două clustere în acest set de date. Centroizii iniţiali sunt chiar punctele P_1 şi P_{10}. Se folosesc distanţe euclidiene.

Care este compoziţia clusterelor după prima iteraţie? Dar la convergenţa algoritmului?

Răspuns:

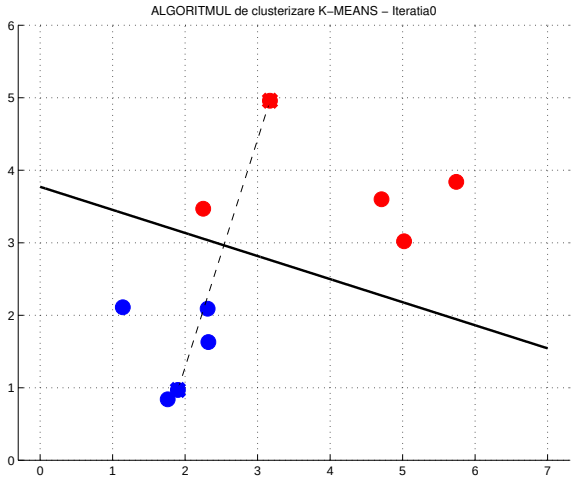
La prima iteraţie a algoritmului K-means, fiecare punct P_i va fi asignat unuia dintre cele două clustere (C_1 şi C_2), în funcţie de distanţele dintre punctul respectiv şi centroizii iniţiali.

Construim un tabel cu aceste distanţe şi asignările deduse din relaţia de ordine dintre ele. Vom nota cu $d(P_i, P_j)$ distanţa de la punctul P_i la punctul P_j.

P_i	$d(P_i, P_1)$	$d(P_i, P_{10})$	$d(P_i, P_1) \vdots d(P_i, P_{10})$	**Clusterul**
P_1	0	4.18	$<$	C_1
P_2	0.19	4.35	$<$	C_1
P_3	0.78	3.43	$<$	C_1
P_4	1.19	2.99	$<$	C_1
P_5	1.37	3.49	$<$	C_1
P_6	3.73	2.68	$>$	C_2
P_7	4.97	2.80	$>$	C_2
P_8	2.52	1.75	$>$	C_2
P_9	3.84	2.05	$>$	C_2
P_{10}	4.18	0	$>$	C_2

Aşadar, la finalul primei iteraţii obţinem clusterele $C_1 = \{P_1, P_2, P_3, P_4, P_5\}$ şi $C_2 = \{P_6, P_7, P_8, P_9, P_{10}\}$.

Acelaşi rezultat se poate obţine şi procedând *geometric*. Pentru aceasta, în planul euclidian bidimensional trasăm mediatoarea segmentului P_1P_{10} şi apoi analizăm poziţia fiecărui punct dat în raport cu această mediatoare. Rezultatul este arătat în figura alăturată.



Poziţiile centroizilor (marcate pe grafic prin caracterul $\times$) se recalculează astfel:

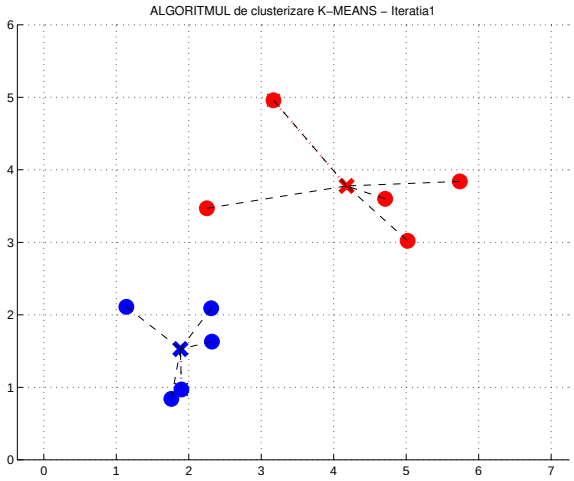
$$x_{\mu_1} = \frac{x_{P_1} + x_{P_2} + x_{P_3} + x_{P_4} + x_{P_5}}{5} = 1.886$$
$$y_{\mu_1} = \frac{y_{P_1} + y_{P_2} + y_{P_3} + y_{P_4} + y_{P_5}}{5} = 1.528$$

Prin urmare, $\mu_1 = (1.886, 1.528)$. Analog, obţinem $\mu_2 = (4.178, 3.778)$.

Dacă am decis să procedăm *analitic*, în continuare vom calcula distanţele de la fiecare punct P_i la centroizii μ_1 şi respectiv μ_2.

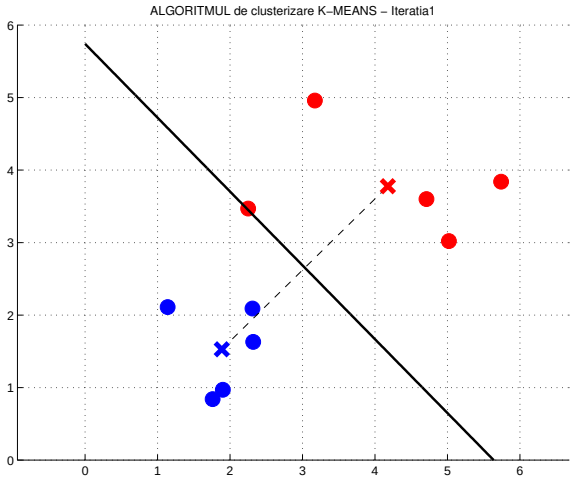
În figura alăturată am marcat noile poziţii ale centroizilor (μ_1 şi μ_2), precum şi asignarea fiecăruia dintre punctele P_i la cel mai apropiat centroid, în funcţie de distanţe.

Aşadar, punctele $P_1, \ldots, P_5$ formează — şi de data aceasta! — primul cluster (cel cu centroidul μ_1), iar restul punctelor, $P_6, \ldots, P_{10}$, constituie cel de-al doilea cluster (cel cu centroidul μ_2).



Alternativ, adică în ipoteza că am optat pentru o *rezolvare geometrică*, am observa că punctul $P_8(2.25, 3.47)$ are o poziţie „la limită“ faţă de mediatoarea segmentului care uneşte centroizii μ_1 şi μ_2, aşa cum se observă din figura de mai jos.

Pentru a decide cărui cluster îi va aparţine acest punct, fie calculăm $d^2(P_8, \mu_1)$ şi $d^2(P_8, \mu_2)$ — şi vom obţine valorile 3.904 şi respectiv 3.812 —, fie comparăm semnul expresiei $f(2.52, 1.75)$ cu cel al lui $f(x_{\mu_1}, y_{\mu_1})$ (sau $f(x_{\mu_2}, y_{\mu_2})$), unde f este funcţia din membrul stâng al ecuaţiei $f(x, y) = 0$ care defineşte mediatoarea segmentului determinat de punctele μ_1 şi μ_2.



În ambele variante se ajunge la aceeaşi concluzie: punctul P_8 va aparţine clusterului cu centroidul μ_2. (Valoarea expresiei $f(2.52, 1.75)$ are acelaşi semn cu $f(x_{\mu_2}, y_{\mu_2})$).)

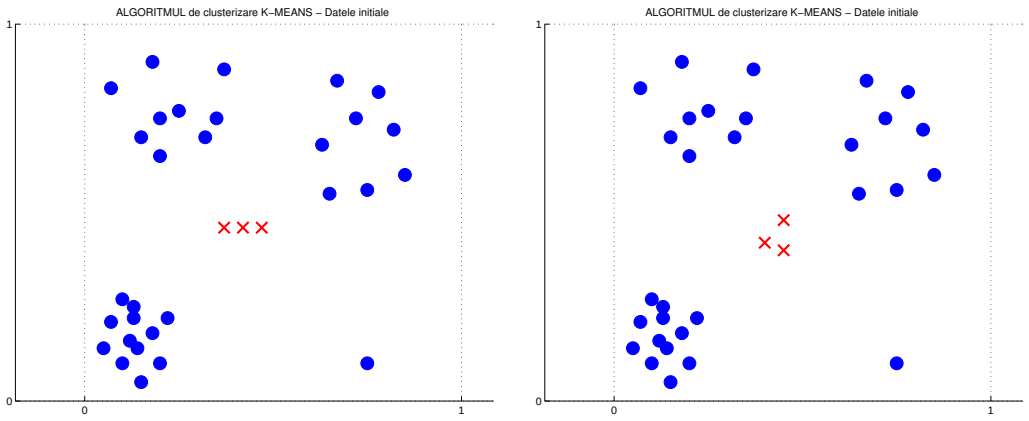
Clusterele finale sunt deci: $\{P_1, P_2, P_3, P_4, P_5\}$ şi $\{P_6, P_7, P_8, P_9, P_{10}\}$.

10. **(Algoritmul K-means: aplicare în $\mathbb{R}^2$; compararea rezultatelor obţinute în cazul a două iniţializări diferite ale centroizilor)**

■ • ***CMU, 2006 spring, Carlos Guestrin, HW5, pr. 1***

Se consideră setul de date reprezentat în cele două figuri de mai jos. Sunt date două iniţializări posibile ale centroizilor, câte una în fiecare din cele două figuri. Instanţele sunt reprezentate de cerculeţe umplute (•), iar simbolurile

× corespund poziţiilor iniţiale ale centroizilor clusterelor.[846]



a-b. Executaţi / aplicaţi algoritmul K-means separat pentru fiecare dintre cele două iniţializări ale centroizilor, arătând la fiecare iteraţie cum evoluează centroizii şi separatorii, până la convergenţă.

***Observaţie*: La execuţia algoritmului se va considera că atunci când un centroid nu are puncte asignate lui, atunci el va rămâne pe loc la iteraţia respectivă.**

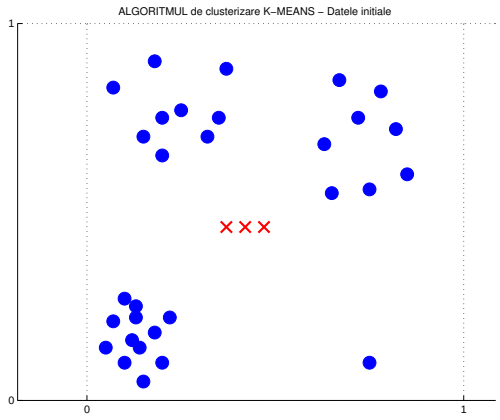
c. Având în vedere rezultatele obţinute în cele două cazuri, ce puteţi spune despre comportamentul algoritmului K-means?

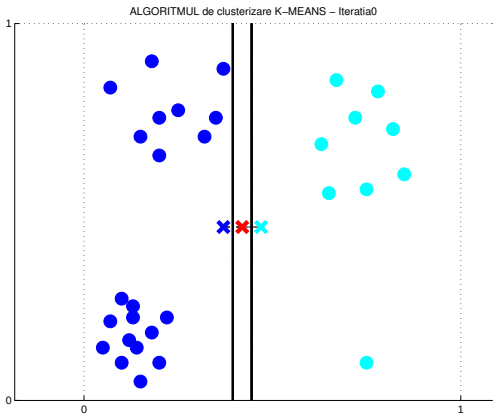
Răspuns:

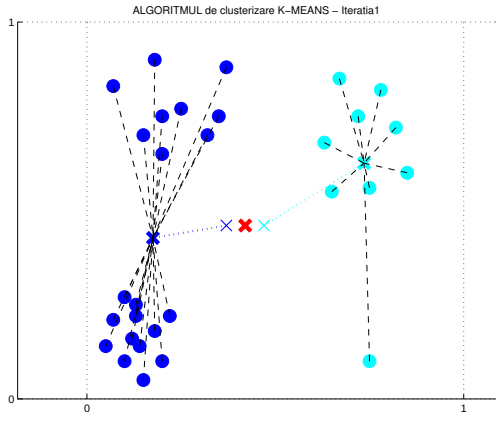
a. Pentru prima iniţializare a centroizilor, algoritmul K-means parcurge următoarele iteraţii:[847]

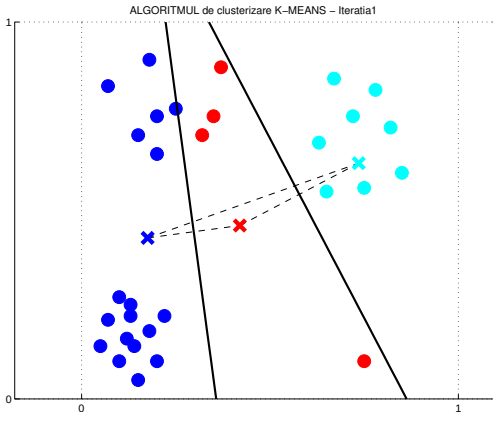
[846] Coordonatele exacte ale instanţelor, precum şi cele ale centroizilor vă sunt puse la dispoziţie în următoarele fişiere, depuse pe site-ul acestei cărţi:
http://profs.info.uaic.ro/∼ciortuz/ML.ex-book/res/CMU.2006s.HW5.pr1.cl.dat,
http://profs.info.uaic.ro/∼ciortuz/ML.ex-book/res/CMU.2006s.HW5.pr1.a.init.dat,
http://profs.info.uaic.ro/∼ciortuz/ML.ex-book/res/CMU.2006s.HW5.pr1.b.init.dat.

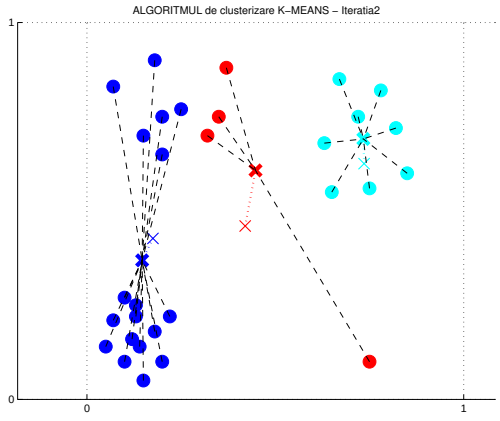
[847] La fiecare iteraţie, în afară de cea iniţială (care a fost numerotată cu 0), am desenat în graficul din partea stângă noile poziţii ale centroizilor — vedeţi simbolul × mai îngroşat, spre deosebire de vechea lui poziţie, care a fost redată neîngroşat pentru a sugera mişcarea —, iar în graficul din partea dreaptă am indicat noua componenţă a clusterelor, în funcţie de poziţiile actuale ale separatorilor (mediatoarele segmentelor care unesc centroizii).

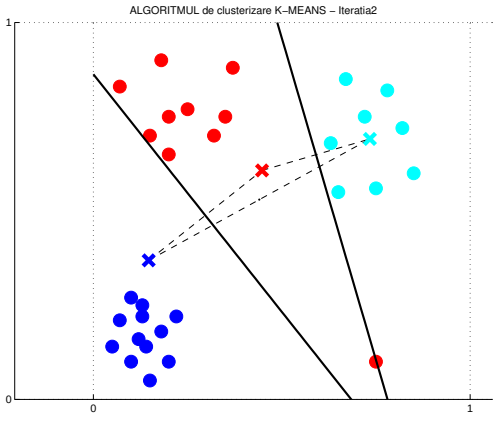

ALGORITMUL de clusterizare K–MEANS – Datele initiale
1
0
0
1

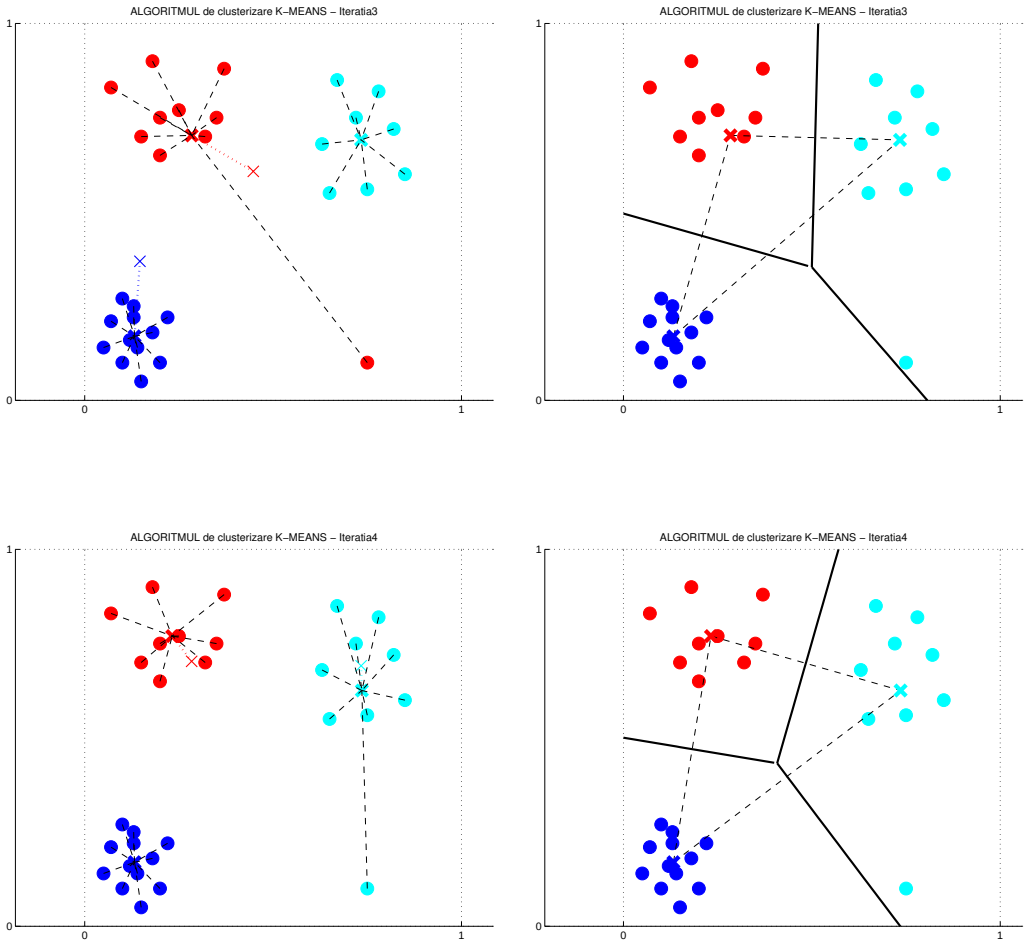

ALGORITMUL de clusterizare K–MEANS – Iteratia0
1
0
0
1


ALGORITMUL de clusterizare K–MEANS – Iteratia1
1
0
0
1

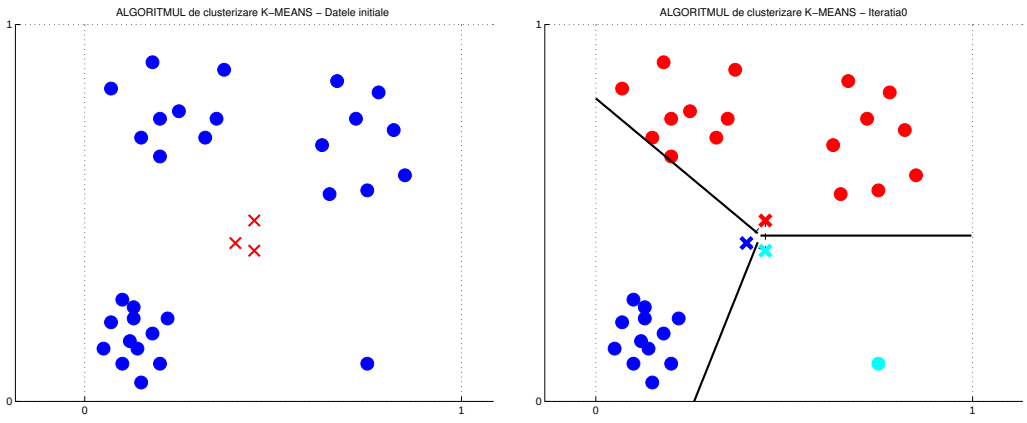

ALGORITMUL de clusterizare K–MEANS – Iteratia1
1
0
0
1

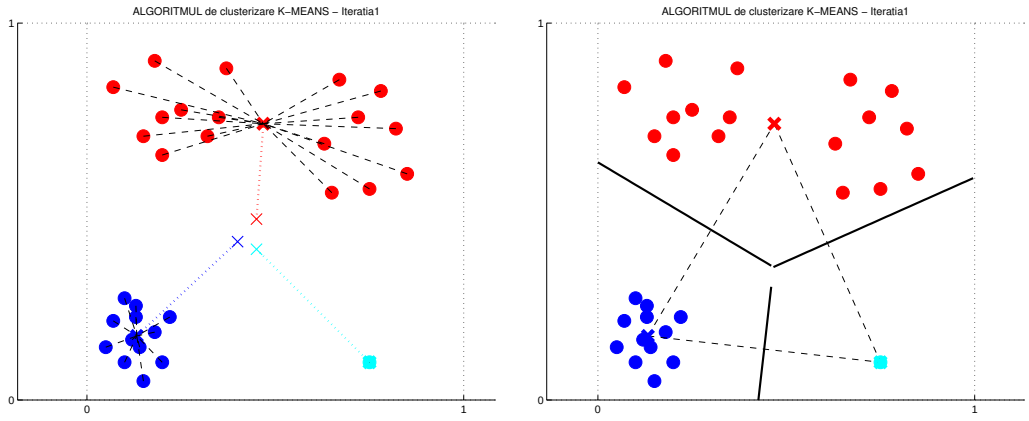

ALGORITMUL de clusterizare K–MEANS – Iteratia2
1
0
0
1


ALGORITMUL de clusterizare K–MEANS – Iteratia2
1
0
0
1



b. Pentru a doua iniţializare a centroizilor, algoritmul K-means parcurge următoarele iteraţii:





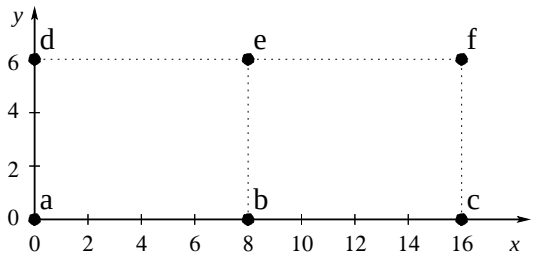
c. Este evident din acest exerciţiu că rezultatul algoritmului K-means depinde de poziţionarea iniţială a centroizilor. În cazul iniţializării de la punctul a au fost necesare 4 iteraţii până a se ajunge la convergenţă, pe când la punctul b algoritmul a convers după doar o iteraţie.

Mai este încă ceva important *de remarcat*: faptul că la prima variantă de iniţializare, punctul din dreapta jos, care este un *outlier* (rom., excepţie, caz particular, aberaţie) este până la urmă asociat clusterului format de punctele din partea dreaptă (sus), în vreme ce la cea de-a doua variantă de iniţializare el constituie un cluster aparte / "singleton", obligând în mod indirect grupările de puncte din stânga-sus şi dreapta-sus să formeze împreună un singur cluster. Detecţia outlier-elor este un capitol important din învăţarea automată. Ignorarea lor poate conduce la rezultate eronate sau chiar aberante ale modelărilor.

11. **(Algoritmul K-means: aplicare în $\mathbb{R}^2$; realizarea corespondenţei dintre K-configuraţia [obţinută la o iteraţie] curentă a algoritmului şi una sau mai multe K-configuraţii de start)**

prelucrare făcută de Liviu Ciortuz, după
• ○ *CMU, 2003 fall, T. Mitchell, A. Moore, final exam, pr. 9*

Considerăm mulţimea S formată de următoarele 6 puncte din plan: $a = (0, 0)$, $b = (8, 0)$, $c = (16, 0)$, $d = (0, 6)$, $e = (8, 6)$ şi $f = (16, 6)$. Reprezentarea datelor în planul euclidian este dată în figura alăturată. Se rulează algoritmul $K-$means, unde $K = 3$, şi se foloseşte distanţa euclidiană.



Introducem următoarele două *noţiuni*:

- **K-*partiţie*: desemnează o partiţionare a mulţimii S în K submulţimi nevide. Spre exemplu, $\{a, b, e\}, \{c, d\}, \{f\}$ este o 3-partiţie;**

• ***K-configuraţie de start*: desemnează o submulţime de K puncte din mulţimea S, care vor juca rolul de centroizi iniţiali. Iată un exemplu de 3-configuraţie de start, care generează una dintre 3-partiţiile considerate mai jos (şi anume, $\{a,b\},\{d,e\},\{c,f\}$): $\{a,d,c\}$.**

În mod evident, orice K-partiţie induce o mulţime de K centroizi. O *$K-$partiţie* este numită *stabilă* dacă după execuţia unei noi iteraţii a algoritmului $K-$means, partiţia respectivă rămâne neschimbată.

a. Câte 3-configuraţii de start există pentru mulţimea S dată mai sus?

b. Completaţi tabelul următor:

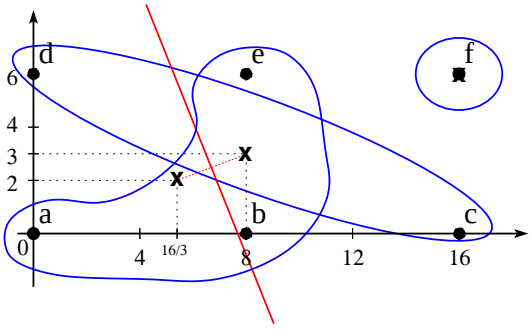
3-partiţie	Stabilitate?	Un exemplu de 3-configuraţie de start care poate determina 3-partiţia din prima coloană, după 0 sau mai multe iteraţii ale algoritmului K-means. (Dacă nu există o asemenea configuraţie, se va trece $-$.)	Numărul de 3-configuraţii de start care pot genera această 3-partiţie
$\{a,b,e\},\{c,d\},\{f\}$	**Nu**	$-$	0
$\{a,b\},\{d,e\},\{c,f\}$	**Da**	$\{b,c,e\}$	4
$\{a,d\},\{b,e\},\{c,f\}$			
$\{a\},\{d\},\{b,c,e,f\}$			
$\{a,b\},\{d\},\{c,e,f\}$			
$\{a,b,d\},\{c\},\{e,f\}$			

Răspuns:

a. Pentru a forma o 3-configuraţie de start, se aleg oricare 3 puncte distincte din mulţimea S (care are 6 puncte), deci numărul acestor 3-configuraţii este $C_6^3 = \dfrac{6!}{3!\cdot 3!} = \dfrac{6\cdot 5\cdot 4}{1\cdot 2\cdot 3} = 20$.

b. Vom analiza pe rând fiecare dintre cele şase cazuri din tabel, inclusiv primele două cazuri pentru care avem deja anumite informaţii date în tabel, dar care merită a fi discutate. În figurile următoare, centroizii iniţiali sunt reprezentaţi cu cerculeţe, iar centroizii actuali cu ×.

Se poate verifica uşor faptul că în *cazul 1*, 3-partiţia dată nu este stabilă. Ea „evoluează“ în partiţia $\{a,d\}$, $\{b,e\}$, $\{c,f\}$. Însă, ţinând cont de faptul că poziţiile iniţiale ale centroizilor sunt alese doar din mulţimea S, vom putea arăta mai jos că *nu* există nicio 3-configuraţie de start care să rezulte în 3-partiţia dată.

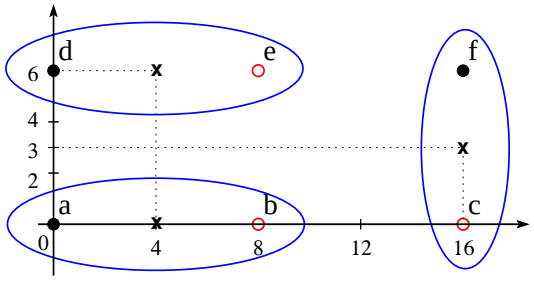


Pentru algoritmul K-means, atunci când se foloseşte distanţa euclidiană, are loc următoarea *proprietate*: la finalul oricărei iteraţii a algoritmului (deci şi la terminarea algoritmului), orice două clustere sunt separabile liniar (şi anume, cu ajutorul mediatoarei segmentului care uneşte centroizii acestor două clustere, dacă instanţele sunt din $\mathbb{R}^2$). Demonstrarea acestei proprietăţi

este imediată, ţinând cont de modul în care sunt asignate instanţele la centroizi. (Vedeţi pasul al doilea al corpului iterativ al algoritmului K-means, de exemplu în formularea din enunţul problemei 12.)

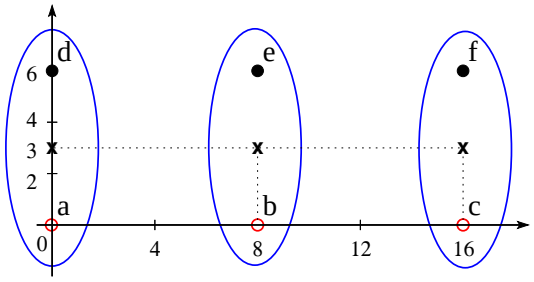
În condiţiile de faţă (*cazul 1*), se observă imediat că mediatoarea segmentului care uneşte centroizii clusterelor $\{a, b, e\}$ şi $\{c, d\}$ le separă pe fiecare în câte două submulţimi nevide, în loc să situeze un cluster de o parte a ei (adică, a mediatoarei) şi celălalt cluster de cealaltă parte.

În *cazul 2*, se observă imediat că în afară de 3-configuraţia de start dată ($\{b, c, e\}$, deja stabilă), mai există 3 alte asemenea configuraţii de start care generează 3-partiţia considerată ($\{a, b\}, \{d, e\}, \{c, f\}$): $\{b, e, f\}$, $\{a, d, c\}$ şi $\{a, d, f\}$.

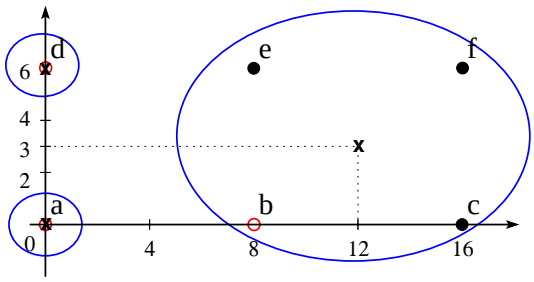


***Observaţie*: În acest caz,[848] am presupus că este permisă libertatea de asociere a instanţelor care sunt egal distanţate faţă de doi centroizi. Aceeaşi presupoziţie o vom folosi şi în continuare, acolo unde va fi nevoie (*cazul 5* şi *cazul 6*).**

În *cazul 3*, se observă că 3-partiţia dată ($\{a, d\}, \{b, e\}, \{c, f\}$) este stabilă şi sunt uşor de enumerat cele 8 configuraţii de start care „termină“ în această 3-partiţie a mulţimii S: $\{a, b, c\}$, $\{a, b, f\}$, $\{a, e, c\}$, $\{a, e, f\}$, $\{d, b, c\}$, $\{d, b, f\}$, $\{d, e, c\}$, $\{d, e, f\}$.



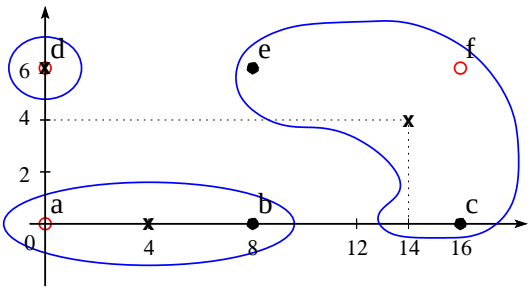
În *cazul 4*, unde 3-partiţia de start $(\{a\}, \{d\}, \{b, c, e, f\})$ este de asemenea stabilă, este imediat că instanţele a şi d trebuie să facă parte în mod obligatoriu din configuraţia (eventual, configuraţiile) de start. Într-adevăr, dacă măcar una dintre instanţele a şi d ar lipsi din configuraţia de start, se observă că algoritmul K-means nu ar putea să ajungă în 3-partiţia dată.



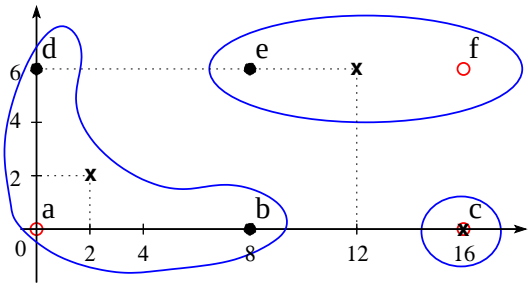
Pe de altă parte, c şi f nu pot să apară în configuraţia de start (din exact acelaşi motiv). Rămân de analizat poziţiile b şi e; se verifică uşor că acestea pot completa (pe rând) 3-configuraţiile de start, care sunt prin urmare $\{a, b, d\}$ şi $\{a, d, e\}$.

[848] Vedeţi configuraţiile de start $\{b, c, e\}$ şi $\{b, e, f\}$ menţionate mai sus.

În *cazul 5*, 3-partiţia dată ($\{a,b\},\{d\},\{c,e,f\}$) este stabilă. Se observă că pe de o parte instanţa d trebuie să facă parte din orice configuraţie de start care ar putea să genereze 3-partiţia dată, iar pe de altă parte instanţele b, c şi e nu pot face parte din nicio astfel de configuraţie de start. Celelalte instanţe din S formează singura 3-configuraţie de start validă: $\{a,d,f\}$.



***Cazul 6* este similar *cazului 5* (datorită simetriei celor două 3-partiţii date faţă de punctul $(8;3)$), aşadar singura 3-configuraţie de start posibilă este $\{a,c,f\}$.**



În final, centralizăm în tabel rezultatele obţinute:

Caz	3-partiţie	Stabilitate?	Un exemplu de 3-configuraţie de start care poate determina 3-partiţia dată, după 0 sau mai multe iteraţii ale algoritmului K-means.	Numărul de 3-configuraţii de start care generează această 3-partiţie
1	$\{a,b,e\},\{c,d\},\{f\}$	**Nu**	–	0
2	$\{a,b\},\{d,e\},\{c,f\}$	**Da**	$\{b,c,e\}$	4
3	$\{a,d\},\{b,e\},\{c,f\}$	**Da**	$\{a,b,c\}$	8
4	$\{a\},\{d\},\{b,c,e,f\}$	**Da**	$\{a,b,d\}$	2
5	$\{a,b\},\{d\},\{c,e,f\}$	**Da**	$\{a,d,f\}$	1
6	$\{a,b,d\},\{c\},\{e,f\}$	**Da**	$\{a,c,f\}$	1

***Observaţie*: Restul de 5 configuraţii de start — în total, în varianta completă (vedeţi coloana a cincea) a tabelului de mai sus sunt 15 configuraţii de start, dintre care una, $\{a,d,f\}$, apare de două ori — conduc la alte trei 3-partiţii decât cele menţionate în coloana a doua a tabelului.**

***Observaţie importantă*: Este util de precizat acum — odată ce au fost „fixate" noţiunile de K-partiţie şi K-configuraţie — că algoritmul K-means poate fi văzut ca un *algoritm de căutare*. K-means porneşte de la o K-configuraţie (aleasă eventual în mod arbitrar) şi o îmbunătăţeşte în mod succesiv, prin intermediul K-partiţiilor corespunzătoare, sau invers. *Spaţiul de căutare* corespunzător algoritmului K-means este mulţimea tuturor K-partiţiilor care se pot forma peste X, setul de instanţe de clusterizat.**

12. **(K-means, ca algoritm de optimizare a criteriului *coeziunii intra-clustere* („suma celor mai mici pătrate“))**

prelucrare de Liviu Ciortuz, după
■ ***CMU, 2009 spring, Ziv Bar-Joseph, HW5, pr. 2.1***

Vă reamintim algoritmul K-means, datorat lui Lloyd,[849] predat la curs:

***Input*:** $x_1, \ldots, x_n \in \mathbb{R}^d$ şi $K \in \mathbb{N}^*$.

***Output*:** o anumită K-partiţie pentru $\{x_1, \ldots, x_n\}$, adică o descompunere a acestei mulţimi într-o colecţie de K mulţimi disjuncte (care nu sunt în mod neapărat nevide).

***Procedură*:**

[Iniţializare / Iteraţia 0:] $t \leftarrow 0$;
se fixează în mod arbitrar $\mu_1^0, \ldots, \mu_K^0$, centroizii iniţiali ai clusterelor, şi
se asignează fiecare instanţă x_i la centroidul cel mai apropiat, formând astfel clusterele $C_1^0, \ldots, C_K^0$.

[Corpul iterativ:] Se execută iteraţia $++\, t$:

***Pasul 1*:** se calculează noile poziţii ale centroizilor:[850]
$\mu_j^t = \frac{1}{|C_j^{t-1}|} \sum_{x_i \in C_j^{t-1}} x_i$ pentru $j = \overline{1, K}$;

***Pasul 2*:**
se reasignează fiecare x_i la [clusterul cu] centroidul cel mai apropiat, adică se stabileşte noua componenţă a clusterelor la iteraţia t: $C_1^t, \ldots, C_K^t$;

[Terminare:] până când o anumită condiţie este îndeplinită
(de exemplu: până când poziţiile centroizilor — sau: componenţa clusterelor — nu se mai modifică de la o iteraţie la alta).

a. Demonstraţi că, de la o iteraţie la alta, algoritmul K-means măreşte *coeziunea de ansamblu* a clusterelor. Veţi proceda astfel: considerând funcţia

$$J(C^t, \mu^t) \overset{def.}{=} \sum_{i=1}^{n} ||x_i - \mu^t_{C^t(x_i)}||^2 \overset{def.}{=} \sum_{i=1}^{n} \left(x_i - \mu^t_{C^t(x_i)}\right) \cdot \left(x_i - \mu^t_{C^t(x_i)}\right),$$

unde

$C^t = (C_1^t, C_2^t, \ldots, C_K^t)$ **este colecţia de clustere (i.e., K-partiţia) la momentul** t,
$\mu^t = (\mu_1^t, \mu_2^t, \ldots, \mu_K^t)$ **este colecţia de centroizi ai clusterelor (K-configuraţia) la momentul** t,
$C^t(x_i)$ **desemnează clusterul la care este asignat elementul** x_i **la iteraţia** t,
operatorul $\cdot$ **desemnează produsul scalar al vectorilor din** $\mathbb{R}^d$,

arătaţi că $J(C^t, \mu^t) \geq J(C^{t+1}, \mu^{t+1})$ **pentru orice** t**.**

***Indicaţie*:** Inegalitatea de mai sus rezultă din două inegalităţi (care corespund paşilor 1 şi 2 de la iteraţia t):

$$J(C^t, \mu^t) \overset{(1)}{\geq} J(C^t, \mu^{t+1}) \overset{(2)}{\geq} J(C^{t+1}, \mu^{t+1}).$$

[849] Lloyd, S. P. (1957). "Least square quantization in PCM". Bell Telephone Laboratories Paper.

[850] Formula aceasta corespunde folosirii distanţei euclidiene. Pentru alte măsuri de distanţă, este posibil să fie necesare alte formule (vedeţi problema 49.B).

La prima inegalitate (cea corespunzătoare pasului 1) se poate considera că parametrul C^t este fixat iar μ este variabil, în vreme ce la a doua inegalitate (cea corespunzătoare pasului 2) se consideră μ^t fixat şi C variabil. Prima inegalitate se poate obţine însumând o serie de inegalităţi, şi anume câte una pentru fiecare cluster C_j^t: se demonstrează (de exemplu, cu ajutorul proprietăţilor derivatei) că $J(C_j^t, \mu) \geq J(C_j^t, \mu^{t+1})$ pentru $\forall \mu$, deci în particular şi pentru μ^t. A doua inegalitate se demonstrează imediat.

b. Ce puteţi spune despre oprirea algoritmului K-means? Termină oare acest algoritm într-un număr finit de paşi, ori dimpotrivă — dat fiind faptul că există doar un număr finit de K-partiţii ale mulţimii de instanţe $\{x_1, \ldots, x_n\}$, şi anume K^n — este posibil ca el să reviziteze de o infinitate de ori o K-configuraţie anterioară, $\mu = (\mu_1, \ldots, \mu_K)$?

Răspuns:

a. Pentru convenienţă, ne vom limita la cazul $d = 1$.[851] Extinderea demonstraţiei la cazul $d > 1$ nu comportă dificultăţi.[852]

Vom demonstra mai întâi inegalitatea (1): $J(C^t, \mu^t) \geq J(C^t, \mu^{t+1})$.

După definiţie, $J(C^t, \mu^t) = \sum_{i=1}^{n} \left(x_i - \mu^t_{C^t(x_i)} \right)^2$.

Conform notaţiei din enunţ, C_j^t reprezintă clusterul j de la iteraţia t, iar μ_j^t este centroidul corespunzător acestui cluster. Dacă notăm cu $x_{i_1}, x_{i_2}, \ldots, x_{i_l}$ instanţele din componenţa clusterului C_j^t, unde $l \overset{not.}{=} |C_j^t|$, atunci (printr-un uşor abuz de notaţie) vom putea scrie:

$$J(C_j^t, \mu_j^t) = \sum_{p=1}^{l} \left(x_{i_p} - \mu_j^t \right)^2,$$

iar $J(C^t, \mu^t)$ se va rescrie sub forma $J(C^t, \mu^t) = \sum_{j=1}^{K} J(C_j^t, \mu_j^t)$.

Dacă se consideră C_j^t fixat, iar μ_j^t variabil (vedeţi pasul 1 al iteraţiei t), atunci putem minimiza funcţia de gradul al doilea care „măsoară" *coeziunea* din interiorul clusterului C_j^t:

$$f(\mu) \overset{def.}{=} J(C_j^t, \mu) = l\mu^2 - 2\Big(\sum_{p=1}^{l} x_{i_p} \Big) \cdot \mu + \sum_{p=1}^{l} x_{i_p}^2$$

fie în mod direct (adică făcând apel la proprietăţile funcţiei de gradul al doilea):

$$\arg\min_{\mu} J(C_j^t, \mu) = \frac{1}{l} \sum_{p=1}^{l} x_{i_p},$$

fie cu ajutorul derivatei vectoriale:[853]

$$\frac{\partial}{\partial \mu} f(\mu) = \sum_{p=1}^{l} (-2) \left(x_{i_p} - \mu \right) = -2 \left(\sum_{p=1}^{l} x_{i_p} - l\mu \right),$$

[851] Pentru o discuţie interesantă asupra aplicării algoritmului K-means în cazul $d = 1$, vedeţi problema 47.
[852] Se dezvoltă în mod corespunzător expresia lui J sau se folosesc derivatele sale parţiale de ordinul întâi.
[853] Vedeţi formula (5g) din documentul *Matrix Identities* de Sam Roweis.

ştiind că punctul de minim al funcţiei $f(\mu) = J(C_j^t, \mu)$ se obţine pentru rădăcina ecuaţiei $\frac{\partial}{\partial \mu} f(\mu) = 0$, adică

$$\mu = \frac{1}{l} \sum_{p=1}^{l} x_{i_p} = \frac{1}{|C_j^t|} \sum_{p=1}^{l} x_{i_p} \stackrel{def.}{=} \mu_j^{t+1}.$$

Aceasta înseamnă că $J(C_j^t, \mu) \geq J(C_j^t, \mu_j^{t+1})$, pentru $\forall \mu$. În particular, pentru $\mu = \mu_j^t$ vom avea: $J(C_j^t, \mu_j^t) \geq J(C_j^t, \mu_j^{t+1})$. Inegalitatea aceasta este valabilă pentru fiecare dintre clusterele $j = 1, \ldots, K$. Dacă însumăm toate aceste inegalităţi, rezultă: $J(C^t, \mu^t) \geq J(C^t, \mu^{t+1})$.

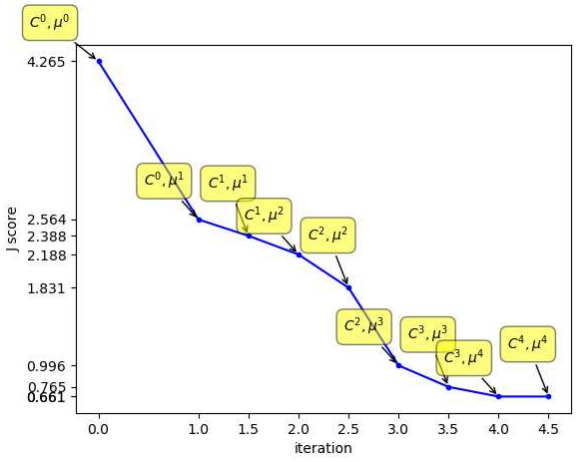
Mai rămâne de demonstrat inegalitatea (2): $J(C^t, \mu^{t+1}) \geq J(C^{t+1}, \mu^{t+1})$, corespunzătoare *pasului 2* din algoritm.

La acest pas, o instanţă oarecare x_i, unde $i \in \{1, \ldots, n\}$, este reasignată de la clusterul cu centroidul μ_j^{t+1}, la un alt centroid μ_q^{t+1}, dacă $||x_i - \mu_{j'}^{t+1}||^2 \geq ||x_i - \mu_q^{t+1}||^2$ pentru orice $j' = 1, \ldots, K$. Această condiţie este echivalentă cu următoarea: $(x_i - \mu_{j'}^{t+1})^2 \geq (x_i - \mu_q^{t+1})^2$ pentru orice j'. În contextul iteraţiei t, acest lucru implică

$$\left(x_i - \mu_{C^t(x_i)}^{t+1}\right)^2 \geq \left(x_i - \mu_{C^{t+1}(x_i)}^{t+1}\right)^2.$$

Sumând membru cu membru inegalităţile de acest tip obţinute pentru $i = \overline{1, n}$, rezultă: $J(C^t, \mu^{t+1}) \geq J(C^{t+1}, \mu^{t+1})$, ceea ce era de demonstrat.

***Exemplu*:**
Pentru a ilustra grafic monotonia criteriului J, alăturat vă punem la dispoziţie rezultatul obţinut pe datele de la problema 10.a.[854]



b. Evident, numărul K-partiţiilor care se pot forma cu cele n instanţe date este finit (K^n). A căuta minimul „criteriului“ J se poate face parcurgând în mod exhaustiv mulţimea acestor K-partiţii,[855] **dar acest proces ar fi ineficient (sau, practic imposibil de realizat) pentru valori mari ale lui n. Algoritmul**

[854] Graficul a fost realizat de către studentul Gheorghe Balan de la Universitatea „Al. I. Cuza“ din Iaşi în semestrul I al anului universitar 2017-2018.

[855] Este suficient să explorăm doar spaţiul format de valorile primului argument al criteriului J deoarece, conform inegalităţii (1), pentru orice poziţionare a centroizilor μ, valoarea $J(C, \mu)$ este minorată de $J(C, \mu')$ unde μ' reprezintă poziţiile recalculate ale centroizilor [adică, centrele de greutate ale] clusterelor din K-partiţia C, după cum se procedează la pasul 1 al algoritmului K-means.

K-means explorează — pornind de la o anumită iniţializare a celor K centroizi —, doar un subset de K-partiţii, asigurându-ne însă că are loc proprietatea $J(C^0, \mu^1) \geq J(C^1, \mu^2) \geq \ldots \geq J(C^{t-1}, \mu^t) \geq J(C^t, \mu^{t+1})$, conform punctului precedent al acestei probleme. (În inegalitatea multiplă de mai sus, μ^{i+1} constituie vectorul de centroizi, i.e., mediile instanţelor din clusterele care compun K-partiţia C^i.)

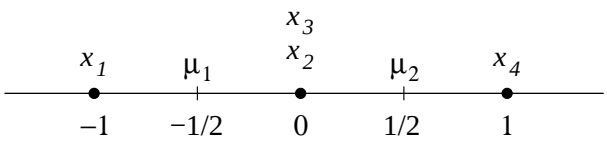
Dacă algoritmul revizitează o K-partiţie, atunci rezultă că pentru un anumit t avem $J(C^{t-1}, \mu^t) = J(C^t, \mu^{t+1})$. Este posibil ca acest fapt să se întâmple, şi anume atunci când:

– există instanţe multiple (i.e., $x_i = x_j$, deşi $i \neq j$),

– criteriul de oprire al algoritmului K-means este de forma „până când componenţa clusterelor nu se mai modifică“,

– se presupune că, în cazul în care o instanţă x_i este situată la egală distanţă faţă de doi sau mai mulţi centroizi, ea poate fi asignată în mod aleatoriu la oricare dintre ei.

Aşa se întâmplă în *exemplul* din figura următoare



dacă se consideră că la o iteraţie t avem $x_2 = 0 \in C_1^t$ şi $x_3 = 0 \in C_2^t$, iar la iteraţia următoare alegem ca $x_3 = 0 \in C_1^{t+1}$ şi $x_2 = 0 \in C_2^{t+1}$ şi, din nou, invers la iteraţia $t + 2$.

Evident, dacă se impune restricţia ca $x_2 = 0$ şi $x_3 = 0$ să fie asignaţi la un acelaşi cluster,[856] atunci nu vom mai avea ciclare (iar $J(C, \mu)$ va avea în final valoarea minimă: $0 + \left(\frac{2}{3}\right)^2 + 2\left(\frac{1}{3}\right)^2 = \frac{2}{3}$.

***Observaţia 1*: Dacă se păstrează criteriul dat ca exemplu în enunţul problemei – adică se iterează până când centroizii „staţionează“ – algoritmul se poate opri fără ca la ultima iteraţie $J(C, \mu)$ să fi atins minimul posibil. În cazul exemplului de mai sus, vom avea $\frac{1}{4} + 2 \cdot \frac{1}{4} + \frac{1}{4} = 1 > \frac{2}{3}$.**

***Observaţia 2*: Dacă nu există instanţe multiple care să fie situate la distanţe egale faţă de doi sau mai mulţi centroizi la o iteraţie oarecare a algoritmului K-means (precum sunt x_2 şi x_3 în *exemplul* de mai sus), sau dacă se impune *restricţia* ca în astfel de situaţii instanţele identice să fie asignate la un singur / acelaşi cluster, este evident că algoritmul K-means se opreşte într-un număr finit de paşi.**

***Observaţia 3*: Atingerea minimului global al funcţiei $J(C, \mu)$ — unde C este o variabilă care parcurge mulţimea tuturor K-partiţiilor care se pot forma cu instanţele $\{x_1, \ldots, x_n\}$ — nu este garantată pentru algoritmul K-means. Valoarea funcţiei J care se obţine la oprirea algoritmului K-means este dependentă de plasarea iniţială a centroizilor μ, precum şi de modul concret în care sunt**

[856] O condiţie mai generală poate fi următoarea: asignarea instanţelor x_i la centroizi să se facă astfel încât la pasul 1 inegalitatea (1) să fie satisfăcută în sens strict: $J(C^t, \mu^t) > J(C^t, \mu^{t+1})$.

alcătuite clusterele în cazul în care o instanţă oarecare se află la distanţă egală de doi sau mai mulţi centroizi, după cum am arătat în exemplul de mai sus.

***Observaţia 4*: Datorită rezultatului obţinut la punctul a al acestui exerciţiu, algoritmul de clusterizare K-means poate fi văzut [şi reformulat] ca un *algoritm de optimizare*, folosind metoda *descreşterii pe coordonate*.**[857] **Ca *input*, vom considera (ca şi în pseudo-codul anterior) punctele $x_1, \ldots, x_n \in \mathbb{R}^d$, împreună cu numărul natural $K \geq 1$. *Obiectivul* este acela de a minimiza o funcţie obiectiv care măsoară (indirect) coeziunea intra-clustere. Atunci când se foloseşte distanţa euclidiană, această funcţie este**

$$J(L, \mu) = \sum_{i=1}^{n} ||x_i - \mu_{l_i}||^2,$$

unde $\mu = (\mu_1, \ldots, \mu_K)$ sunt centroizii clusterelor ($\mu_j \in \mathbb{R}^d$), iar lista $L = (l_1, \ldots, l_n)$ memorează asignările instanţelor x_i la clustere ($l_i \in \{1, \ldots, K\}$).[858]

Algoritmul K-means face *iniţializarea* centroizilor clusterelor μ cu anumite valori, după care procedează astfel:

***Pasul 1*: Păstrând μ fixat, găseşte acea asignare L a instanţelor la clustere care minimizează funcţia $J(L, \mu)$;**[859]

***Pasul 2*: Păstrând asignarea L fixată, găseşte acea valoare pentru μ pentru care se minimizează $J(L, \mu)$.**[860]

***Criteriul de oprire*: Dacă [aceasta nu este prima iteraţie şi] niciuna dintre asignările din lista L nu s-a modificat în raport cu precedenta iteraţie, se trece la pasul următor (Terminare); altfel se repetă de la Pasul 1.**

***Terminare*: Returnează L şi μ.**

13. (Algoritmul K-means: o proprietate de [*anti*]*monotonie* a valorilor minime pentru criteriul „sumei celor mai mici pătrate“ (J) în funcţie de K)

• ○ ***CMU, 2012 fall, E. Xing, A. Singh, HW3, pr. 1.1.d***
CMU, 2010 fall, Aarti Singh, HW3, pr. 5.4

Cum evoluează valoarea minimă a funcţiei obiectiv J, aşa-numita sumă a celor mai mici pătrate (vedeţi problema 12), la execuţia algoritmului K-means pe un set de date arbitrar ales (dar fixat) atunci când valoarea lui K creşte de la 1 la n: creşte, scade, rămâne constantă, variază arbitrar ori conform unei anumite legi / proprietăţi? Justificaţi în mod riguros.

[857] Ceea ce urmează în continuare este preluat — cu uşoare adaptări — din CMU, 2012 fall, E. Xing, A. Singh, HW3, pr. 1.

[858] Găsirea minimului global al acestei funcţii este un task *dificil* din punct de vedere computaţional.

[859] Aceasta revine la a găsi pentru fiecare instanţă x_i care este cel mai apropiat centroid. În cazul în care instanţa se află la distanţă minimă faţă de mai mulţi centroizi, se pot folosi diverse euristici pentru a decide cum anume se face asignarea.

[860] Acesta este un pas simplu, care revine — în cazul în care se foloseşte distanţa euclidiană — la a calcula pentru fiecare cluster media instanţelor (văzute ca vectori) care alcătuiesc clusterul respectiv.

***Indicaţie*: În răspunsul pe care-l veţi da, veţi nota cu $X = \{x_1, \ldots, x_n\}$ setul de instanţe de clusterizat şi cu $\underline{J}_K(X)$ valoarea minimă a criteriului J pe mulţimea tuturor K-partiţiilor C ale lui X. Formal, putem scrie astfel:**

$$\begin{aligned} \underline{J}_K(X) &= \min\{J_K(X,C) \mid C \text{ este } K\text{-partiţie a lui } X\} \\ \text{cu } J_K(X,C) &= \sum_{i=1}^{n} \|x_i - \mu_{C(x_i)}\|^2, \end{aligned}$$

unde $\mu_{C(x_i)}$ este centroidul clusterului $C(x_i)$, la care este asignat x_i în K-partiţia (adică setul de clustere) C. Conform problemei 12.a, atunci când se foloseşte distanţa euclidiană se consideră $\mu_{C(x_i)} = \frac{1}{|C(x_i)|} \sum_{x_j \in C(x_i)} x_j$.

Răspuns:

Este imediat că orice K-partiţie C a lui X poate fi pusă în mod natural în corespondenţă cu o $(K+1)$-partiţie C' a lui X (să o notăm pe aceasta din urmă cu $(C_1, \ldots, C_K, C_{K+1})$), în care un cluster (de exemplu C_{K+1}) este vid, iar celelalte clustere sunt în corespondenţă bijectivă de tipul funcţiei *identitate* cu clusterele K-partiţiei C. Evident, $J_K(X,C) = J_K(X,C')$.

În acest fel, mulţimea tuturor K-partiţiilor lui X (notată cu $\mathcal{P}$) este pusă în corespondenţă bijectivă cu o *submulţime* din mulţimea tuturor $(K+1)$-partiţiilor lui X (ultima, notată cu $\mathcal{P}'$).[861] Aşadar, putem spune că într-un anumit sens, mulţimea $\mathcal{P}$ este mai amplă decât mulţimea $\mathcal{P}'$. Rezultă că

$$\underline{J}_{K+1}(X) \leq \underline{J}_K(X), \text{ pentru orice } K = 1, \ldots, n-1.$$

Aceasta este o proprietate de *anti-monotonie* a valorilor minime pentru criteriul „sumei celor mai mici pătrate“ (J) în funcţie de K.

Altă variantă de demonstraţie:
Se poate vedea că orice K-partiţie poate fi pusă în corespondenţă cu o $(K+1)$-partiţie al cărei scor (ca „sumă a pătratelor erorilor“) este mai mic sau egal cu cel al K-partiţiei.
În cazul $K < n$ (unde n este numărul de instanţe de clusterizat), este clar că există o instanţă care poate fi extrasă din clusterul ei ca să devină [ea însăşi] un cluster singleton. Clusterul din care a provenit instanţa respectivă va avea acum valoarea „sumei pătratelor erorilor“ mai mică decât [sau egală cu] vechea valoare (ceea ce se justifică imediat, fiindcă centroidul clusterului se va muta în poziţia optimă, adică poziţia care corespunde minimului „sumei pătratelor erorilor“ pentru cluserul care acum are cu o instanţă mai puţin decât avea înainte), iar clusterul singleton va avea valoarea „sumei pătratelor erorilor“ 0.
În cazul $K = n$, toate $(K+1)$-partiţiile vor avea un cluster vid, în consecinţă, valoarea minimă a criteriului J_{K+1} este aceeaşi cu minimul lui J_K.

[861] Matematicienii spun că mulţimea $\mathcal{P}$ este „scufundată“ în mulţimea $\mathcal{P}'$.

14. **(Clusterizarea ierarhică şi algoritmul K-means: comparaţii)**

□ • ○ ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, midterm, pr. 4.b***

Enunţaţi câteva avantaje ale clusterizării ierarhice în raport cu clusterizarea bazată pe algoritmul K-means. Similar, enunţaţi câteva avantaje ale clusterizării bazate pe algoritmul K-means în raport cu clusterizarea ierarhică.

Răspuns:

Iată câteva avantaje ale clusterizării ierarhice:

– nu necesită fixarea dinainte a unui anumit număr (K) de clustere pe care dorim să le obţinem;
– arborele ierarhic obţinut poate fi tăiat la orice nivel, pentru a obţine câte clustere dorim;
– pentru multe aplicaţii, ierarhia obţinută este uşor de interpretat;
– se poate lucra cu date dispuse spaţial într-o formă elongată (engl., long stringy data).

Şi câteva avantaje ale clusterizării bazate pe algoritmul K-means:

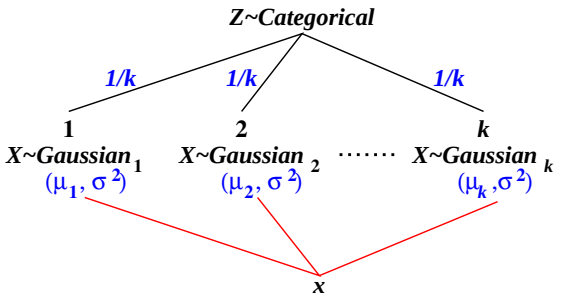
– pe anumite seturi de date, rezultatul poate fi obţinut mult mai rapid decât clusterizarea ierarhică;
– beneficiază de un cadru teoretic elegant;
– se pot incorpora uşor date noi şi, de asemenea, este posibilă realcătuirea clusterelor în mod facil;
– poate fi kernel-izat (vedeţi ex. 51).

7.1.3 Algoritmul EM pentru modele de mixturi gaussiene

15. **(Algoritmul EM pentru GMM, cazul unidimensional, cu $\sigma_1 = \ldots = \sigma_k$ şi $\pi_1 = \ldots = \pi_k$: demonstraţia formulelor de „actualizare“)**

prelucrare de Liviu Ciortuz, după
■ • ○ ***Machine Learning, Tom Mitchell, 1997, pag. 193, 195–196.***

Ne propunem să facem estimarea parametrilor unui model de mixtură de k distribuţii gaussiene (engl., Gaussian Mixture Model, GMM), despre care se presupune că au [toate] aceeaşi varianţă σ^2, iar probabilităţile a priori de selecţie sunt egale ($1/k$).



Prin urmare, se vor estima doar mediile acestor k distribuţii gaussiene. Pentru aceasta, vom considera următorul *algoritm EM*:

***Iniţializare*: Date fiind instanţele $x_1, \ldots, x_n \in \mathbb{R}$, se aleg în mod arbitrar valori pentru mediile $\mu \overset{not.}{=} (\mu_1, \ldots, \mu_k)$. Vom nota aceste valori iniţiale cu $\mu_1^{(0)}, \ldots, \mu_k^{(0)}$. Pentru $t = 0, 1, \ldots$, atâta timp cât nu este îndeplinit *criteriul de oprire*,[862] execută:**

***Pasul E*:**

$$E[Z_{ij}] \overset{not.}{=} E[Z_{ij}|X = x_i; \mu^{(t)}, \sigma^2] = \ldots = P(Z_{ij} = 1|X = x_i; \mu^{(t)}, \sigma^2)$$

$$\overset{F.\ Bayes}{=} \ldots$$

$$\overset{p.d.f.}{=} \ldots \qquad \text{pentru } i = 1, \ldots, n \text{ şi } j = 1, \ldots, k.$$

***Pasul M*:**

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^n E[Z_{ij}]x_i}{\sum_{i=1}^n E[Z_{ij}]}, \qquad \text{pentru } j = 1, \ldots, k, \tag{367}$$

unde Z_{ij}, cu $i \in \{1, \ldots, n\}$ şi $j \in \{1, \ldots, k\}$, sunt variabile-indicator *neobservabile* (sau „ascunse" / „latente"), luând valoarea 1 dacă x_i a fost generat de gaussiana j şi 0 în caz contrar.

Elaboraţi formula de la pasul E şi demonstraţi formula (367) pentru pasul M.

***Sugestie*: Pentru pasul M,**

- **veţi scrie mai întâi $p(y_i|\mu)$ verosimilitatea unei instanţe „complete", $y_i \overset{not.}{=} (x_i, z_{i1}, \ldots, z_{ik})$;**[863]
- **apoi veţi scrie $\ln P(Y|\mu)$, funcţia de log-verosimilitate a datelor complete $Y \overset{not.}{=} \{y_1, \ldots, y_n\}$;**
- **veţi calcula funcţia „auxiliară" $Q(\mu|\mu^{(t)}) \overset{def.}{=} E[\ln P(Y|\mu)]$, care este media funcţiei de log-verosimilitate a datelor complete în raport cu distribuţia probabilistă a posteriori a variabilelor neobservabile Z_{ij} în raport cu datele observabile x_i şi cu valorile parametrilor μ de la iteraţia t, adică $\mu^{(t)}$;**
- **în final, veţi calcula valorile mediilor μ_j pentru care se obţine valoarea optimă a funcţiei „auxiliare" $Q(\mu|\mu^{(t)})$.**

Răspuns:

Pasul E:

Aşa cum a fost sugerat în enunţ, mai întâi arătăm că media variabilei-indicator Z_{ij} (văzută ca variabilă aleatoare) este $P(Z_{ij} = 1 \mid x_i, \mu^{(t)})$.[864] **Apoi calculăm această probabilitate condiţionată folosind formula lui Bayes combinată cu formula probabilităţii totale, iar la final ţinem cont de expresia funcţiei de densitate pentru distribuţia gaussiană.**[865]

$$E[Z_{ij}] \overset{def.}{=} 0 \cdot P(Z_{ij} = 0 \mid x_i, \mu^{(t)}) + 1 \cdot P(Z_{ij} = 1 \mid x_i, \mu^{(t)}) = P(Z_{ij} = 1 \mid x_i, \mu^{(t)})$$

[862] De exemplu, se execută un anumit număr de iteraţii, fixat în prealabil.

[863] Pentru un exemplu simplu de definire şi calculare a funcţiei de verosimilitate, vedeţi ex. 42 de la capitolul *Fundamente*.

[864] Pentru o formulare generală a acestei proprietăţi, vedeţi problema 98 de la capitolul de *Fundamente*.

[865] Vedeţi problema 32 de la capitolul de *Fundamente*.

$$\overset{T.Bayes}{=} \frac{P(x_i \mid Z_{ij}=1;\mu^{(t)})\cdot \overbrace{P(Z_{ij}=1 \mid \mu^{(t)})}^{1/k}}{\sum_{l=1}^{k} P(x_i \mid Z_{il}=1;\mu^{(t)})\cdot \underbrace{P(Z_{il}=1 \mid \mu^{(t)})}_{1/k}} = \frac{\mathcal{N}(X=x_i|\mu=\mu_j^{(t)})}{\sum_{l=1}^{k}\mathcal{N}(X=x_i|\mu=\mu_l^{(t)})}$$

$$\overset{p.d.f.}{=} \frac{\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2\sigma^2}(x_i-\mu_j^{(t)})^2}}{\sum_{l=1}^{k}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{1}{2\sigma^2}(x_i-\mu_l^{(t)})^2}} = \frac{e^{-\frac{1}{2\sigma^2}(x_i-\mu_j^{(t)})^2}}{\sum_{l=1}^{k} e^{-\frac{1}{2\sigma^2}(x_i-\mu_l^{(t)})^2}}.$$

Vă readucem aminte că probabilităţile a priori $P(Z_{il}=1 \mid \mu^{(t)})$ au fost considerate de la început identice, indiferent de valoarea lui l.

Pasul M:

Pentru a calcula probabilitatea unei date complete $y_i \overset{not.}{=} (x_i, z_{i1}, \ldots, z_{ik})$, vom folosi regula de înmulţire a probabilităţilor şi, din nou, expresia funcţiei de densitate pentru distribuţia gaussiană.

$$p(y_i|\mu) \overset{not.}{=} p(x_i, z_{i1}, \ldots, z_{ik}|\mu) = p(x_i|z_{i1}, \ldots, z_{ik};\mu)\underbrace{p(z_{i1}, \ldots, z_{ik}|\mu)}_{1/k}$$

$$= \frac{1}{k}\frac{1}{\sqrt{2\pi}\sigma}\ e^{-\frac{1}{2\sigma^2}\sum_{j=1}^{k} z_{ij}(x_i-\mu_j)^2}$$

Observaţi introducerea în ultima expresie a variabilelor-indicator Z_{ij}. Se ţine cont de faptul că pentru fiecare $i=1,\ldots,n$, avem $\sum_{j=1}^{k} Z_{ij}=1$, deci doar una dintre aceste variabile-indicator are valoarea 1 (şi anume, cea care corespunde gaussienei care a generat instanţa x_i), în vreme ce toate celelalte variabile-indicator au valoarea 0.

Funcţia de log-verosimilitate a datelor complete $Y=\{y_1,\ldots,y_n\}$ se calculează uşor, ţinând cont de independenţa creării datelor x_i, precum şi de rezultatul obţinut anterior.

$$\ln P(Y|\mu) \overset{i.i.d.}{=} \ln\prod_{i=1}^{n} p(y_i|\mu) = \sum_{i=1}^{n}\ln p(y_i|\mu)$$

$$= \sum_{i=1}^{n}(-\ln k + \ln\frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} z_{ij}(x_i-\mu_j)^2).$$

Media funcţiei de log-verosimilitate a datelor complete (aşa-numita *funcţie auxiliară*) se obţine aplicând proprietatea de *liniaritate a mediei*.[866]

$$Q(\mu|\mu^{(t)}) = E[\ln P(Y|\mu)] \overset{lin.\ med.}{=} \sum_{i=1}^{n}(-\ln k + \ln\frac{1}{\sqrt{2\pi}\sigma} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} E[Z_{ij}](x_i-\mu_j)^2). \quad (368)$$

Se poate observa că aceasta este o funcţie de gradul al doilea în raport cu variabilele μ_j.[867] **Se constată apoi că optimul ei se poate obţine prin maximizare separată (sau, pe rând) în raport cu fiecare variabilă μ_j.**

[866] Vedeţi problema 9.a de la capitolul de *Fundamente*.
[867] Aşadar, pentru $k=2$ graficul unei astfel de funcţii este un *paraboloid*.

$$\begin{aligned} Q(\mu|\mu^{(t)}) &= n(-\ln k + \ln \frac{1}{\sqrt{2\pi}\sigma}) - \frac{1}{2\sigma^2}\sum_{i=1}^{n}\sum_{j=1}^{k} E[Z_{ij}](x_i-\mu_j)^2 \\ &= n(-\ln k + \ln \frac{1}{\sqrt{2\pi}\sigma}) - \frac{1}{2\sigma^2}\sum_{j=1}^{k}\sum_{i=1}^{n} E[Z_{ij}](x_i-\mu_j)^2 \\ &= \textbf{\textit{const}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k}\sum_{i=1}^{n} E[Z_{ij}](x_i^2 - 2\mu_j x_i + \mu_j^2)] \\ &= \textbf{\textit{const}}' - \frac{1}{2\sigma^2}\sum_{j=1}^{k}\Big[\Big(\sum_{i=1}^{n} E[Z_{ij}]\Big)\mu_j^2 - 2\Big(\sum_{i=1}^{n} E[Z_{ij}]x_i\Big)\mu_j\Big]. \end{aligned}$$

Pentru fiecare $j = 1, \ldots, k$, coeficientul dominant al lui μ_j este strict negativ (fiindcă $E[Z_{ij}] = P(Z_{ij} = 1 \mid x_i, \mu^{(t)}) > 0$, conform rezultatului de la pasul E). Aşadar, soluţia optimă este dată de relaţia

$$\mu_j^{(t+1)} \overset{def.}{=} \operatorname*{argmax}_{\mu_j} Q(\mu|\mu^{(t)}) = -\frac{2\frac{1}{2\sigma^2}\sum_{i=1}^{n} E[Z_{ij}]\, x_i}{-2\frac{1}{2\sigma^2}\sum_{i=1}^{n} E[Z_{ij}]} = \frac{\sum_{i=1}^{n} E[Z_{ij}]\, x_i}{\sum_{i=1}^{n} E[Z_{ij}]}.$$

***Observaţie*: Rezultatul acesta (în enunţ, formula (367)) reprezintă media ponderată a instanţelor x_i, $i = 1, \ldots, n$. Aşadar, această formulă reprezintă o generalizare (sau, o variantă ponderată) a formulei pentru estimarea mediei (în sens MLE) pentru o distribuţie gaussiană unidimensională (vedeţi rezolvarea problemei 50.a de la capitolul *Fundamente*).**

16. **(Algoritmul EM/GMM, cazul $\sigma_1^2 = \sigma_2^2 = 1$, $\pi_1 = \pi_2 = 1/2$: executarea unei iteraţii)**

prelucrare de Liviu Ciortuz, după
■ • ○ ***CMU, 2012 spring, Ziv Bar-Joseph, final exam, pr. 3.1***

Fie un model de mixtură gaussiană (engl., Gaussian mixture model, GMM) cu două componente având varianţe cunoscute şi probabilităţi *a priori* egale pentru selecţia celor două distribuţii:

$$\frac{1}{2}\mathcal{N}(x|\mu_1, 1) + \frac{1}{2}\mathcal{N}(x|\mu_2, 1),\ x \in \mathbb{R}.$$

În continuare se va considera că $n = 2$, $x_1 = 0.5$ şi $x_2 = 2$, iar valorile iniţiale pentru μ_1 şi μ_2 sunt 1 şi respectiv 2.

Executaţi în mod manual o iteraţie a algoritmului EM din enunţul problemei 15 (preluat din cartea *Machine Learning* a lui Tom Mitchell, pag. 193) pe aceste date, astfel:

a. (Pasul E) Calculaţi mai întâi $P(Z_{ij} = 1|x_i, \mu)$, probabilităţile a posteriori de apartenenţă a datelor observate (x_1 şi x_2) la cele două componente ale mixturii. Am folosit notaţia $\mu = (\mu_1, \mu_2)$, iar Z_{ij} este variabilă-indicator având valoarea 1 dacă instanţa x_i a fost generată de gaussiana j şi 0 în cazul contrar.

***Indicaţie*: În vederea efectuării calculelor, pentru convenienţă puteţi considera valorile distribuţiei normale / gaussiene standard $\mathcal{N}(x|\mu=0,\sigma^2=1)$ în punctele 0, 0.5, 1, 1.5 şi 2 ca fiind respectiv 0.4, 0.35, 0.24, 0.13 şi 0.05.**

b. (Pasul M) Re-calculaţi valorile parametrilor μ_1 şi μ_2 în funcţie de probabilităţile calculate la punctul precedent. Care credeţi că va fi tendinţa de mişcare a mediilor la următoarele iteraţii?

c. Funcţia de log-verosimilitate a datelor observabile este definită aşa cum este de aşteptat, ca logaritmul unei funcţii de probabilitate (marginală în raport cu distribuţia comună $P(x,z|\mu)$):

$$\ell(\mu_1,\mu_2) \overset{def.}{=} \ln P(x_1,x_2|\mu_1,\mu_2) \overset{indep.}{=} \sum_{i=1}^{2} \ln P(x_i|\mu_1,\mu_2) = \sum_{i=1}^{2} \ln \Big(\sum_{z_{ij}} P(x_i,z_{ij}|\mu_1,\mu_2) \Big),$$

unde simbolul $\sum_{zij}$ desemnează parcurgerea tuturor asignărilor posibile pentru variabilele neobservabile Z_{ij} (adică, mai întâi $z_{11}=1$ şi $z_{21}=1$, apoi $z_{11}=1$ şi $z_{21}=0$, după care $z_{11}=0$ şi $z_{21}=1$ şi, în final, $z_{11}=0$ şi $z_{21}=0$). Explicitaţi expresia acestei funcţii (păstrând μ_1 şi μ_2 nespecificaţi). După aceea, calculaţi valoarea ei la începutul şi respectiv la sfârşitul primei iteraţii a algoritmului EM pe datele x_1 şi x_2 specificate în prima parte a acestui exerciţiu.

Răspuns:

a. Folosind teorema lui Bayes, vom exprima probabilităţile de apartenenţă ale instanţelor x_1 şi respectiv x_2 la clusterul reprezentat de prima gaussiană:

$$\begin{aligned} P(Z_{i1}=1|x_i,\mu) &\overset{T.B.}{=} \frac{P(x_i|Z_{i1}=1,\mu_1)P(Z_{i1}=1)}{P(x_i|Z_{i1}=1,\mu_1)P(Z_{i1}=1)+P(x_i|Z_{i2}=1,\mu_2)P(Z_{i2}=1)} \\ &= \frac{P(x_i|Z_{i1}=1,\mu_1)\cdot\frac{1}{2}}{P(x_i|Z_{i1}=1,\mu_1)\cdot\frac{1}{2}+P(x_i|Z_{i2}=1,\mu_2)\cdot\frac{1}{2}} \\ &= \frac{P(x_i|Z_{i1}=1,\mu_1)}{P(x_i|Z_{i1}=1,\mu_1)+P(x_i|Z_{i2}=1,\mu_2)} \text{ pentru } i\in\{1,2\}. \end{aligned}$$

Ţinând cont de faptul că valorile oricărei *distribuţii gaussiene* unidimensionale pot fi puse în corespondenţă cu valorile *distribuţiei gaussiene standard*,[868] **utilizând valorile furnizate în *indicaţia* din enunţ şi folosind proprietatea de simetrie a valorilor distribuţiei gaussiene standard în raport cu originea, vom avea:**

$$\begin{aligned} P(Z_{11}=1|x_1,\mu) &= \frac{\mathcal{N}(0.5|1,1)}{\mathcal{N}(0.5|1,1)+\mathcal{N}(0.5|2,1)} = \frac{\mathcal{N}(0.5|0,1)}{\mathcal{N}(0.5|0,1)+\mathcal{N}(1.5|0,1)} \\ &= \frac{0.35}{0.35+0.13} = \frac{0.35}{0.48} = \frac{35}{48} \\ P(Z_{21}=1|x_2,\mu) &= \frac{\mathcal{N}(2|1,1)}{\mathcal{N}(2|1,1)+\mathcal{N}(2|2,1)} = \frac{\mathcal{N}(1|0,1)}{\mathcal{N}(1|0,1)+\mathcal{N}(0|0,1)} \\ &= \frac{0.24}{0.24+0.4} = \frac{0.24}{0.64} = \frac{3}{8} \end{aligned}$$

[868] Vom folosi formula pentru „standardizare“ $\mathcal{N}(x|\mu,\sigma^2) = \frac{1}{\sigma}\mathcal{N}(\frac{x-\mu}{\sigma}|0,1)$, care este uşor de demonstrat, precum şi proprietatea $\mathcal{N}(x|0,1)=\mathcal{N}(-x|0,1)$.

În sfârşit, datorită faptului că $Z_{i1} + Z_{i2} = 1$ pentru $\forall i \in \{1, 2\}$ (cu $Z_{ij} \in \{0, 1\}$) şi, de asemenea, datorită complementarităţii evenimentelor aleatoare, vom obţine imediat şi celelalte două probabilităţi cerute:

$$\begin{aligned} P(Z_{12} = 1|x_1, \mu) &= P(Z_{11} = 0|x_1, \mu) = 1 - P(Z_{11} = 1|x_1, \mu_1) = \frac{13}{48} \\ P(Z_{22} = 1|x_2, \mu) &= P(Z_{21} = 0|x_2, \mu) = 1 - P(Z_{21} = 1|x_2, \mu_1) = \frac{5}{8} \end{aligned}$$

b. Formulele (367) pentru actualizarea mediilor μ_j pot fi scrise folosind notaţiile de aici (adică, explicitând direct mediile variabilelor-indicator Z_{ij}), astfel:[869]

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^{2} P(Z_{ij} = 1|x_i, \mu^{(t)})\, x_i}{\sum_{i=1}^{2} P(Z_{ij} = 1|x_i, \mu^{(t)})}$$

Prin urmare,

$$\mu_1^{(1)} = \frac{\frac{35}{48} \cdot 0.5 + \frac{3}{8} \cdot 2}{\frac{35}{48} + \frac{3}{8}} = \frac{107}{106} \approx 1.009 \text{ şi } \mu_2^{(1)} = \frac{\frac{13}{48} \cdot 0.5 + \frac{5}{8} \cdot 2}{\frac{13}{48} + \frac{5}{8}} = \frac{133}{86} \approx 1.547$$

Se observă că, în raport cu poziţiile iniţiale, mediile celor două gaussiene au „migrat“ astfel: μ_1 foarte puţin spre dreapta, iar μ_2 considerabil mai la stânga. În noile poziţii, mediile $\mu_1^{(1)}$ şi $\mu_2^{(1)}$ sunt aproape simetrice în raport cu mijlocul segmentului determinat de punctele $x_1 = 0.5$ şi $x_2 = 2$. Aceasta implică faptul că la pasul următor vom avea $P(Z_{11} = 1|x_1, \mu^{(1)}) \approx P(Z_{22} = 1|x_2, \mu^{(1)})$ şi $P(Z_{12} = 1|x_1, \mu^{(1)}) \approx P(Z_{21} = 1|x_2, \mu^{(1)})$. Executând încă o iteraţie (nu dăm aici detaliile) obţinem $\mu_1^{(2)} \approx 1.110$ şi $\mu_2^{(2)} \approx 1.390$. Ne aşteptăm ca pe parcursul următoarelor iteraţii mediile $\mu_1^{(t)}$ şi $\mu_2^{(t)}$ să migreze către mijlocul intervalului $[0.5,\ 2]$.

c. Vom face mai întâi explicitarea expresiei care a fost dată în enunţ pentru funcţia de verosimilitate ℓ, specificând apoi termenii ei de bază într-un tabel:

$$\begin{aligned} \ell(\mu_1, \mu_2) &= \sum_{i=1}^{2} \ln\Big(\sum_{z_{ij}} P(x_i, Z_{ij} = z_{ij}|\mu_1, \mu_2)\Big) \\ &= \sum_{i=1}^{2} \ln\Big(\sum_{z_{ij}} P(x_i|z_{ij}, \mu_1, \mu_2) \cdot \underbrace{P(z_{ij}|\mu_1, \mu_2)}_{1/2}\Big) \\ &= \sum_{i=1}^{2} \ln\Big(\frac{1}{2}\sum_{z_{ij}} P(x_i|z_{ij}, \mu_1, \mu_2)\Big) \\ &= \sum_{i=1}^{2} \Big[-\ln 2 + \ln\Big(\sum_{j=1}^{2} P(x_i|z_{ij} = 1, \mu_j)\Big)\Big] \end{aligned}$$

[869] Expresia *funcţiei auxiliare* $Q(\mu, \mu^{(t)})$, al cărei punct de optim are abscisa $\mu^{(t+1)} \overset{not.}{=} (\mu_1^{(t+1)}, \mu_2^{(t+1)})$, se obţine — pentru acest caz particular — din formula (368) de la problema 15.

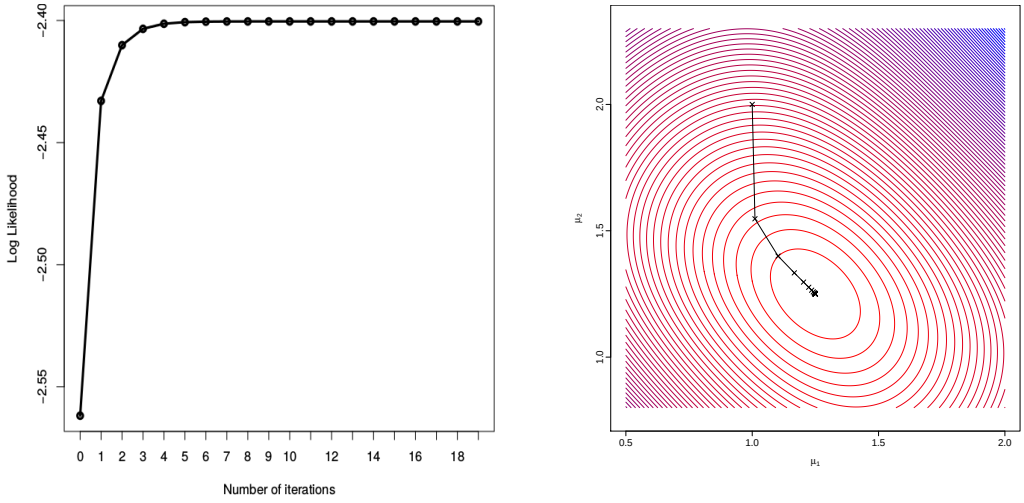
z_{ij}	$P(x_i \mid z_{ij}, \mu_1, \mu_2)$
$z_{11} = 1,\ z_{12} = 0$	$\frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}(x_1 - \mu_1)^2)$
$z_{11} = 0,\ z_{12} = 1$	$\frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}(x_1 - \mu_2)^2)$
$z_{21} = 1,\ z_{22} = 0$	$\frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}(x_2 - \mu_1)^2)$
$z_{21} = 0,\ z_{22} = 1$	$\frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}(x_2 - \mu_2)^2)$

Prin urmare,

$$\begin{aligned}\ell(\mu_1, \mu_2) =& -2\ln 2 + \ln\Big(\frac{1}{\sqrt{2\pi}}\Big(\exp\big(-\frac{1}{2}(x_1-\mu_1)^2\big) + \exp\big(-\frac{1}{2}(x_1-\mu_2)^2\big)\Big)\Big)\\ &+ \ln\Big(\frac{1}{\sqrt{2\pi}}\Big(\exp\big(-\frac{1}{2}(x_2-\mu_1)^2\big) + \exp\big(-\frac{1}{2}(x_2-\mu_2)^2\big)\Big)\Big)\\ =& -2\ln 2 - \ln(2\pi) + \ln\Big(\exp\big(-\frac{1}{2}(x_1-\mu_1)^2\big) + \exp\big(-\frac{1}{2}(x_1-\mu_2)^2\big)\Big)\\ &+ \ln\Big(\exp\big(-\frac{1}{2}(x_2-\mu_1)^2\big) + \exp\big(-\frac{1}{2}(x_2-\mu_2)^2\big)\Big)\end{aligned}$$

Valoarea acestei funcţii ca urmare a iniţializării mediilor μ_1 şi μ_2 cu valorile 1 şi respectiv 2 este -2.561833, iar la finalul primei iteraţii a algoritmului EM ea devine -2.462877. Se observă că valoarea funcţiei de log-verosimilitate creşte, aşa cum era de aşteptat (vedeţi problema 2 de la capitolul *Schema algoritmică EM*).

Pentru [confirmarea şi] *extinderea* acestui rezultat, prezentăm mai jos rezultatele obţinute cu ajutorul unei implementări a algoritmului EM:[870]

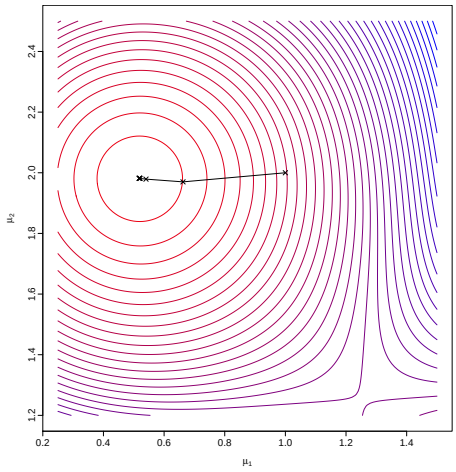


– în graficul din partea stângă sunt reprezentate valorile funcţiei de log-verosimilitate obţinute de EM la iniţializare şi apoi la finalul fiecăreia din primele 19 iteraţii,

– în graficul din partea dreaptă avem reprezentarea sub formă de curbe de izocontur a valorilor log-verosimilităţii în funcţie de cei doi parametri, μ_1 şi

[870] Această implementare a fost realizată în 2018 de către Sebastian Ciobanu.

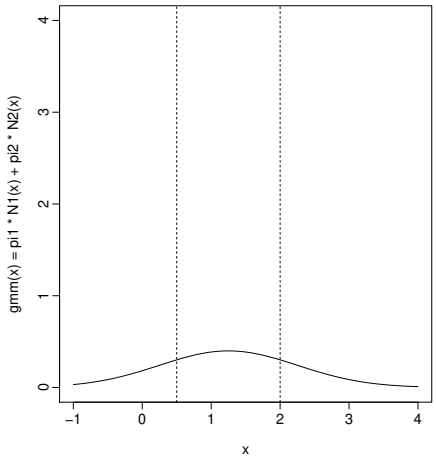
μ_2. Pe acest al doilea grafic a fost adăugat un „drum“ care pune în evidenţă succesiunea de valori pentru perechile (μ_1, μ_2) de-a lungul iteraţiilor executate de algoritmul EM.



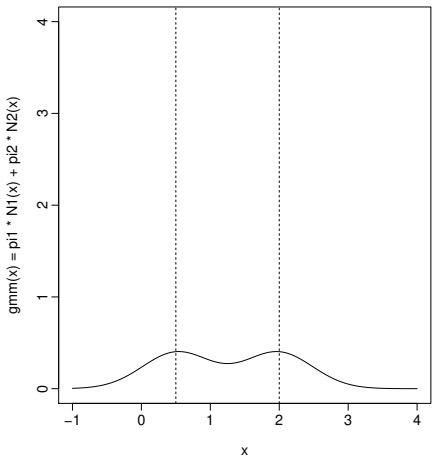
***Observaţie* (1): Se poate vedea în figura de mai sus, partea dreaptă, că mediile μ_1 şi μ_2 vor tinde la convergenţă la aceeaşi valoare, 1.25, valoare care reprezintă jumătatea intervalului $[0.5, 2]$. Este interesant de constatat că dacă am fi lucrat cu valori mici ale varianţei (de exemplu, $\sigma^2 = 1/4$, caz ilustrat în figura alăturată), atunci la convergenţă μ_1 ar fi tins la valoarea $x_1 = 0.5$, în vreme ce μ_2 ar fi tins la valoarea $x_2 = 2$. La problema 65 se va vedea că în general, atunci când $\sigma^2 \to 0$, algoritmul EM/GMM tinde să obţină (la convergenţă) aceleaşi rezultate ca şi algoritmul K-means!**

***Observaţie* (2): Modelele probabiliste obţinute „învăţate“ în urma aplicării algoritmului EM pentru cele două cazuri ($\sigma^2 = 1$ şi $\sigma^2 = 1/4$) sunt prezentate în graficele următoare.[871] În primul caz se poate observa că mixtura obţinută coincide practic cu o distribuţie gaussiană a cărei medie este situată la jumătatea intervalului determinat de cele două instanţe, x_1 şi x_2. În cel de-al doilea caz se poate constata că mediile celor două gaussiene care compun mixtura coincid cu (de fapt, sunt situate foarte aproape de) cele două instanţe, x_1 şi x_2.**

$\sigma^2 = 1$



$\sigma^2 = 1/4$



[871] Aceste grafice, ca şi cel precedent, au fost realizate de Sebastian Ciobanu.

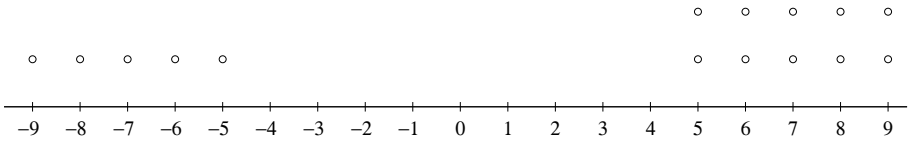
17. **(Algoritmii K-means şi EM pentru GMM, cazul unidimensional: aplicare)**

prelucrare făcută de L. Ciortuz şi A. Munteanu, după Edinburgh, 2009 fall, C. Williams, V. Lavrenko, HW4, pr. 3

Se consideră următorul set de instanţe – puncte în spaţiul unidimensional:

$$-9, -8, -7, -6, -5, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9$$

Aceste date au reprezentarea următoare:



Obiectivul este să se aplice pe de o parte algoritmul K-means şi pe de altă parte algoritmul EM pentru un model de mixtură de distribuţii gaussiene (GMM) ca să împărţim acest set de date în două clustere. Drept centroizi iniţiali se consideră punctele -20 şi -10. Este evident că această alegere este una nefavorabilă / nestandard, însă dorim să comparăm evoluţiile celor doi algoritmi în acest caz.

a. Să se elaboreze calculele numerice corespunzătoare aplicării algoritmului K-means pe acest set de date. Care sunt clusterele finale şi centroizii corespunzători?

b. Se consideră algoritmul EM/GMM descris în enunţul problemei 15. El a fost conceput pentru a face estimarea parametrilor unui model de mixtură de k (aici $k = 2$) distribuţii gaussiene care au aceeaşi varianţă σ^2, iar probabilităţile a priori de selecţie sunt egale ($1/2$). Prin urmare, se vor estima doar mediile celor două distribuţii gaussiene (i.e., centroizii clusterelor corespunzătoare), aplicându-se formulele următoare:[872]

Pasul E:

$$E[Z_{ij}] = P(Z_{ij} = 1 | X = x_i; \mu, \sigma^2) \overset{F.Bayes}{=} \frac{\exp\Big(-\frac{1}{2\sigma^2}(x_i - \mu_j)^2\Big)}{\sum_{p=1}^{k} \exp\Big(-\frac{1}{2\sigma^2}(x_i - \mu_p)^2\Big)}$$

Pasul M:

$$\mu_j = \frac{\sum_{i=1}^{n} E[Z_{ij}] x_i}{\sum_{i=1}^{n} E[Z_{ij}]}$$

unde simbolul exp **denotă funcţia exponenţială (e^x), $n = 15$ este numărul de puncte, $k = 2$ este numărul de clustere, $\mu \overset{not.}{=} (\mu_1, \mu_2)$, iar Z_{ij} sunt variabilele-indicator „ascunse“ / „latente“, cu $i = \overline{1, n}$ şi $j = \overline{1, k}$.**

Să se calculeze valorile numerice corespunzătoare execuţiei primei iteraţii a acestui algoritm, pornind ca şi la punctul a cu valorile iniţiale $\mu_1 = -20$, $\mu_2 = -10$ şi considerând varianţa (fixată) $\sigma^2 = 1$.

[872] Pentru deducerea acestor formule, vedeţi problema 15 sau cartea *Machine Learning*, de Tom Mitchell, 1997, pag. 193, 195–196.

c. Se consideră o altă variantă a algoritmului EM (de asemenea pentru GMM, cazul unidimensional), în care însă se estimează toţi parametrii (μ, σ, π). Ecuaţiile de actualizare pentru parametrii primei distribuţii gaussiene (desemnată prin a) sunt date mai jos, iar cele pentru cea de-a doua gaussiană (b) sunt obţinute pur şi simplu înlocuind a cu b:[873]

Pasul E:

$$a_i \overset{not.}{=} P(a \mid x_i) \overset{F.Bayes}{=} \frac{p(x_i \mid a) \cdot \pi_a}{p(x_i \mid a) \cdot \pi_a + p(x_i \mid b) \cdot \pi_b}, \text{ unde}$$

$$p(x_i \mid a) \overset{def.}{=} \frac{1}{\sqrt{2\pi}\sigma_a} \cdot \exp\Big(-\frac{(x_i - \mu_a)^2}{2\sigma_a^2}\Big)$$

Pasul M:

$$\mu_a = \frac{a_1x_1 + a_2x_2 + \cdots + a_nx_n}{a_1 + a_2 + \cdots + a_n}$$

$$\sigma_a^2 = \frac{a_1(x_1 - \mu_a)^2 + a_2(x_2 - \mu_a)^2 + \cdots + a_n(x_n - \mu_a)^2}{a_1 + a_2 + \cdots + a_n}$$

$$\pi_a = (a_1 + a_2 + \cdots + a_n)/n$$

Presupunem că valorile iniţiale pentru parametri sunt: $\mu_a = -20, \mu_b = -10, \sigma_a^2 = \sigma_b^2 = 1$ şi $\pi_a = \pi_b = 0.5$. Efectuaţi o singură iteraţie a acestui algoritm EM, pentru a determina noile valori ale parametrilor μ_a, μ_b, σ_a^2, σ_b^2, π_a şi π_b. Continuând rularea acestui algoritm EM, valorile parametrilor vor rămane neschimbate? Dacă da, de ce? Dacă nu, de ce?

Răspuns:

a. Algoritmul K-means va executa următoarele iteraţii:
Iteraţia 1:

$$\mu_1 = -20 \qquad C_1 = \emptyset$$
$$\mu_2 = -10 \qquad C_2 = \{-9, -8, -7, -6, -5, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9\}$$

Iteraţia 2:

$$\mu_1 = -20 \qquad C_1 = \{-9\}$$
$$\mu_2 = \frac{-9 + \ldots + 9}{15} = \frac{7}{3} = 2.(3) \qquad C_2 = \{-8, -7, -6, -5, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9\}$$

Iteraţia 3:

$$\mu_1 = -9 \qquad C_1 = \{-9, -8, -7, -6, -5\}$$
$$\mu_2 = \frac{-8 + \ldots + 9}{14} = \frac{44}{14} = \frac{22}{7} \approx 3.143 \qquad C_2 = \{5, 5, 6, 6, 7, 7, 8, 8, 9, 9\}$$

Iteraţia 4:

$$\mu_1 = \frac{-9 - 8 - 7 - 6 - 5}{5} = -7 \qquad C_1 = \{-9, -8, -7, -6, -5\}$$
$$\mu_2 = \frac{2(5 + 6 + 7 + 8 + 9)}{10} = 7 \qquad C_2 = \{5, 5, 6, 6, 7, 7, 8, 8, 9, 9\}$$

[873] Pentru deducerea acestor formule, vedeţi rezolvarea problemei 18.

După această iteraţie algoritmul K-means se opreşte, întrucât centroizii nu se mai modifică. Aşadar, clusterele finale sunt:

$$C_1 = \{-9, -8, -7, -6, -5\} \text{ şi } C_2 = \{5, 5, 6, 6, 7, 7, 8, 8, 9, 9\}, \text{ cu } \mu_1 = -7 \text{ şi } \mu_2 = 7.$$

b. O iteraţie a algoritmului EM/GMM în această variantă, cu valorile iniţiale $\mu_1 = -20$, $\mu_2 = -10$ şi $\sigma^2 = 1$ constă în calcularea mediilor variabilelor „ascunse" Z_{ij} (pasul E) şi apoi recalcularea mediilor, adică a parametrilor μ_1 şi μ_2 (pasul M).

Aşadar, la pasul E, pentru punctul $x_1 = -9$, vom avea:

$$E[Z_{11}] = \frac{\exp(-\frac{1}{2}(-9+20)^2)}{\exp(-\frac{1}{2}(-9+10)^2) + \exp(-\frac{1}{2}(-9+20)^2)} = \frac{\exp(-\frac{121}{2})}{\exp(-\frac{1}{2}) + \exp(-\frac{121}{2})} = \frac{e^{-60}}{1+e^{-60}}$$

$$E[Z_{12}] = \frac{\exp(-\frac{1}{2}(-9+10)^2)}{\exp(-\frac{1}{2}(-9+10)^2) + \exp(-\frac{1}{2}(-9+20)^2)} = \frac{\exp(-\frac{1}{2})}{\exp(-\frac{1}{2}) + \exp(-\frac{121}{2})} = \frac{1}{1+e^{-60}}$$

Este evident că $P(Z_{11} = 1|x_1) = E[z_{11}] \approx 0$, iar $P(Z_{12} = 1|x_1) = E[Z_{12}] \approx 1$. Aşadar, dacă decidem să asociem fiecare instanţă x_i la clusterul / centroidul desemnat de $\text{argmax}_j E[Z_{ij}]$, punctul $x_1 = -9$ va fi asignat celui de-al doilea cluster.

Pentru punctul $x_2 = -8$:

$$E[Z_{21}] = \frac{\exp(-\frac{1}{2} \cdot 144)}{\exp(-\frac{1}{2} \cdot 4) + \exp(-\frac{1}{2} \cdot 144)} = \frac{e^{-70}}{1+e^{-70}}$$

$$E[Z_{22}] = \frac{\exp(-\frac{1}{2} \cdot 4)}{\exp(-\frac{1}{2} \cdot 4) + \exp(-\frac{1}{2} \cdot 144)} = \frac{1}{1+e^{-70}}$$

Ca şi mai sus, este evident că $E[Z_{21}] \approx 0$, iar $E[Z_{22}] \approx 1$. Deci punctul $x_2 = -8$ va fi asociat tot celui de-al doilea cluster.

Pentru punctul $x_3 = -7$:

$$E[Z_{31}] = \frac{\exp(-\frac{1}{2} \cdot 169)}{\exp(-\frac{1}{2} \cdot 9) + \exp(-\frac{1}{2} \cdot 169)} = \frac{e^{-80}}{1+e^{-80}}$$

$$E[Z_{32}] = \frac{\exp(-\frac{1}{2} \cdot 9)}{\exp(-\frac{1}{2} \cdot 9) + \exp(-\frac{1}{2} \cdot 169)} = \frac{1}{1+e^{-80}}$$

Este evident că $E[Z_{31}] \approx 0$, iar $E[Z_{32}] \approx 1$. Deci punctul $x_3 = -7$ va fi asociat de asemenea celui de-al doilea cluster.

Se observă că pentru celelalte puncte se va păstra acest comportament, deci $C_1 = \emptyset$ şi $C_2 = \{-9, -8, -7, -6, -5, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9\}$, iar $E[Z_{i1}] \approx 0, E[Z_{i2}] \approx 1$ pentru $i = \overline{1, 15}$.

Apoi, la pasul de *maximizare*, trebuie să calculăm noile valori ale mediilor:

$$\mu_1 = \frac{\frac{e^{-60}}{1+e^{-60}} \cdot (-9) + \frac{e^{-70}}{1+e^{-70}} \cdot (-8) + \frac{e^{-80}}{1+e^{-80}} \cdot (-7) + \dots}{\frac{e^{-60}}{1+e^{-60}} + \frac{e^{-70}}{1+e^{-70}} + \frac{e^{-80}}{1+e^{-80}} + \dots} \approx -9$$

$$\mu_2 \approx \frac{1 \cdot (-9) + \dots + 1 \cdot 9}{15} = \frac{35}{15} = \frac{7}{3} = 2.(3)$$

***Observaţie* (1):** Folosind o implementare în Matlab a acestui algoritm EM/GMM, valorile numerice pe care le-am obţinut pentru această iteraţie sunt:

x_i	-9	-8	-7	-6	-5
$E[Z_{i1}]$	$8.75 \cdot 10^{-27}$	$3.97 \cdot 10^{-31}$	$1.8 \cdot 10^{-35}$	$8.19 \cdot 10^{-40}$	$3.72 \cdot 10^{-44}$
$E[Z_{i2}]$	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1

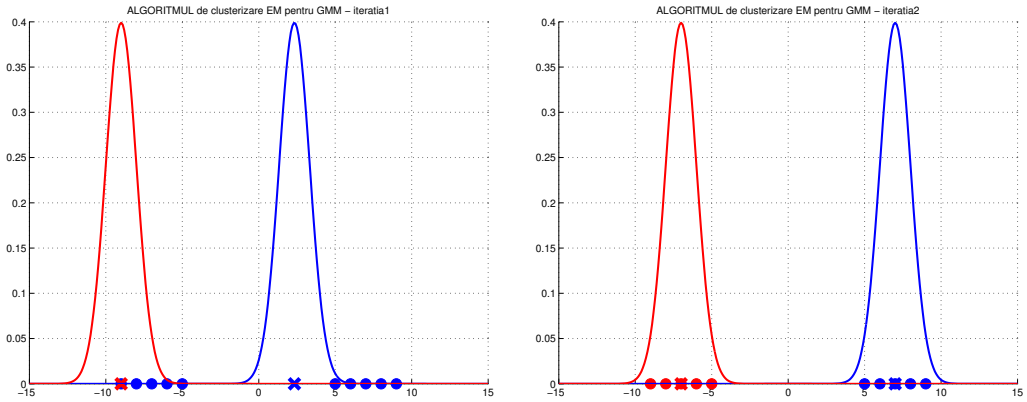
x_i	5	6	7	8	9
$E[Z_{i1}]$	$1.38 \cdot 10^{-87}$	$6.28 \cdot 10^{-92}$	$2.85 \cdot 10^{-96}$	$1.29 \cdot 10^{-100}$	$5.87 \cdot 10^{-105}$
$E[Z_{i2}]$	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1

$$\mu_1 = -8.999955 \text{ şi } \mu_2 = 2.333333$$

La următoarea iteraţie a acestui algoritm EM/GMM, componenţa clusterelor devine:

$$C_1 = \{-9, -8, -7, -6, -5\} \text{ şi } C_2 = \{5, 5, 6, 6, 7, 7, 8, 8, 9, 9\}, \text{ cu } \mu_1 \approx -7,\ \mu_2 \approx 7,$$

după cum se poate observa şi din graficul de mai jos, partea dreaptă:[874]



La următoarea iteraţie, algoritmul converge.[875]

***Observaţie* (2):** Aşadar, pentru cazul $\sigma^2 = 1$, soluţia obţinută de acest algoritm EM/GMM coincide cu cea a algoritmului K-means (deşi numărul de iteraţii diferă). În cele ce urmează vom arăta că pentru alte valori ale lui σ^2 se pot obţine rezultate semnificativ diferite pentru *valorile finale ale mediilor*:

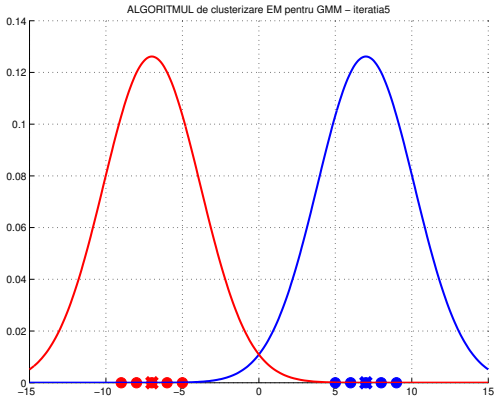
[874] Graficul din partea stângă corespunde finalului primei iteraţii.

[875] *Criteriul de oprire* pe care l-am folosit în implementarea noastră pentru algoritmul EM/GMM a fost următorul: la două iteraţii succesive, valorile mediilor nu se modifică cu mai mult de 10^{-5}.

Pentru $\sigma^2 = 20$, algoritmul EM/GMM converge după 10 iteraţii, obţinându-se aceleaşi clustere C_1 şi C_2, dar cu

$$\mu_1 = -6.657120 \text{ şi } \mu_2 = 6.940543.$$

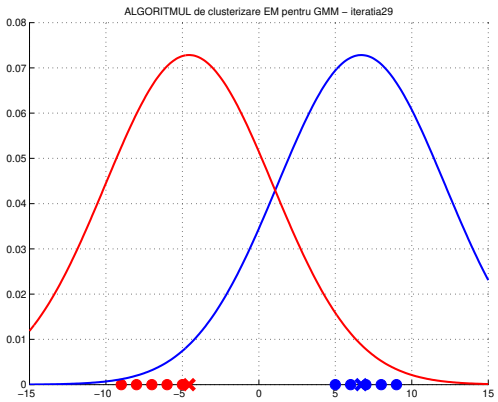
Se observă că aceste medii sunt foarte aproape de valorile 7 şi -7 obţinute de către algoritmul K-means, doar că au fost puţin „atrase“ spre punctele din clusterul opus (şi anume, mai mult μ_2 decât μ_1, pentru că în clusterul din dreapta sunt de două ori mai multe puncte decât în clusterul din stânga).



Pentru $\sigma^2 = 30$, algoritmul EM/GMM converge după 29 iteraţii, obţinându-se aceleaşi clustere ca mai sus, dar cu

$$\mu_1 = -4.564930 \text{ şi } \mu_2 = 6.698046.$$

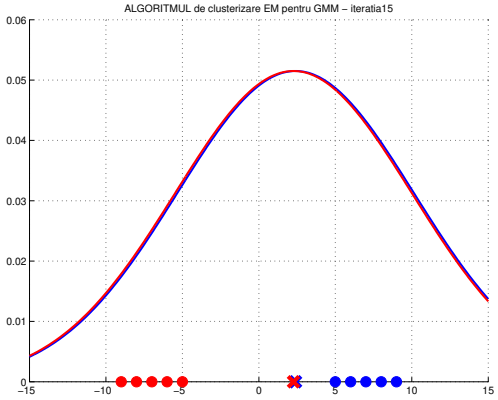
Media primei distribuţii gaussiene a migrat în mod considerabil spre cele 10 puncte din cel de-al doilea cluster, pe când media celei de-a doua distribuţii gaussiene este mult mai puţin „atras“ de cele 5 puncte din clusterul opus.



Pentru $\sigma^2 = 60$, algoritmul EM/GMM converge după 45 iteraţii, obţinându-se medii aproape identice:

$$\mu_1 = 2.333316 \text{ şi } \mu_2 = 2.333351.$$

Cu toate acestea, probabilităţile de apartenenţă ale punctelor la cele două clustere sunt sensibil diferite. De exemplu, pentru punctul -9 aceste probabilităţi sunt $E[Z_{11}] = 0.5000022$ şi $E[Z_{12}] = 0.4999978$. Acest fapt determină împărţirea punctelor în aceleaşi două clustere ca mai sus.

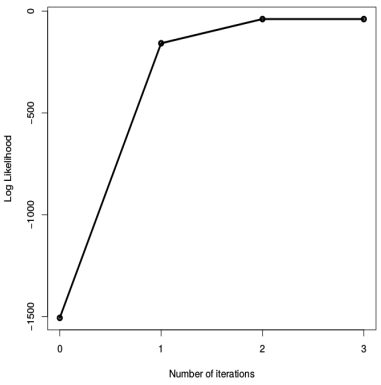


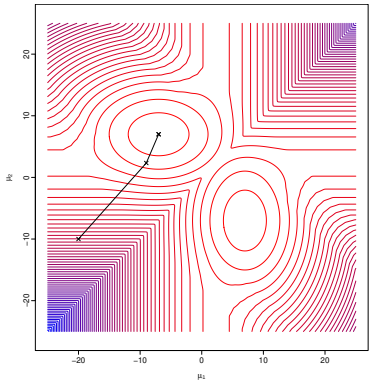
În imaginile următoare prezentăm,[876] pentru diferitele cazuri de mai sus (*i.* $\sigma^2 = 1$, *ii.* $\sigma^2 = 20$, *iii.* $\sigma^2 = 30$, *iv.* $\sigma^2 = 60$), valorile funcţiilor de log-verosimilitate a datelor „observabile“ obţinute de algoritmul EM la iteraţii succesive (vedeţi partea stângă) şi respectiv graficele acestor funcţii de log-verosimilitate reprezentate prin curbe de izocontur, pe care au fost puse în evidenţă valorile parametrilor μ_1 şi μ_2 obţinute de EM la iteraţii succesive (partea dreaptă).[877]

[876] Graficele au fost realizate de către studentul Sebastian Ciobanu în 2018.

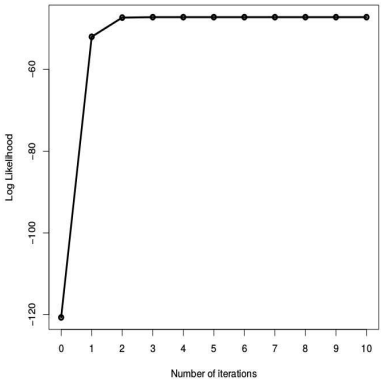
[877] Pentru un exemplu relativ simplu de calculare a funcţiei de log-verosimilitate a datelor observabile în cazul unei mixturi de două distribuţii gaussiene unidimensionale, vedeţi ex. 16.c.

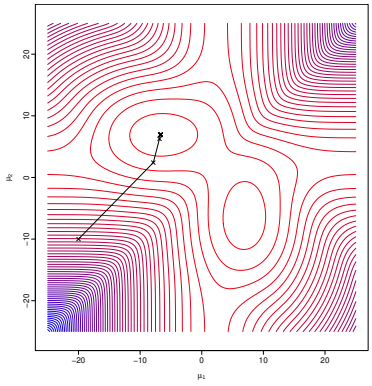
Cazul *i*.
$\sigma^2 = 1$.



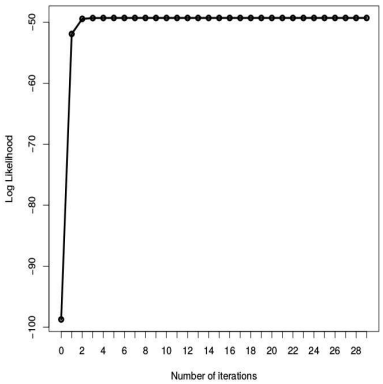


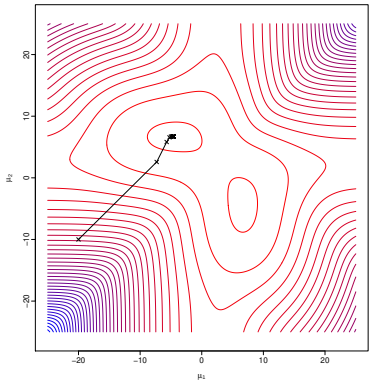
Cazul *ii*.
$\sigma^2 = 20$.



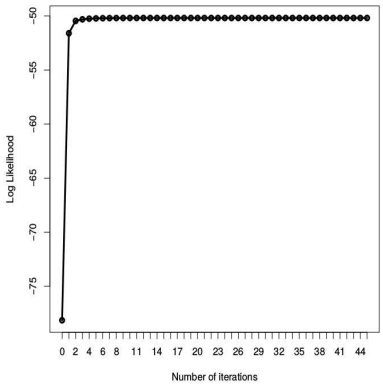


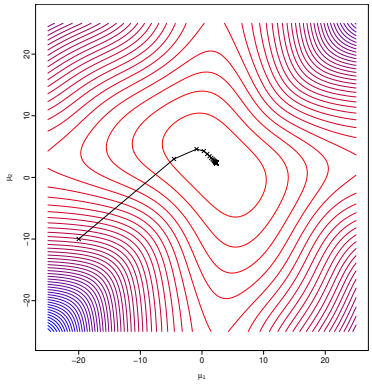
Cazul *iii*.
$\sigma^2 = 30$.





Cazul *iv*.
$\sigma^2 = 60$.





Se observă, ca şi la problema 16, că valorile funcţiei de log-verosimilitate a datelor „observabile“ obţinute de algoritmul EM cresc de la o iteraţie la alta, aşa cum era de aşteptat (vedeţi problema 2 de la capitolul *Schema algoritmică EM*).

c. Pentru a realiza o iteraţie a acestei variante algoritmului EM/GMM trebuie să calculăm „densităţile“ (adică, valorile funcţiilor densitate de probabilitate ale celor două gaussiene) $p(x_i \mid a)$ şi $p(x_i \mid b)$ şi probabilităţile a posteriori $a_i \overset{not.}{=} P(a \mid x_i)$ şi $b_i \overset{not.}{=} P(b \mid x_i)$ corespunzătoare fiecărui punct x_i din setul de date considerat.

Spre exemplu, pentru punctul -9, vom obţine:

$$p(-9 \mid a) = \frac{1}{\sqrt{2\pi}} \cdot \exp\left(-\frac{(-9+20)^2}{2}\right) = \frac{1}{e^{60}\sqrt{2\pi e}} \approx 2 \cdot 10^{-27}$$

$$p(-9 \mid b) = \frac{1}{\sqrt{2\pi}} \cdot \exp\left(-\frac{(-9+10)^2}{2}\right) = \frac{1}{\sqrt{2\pi e}} \approx 0.24$$

$$a_1 \overset{T.\ Bayes}{=} \frac{2 \cdot 10^{-27} \cdot 0.5}{0.24 \cdot 0.5 + 2 \cdot 10^{-27} \cdot 0.5} = 8 \cdot 10^{-27}$$

$$b_1 \overset{T.\ Bayes}{=} \frac{0.24 \cdot 0.5}{0.24 \cdot 0.5 + 2 \cdot 10^{-27} \cdot 0.5} \approx 1$$

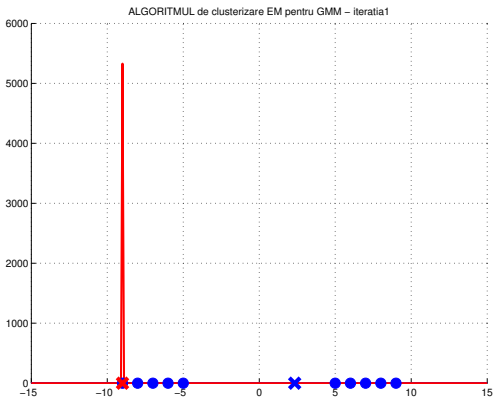
În mod analog se fac calculele pentru toate punctele. Folosind o [altă] implementare în Matlab, am obţinut:

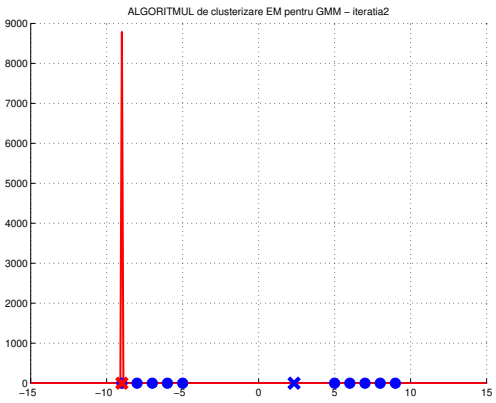
x_i	-9	-8	-7	-6	-5
$p(x_i \mid a)$	$2 \cdot 10^{-27}$	$2 \cdot 10^{-32}$	$8 \cdot 10^{-38}$	10^{-43}	10^{-50}
$p(x_i \mid b)$	0.24	0.05	0.004	0.0001	$1.4 \cdot 10^{-6}$
a_i	$8 \cdot 10^{-27}$	10^{-31}	10^{-35}	10^{-40}	10^{-44}
b_i	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1

x_i	5	6	7	8	9
$p(x_i \mid a)$	10^{-137}	10^{-148}	10^{-159}	10^{-171}	10^{-184}
$p(x_i \mid b)$	10^{-50}	10^{-56}	10^{-64}	10^{-71}	10^{-79}
a_i	10^{-87}	10^{-92}	10^{-96}	10^{-100}	10^{-105}
b_i	≈ 1	≈ 1	≈ 1	≈ 1	≈ 1

Apoi se calculează $\pi_a = \frac{1}{n}\sum_{i=1}^{15} a_i \simeq 5 \cdot 10^{-28}$ **şi** $\pi_b = \frac{1}{n}\sum_{i=1}^{15} b_i \simeq 1$**, deci toate punctele vor fi asignate clusterului** b**, iar noile medii şi varianţe vor fi:**[878]

$$\begin{array}{ll} \mu_a \simeq -9, & \mu_b = 2.(3) \\ \sigma_a = 0.000045, & \sigma_b = 45.55 \end{array}$$





La cea de-a doua iteraţie, varianţa componentei a **se va apropia foarte mult de** 0**, ceea ce va cauza un fapt suprinzător: cu toate că** $\mu_a \simeq -9$**, punctul** $x_1 = -9$ **va fi asociat celuilalt cluster, datorită varianţei foarte mici a lui** a**, precum şi datorită probabilităţii foarte mici de selecţie a acestei gaussiene** ($\pi_a \simeq 0$)**. Deci clusterele finale vor fi:**

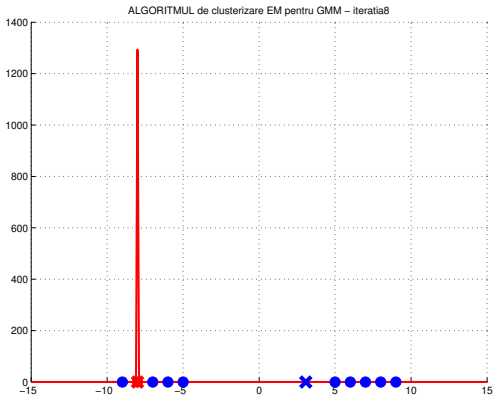
$$C_a = \emptyset \text{ şi } C_b = \{-9, -8, -7, -6, -5, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9\}$$

***Observaţie* (3):** Soluţia pe care tocmai am obţinut-o cu această variantă a algoritmului EM/GMM nu este satisfăcătoare. Totuşi, ca şi la punctul b, vom arăta că rezultatul se poate schimba, în funcţie de valorile iniţiale considerate. Concret, dacă se variază iniţializarea varianţelor, se pot obţine alte clustere, ca în exemplele următoare:

Pentru $\sigma_a^2 = \sigma_b^2 = 9$, algoritmul converge după 8 iteraţii, obţinându-se:

$$\begin{array}{ll} \mu_a \simeq -8, & \mu_b = 3.071348 \\ \sigma_a = 0.000308, & \sigma_b = 40.638356 \\ \pi_a = 0.066659, & \pi_b = 0.933340 \end{array}$$

Punctul $x_2 = -8$ va fi asociat gaussienei a, întrucât probabilităţile de apartenenţă la cele două clustere sunt: $a_2 = E[Z_{21}] = 0.999897$, $b_2 = E[Z_{22}] = 0.000102$, însă restul punctelor vor fi asignate celuilalt cluster.



Deci, la finalul execuţiei algoritmului vom avea:

$$C_a = \{-8\} \text{ şi } C_b = \{-9, -7, -6, -5, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9\}.$$

[878] La următoarele trei grafice nu am pus în evidenţă p.d.f.-urile corespunzătoare gaussienei b din cauza varianţelor prea mari. (În astfel de situaţii, curbele gaussiene nu se disting în mod semnificativ de axa Ox.)

Este un rezultat interesant, care se datorează nerestricţionării varianţelor celor două distribuţii. Se pune astfel în evidenţă faptul că spre deosebire de cazul algoritmului K-means, *suprafeţele de separare determinate de algoritmul EM nu sunt în mod neapărat liniare!*

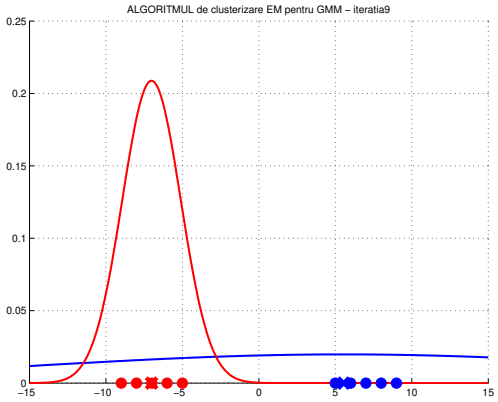
***Observaţie* (4):** La ultimele două rezultate se observă că valoarea funcţiei densitate de probabilitate (p.d.f.) poate fi foarte mare dacă parametrul σ ia valori foarte mici. Într-o astfel de situaţie, valoarea funcţiei de log-verosimilitate a datelor complete poate deveni foarte mare.[879] Astfel de situaţii (caracterizate de „puncte de singularitate“ pentru p.d.f.) nu sunt însă în general dezirabile pentru clusterizare.

Pentru $\sigma_a^2 = \sigma_b^2 = 20$, algoritmul EM/GMM converge după 9 iteraţii, obţinându-se:

$$\begin{aligned} \mu_a &= -7.018194, & \mu_b &= 5.550311 \\ \sigma_a &= 1.910617, & \sigma_b &= 20.137057 \\ \pi_a &= 0.255955, & \pi_b &= 0.744044 \end{aligned}$$

Clusterele obţinute în acest caz sunt cele aşteptate, şi anume:

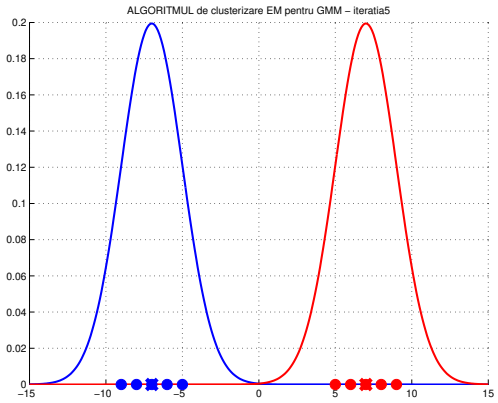
$$\begin{aligned} C_a &= \{-9, -8, -7, -6, -5\} \\ C_b &= \{5, 5, 6, 6, 7, 7, 8, 8, 9, 9\} \end{aligned}$$



Pentru $\sigma_a^2 = 20$ şi $\sigma_b^2 = 4$, acest algoritm EM/GMM converge după doar 5 iteraţii, obţinându-se un rezultat similar cu cel al algoritmului de la punctul b (unde s-au folosit valori identice pentru σ^2):

$$\begin{aligned} \mu_a &= 7, & \mu_b &= -7 \\ \sigma_a &= 2, & \sigma_b &= 2 \\ \pi_a &= 0.(6), & \pi_b &= 0.(3) \end{aligned}$$

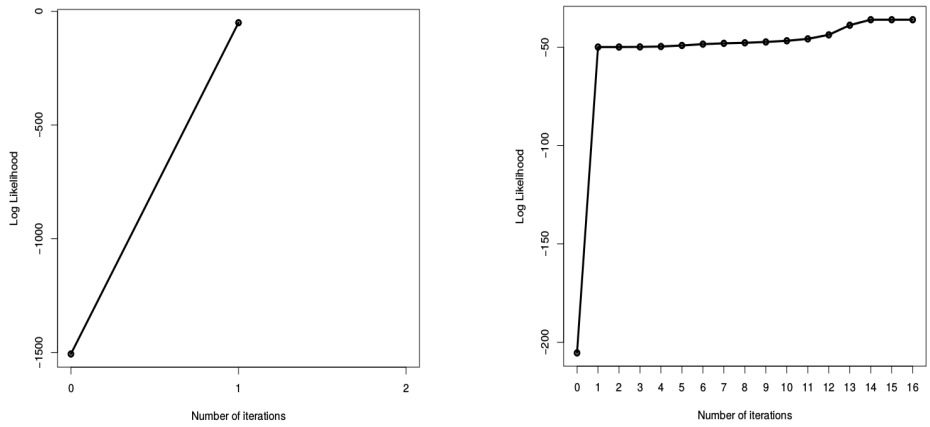
Mai mult, clusterele şi mediile / centroizii sunt aceiaşi ca şi în cazul algoritmului K-means, însă — interesant! — ele devin inversate: $C_a = \{5, 5, 6, 6, 7, 7, 8, 8, 9, 9\}$, $C_b = \{-9, -8, -7, -6, -5\}$



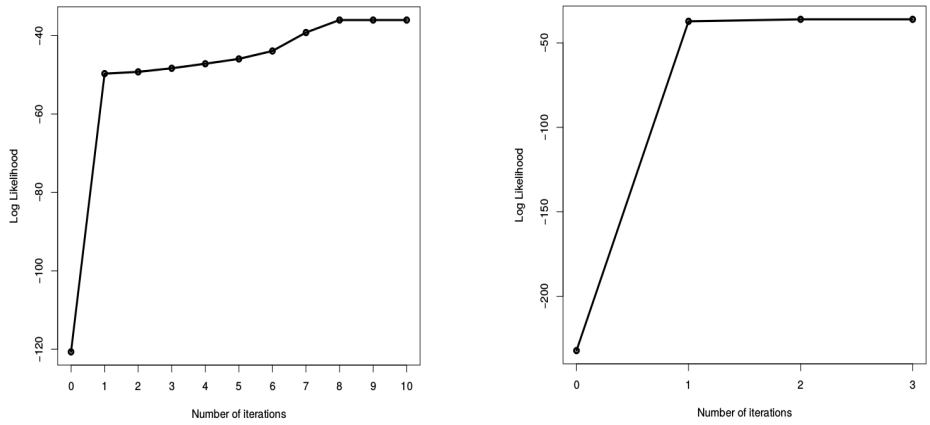
În imaginile următoare prezentăm,[880] pentru diferitele cazuri de mai sus (*i.* $\sigma_a^2 = \sigma_b^2 = 1$, *ii.* $\sigma_a^2 = \sigma_b^2 = 9$, *iii.* $\sigma_a^2 = \sigma_b^2 = 20$, *iv.* $\sigma_a^2 = 20, \sigma_b^2 = 4$), valorile funcţiilor de log-verosimilitate obţinute de algoritmul EM la iteraţii succesive.

[879] La distribuţii discrete, funcţia de log-verosimilitate a datelor complete (a cărei medie este maximizată de algoritmul EM) nu poate avea decât valori negative. (Justificarea este imediată, fiindcă funcţia masă de probabilitate nu poate lua valori decât in intervalul $[0, 1]$.) La distribuţii continue, funcţia de log-verosimilitate a datelor complete poate lua şi valori pozitive.

[880] Graficele au fost realizate de către studentul Sebastian Ciobanu în 2018.



Cazurile *i*. $\sigma_a^2 = \sigma_b^2 = 1$ **(partea stângă) şi, *ii*.** $\sigma_a^2 = \sigma_b^2 = 9$ **(partea dreaptă).**



Cazurile *iii*. $\sigma_a^2 = \sigma_b^2 = 20$ **(partea stângă) şi, *iv*.** $\sigma_a^2 = 20, \sigma_b^2 = 4$ **(partea dreaptă).**

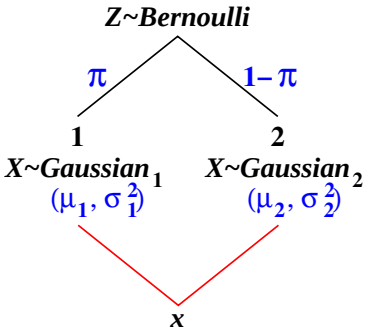
***Observaţie* (5): T. Hastie, R. Tibshirani şi J. Friedman recomandă în cartea *The Elements of Statistical Learning* (Springer, ed. a II-a, 2009) ca la pasul de iniţializare în algoritmul EM/GMM să se folosească pentru varianţele celor două distribuţii gaussiene individuale valoarea varianţei empirice a întregului set de date.**[881]

[881] Pentru un exemplu de calcul al varianţei empirice, vedeţi problema 124.b de la capitolul de *Fundamente*.

18. **(Algoritmul EM: estimarea tuturor parametrilor unei mixturi de două distribuţii gaussiene unidimensionale)**

Liviu Ciortuz, 2012, după
■ ● ○ *MIT, ML course 6768, 2012 fall, Dahua Lin, An Introduction to Expectation-Maximization*

Folosind algoritmul EM, rezolvaţi problema estimării parametrilor unei mixturi de două distribuţii gaussiene unidimensionale în cazul cel mai general, adică lăsând liberi toţi parametrii (μ – mediile, σ^2 – varianţele şi π – probabilităţile a priori de selecţie a celor două gaussiene).



Răspuns:

Vom urma etapele indicate de schema algoritmică EM, prezentată în secţiunea 6.12.2 din cartea *Machine Learning* de Tom Mitchell (sau, echivalent problema 1, pag. 981 de la capitolul *Schema algoritmică EM* din această culegere).

Pasul E:

Vom considera instanţele $x_1, \ldots, x_n \in \mathbb{R}$, **vom nota parametrii** $\mu = (\mu_1, \mu_2)$, $\sigma = (\sigma_1, \sigma_2)$ **şi** $\pi = (\pi_1, \pi_2)$ **şi vom folosi variabilele aleatoare** $Z_{i1}, Z_{i2} \in \{0, 1\}$ **cu restricţia** $Z_{i1} + Z_{i2} = 1$ **pentru orice** $i \in \{1, \ldots, n\}$. **În aceste condiţii, probabilităţile / mediile**

$$p_{ij} \overset{not.}{=} P(Z_{ij} = 1 \mid X_i, \mu, \sigma, \pi) \overset{calcul}{=} E[Z_{ij} \mid X_i, \mu, \sigma, \pi], \text{ pentru } i = 1, \ldots, n \text{ şi } j = 1, 2,$$

se vor obţine imediat folosind formula lui Bayes combinată cu formula probabilităţii totale:

$$\begin{aligned}
p_{ij} &= \frac{P(X_i = x_i \mid Z_{ij} = 1, \mu, \sigma, \pi) \cdot P(Z_{ij} = 1 \mid \mu, \sigma, \pi)}{\sum_{j'} P(X_i = x_i \mid Z_{ij'} = 1, \mu, \sigma, \pi) \cdot P(Z_{ij'} = 1 \mid \mu, \sigma, \pi)} \\
&= \frac{\frac{1}{\sqrt{2\pi}\sigma_j} \cdot \exp\left(-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}\right) \cdot \pi_j}{\frac{1}{\sqrt{2\pi}\sigma_1} \cdot \exp\left(-\frac{(x_i - \mu_1)^2}{2\sigma_1^2}\right) \cdot \pi_1 + \frac{1}{\sqrt{2\pi}\sigma_2} \cdot \exp\left(-\frac{(x_i - \mu_2)^2}{2\sigma_2^2}\right) \cdot \pi_2} \\
&= \frac{\frac{\pi_j}{\sigma_j} \cdot \exp\left(-\frac{(x_i - \mu_j)^2}{2(\sigma_j)^2}\right)}{\frac{\pi_1}{\sigma_1} \cdot \exp\left(-\frac{(x_i - \mu_1)^2}{2(\sigma_1)^2}\right) + \frac{\pi_2}{\sigma_2} \cdot \exp\left(-\frac{(x_i - \mu_2)^2}{2(\sigma_2)^2}\right)}.
\end{aligned}$$

***Verosimilitatea* oricărei instanţe „complete“** $y_i \overset{not.}{=} (x_i, z_{i1}, z_{i2})$ **se va exprima sub forma:**

$$\begin{aligned}
&P(X_i = x_i, Z_{i1} = z_{i1}, Z_{i2} = z_{i2} \mid \mu, \sigma, \pi) \\
&\quad = P(X_i = x_i \mid Z_{i1} = z_{i1}, Z_{i2} = z_{i2}, \mu_i, \sigma_i, \pi_i) \cdot P(Z_{i1} = z_{i1}, Z_{i2} = z_{i2} \mid \mu_i, \sigma_i, \pi_i)
\end{aligned}$$

$$= \frac{1}{\sqrt{2\pi}\sigma_{j'}} \cdot \exp\Big(- \frac{(x_i - \mu_{j'})^2}{2\sigma_{j'}^2} \Big) \cdot \pi_{j'}, \text{ unde } z_{ij'} = 1, \text{ iar } z_{ij''} = 0 \text{ pentru } j'' \neq j'.$$

Ţinând cont de egalitatea $z_{i1} + z_{i2} = 1$ şi de faptul că $z_{ij} \in \{0, 1\}$, putem continua dezvoltarea acestei expresii astfel:

$$\begin{aligned} &P(X_i = x_i, Z_{i1} = z_{i1}, Z_{i2} = z_{i2} \mid \mu, \sigma, \pi) \\ &\quad = \frac{1}{\sqrt{2\pi}\ \sigma_1^{z_{i1}}\ \sigma_2^{z_{i2}}} \cdot \exp\Big(- \frac{1}{2} \sum_{j \in \{1,2\}} z_{ij} \frac{(x_i - \mu_j)^2}{\sigma_j^2} \Big) \cdot \pi_1^{z_{i1}} \pi_2^{z_{i2}}. \end{aligned}$$

***Observaţie* (1): Acest mod de a scrie $P(X_i, Z_{i1}, Z_{i2} \mid \mu, \sigma, \pi)$, deşi pare cumva artificial, este de un ajutor crucial pentru că include atât Z_{i1} cât şi Z_{i2}. Ne pregătim astfel pentru aplicarea [ulterioră a] proprietăţii de liniaritate a mediei, la construirea funcţiei „auxiliare" Q.**

Logaritmând ultima expresie obţinută mai sus, vom obţine *log-verosimilitatea* instanţei „complete" $y_i = (x_i, z_{i1}, z_{i2})$:

$$\begin{aligned} &\ln P(X_i = x_i, Z_{i1} = z_{i1}, Z_{i2} = z_{i2} \mid \mu, \sigma, \pi) \\ &\quad = -\frac{1}{2} \ln(2\pi) - \sum_{j=1}^{2} z_{ij} \ln \sigma_j - \frac{1}{2} \sum_{j=1}^{2} z_{ij} \frac{(x_i - \mu_j)^2}{\sigma_j^2} + \sum_{j=1}^{2} z_{ij}\ \ln \pi_j. \end{aligned}$$

Notând $X = (X_1, \ldots, X_n)$, $Z_1 = (Z_{11}, Z_{21}, \ldots, Z_{n1})$ şi $Z_2 = (Z_{12}, Z_{22}, \ldots, Z_{n2})$, şi ţinând cont de independenţa statistică a datelor (x_i, z_{i1}, z_{i2}) cu $i = 1, \ldots, n$, *log-verosimilitatea datelor complete* va fi dată de expresia:

$$\begin{aligned} &\ln P(X, Z_1, Z_2 \mid \mu, \sigma, \pi) = \ln \prod_{i=1}^{n} P(X_i = x_i, Z_{i1}, Z_{i2} \mid \mu, \sigma, \pi) = \\ &\sum_{i=1}^{n} \ln P(X_i = x_i, Z_{i1}, Z_{i2} \mid \mu, \sigma, \pi) = \\ &-\frac{n}{2}\ \ln(2\pi) - \sum_{i=1}^{n} \sum_{j=1}^{2} Z_{ij} \ln \sigma_j - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{2} Z_{ij} \frac{(x_i - \mu_j)^2}{\sigma_j^2} + \sum_{i=1}^{n} \sum_{j=1}^{2} Z_{ij}\ \ln \pi_j. \end{aligned}$$

Mai departe, pentru a calcula *media log-verosimilităţii datelor complete*, vom ţine cont de proprietatea de liniaritate a mediilor. Aşadar, vom avea:

$$\begin{aligned} &E[\ln P(X, Z_1, Z_2 \mid \mu, \sigma, \pi)] = \\ &\quad -\frac{n}{2} \ln(2\pi) - \sum_{i=1}^{n} \sum_{j=1}^{2} E[Z_{ij}] \ln \sigma_j - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{2} E[Z_{ij}] \frac{(x_i - \mu_j)^2}{\sigma_j^2} + \sum_{i=1}^{n} \sum_{j=1}^{2} E[Z_{ij}]\ \ln \pi_j. \end{aligned}$$

Acum vom pune în evidenţă faptul că media aceasta se calculează în funcţie de probabilitatea condiţionată $P_{Z|X,\, \mu^{(t)}, \sigma^{(t)}, \pi^{(t)}}$, unde $\mu^{(t)}, \sigma^{(t)}, \pi^{(t)}$ notează valorile parametrilor la iniţializare ($t = 0$) sau, mai general, la iteraţia precedentă:

$$Q(\mu, \sigma, \pi \mid \mu^{(t)}, \sigma^{(t)}, \pi^{(t)}) \overset{not.}{=} E_{Z|X,\, \mu^{(t)}, \sigma^{(t)}, \pi^{(t)}}[\ln P(X, Z_1, Z_2 \mid \mu, \sigma, \pi)] =$$

$$-\frac{n}{2}\ln(2\pi) - \sum_{i=1}^{n}\sum_{j=1}^{2} E[Z_{ij} \mid X_i, \mu^{(t)}, \sigma^{(t)}, \pi^{(t)}] \ln \sigma_j$$
$$-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{2} E[Z_{ij} \mid X_i, \mu^{(t)}, \sigma^{(t)}, \pi^{(t)}] \frac{(x_i-\mu_j)^2}{\sigma_j^2}$$
$$+\sum_{i=1}^{n}\sum_{j=1}^{2} E[Z_{ij} \mid X_i, \mu^{(t)}, \sigma^{(t)}, \pi^{(t)}] \ln \pi_j.$$

Extinzând în mod natural notaţia $p_{ij} = E[Z_{ij} \mid X_i, \mu, \sigma, \pi]$, care a fost introdusă mai sus la $p_{ij}^{(t)} = E[Z_{ij} \mid X_i, \mu^{(t-1)}, \sigma^{(t-1)}, \pi^{(t-1)}]$, vom rescrie în mod simplificat expresia aşa-numitei *funcţii auxiliare* Q:

$$Q(\mu, \sigma, \pi \mid \mu^{(t)}, \sigma^{(t)}, \pi^{(t)}) =$$
$$-\frac{n}{2}\ln(2\pi) - \sum_{i=1}^{n}\sum_{j=1}^{2} p_{ij}^{(t)} \ln \sigma_j - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{2} p_{ij}^{(t)} \frac{(x_i-\mu_j)^2}{\sigma_j^2} + \sum_{i=1}^{n}\sum_{j=1}^{2} p_{ij}^{(t)} \ln \pi_j. \quad (369)$$

Pasul M:

Vom căuta optimul funcţiei $Q(\mu, \sigma, \pi \mid \mu^{(t)}, \sigma^{(t)}, \pi^{(t)})$ cu ajutorul derivatelor parţiale de ordinul întâi.

Pentru calculul derivatelor parţiale în raport cu π_j, vom ţine cont de restricţia $\pi_1 + \pi_2 = 1$:

$$Q(\mu, \sigma, \pi \mid \mu^{(t)}, \sigma^{(t)}, \pi^{(t)}) = -\frac{n}{2}\ln 2\pi - \sum_{i=1}^{n}\sum_{j=1}^{2} p_{ij}^{(t)} \ln \sigma_j$$
$$-\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{2} p_{ij}^{(t)} \frac{(x_i-\mu_j)^2}{\sigma_j^2} + \sum_{i=1}^{n}(p_{i1}^{(t)} \ln \pi_1 + p_{i2}^{(t)} \ln(1-\pi_1)).$$

Aşadar,

$$\frac{\partial}{\partial \pi_1} Q(\mu, \sigma, \pi \mid \mu^{(t)}, \sigma^{(t)}, \pi^{(t)}) = 0 \Leftrightarrow \frac{1}{\pi_1}\sum_{i=1}^{n} p_{i1}^{(t)} = \frac{1}{1-\pi_1}\sum_{i=1}^{n} p_{i2}^{(t)} \Leftrightarrow$$
$$\sum_{i=1}^{n} p_{i1}^{(t)} = \pi_1\Big(\sum_{i=1}^{n} p_{i1}^{(t)} + \sum_{i=1}^{n} p_{i2}^{(t)}\Big) \Leftrightarrow \sum_{i=1}^{n} p_{i1}^{(t)} = \pi_1 \sum_{i=1}^{n} \underbrace{(p_{i1}^{(t)} + p_{i2}^{(t)})}_{1} \Leftrightarrow \sum_{i=1}^{n} p_{i1}^{(t)} = n\pi_1,$$

de unde rezultă $\pi_1^{(t+1)} \leftarrow \frac{1}{n}\sum_{i=1}^{n} p_{i1}^{(t)}$. Apoi, ţinând cont de relaţiile $p_{i1}^{(t)} + p_{i2}^{(t)} = 1$ pentru $i = 1, \ldots, n$ şi $\pi_1^{(t+1)} + \pi_2^{(t+1)} = 1$, vom obţine imediat $\pi_2^{(t+1)} \leftarrow \frac{1}{n}\sum_{i=1}^{n} p_{i2}^{(t)}$. (Se poate demonstra uşor că $\sum_{i=1}^{n} p_{i1}^{(t)} \in (0, n)$ şi $\sum_{i=1}^{n} p_{i2}^{(t)} \in (0, n)$, deci $\pi_1^{(t+1)} \in (0, 1)$ şi $\pi_2^{(t+1)} \in (0, 1)$.)

Mai departe, vom deriva funcţia Q în raport cu μ_1:
$\frac{\partial}{\partial \mu_1} Q(\mu, \sigma, \pi \mid \mu^{(t)}, \sigma^{(t)}, \pi^{(t)}) = 0 \Leftrightarrow \frac{1}{\sigma_1^2}\sum_{i=1}^{n} p_{i1}^{(t)}(x_i - \mu_1) = 0 \Leftrightarrow \sum_{i=1}^{n} p_{i1}^{(t)}(x_i - \mu_1) = 0$,

de unde avem $\mu_1^{(t+1)} \leftarrow \frac{\sum_{i=1}^{n} p_{i1}^{(t)} x_i}{\sum_{i=1}^{n} p_{i1}^{(t)}}$. Similar, vom obţine $\mu_2^{(t+1)} \leftarrow \frac{\sum_{i=1}^{n} p_{i2}^{(t)} x_i}{\sum_{i=1}^{n} p_{i2}^{(t)}}$.

În final, vom deriva funcţia Q în raport cu σ_1:

$$\frac{\partial}{\partial \sigma_1} Q(\mu, \sigma, \pi \mid \mu^{(t)}, \sigma^{(t)}, \pi^{(t)}) = 0 \Leftrightarrow -\frac{1}{\sigma_1} \sum_{i=1}^{n} p_{i1}^{(t)} + \frac{1}{\sigma_1^3} \sum_{i=1}^{n} p_{i1}^{(t)} (x_i - \mu_1)^2 = 0,$$

de unde rezultă că $\left(\sigma_1^{(t+1)}\right)^2 \leftarrow \frac{\sum_{i=1}^{n} p_{i1}^{(t)} (x_i - \mu_1^{(t+1)})^2}{\sum_{i=1}^{n} p_{i1}^{(t)}}$.

Similar, vom obţine $\left(\sigma_2^{(t+1)}\right)^2 \leftarrow \frac{\sum_{i=1}^{n} p_{i2}^{(t)} (x_i - \mu_2^{(t+1)})^2}{\sum_{i=1}^{n} p_{i2}^{(t)}}$.
Evident, cele două fracţii de mai sus sunt strict pozitive.

***Observaţie* (2):**
Cititorul atent va sesiza că în final, la calculul expresiei lui $\sigma_1^{(t+1)}$, s-a înlocuit μ_1 cu $\mu_1^{(t+1)}$. Explicaţia, în manieră *intuitivă*, este următoarea: per ansamblu, aici calculăm maximul funcţiei Q nu doar în raport cu σ_1 ci şi cu ceilalţi parametri, deci în particular şi cu μ_1, a cărui valoare optimă la iteraţia $t+1$ este $\mu_1^{(t+1)}$. La calculul celorlalte valori optime ale parametrilor ($\pi^{(t+1)}$ şi $\mu^{(t+1)}$) nu a fost necesară o astfel de coroborare a soluţiilor. Cazul lui $\sigma^{(t+1)}$ este însă uşor diferit faţă de cazul celorlalţi parametri. În mod *riguros*, se poate demonstra că rădăcinile derivatelor parţiale de ordinul întâi ale funcţiei Q constituie într-adevăr valorile [parametrilor] pentru care această funcţie îşi atinge *maximul*.[882]

***Observaţie* (3):**
Este foarte util să comparăm *i.* forma relaţiilor de actualizare pentru parametrii μ_j şi σ_j^2 de la pasul M al algoritmului EM pentru modelarea unei mixturi de distribuţii gaussiene unidimensionale cu *ii.* formulele obţinute la estimarea în sens MLE a parametrilor μ şi σ^2 ai unei distribuţii gaussiene tot unidimensionale (vedeţi problemele 50.a şi 51.a de la capitolul de *Fundamente*). Se poate observa uşor că aceste două seturi de formule sunt foarte asemănătoare! Aşa cum era aşteptat, deosebirea dintre ele este dată de prezenţa probabilităţilor p_{ij} (calculate la pasul E) din algoritmul EM. Aşadar, formulele *i.* sunt versiuni *ponderate* ale formulelor *ii.* (deci reprezintă *medii ponderate*, respectiv *varianţe ponderate*).

***Observaţie* (4):**
O comparaţie similară se poate face şi între formula de actualizare a mediilor μ_j de la pasul M al algoritmului EM şi formula pentru recalcularea centroizilor μ_j la pasul al doilea al algoritmului K-means (a se vedea *în special* varianta lui K-means de la problema 45, unde se folosesc variabilele-indicator γ_{ij}).

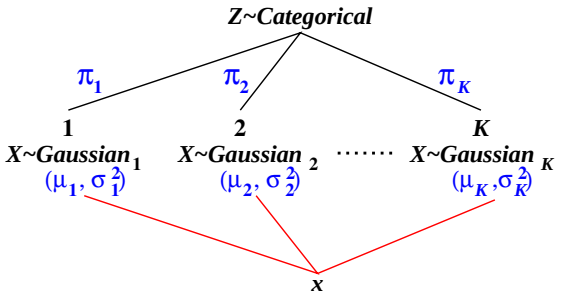
[882] Ideea demonstraţiei este similară cu cea de la problema 53.a de la capitolul de *Fundamente*. Şi anume, după fiecare etapă de calcul al unei derivate parţiale de la pasul M se pot scrie relaţii de forma

$$Q(\pi_1, \pi_2, \mu, \sigma | \pi^{(t)}, \mu^{(t)}, \sigma^{(t)}) \leq Q(\pi_1^{(t+1)}, \pi_2, \mu, \sigma | \pi^{(t)}, \mu^{(t)}, \sigma^{(t)}),\ \forall \pi_1, \pi_2, \mu, \sigma$$
$$Q(\pi_1^{(t+1)}, \pi_2, \mu, \sigma | \pi^{(t)}, \mu^{(t)}, \sigma^{(t)}) \leq Q(\pi_1^{(t+1)}, \pi_2^{(t+1)}, \mu, \sigma | \pi^{(t)}, \mu^{(t)}, \sigma^{(t)}),\ \forall \pi_2, \mu, \sigma$$

ş.a.m.d.

În final, putem face următoarea *generalizare*, pentru mixturi de $K > 2$ gaussiene unidimensionale:

$$\sum_{j=1}^{K} \pi_j N(x; \mu_j, \sigma_j^2),$$



cu $\sum_{j=1}^{K} \pi_j = 1$.

Este de observat faptul că, în acest caz, în mixtură, distribuţia Bernoulli este înlocuită cu o distribuţie categorială, ale cărei valori $(1, \ldots, K)$ sunt luate cu probabilităţile $\pi_1, \ldots, \pi_K$. Pentru deducerea algoritmului EM în acest caz, singura schimbare de fond necesară în raport cu demonstraţia de mai sus se referă la actualizarea parametrilor π_j:

Întrucât $\pi_1 + \ldots + \pi_K = 1$, va trebui să rezolvăm următoarea *problemă de optimizare cu restricţii*:

$$\max_{\pi,\mu,\sigma} Q(\pi, \mu, \sigma | \pi^{(t)}, \mu^{(t)}, \sigma^{(t)})$$

$$\text{cu restricţia } \sum_{j=1}^{K} \pi_j = 1,$$

în care expresia lui Q provine din relaţia (369) înlocuind simbolurile de sumare $\sum_{j=1}^{2}$ cu $\sum_{j=1}^{K}$.[883] Folosind *metoda multiplicatorilor Lagrange*, vom introduce variabila $\lambda \in \mathbb{R}$, iar problema de mai sus va deveni:

$$\max_{\pi,\mu,\sigma} \Big(Q(\pi, \mu, \sigma | \pi^{(t)}, \mu^{(t)}, \sigma^{(t)}) + \lambda (1 - \sum_{j=1}^{K} \pi_j) \Big).$$

Derivând funcţia obiectiv a acestei probleme în raport cu variabila π_j, unde $j \in \{1, \ldots, K\}$, obţinem:

$$\frac{\partial}{\partial \pi_j} Q(\mu, \sigma, \pi \mid \mu^{(t)}, \sigma^{(t)}, \pi^{(t)}) = 0 \Leftrightarrow \sum_{i=1}^{n} p_{ij}^{(t)} \frac{1}{\pi_j} = \lambda \Leftrightarrow \pi_j^{(t+1)} = \frac{1}{\lambda} \sum_{i=1}^{n} p_{ij}^{(t)}.$$

Întrucât $\sum_{j=1}^{K} \pi_j^{(t+1)} = 1$, urmează că

$$\lambda = \sum_{j=1}^{K} \sum_{i=1}^{n} p_{ij}^{(t)} = \sum_{i=1}^{n} \underbrace{\sum_{j=1}^{K} p_{ij}^{(t)}}_{1} = \sum_{i=1}^{n} 1 = n.$$

Aşadar,

$$\pi_j^{(t+1)} \leftarrow \frac{1}{n} \sum_{i=1}^{n} p_{ij}^{(t)}.$$

Se observă că $0 \leq \pi_j^{(t+1)} \leq 1$ pentru orice j, întrucât termenii $p_{ij}^{(t)}$ reprezintă nişte probabilităţi (vedeţi pasul E).

[883] *Observaţie*: Nu mai adăugăm la această problemă de optimizare şi restricţiile $\pi_j \geq 0$ pentru $j = 1, \ldots, K$, întrucât vom arăta la final că soluţiile obţinute satisfac în mod implicit aceste restricţii.

În concluzie, corpul iterativ al algoritmului EM pentru estimarea [tuturor] parametrilor unei mixturi de K distribuţii gaussiene este:

Pasul E: Pentru $i = 1, \ldots, n$ şi $j = 1, \ldots, K$, calculează

$$p_{ij}^{(t)} \overset{not.}{=} P(z_{ij} = 1 \mid x_i; \mu^{(t)}, (\sigma^2)^{(t)}, \pi^{(t)}) = \frac{\mathcal{N}(x_i \mid \mu_j^{(t)}, (\sigma_j^2)^{(t)}) \cdot \pi_j^{(t)}}{\sum_{l=1}^{K} \mathcal{N}(x_i \mid \mu_l^{(t)}, (\sigma_l^2)^{(t)}) \cdot \pi_l^{(t)}}$$

$$\textbf{unde } \mathcal{N}(x_i \mid \mu_j, \sigma_j^2) \overset{def.}{=} \frac{1}{\sqrt{2\pi}\sigma_j} \cdot \exp\Big(- \frac{(x_i - \mu_j)^2}{2\sigma_j^2} \Big).$$

Pasul M: Pentru $j = 1, \ldots, K$, calculează

$$\pi_j^{(t+1)} \leftarrow \frac{1}{n} \sum_{i=1}^{n} p_{ij}^{(t)}$$

$$\mu_j^{(t+1)} \leftarrow \frac{\sum_{i=1}^{n} p_{ij}^{(t)} x_i}{\sum_{i=1}^{n} p_{ij}^{(t)}}$$

$$\Big(\sigma_j^{(t+1)}\Big)^2 \leftarrow \frac{\sum_{i=1}^{n} p_{ij}^{(t)} (x_i - \mu_j^{(t+1)})^2}{\sum_{i=1}^{n} p_{ij}^{(t)}}$$

Evident, aceste formule le generalizează pe cele care au fost deduse anterior, când s-a considerat $K = 2$.

***Observaţie* (5): Se va putea vedea la problema 24 că formulele care au fost obţinute aici pot fi derivate şi pe altă cale, şi anume particularizând numărul de dimensiuni (d) la valoarea 1 în formulele care care alcătuiesc corpul iterativ al algoritmului EM pentru mixturi de distribuţii gaussiene multidimensionale (vedeţi pag. 921).**

19. **(Algoritmul EM: chestiuni calitative / metodologice privind aplicarea în cazul unei mixturi de două distribuţii gaussiene unidimensionale presupunând probabilităţile de selecţie egale şi fixate)**

■ • ○ ***CMU, 2007 spring, Eric Xing, final exam, pr. 1.8***

A fost odată, cu mult timp în urmă, un sat care era situat într-o regiune cu sute de lacuri. În acele lacuri trăiau două specii de peşti, însă în fiecare lac, nu erau decât peşti dintr-o singură specie. (Peştii din lacuri diferite puteau fi de specii diferite.) Peştii din aceste două specii arătau identic, miroseau identic atunci când erau gătiţi şi aveau exact acelaşi gust delicios, însă una dintre specii era otrăvitoare şi oricine mânca peşti din specia respectivă murea. Singura diferenţă observabilă la aceste specii de peşti consta în efectul asupra nivelului pH-ului (aciditatea) apei din lacul în care trăiau. Lacurile cu peşti neotrăvitori aveau pH-ul distribuit conform unei distribuţii gaussiene de medie (μ_{sigur}) şi varianţă (σ^2_{sigur}) necunoscute, iar pH-ul din lacurile cu peşti otrăvitori era distribuit conform unei alte distribuţii gaussiene de medie (μ_{mortal}) şi varianţă (σ^2_{mortal}) de asemenea necunoscute. (Peştii otrăvitori cauzau o aciditate care era în general mai mare / ridicată decât în cazul celorlalţi peşti.)

Aşa cum era firesc, pentru a ieşi din încurcătură — adică pentru a determina caracterul otrăvitor ori neotrăvitor al peştilor din fiecare lac (sau dintr-un lac oarecare, din care încă nu s-au consumat peşti) în funcţie de nivelul pH-ului apei —, sătenii au apelat la învăţarea automată. Cu toate acestea, au avut loc dezbateri aprinse cu privire la modalitatea corectă de aplicare a algoritmului Expectation-Maximization la problema lor.

Pentru fiecare dintre modalităţile prezentate mai jos, indicaţi dacă ea constituie o aplicare corectă a algoritmului EM şi dacă va conduce la o estimare rezonabilă pentru parametrii μ şi σ^2 ai fiecărei clase.

a. Se ghicesc valorile iniţiale pentru parametrii μ şi σ^2 corespunzători fiecărei specii. (1) Pentru fiecare lac, se determină — folosind teorema lui Bayes şi distribuţiile gaussiene de parametri μ şi σ^2 — care este cea mai probabilă specie de peşti asociată lacului respectiv. (2) Se recalculează valorile pentru μ şi σ^2 utilizând metoda estimării de verosimilitate maximă. Se repetă iterativ paşii (1) şi (2) până se ajunge la convergenţă.

b. Pentru fiecare lac, se ghiceşte probabilitatea (iniţială) că lacul respectiv ar fi populat cu peşti neotrăvitori. (1) Folosind aceste probabilităţi, se estimează în sensul verosimilităţii maxime valorile μ şi σ^2 corespunzătoare fiecărei clase. (2) Utilizând aceste estimări pentru μ şi σ^2, se recalculează probabilităţile de 'siguranţă' pentru fiecare lac. Se repetă iterativ paşii (1) şi (2) până se ajunge la convergenţă.

c. Se calculează media şi varianţa nivelului de pH pentru toată mulţimea lacurilor. Se folosesc aceste valori pentru a iniţializa parametrii μ şi σ^2 corespunzători fiecărei specii de peşti. (1) Folosind aceste valori pentru μ şi σ^2, se calculează pentru fiecare lac probabilitatea să conţină peşti otrăvitori. (2) Se găsesc valorile de verosimilitate maximă pentru μ şi σ^2. Se repetă iterativ paşii (1) şi (2) până se ajunge la convergenţă.

Răspuns:

Din enunţul problemei, în mod implict putem presupune că $P(sigur) = P(mortal) = 1/2$. Varianta algoritmui EM/GMM utilizată în acest caz diferă doar uşor de varianta (mai generală) formulată la problema 18: probabilităţile π_{sigur} şi π_{mortal} se consideră totdeauna $1/2$ (deci nu se recalculează la pasul M).

a. Această modalitate de aplicare a algoritmului este cea clasică, deci corectă. Pasul (1) face în mod implicit calculul mediilor $E[z_{i,sigur}] = p(z_{i,sigur} = 1|x_i; \mu, \sigma^2)$ şi $E[z_{i,mortal}] = p(z_{i,mortal} = 1|x_i; \mu, \sigma^2)$, unde notaţiile folosite sunt cele clasice, iar pasul (2) reprezintă maximizarea funcţiei auxiliare, în speţă recalcularea parametrilor μ şi σ^2.

b. Această metodă de aplicare a algoritmului EM nu este la fel de evidentă ca metoda la punctul precedent, dar vom arăta că este totuşi corectă.

O primă diferenţă faţă de metoda anterioară constă în faptul că în loc să se ghicească valorile iniţiale pentru parametrii μ şi σ^2 corespunzători fiecărei clase / specii, se ghicesc probabilităţile iniţiale pentru 'siguranţa' fiecărui lac, adică $p(z_{i,sigur} = 1|x_i; \mu, \sigma^2)$ (chiar dacă numărul acestor probabilităţi este în mod normal mult mai mare decât al parametrilor μ şi σ^2). Ştim că $z_{i,sigur} + z_{i,mortal} = 1$, aşadar ghicirea probabilităţii $p(z_{i,sigur} = 1|x_i; \mu, \sigma^2)$ antrenează în mod implicit setarea probabilităţii $p(z_{i,mortal} = 1|x_i; \mu, \sigma^2)$. Prin urmare, la acest pas de

iniţializare se încearcă să se „ghicească“ valorile care ar trebui calculate la prima execuţie a pasului E din varianta clasică a algoritmului EM, dacă s-ar dispune de valori iniţiale pentru parametrii μ şi σ^2.

A doua diferenţă faţă de metoda clasică constă în inversarea paşilor din corpul iterativ al algoritmului EM: pasul (1) reprezintă maximizarea, iar pasul (2) estimarea.

Se observă că aceste două „diferenţe“ se compensează reciproc: valorile pentru parametrii μ şi σ^2 care sunt calculate / estimate la prima execuţie a pasului de maximizare (1) — chiar dacă probabilităţile „ghicite“ iniţial, pe care se bazează aceste calcule, sunt imperfecte sau chiar grav eronate — vor juca rolul valorilor iniţiale din varianta (clasică) de la punctul a. După acest prim pas, prezenta variantă a algoritmului EM se comportă aidoma cu cea de la punctul a.

c. Această metodă nu este corectă / convenabilă deoarece utilizează valori iniţiale egale pentru media (şi respectiv pentru varianţa) nivelului de pH pentru ambele clase de lacuri. Vom arăta că din acest motiv algoritmul rămâne „blocat“ la valorile calculate pentru μ şi σ^2 la prima iteraţie. Iată explicaţia:

La pasul (1) – estimare, se calculează:

$$P(mortal \mid x_i; \mu, \sigma^2) = \frac{P(x_i \mid mortal, \mu, \sigma^2) \cdot P(mortal)}{P(x_i \mid mortal, \mu, \sigma^2) \cdot P(mortal) + P(x_i \mid sigur, \mu, \sigma^2) \cdot P(sigur)}$$

Am văzut că din enunţul problemei putem presupune că $P(sigur) = P(mortal) = 1/2$. Întrucât s-a considerat $\mu_{sigur} = \mu_{mortal}$ şi $\sigma_{sigur} = \sigma_{mortal}$, rezultă că $P(x \mid sigur, \mu, \sigma^2) = P(x \mid mortal, \mu, \sigma^2)$ pentru orice x. Deci, conform formulei lui Bayes, vom obţine $P(sigur \mid x; \mu, \sigma^2) = 1/2$; prin urmare, şi $P(mortal \mid x; \mu, \sigma^2) = 1/2$ pentru orice x.

La pasul (2) – maximizare, valorile obţinute pentru parametrii μ şi σ^2 vor fi:[884] **$\mu_{sigur} = \mu_{mortal} = \frac{1}{n}\sum_{i=1}^{n} x_i$, iar $\sigma^2_{sigur} = \sigma^2_{mortal} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_{sigur})^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu_{mortal})^2$, unde x_i sunt nivelurile de pH ale lacurilor (şi anume, cele pentru care s-a măsurat acest nivel).**

Se observă uşor că valorile obţinute pentru $P(sigur \mid x, \mu, \sigma^2)$, $P(mortal \mid x, \mu, \sigma^2)$, μ_{sigur}, μ_{mortal}, σ^2_{sigur} şi σ^2_{mortal} vor rămâne neschimbate după execuţia primei iteraţii, deci practic algoritmul „staţionează“ / „buclează“.

20. (Algoritmul EM/GMM, cazul multidimensional, cu $\Sigma_j = \sigma^2 I$ pt. $j = 1, \ldots, K$: deducerea formulelor de actualizare)

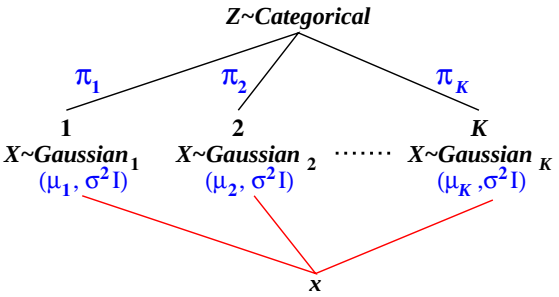
prelucrare făcută de Liviu Ciortuz, după
■ ○ U. Utah, 2009 fall, ML (CS5350/6350), Piyush Rai

A. Fie mixtura de distribuţii gaussiene

$$gmm(x) = \sum_{j=1}^{K} \pi_j \, \mathcal{N}(x|\mu_j, \sigma^2 I),$$

[884] A se vedea regulile de actualizare de la problema 18.

unde $x \in \mathbb{R}^d$, probabilităţile a priori de selecţie $\pi_j \in \mathbb{R}$ satisfac (ca de obicei) restricţiile $\pi_j \geq 0$ pentru $j = 1, \ldots, K$ şi $\sum_{j=1}^{K} \pi_j = 1$, mediile gaussienelor $j = 1, \ldots, K$ sunt vectorii $\mu_j \in \mathbb{R}^d$, iar matricele de covarianţă ale acestor gaussiene sunt identice, ba chiar au forma particulară $\sigma^2 I$, cu $\sigma \in \mathbb{R}$ şi $\sigma > 0$, matricea I fiind matricea identitate.



Se consideră instanţele $x_1, \ldots, x_n \in \mathbb{R}^d$ generate cu distribuţia probabilistă de mai sus (*gmm*). Vom asocia fiecărei instanţe x_i un vector-indicator (mai precis, un vector de variabile aleatoare $z_i \in \{0,1\}^K$, cu $z_i \overset{not.}{=} (z_{i1}, \ldots, z_{iK})$ şi $z_{ij} = 1$ dacă şi numai dacă x_i a fost generat de gaussiana $\mathcal{N}(x|\mu_j, \sigma^2 I)$.

Deduceţi regulile de actualizare din cadrul [pasului E şi al pasului M al] algoritmului EM. După cum ştim, acest algoritm face estimarea parametrilor $\pi \overset{not.}{=} (\pi_1, \ldots, \pi_K), \mu \overset{not.}{=} (\mu_1, \ldots, \mu_K)$ şi σ, asigurând la convergenţă atingerea unui maxim local pentru funcţia de log-verosimilitate a datelor „observabile“ $x_1, \ldots, x_n$.

Sugestie: Pentru rezolvarea problemei, vă recomandăm să parcurgeţi următoarele etape:

a. Se ştie că expresia funcţiei de densitate a distribuţiei gaussiene multidimensionale (mai precis, d-dimensionale) de medie $\mu \in \mathbb{R}^d$ şi matrice de covarianţă $\Sigma \in \mathbb{R}^{d \times d}$ este

$$\frac{1}{(\sqrt{2\pi})^d \sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1}(x-\mu)\right),$$

unde x şi μ sunt consideraţi vectori-coloană din $\mathbb{R}^d$, iar operatorul $\top$ desemnează operaţia de transpunere a vectorilor / matricelor.

Aduceţi expresia de mai sus la forma cea mai simplă pentru cazul $\Sigma = \sigma^2 I$.
Vă recomandăm să folosiţi faptul că $\|x-\mu\|^2 = (x-\mu)^\top (x-\mu) = (x-\mu) \cdot (x-\mu)$, unde operatorul $\cdot$ desemnează produsul scalar al vectorilor.

b. Pentru *pasul E* al algoritmului EM, veţi demonstra mai întâi că media $E[z_{ij}] \overset{not.}{=} E[z_{ij}|x_i; \pi, \mu, \sigma]$, unde $x_i = (x_{i,1}, \ldots, x_{i,d}) \in \mathbb{R}^d$, $\mu \overset{not.}{=} (\mu_1, \ldots, \mu_K) \in (\mathbb{R}^d)^K$ şi $\sigma \in \mathbb{R}^+$, are valoarea $P(z_{ij} = 1|x_i; \pi, \mu, \sigma)$, iar apoi veţi elabora formula de calcul a acestei probabilităţi, folosind teorema lui Bayes.

c. Arătaţi că expresia funcţiei de log-verosimilitate a datelor „complete“ în raport cu parametrii π, μ şi σ este

$$\ln p(x, z|\pi, \mu, \sigma) = \sum_{i=1}^{n} \sum_{j=1}^{K} z_{ij} (\ln \pi_j + \ln \mathcal{N}(x_i|\mu_j, \sigma^2 I)),$$

unde $x \overset{not.}{=} (x_1, \ldots, x_n)$ şi $z \overset{not.}{=} (z_1, \ldots, z_n)$.

Deduceţi apoi expresia „funcţiei auxiliare“ $Q(\pi, \mu, \sigma|\pi^{(t)}, \mu^{(t)}, \sigma^{(t)}) \overset{def.}{=} E[\ln p(x, z|\pi, \mu, \sigma)]$, cu precizarea că media aceasta este calculată în raport cu distribuţia / distribuţiile $P(z_{ij}|x_i, \pi^{(t)}, \mu^{(t)}, \sigma^{(t)})$.

Pasul M: Problema de optimizare EM în acest context este

$$(\pi^{(t+1)}, \mu^{(t+1)}, \sigma^{(t+1)}) = \underset{\pi,\mu,\sigma}{\operatorname{argmax}} Q(\pi, \mu, \sigma | \pi^{(t)}, \mu^{(t)}, \sigma^{(t)}), \tag{370}$$

$$\textit{cu restricţiile } \pi_j^{(t+1)} \geq 0 \textit{ pentru } j = 1, \ldots, K \textit{ şi } \sum_{j=1}^{K} \pi_j^{(t+1)} = 1.$$

Această problemă se rezolvă optimizând funcţia ei obiectiv în mod separat în raport cu variabilele π, μ şi σ.

d. Aplicaţi metoda multiplicatorilor lui Lagrange pentru a rezolva problema de optimizare cu restricţii (370) în raport (doar) cu variabilele π.

e. Optimizaţi funcţia Q (i.e., rezolvaţi problema de optimizare (370)) în raport cu variabilele μ.

Indicaţie: Următoarea formulă de *derivare vectorială* (preluată din documentul *Matrix Identities*, de Sam Roweis, 1999) vă poate fi de folos:

(5g) $$\frac{\partial}{\partial X}(Xa+b)^\top C(Xa+b) = (C + C^\top)(Xa+b)a^\top$$

f. Optimizaţi funcţia Q (i.e., rezolvaţi problema de optimizare (370)) în raport cu variabila σ.

g. Sumarizaţi rezultatele obţinute la punctele de mai sus (b şi d-f), redactând în pseudocod algoritmul EM pentru rezolvarea mixturii de gaussiene din enunţ.

B. Analizaţi ce se întâmplă atunci când modelul de mixtură de mai sus este generalizat la gaussiene cu matrice de covarianţă diagonale oarecare (adică acestea nu mai sunt de forma $\sigma^2 I$). Concret, modelul devine:

$$z_i \sim \textit{Categorical}(p_1, \ldots, p_K)$$

$$x_i | z_i = j \sim \mathcal{N}\left(\begin{bmatrix} \mu_{j,1} \\ \vdots \\ \mu_{j,d} \end{bmatrix}, \begin{bmatrix} (\sigma_{j,1})^2 & 0 & \ldots & 0 \\ 0 & (\sigma_{j,2})^2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \ldots & 0 & (\sigma_{j,d})^2 \end{bmatrix} \right)$$

Răspuns:

A. a. Întrucât $\Sigma = \sigma^2 I$, rezultă că $\Sigma^{-1} = \frac{1}{\sigma^2} I$. Avem, de asemenea, $|\Sigma| = (\sigma^2)^d$, ceea ce implică $\sqrt{|\Sigma|} = \sigma^d$, întrucât ştim că $\sigma > 0$, prin definiţie. Aşadar,

$$\begin{aligned} \mathcal{N}(x | \mu, \Sigma = \sigma^2 I) &\overset{def.}{=} \frac{1}{(\sqrt{2\pi})^d \sqrt{|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)^\top \Sigma^{-1} (x-\mu) \right) \\ &= \frac{1}{(\sqrt{2\pi})^d \sigma^d} \exp\left(-\frac{1}{2}(x-\mu)^\top \frac{1}{\sigma^2}(x-\mu) \right) \\ &= \frac{1}{(\sqrt{2\pi}\sigma)^d} \exp\left(-\frac{1}{2\sigma^2}\|x-\mu\|^2 \right). \end{aligned}$$

b. Pentru comoditate, în cele ce urmează vom renota $E[z_{ij} | x_i; \pi\mu, \sigma]$ prin $E[z_{ij}]$.

Aşadar,

$$\begin{aligned}
E[z_{ij}] &\overset{not.}{=} E[z_{ij}|x_i;\pi,\mu,\sigma] \overset{def.}{=} 0\cdot P(z_{ij}=0|x_i;\pi,\mu,\sigma)+1\cdot P(z_{ij}=1|x_i;\pi,\mu,\sigma)\\
&= P(z_{ij}=1|x_i;\pi,\mu,\sigma)\\
&\overset{F.\ Bayes}{=} \frac{P(x_i|z_{ij}=1;\pi,\mu,\sigma)\cdot P(z_{ij}=1;\pi,\mu,\sigma)}{P(x_i;\pi,\mu,\sigma)}\\
&\overset{F.P.T.}{=} \frac{P(x_i|z_{ij}=1;\pi,\mu,\sigma)\cdot P(z_{ij}=1;\pi,\mu,\sigma)}{\sum_{j'=1}^{K} P(x_i|z_{ij'}=1;\pi,\mu,\sigma)\cdot P(z_{ij'}=1;\pi,\mu,\sigma)}\\
&\overset{a.}{=} \frac{\frac{1}{(\sqrt{2\pi}\sigma)^d}\exp\left(-\frac{1}{2\sigma^2}\|x_i-\mu_j\|^2\right)\ \pi_j}{\sum_{j'=1}^{K}\frac{1}{(\sqrt{2\pi}\sigma)^d}\exp\left(-\frac{1}{2\sigma^2}\|x_i-\mu_{j'}\|^2\right)\ \pi_{j'}}\\
&= \frac{\pi_j\ \exp\left(-\frac{1}{2\sigma^2}\|x_i-\mu_j\|^2\right)}{\sum_{j'=1}^{K}\pi_{j'}\ \exp\left(-\frac{1}{2\sigma^2}\|x_i-\mu_{j'}\|^2\right)}.
\end{aligned} \tag{371}$$

c. Log-verosimilitatea datelor complete este

$$\begin{aligned}
\ln p(x,z|\pi,\mu,\sigma) &\overset{def.}{=} \ln p((x_1,z_1),\ldots,(x_n,z_n)|\pi,\mu,\sigma) \overset{i.i.d.}{=} \ln\prod_{i=1}^{n} p(x_i,z_i|\pi,\mu,\sigma)\\
&= \sum_{i=1}^{n}\ln p(x_i,z_i|\pi,\mu,\sigma) \overset{reg.\ mult.}{=} \sum_{i=1}^{n}\ln\left(p(x_i|z_i;\pi,\mu,\sigma)\cdot\underbrace{p(z_i|\pi,\mu,\sigma)}_{\pi_j}\right)\\
&= \sum_{i=1}^{n}\sum_{j=1}^{K} z_{ij}\left[\ln\mathcal{N}(x_i|\mu_j,\sigma)+\ln\pi_j\right],
\end{aligned}$$

întrucât $z_i=(z_{i,1},\ldots,z_{i,j},\ldots,z_{i,K})$**, cu** $z_{i,j}=1$ **şi** $z_{i,j'}=0$ **pentru orice** $j'\neq j$**. Acum, folosind rezultatul pe care l-am obţinut la punctul** a**, putem scrie:**

$$\ln p(x,z|\pi,\mu,\sigma)=\sum_{i=1}^{n}\sum_{j=1}^{K} z_{ij}\left[-\frac{d}{2}\ln(2\pi\sigma^2)-\frac{1}{2\sigma^2}\|x_i-\mu_j\|^2+\ln\pi_j\right].$$

În fine, folosind proprietatea de liniaritate a mediei, obţinem:

$$\begin{aligned}
Q(\pi,\mu,\sigma|\pi^{(t)},\mu^{(t)},\sigma^{(t)}) &\overset{def.}{=} E[\ln p(x,z|\pi,\mu,\sigma)]\\
&= \sum_{i=1}^{n}\sum_{j=1}^{K} E[z_{ij}]\left[-\frac{d}{2}\ln(2\pi\sigma^2)-\frac{1}{2\sigma^2}\|x_i-\mu_j\|^2+\ln\pi_j\right].
\end{aligned}$$

d. Deocamdată nu vom lua în calcul restricţiile $\pi_j\geq 0$**. În consecinţă,** ***funcţia lagrangeană*** **asociată problemei noastre de optimizare este:**

$$Q(\pi,\mu,\sigma|\pi^{(t)},\mu^{(t)},\sigma^{(t)})+\lambda(1-\sum_{j=1}^{K}\pi_j), \tag{372}$$

unde $\lambda \in \mathbb{R}$ este *multiplicator Lagrange*. (După ce vom obţine soluţia acestei probleme de optimizare se va vedea că $\pi_j \geq 0$ pentru orice $j = 1, \ldots, K$.)

Calculând derivata parţială a funcţiei Lagrange (372) **în raport cu π_j (cu j fixat în mulţimea $\{1, \ldots, K\}$) şi apoi egalând-o cu zero,**

$$\frac{\partial}{\partial \pi_j}(Q(\pi, \mu, \sigma | \pi^{(t)}, \mu^{(t)}, \sigma^{(t)}) + \lambda(1 - \sum_{j=1}^{K} \pi_j)) = 0 \Leftrightarrow$$

$$\sum_{i=1}^{n} E[z_{ij}] \frac{1}{\pi_j} = \lambda,$$

vom obţine soluţia

$$\pi_j^{(t+1)} = \frac{1}{\lambda} \sum_{i=1}^{n} E[z_{ij}]. \tag{373}$$

Apoi, ţinând cont că aceste valori trebuie să satisfacă restricţia $\sum_{j=1}^{K} \pi_j^{(t+1)} = 1$, rezultă că

$$\sum_{j=1}^{K} \frac{1}{\lambda} \sum_{i=1}^{n} E[z_{ij}] = 1 \Leftrightarrow \frac{1}{\lambda} \sum_{j=1}^{K} \sum_{i=1}^{n} E[z_{ij}] = 1 \overset{b.}{\Leftrightarrow} \frac{1}{\lambda} \sum_{i=1}^{n} \sum_{j=1}^{K} P(z_{ij} = 1 | x_i, \pi, \mu, \sigma) = 1 \Leftrightarrow$$

$$\frac{1}{\lambda} \sum_{i=1}^{n} \underbrace{\sum_{j=1}^{K} P(z_{ij} = 1 | x_i, \pi, \mu, \sigma)}_{1} = 1 \Leftrightarrow \frac{1}{\lambda} \sum_{i=1}^{n} 1 = 1 \Leftrightarrow \frac{n}{\lambda} = 1 \Leftrightarrow \lambda = n.$$

Înlocuind această valoare a lui λ în relaţia (373)**, obţinem**

$$\pi_j^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} E[z_{ij}]. \tag{374}$$

Este evident acum că $\pi_j^{(t+1)} \geq 0$.

e. Vă readucem aminte notaţia $\mu = (\mu_1, \ldots, \mu_K) \in (\mathbb{R}^d)^K$, cu $\mu_j \in \mathbb{R}^d$ pentru $j = 1, \ldots, K$. În loc să calculăm derivata parţială a funcţiei „auxiliare“ Q în raport cu fiecare componentă a vectorului μ_j, vom calcula direct gradientul lui Q în raport cu μ_j:

$$\begin{aligned}
&\nabla_{\mu_j} Q(\pi, \mu, \sigma | \pi^{(t)}, \mu^{(t)}, \sigma^{(t)}) \\
&= \nabla_{\mu_j} \sum_{i=1}^{n} \sum_{j'=1}^{K} E[z_{ij'}] \left[-\frac{d}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \|x_i - \mu_j'\|^2 + \ln \pi_{j'} \right] \\
&= \sum_{i=1}^{n} E[z_{ij}] \left[-\frac{1}{2\sigma^2} \nabla_{\mu_j} (x_i - \mu_j)^2 \right] \overset{(5g)}{=} -\frac{1}{2\sigma^2} \sum_{i=1}^{n} E[z_{ij}] 2(x_i - \mu_j)(-1) \\
&= \frac{1}{\sigma^2} \sum_{i=1}^{n} E[z_{ij}](x_i - \mu_j) = \frac{1}{\sigma^2} \left[\left(\sum_{i=1}^{n} E[z_{ij}] x_i \right) - \left(\sum_{i=1}^{n} E[z_{ij}] \right) \mu_j \right].
\end{aligned}$$

Egalând această expresie cu zero (de fapt, vectorul-coloană $(0,\ldots,0)^\top \in \mathbb{R}^d$), vom obţine soluţia:

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^n E[z_{ij}]x_i}{\sum_{i=1}^n E[z_{ij}]}. \tag{375}$$

f. Derivata parţială a funcţiei „auxiliare“ $Q(\pi,\mu,\sigma|\pi^{(t)},\mu^{(t)},\sigma^{(t)})$ în raport cu σ este:

$$\begin{aligned}
&\frac{\partial}{\partial\sigma}Q(\pi,\mu,\sigma|\pi^{(t)},\mu^{(t)},\sigma^{(t)}) \\
&= \frac{\partial}{\partial\sigma}\sum_{i=1}^n\sum_{j=1}^K E[z_{ij}]\left[-\frac{d}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|x_i-\mu_j\|^2 + \ln\pi_j\right] \\
&= \sum_{i=1}^n\sum_{j=1}^K E[z_{ij}]\frac{\partial}{\partial\sigma}\left[-\frac{d}{2}\ln(2\pi\sigma^2) - \frac{1}{2\sigma^2}\|x_i-\mu_j\|^2\right] \\
&= \sum_{i=1}^n\sum_{j=1}^K E[z_{ij}]\left[-\frac{d}{2}\cdot\frac{2\sigma}{\sigma^2} - \frac{-2}{2\sigma^3}\|x_i-\mu_j\|^2\right] = \sum_{i=1}^n\sum_{j=1}^K E[z_{ij}]\left[-\frac{d}{\sigma} + \frac{1}{\sigma^3}\|x_i-\mu_j\|^2\right] \\
&= \frac{1}{\sigma^3}(-d\sigma^2\underbrace{\sum_{i=1}^n\sum_{j=1}^K E[z_{ij}]}_{1} + \sum_{i=1}^n\sum_{j=1}^K E[z_{ij}]\|x_i-\mu_j\|^2) \\
&= \frac{1}{\sigma^3}(-nd\sigma^2 + \sum_{i=1}^n\sum_{j=1}^K E[z_{ij}]\|x_i-\mu_j\|^2)
\end{aligned}$$

Egalând această expresie cu 0, obţinem soluţia:

$$(\sigma^{(t+1)})^2 = \frac{1}{nd}\sum_{i=1}^n\sum_{j=1}^K E[z_{ij}]\|x_i-\mu_j^{(t+1)}\|^2 \geq 0. \tag{376}$$

Nu este dificil de demonstrat faptul că această soluţie corespunde maximizării funcţiei Q în raport cu σ^2. (Se ştie că $\sigma > 0$, iar expresia $-nd\sigma^2 + \sum_{i=1}^n\sum_{j=1}^K E[z_{ij}]\|x_i-\mu_j\|^2$, văzută ca funcţie de σ^2, este liniară şi descrescătoare.)

g. Putem acum să scriem pseudo-codul algoritmului EM pentru rezolvarea acestei mixturi de distribuţii gaussiene multidimensionale, cu ajutorul relaţiilor (371), (374), (375) şi (376):

- **Iniţializează [cu valori arbitrare] probabilităţile a priori π, mediile μ şi varianţa σ^2.**
- **Iterează până când o anumită *condiţie de oprire* este satisfăcută:**

  ***Pasul E*: Calculează mediile (adică, probabilităţile a posteriori) pentru variabilele z:**

$$p_{ij}^{(t)} \overset{not.}{=} E[z_{ij}|x_i;\pi^{(t)},\mu^{(t)},\sigma^{(t)}] = \frac{\pi_j^{(t)}\,\mathcal{N}(x_i|\mu_j^{(t)},(\sigma^{(t)})^2I)}{\sum_{j'=1}^K \pi_{j'}^{(t)}\,\mathcal{N}(x_i|\mu_{j'}^{(t)},(\sigma^{(t)})^2I)}; \tag{377}$$

***Pasul M*: Calculează noile valori pentru π, μ şi σ:**

$$\begin{aligned}
\pi_j^{(t+1)} &= \frac{1}{n}\sum_{i=1}^{n} p_{ij}^{(t)}; \\
\mu_j^{(t+1)} &= \frac{\sum_{i=1}^{n} p_{ij}^{(t)}\, x_i}{\sum_{i=1}^{n} p_{ij}^{(t)}}; \\
\left(\sigma^{(t+1)}\right)^2 &= \frac{1}{dn}\sum_{i=1}^{n}\sum_{j=1}^{K} p_{ij}^{(t)}\, \|x_i - \mu_j^{(t)}\|^2.
\end{aligned}$$

B. Se urmează linia demonstraţiei de la secţiunea A.

La calculul probabilităţilor de la pasul E, formula (377) **va fi rescrisă înlocuind în ea** $\mathcal{N}(x_i|\mu_j^{(t)}, (\sigma^{(t)})^2 I)$ **cu** $\mathcal{N}(x_i|\mu_j^{(t)}, \Sigma_j^{(t)})$**, unde** $\Sigma_j^{(t)}) = diag\,((\sigma_{j,1}^{(t)})^2, \ldots, (\sigma_{j,d}^{(t)})^2)$**.**

Se poate demonstra uşor că la pasul M singura *regulă de actualizare* care se schimbă la reformularea algoritmului EM este (376)**. Ea devine:**

$$(\sigma_{jk}^{(t+1)})^2 = \frac{\sum_{i=1}^{n} E[z_{ij}](x_{i,k} - \mu_{j,k}^{(t+1)})^2}{\sum_{i=1}^{n} E[z_{ij}]} \geq 0 \text{ pentru } j = 1, \ldots, K \text{ şi } k = 1, \ldots, d. \quad (378)$$

Acesta este exact lucrul la care ne aşteptam, date fiind rezultatele pe care le-am demonstrat la problema 20 de la capitolul de *Fundamente* şi la problema 18 din prezentul capitol (i.e., *Clusterizare*).

***Observaţie importantă*: La rezolvarea problemelor 21, 22 şi 23 am folosit varianta de algoritm EM/GMM (cazul multidimensional) descris la problema 20, unde se consideră că matricele de covarianţă sunt diagonale, ba chiar** $\Sigma_j = \sigma^2 I_d$**, unde** I_d **este matricea identitate de tip pătratic cu** d **linii şi** d **coloane.**

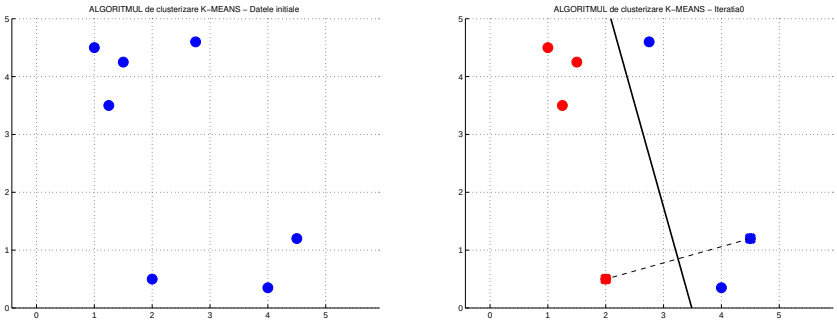
21. **(Algoritmii K-means şi EM/GMM: aplicare pe date din $\mathbb{R}^2$; compararea poziţiilor finale ale centroizilor / mediilor)**

CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, midterm, pr. 4.a

Se consideră un set de date / instanţe din planul bidimensional, conform tabelului de mai jos (partea dreaptă).

Instanţa	x	y
$P1$	1.50	4.25
$P2$	1.25	3.50
$P3$	1.00	4.50
$P4$	2.75	4.60
$P5$	4.50	1.20
$P6$	4.00	0.35
$P7$	2.00	0.50

a. Folosind algoritmul K-means, se vor forma două clustere. Centroizii iniţiali ai celor două clustere sunt instanţele 5 şi respectiv 7. Componenţa iniţială a clusterelor este definită ca de obicei, cu ajutorul unui separator liniar. (Vedeţi imaginea de mai jos, partea dreaptă; instanţele sunt reprezentate prin cerculeţe.)



Pentru fiecare iteraţie executată de algoritmul K-means pe aceste date, veţi desena

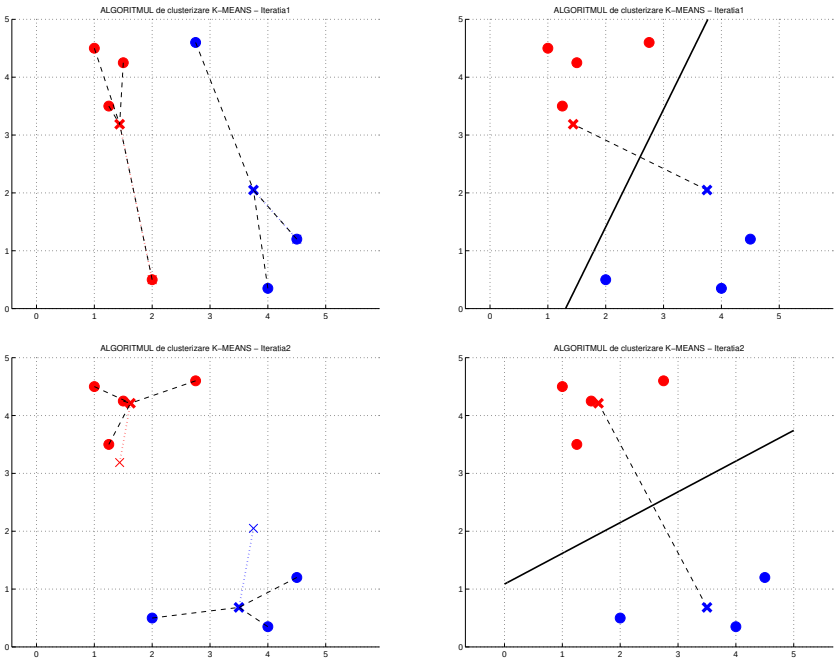
- **centroizii clusterelor (reprezentaţi cu semnul $\times$);**
- **separatorii liniari pentru cele două clustere.**

Folosiţi oricâte diagrame sunt necesare, până se ajunge la convergenţă.

b. Cum va diferi aplicarea pe aceste date a algoritmului EM pentru o mixtură de două distribuţii gaussiene bidimensionale faţă de aplicarea algoritmului K-means?

Răspuns:

a. Aplicând algoritmul K-means, obţinem următoarea evoluţie a clusterelor:

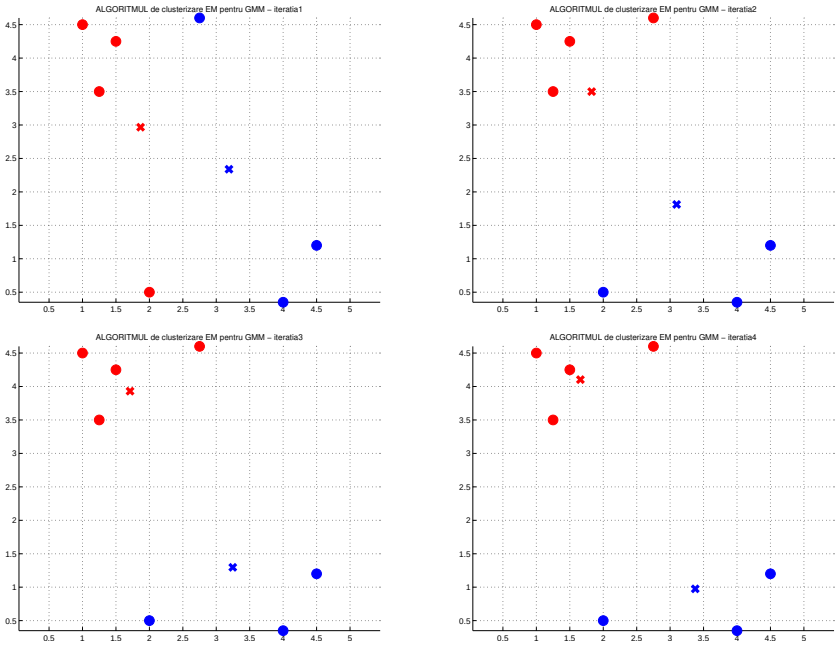


b. La aplicarea algoritmului EM, mediile celor două distribuţii (corespondentul centroizilor clusterelor în K-means) pot evolua mai încet de la o iteraţie la alta, deoarece sunt influenţate de toate instanţele de antrenament (nu doar de

cele din clusterul corespunzător, ca în cazul algoritmului K-means). Aşadar, am putea avea:

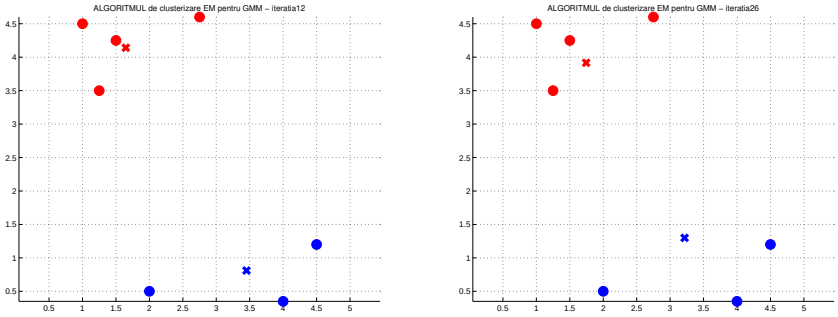
- **mai multe iteraţii până la convergenţă;**
- **o altă componenţă a clusterelor la final;**
- **o altă poziţionare finală a mediilor / centroizilor, chiar dacă se pleacă de la aceleaşi poziţii iniţiale ca pentru algoritmul K-means.**

Pentru exemplificare, dacă se aplică algoritmul EM pe aceste date, considerând varianţa $\sigma^2 = 2$ (valoare fixată şi identică pentru ambele distribuţii gaussiene şi pentru fiecare dintre cele două axe de coordonate), iar probabilităţile a priori de selecţie ale celor gaussiene $1/2$, atunci primele 4 iteraţii vor fi:



Se observă că „deplasarea“ mediilor diferă faţă de cea a centroizilor de la algoritmul K-means este, într-adevăr, mai „lentă“.

Pe datele considerate, algoritmul se opreşte după 12 iteraţii, rezultatul final fiind cel din figura de mai jos, în partea stângă. Se observă că poziţiile finale ale mediilor sunt foarte apropiate de cele ale centroizilor de la terminarea algoritmului K-means.

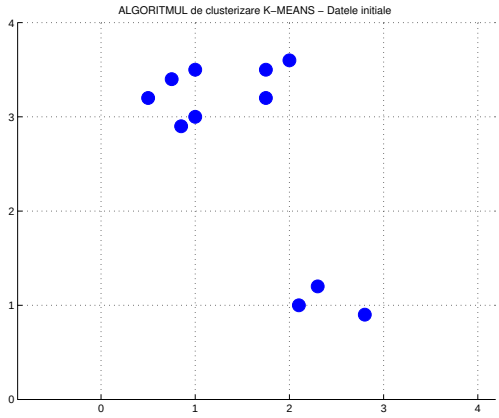


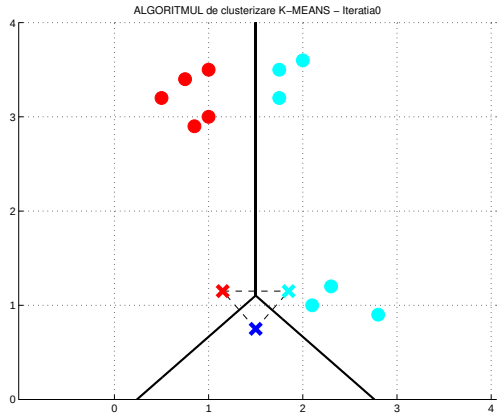
Dacă se consideră $\sigma^2 = 3$, atunci vor fi necesare mai multe iteraţii (şi anume, 26) până la convergenţă, iar rezultatul este cel din figura de mai sus, în partea dreaptă. Se observă că, deşi se obţin aceleaşi clustere, poziţiile finale ale mediilor sunt diferite faţă de cazul anterior ($\sigma^2 = 2$).

22. **(Algoritmii K-means şi EM/GMM: aplicare pe date din $\mathbb{R}^2$; compararea clusterelor finale)**

CMU, 2003 fall, T. Mitchell, A. Moore, HW7, pr. 1.c

Executaţi algoritmul K-means [eventual în mod manual] pe următorul set de date. Cerculeţele reprezintă instanţele de clusterizat, iar cruciuliţele ($\times$) sunt centroizii iniţiali ai clusterelor. Separatorii definesc componenţa iniţială a clusterelor.





În eventualitatea că veţi utiliza o implementare, coordonatele datelor şi a centroizilor iniţiali sunt date în tabelul din partea dreaptă.

Punctele	x	y
1	0.50	3.20
2	0.75	3.40
3	1.00	3.50
4	1.00	3.00
5	0.85	2.90
6	1.75	3.20
7	1.75	3.50
8	2.00	3.60
9	2.10	1.00
10	2.30	1.20
11	2.80	0.90

Centroizii	x	y
μ_1	1.50	0.75
μ_2	1.15	1.15
μ_3	1.85	1.15

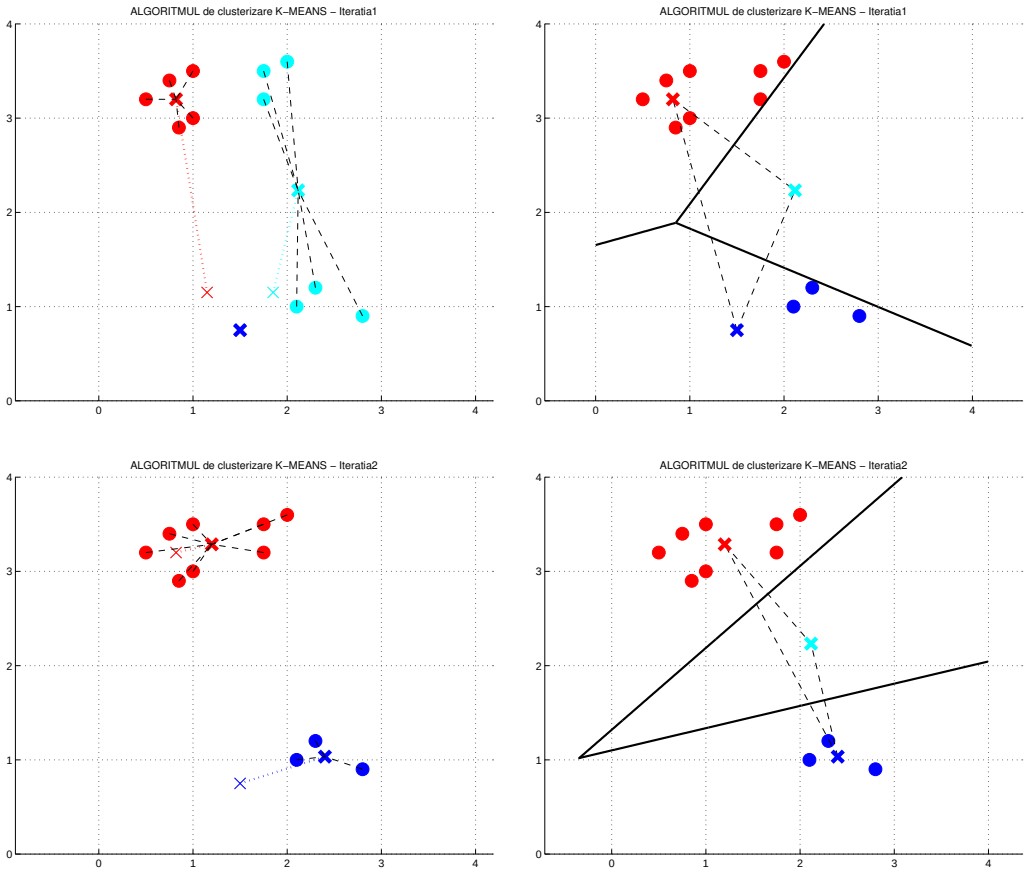
a. Pentru fiecare iteraţie a algoritmului, desenaţi centroizii şi separatorii care definesc fiecare cluster. Folosiţi oricâte imagini sunt necesare până veţi ajunge la convergenţă.

***Observaţie*: La execuţia algoritmului, se consideră că în cazul în care un centroid nu are puncte asignate lui, atunci el rămâne pe loc în iteraţia respectivă.**

b. Dacă am rula algoritmul EM pentru o mixtură de 3 distribuţii gaussiene bidimensionale pe acelaşi set de date, menţinând aceeaşi poziţionare a centroizilor iniţiali ca mai sus, am ajunge la acelaşi rezultat? Justificaţi răspunsul. Precizaţi la ce clustere se va ajunge în final.

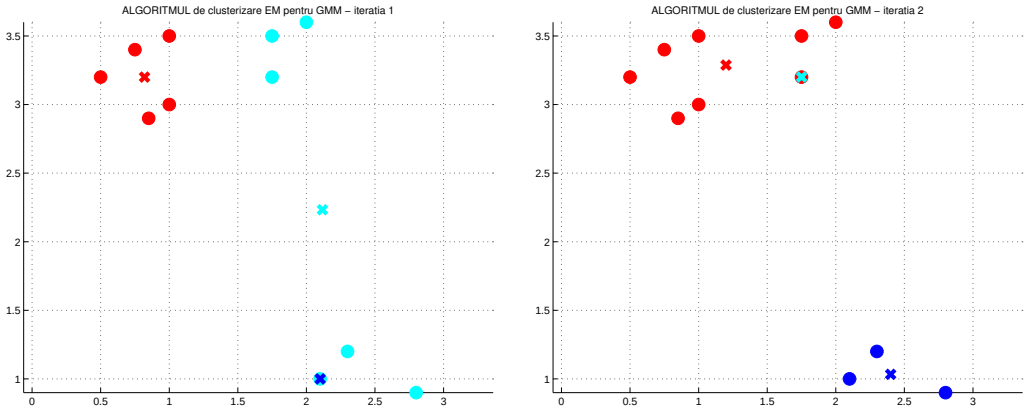
Răspuns:

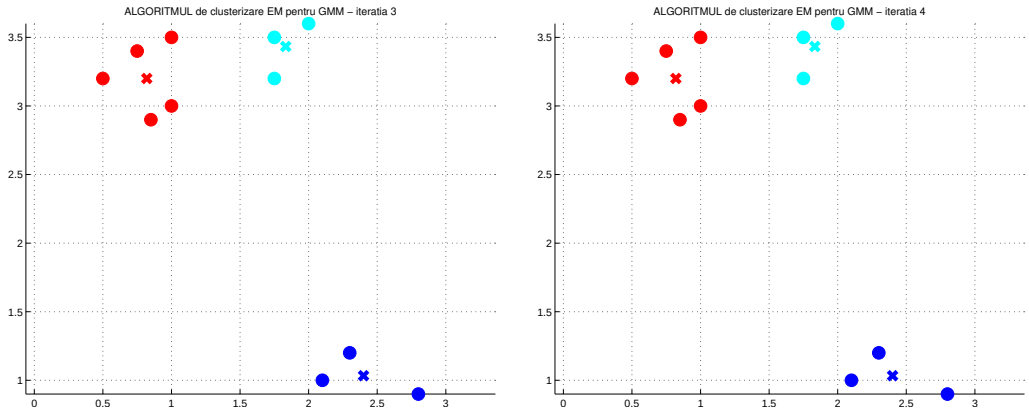
a. Aplicând algoritmul K-means obţinem:



Se observă că algoritmul va împărţi punctele în doar două clustere, al treilea cluster obţinut fiind vid.

b. Figurile de mai jos reprezintă cele 4 iteraţii ale algoritmului EM pe aceleaşi date iniţiale, considerându-se varianţa fixă $\sigma^2 = 0.01$ (valoare identică pentru ambele distribuţii gaussiene şi pentru fiecare din cele două axe de coordonate).



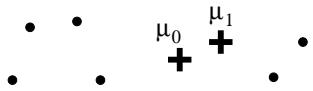


Observăm că se ajunge la o altă configuraţie finală (mai bună) decât cea obţinută de algoritmul K-means: clusterele noi au o mai mare coeziune şi nu mai avem un cluster vid.

23. **(Algoritmul EM pentru GMM: aplicare pe date din $\mathbb{R}^2$; variante de „mişcare“ a mediilor; evoluţia valorilor funcţiei de log-verosimilitate a datelor observabile)**

CMU, 2010 spring, E. Xing, T. Mitchel, A. Singh, midterm, pr. 5.2

Considerăm aplicarea algoritmului EM/GMM pentru a clusteriza datele de mai jos în două clustere. Semnele + indică valorile iniţiale ale mediilor (μ_0 şi respectiv μ_1).[885]



a. În ce direcţie se vor deplasa poziţiile mediilor μ_0 şi μ_1 la execuţia primelor iteraţii ale algoritmului EM?

b. Considerând θ ansamblul parametrilor, precizaţi cum se comportă $\prod_j P(x_j \mid \theta)$, probabilitatea marginală a datelor „observabile“, după prima iteraţie a algoritmului EM: creşte ori scade? Justificaţi pe scurt.

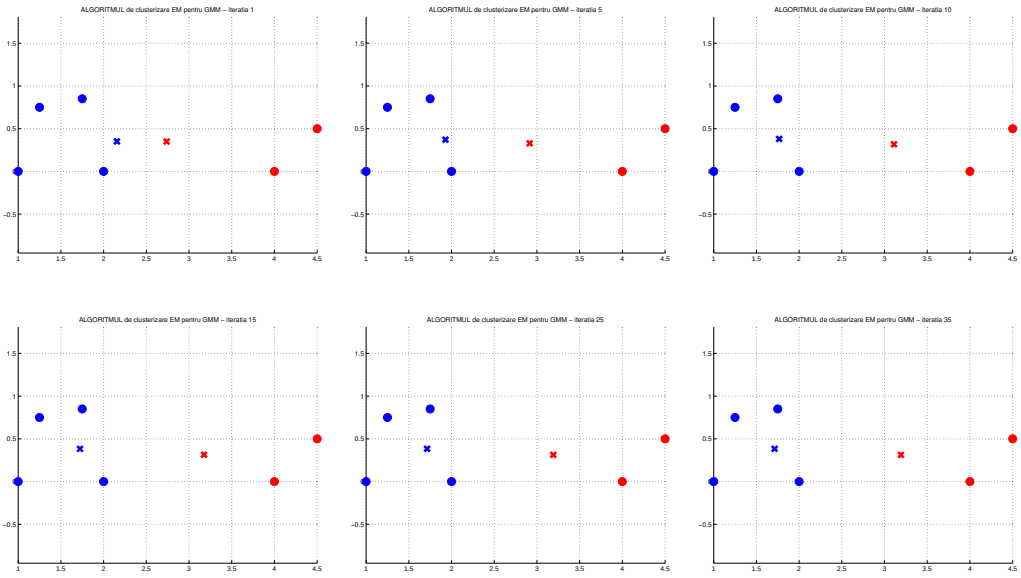
Răspuns:

a. Folosind o implementare a variantei algoritmului EM/GMM prezentată la problema 20, vom arăta că mişcarea mediilor / centroizilor depinde de valorile varianţelor celor două distribuţiile gaussiene. Se disting următoarele cazuri:

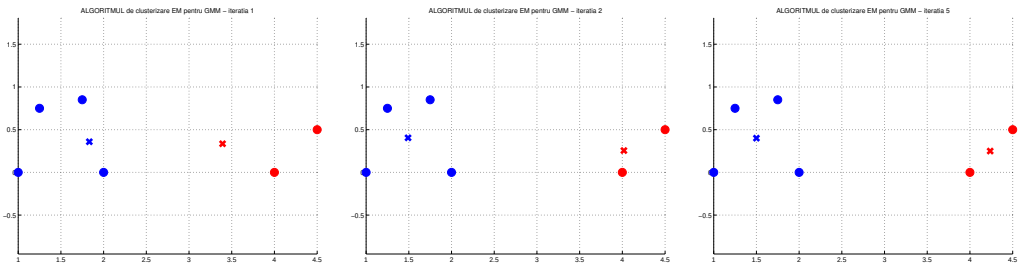
***Cazul 1*: Pentru valori relativ mici ale lui σ^2, se poate observa că la execuţia algoritmului EM μ_0 se va muta din ce în ce mai mult către stânga, în vreme ce μ_1 se va muta iniţial spre stânga, iar după aceea către dreapta.**

[885] Coordonatele asociate acestor instanţe şi respectiv ale mediilor vă sunt puse la dispoziţie în următoarele fişiere, depuse pe site-ul acestei cărţi:
http://profs.info.uaic.ro/~ciortuz/ML.ex-book/res/CMU.2010s.EX+TM+AS.midterm.pr5.2.em.dat,
http://profs.info.uaic.ro/~ciortuz/ML.ex-book/res/CMU.2010s.EX+TM+AS.midterm.pr5.2.init.dat.

De exemplu, pentru $\sigma^2 = 1.5$ se obţin următoarele iteraţii mai relevante până la convergenţă (la care se ajunge în 35 de iteraţii):

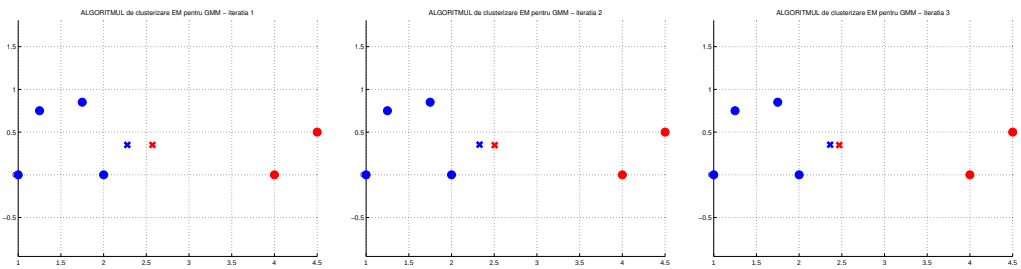


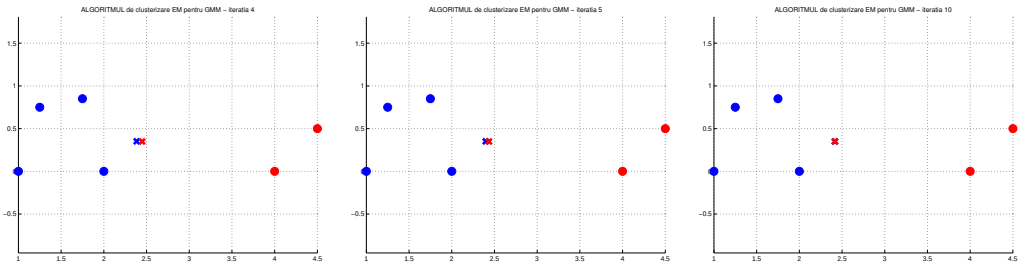
În schimb, pentru $\sigma^2 = 0.5$ (mai mic decât valoarea precedentă), se obţin următoarele iteraţii mai relevante până la convergenţă (la care se ajunge în 5 iteraţii):



***Cazul 2*: Pentru valori mai mari pentru σ^2, se poate observa că μ_0 şi μ_1 se vor deplasa unul spre celălalt, fiind posibil chiar să ajungă practic să coincidă.**

De exemplu, pentru $\sigma^2 = 3$ se obţin următoarele iteraţii mai relevante până la convergenţă (la care se ajunge în 20 de iteraţii):





b. Probabilitatea $\prod_j P(x^j \mid \theta)$ reprezintă verosimilitatea datelor observabile (presupunând că aceste date sunt generate independent unele de altele). Conform rezultatului de convergenţă / corectitudine demonstrat la problema 2 de la capitolul *Schema algoritmică EM*, această probabilitate nu descreşte — adică fie creşte fie rămâne la fel — de la o iteraţie la alta a algoritmului EM. Creşterea se opreşte doar la atingerea unui punct de optim local.

24. **(Algoritmul EM pentru mixturi de distribuţii gaussiene multidimensionale; cazul general)**

prelucrare de Liviu Ciortuz, după
■ ● ○ ***Andrew Ng, Stanford University, ML course, 2009 fall, lecture notes, parts VIII and IX, and CMU, 2010 fall, Aarti Singh, HW4, pr. 1.1***

Presupunem, ca de obicei în învăţarea automată, că avem un set de date $\{x_1, \ldots, x_n\} \subset \mathbb{R}^d$. Ne vom situa în paradigma învăţării nesupervizate, deci vom considera că aceste puncte nu au nicio etichetă ataşată.

Dorim să modelăm aceste date prin specificarea [funcţiei de densitate a] unei distribuţii probabiliste comune

$$p(x_i, z_i) = p(x_i|z_i)\, p(z_i), \text{ pentru } i = 1, \ldots, n,$$
$$\text{cu } z_i \sim \mathit{Categorial}\,(\pi) \text{ şi } \pi \overset{not.}{=} (\pi_1, \ldots, \pi_K).$$

Deci $p(x_i) = \sum_{z_i} p(x_i, z_i) = \sum_{z_i} p(x_i|z_i)\, p(z_i)$, formulă cu care suntem obişnuiţi. K este numărul de valori pe care le pot lua variabilele z_i. Pentru $j = 1, \ldots, K$, probabilitatea a priori $p(z_i = j)$ va fi desemnată prin parametrul π_j. Vom presupune că $(x_i|z_i = j) \sim \mathcal{N}(\mu_j, \Sigma_j)$. Aşadar,

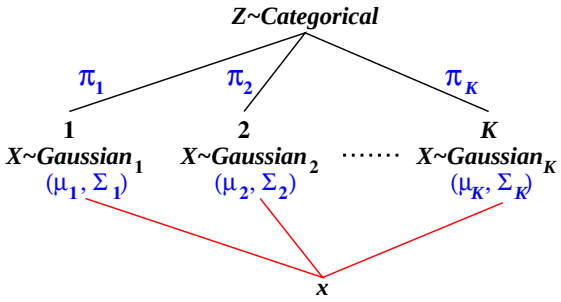
$$p(x_i|z_i = j; \mu_j, \Sigma_j) \overset{def.}{=} \frac{1}{(2\pi)^{d/2}|\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(x - \mu_j)^\top \Sigma_j^{-1}(x - \mu_j)\right), \tag{379}$$

unde $\mu \in \mathbb{R}^d$, iar Σ_j este matrice de dimensiune $d \times d$, simetrică şi pozitiv definită (ultima condiţie este suficientă pentru a asigura inversabilitatea).[886] [887]
Vectorii din $\mathbb{R}^d$ sunt consideraţi vectori-coloană în contextul acestei probleme.

[886] Vedeţi proprietatea *iv* de la problema 36.c de la capitolul de *Fundamente*.

[887] În penultima linie a egalităţii de mai sus, prin simbolul exp am desemnat ca de obicei funcţia exponenţială cu baza e. (Aşadar, $\exp(y) \overset{not.}{=} e^y$ pentru orice $y \in \mathbb{R}$.) Simbolul $\top$ din expresia din dreapta egalităţii (379) reprezintă operaţia de transpunere de matrice / vectori.

Prin urmare, modelul nostru postulează că
– fiecare instanţă x_i a fost generată alegând în mod aleatoriu valorile variabilelor z_i din mulţimea $\{1, \ldots, K\}$,
– iar apoi x_i a fost generat de una dintre cele K distribuţii gaussiene, mai precis cea desemnată de z_i.



Acest model se numeşte *modelul mixturii de [distribuţii] gaussiene*. Vom considera că z_i sunt variabile aleatoare *latente* (se mai spune că sunt variabile *ascunse*, sau *neobservabile*). Datorită acestui fapt, problema estimării în sensul MLE a parametrilor π, μ şi Σ devine dificilă. (Am notat $\mu = (\mu_1, \ldots, \mu_K)$ şi $\Sigma = (\Sigma_1, \ldots, \Sigma_K)$.)

***Observaţie* (1):** Evident, dacă am cunoaşte valorile variabilelor z_i, problema estimării parametrilor modelului nostru (adică π, μ şi Σ) ar fi uşor de rezolvat.[888] Concret, am putea scrie *verosimilitatea* datelor noastre ca

$$l(\pi, \mu, \Sigma) = \sum_{i=1}^{n} \left[\ln p(x_i|z_i; \mu, \Sigma) + \ln p(z_i; \pi)\right].$$

Făcând calculele, vom obţine următoarele estimări MLE ale parametrilor π, μ şi Σ:[889]

$$\pi_j = \frac{1}{n}\sum_{i=1}^{n} 1_{\{z_i=j\}}, \tag{380}$$

$$\mu_j = \frac{\sum_{i=1}^{n} 1_{\{z_i=j\}} x_i}{\sum_{i=1}^{n} 1_{\{z_i=j\}}}, \tag{381}$$

$$\Sigma_j = \frac{\sum_{i=1}^{n} 1_{\{z_i=j\}} (x_i - \mu_j)(x_i - \mu_j)^\top}{\sum_{i=1}^{n} 1_{\{z_i=j\}}}, \tag{382}$$

unde $1_{\{z_i=j\}}$ sunt „funcţii indicator“, care desemnează gaussiana (j) care a generat o instanţă oarecare (x_i) din setul de date pe care îl avem la dispoziţie. Însă, atunci când valorile variabilelor z_i sunt necunoscute, nu este posibil să obţinem estimările în sens MLE ale parametrilor π, μ şi Σ prin calcularea analitică a rădăcinilor derivatelor parţiale de ordinul întâi ale funcţiei de log-verosimilitate $l(\pi, \mu, \Sigma)$.

În astfel de cazuri, *algoritmul EM* constituie o metodă convenabilă pentru estimarea parametrilor în sensul verosimilităţii maxime. *Strategia* sa constă în a construi în mod iterativ o *limită inferioară* pentru funcţia de *log-verosimilitate* a datelor observabile $x_1, \ldots, x_n$ (pasul E), pentru ca apoi să calculeze *maximul* acelei limite inferioare (pasul M).

Folosind schema algoritmică EM,[890] demonstraţi că *regulile de actualizare* pentru parametrii π, μ şi Σ în cazul unei mixturi de distribuţii gaussiene multidi-

[888] Este o situaţie similară cu — şi doar uşor mai elaborată decât — cea din problema 53 (punctul a) de la capitolul de *Fundamente*.
[889] Vedeţi similaritatea dintre formulele (381) şi (60), respectiv (382) şi (64).
[890] Vedeţi cartea *Machine Learning*, a lui Tom Mitchell, 1997, pag. 194-195 sau problema 1 (pag. 981) de la capitolul *Schema algoritmică EM* din această culegere.

mensionale sunt următoarele:[891]

Pasul E:

$$w_{ij} \overset{not.}{=} p(z_i = j|x_i; \pi', \mu', \Sigma') = \frac{p(x_i|z_i = j; \mu', \Sigma')\, p(z_i = j; \pi')}{\sum_{l=1}^{K} p(x_i|z_i = l; \mu', \Sigma')\, p(z_i = l; \pi')}$$

Pasul M:

$$\pi_j = \frac{1}{n} \sum_{i=1}^{n} w_{ij}, \tag{383}$$

$$\mu_j = \frac{\sum_{i=1}^{n} w_{ij} x_i}{\sum_{i=1}^{n} w_{ij}}, \tag{384}$$

$$\Sigma_j = \frac{\sum_{i=1}^{n} w_{ij} (x_i - \mu_j)(x_i - \mu_j)^\top}{\sum_{i=1}^{n} w_{ij}}. \tag{385}$$

În aceste formule, π', μ' şi Σ' reprezintă valorile parametrilor modelului nostru la iteraţia precedentă a algoritmului EM (respectiv, valorile atribuite iniţial parametrilor, pentru prima iteraţie).[892]

***Sugestie*: Întrucât se lucrează cu instanţe din $\mathbb{R}^d$, vă recomandăm să folosiţi derivate [parţiale] vectoriale. Pentru acest tip de derivate, variabila în raport cu care se derivează este din $\mathbb{R}^d$, nu din $\mathbb{R}$ cum suntem obişnuiţi. Unele din următoarele formule (preluate din documentul *Matrix Identities*, de Sam Roweis, 1999) vă pot fi de folos:**

(1e) $(A^{-1})^\top = (A^\top)^{-1}$

(2b) $|A^{-1}| = \dfrac{1}{|A|}$

(4a) $\dfrac{\partial}{\partial X}|AXB| = |AXB|(X^{-1})^\top = |AXB|(X^\top)^{-1}$

(4b) $\dfrac{\partial}{\partial X} \ln |X| = (X^{-1})^\top = (X^\top)^{-1}$

(5a) $\dfrac{\partial}{\partial X} a^\top X = \dfrac{\partial}{\partial X} X^\top a = a$

(5b) $\dfrac{\partial}{\partial X} X^\top A X = (A + A^\top)X$

(5c) $\dfrac{\partial}{\partial X} a^\top X b = ab^\top$

[891] Este util de observat (şi de analizat) corespondenţa dintre formulele (380) şi (383), apoi (381) şi (384) şi, în sfârşit, (382) şi (395).

[892] Pentru ultimele două reguli de actualizare de la pasul M trebuie să fie satisfăcute condiţiile $\sum_{i=1}^{n} w_{ij} \neq 0$, pentru $j \in \{1, \ldots, K\}$. Ţinând cont de definiţia dată mai sus pentru ponderile w_{ij}, cu $i \in \{1, \ldots, n\}$ şi $j \in \{1, \ldots, K\}$, condiţia $\sum_{i=1}^{n} w_{ij} \neq 0$ (pentru j fixat) revine la următoarea proprietate: $\exists i \in \{1, \ldots, n\}$ astfel încât $p(z_i = j|x_i; \pi, \mu, \Sigma) \neq 0$, adică

$$\underbrace{p(x_i|z_i = j; \mu, \Sigma)}_{>0} \underbrace{p(z_i = j|\pi)}_{\pi_j} \neq 0.$$

Restricţia aceasta este satisfăcută (în contextul mixturilor de distribuţii gaussiene) pentru acei j pentru care $\pi_j > 0$.

Este natural să presupunem $\pi_j > 0$ pentru toţi $j \in \{1, \ldots, K\}$. Cazurile $\pi_j = 0$ nu sunt interesante (sunt „degenerate"), deci pot fi eliminate din start. Aşadar, la pasul de iniţializare a algoritmului EM, vom lua toţi $\pi'_j > 0$. Vom arăta la finalul demonstraţiei că această premiză va implica faptul că restricţia $\pi_j > 0$ este satisfăcută la fiecare iteraţie.

(5e) $\dfrac{\partial}{\partial X} a^\top X a = \dfrac{\partial}{\partial X} a^\top X^\top a = a a^\top$

(5g) $\dfrac{\partial}{\partial X} (Xa+b)^\top C (Xa+b) = (C + C^\top)(Xa+b)a^\top$

Răspuns:

***Pasul E* este uşor. Vom calcula**

$$w_{ij} \overset{not.}{=} p(z_i = j | x_i; \pi', \mu', \Sigma').$$

Folosind regula lui Bayes (combinată cu formula probabilităţii totale), obţinem:

$$p(z_i = j | x_i; \pi', \mu', \Sigma') = \frac{p(x_i | z_i = j; \mu', \Sigma')\, p(z_i = j; \pi')}{\sum_{l=1}^{K} p(x_i | z_i = l; \mu', \Sigma')\, p(z_i = l; \pi')}$$

Aici, probabilitatea $p(x_i | z_i = j; \mu', \Sigma')$ este obţinută prin evaluarea densităţii gaussienei având media μ'_j şi matricea de covarianţă Σ'_j pentru argumentul x_i, în vreme ce probabilitatea $p(z_i = j; \pi')$ este dată de π'_j, ş.a.m.d.

Apoi, la *pasul M*, trebuie să maximizăm, în raport cu parametrii π, μ şi Σ, funcţia

$$\sum_{i=1}^{n} \sum_{j} w_{ij} \ln \frac{p(x_i, z_i; \pi, \mu, \Sigma)}{w_{ij}}. \tag{386}$$

Conform schemei algoritmice EM,[893] funcţia (386), ale cărei argumente sunt π, μ şi Σ, constituie o margine inferioară pentru funcţia de log-verosimilitate a datelor observabile, $\sum_{i=1}^{n} \ln p(x_i | \pi, \mu, \Sigma)$.[894] Vom rescrie în mod convenabil expresia acestei funcţii, ţinând cont mai întâi de definiţia probabilităţii condiţionate, apoi de definiţia distribuţiei gaussiene multidimensionale şi, în sfârşit, de proprietăţile logaritmului:

$$\begin{aligned}
&\sum_{i=1}^{n} \sum_{j=1}^{K} w_{ij} \ln \frac{p(x_i, z_i; \pi, \mu, \Sigma)}{w_{ij}} = \sum_{i=1}^{n} \sum_{j=1}^{K} w_{ij} \ln \frac{p(x_i | z_i = j; \mu, \Sigma)\, p(z_i = j; \pi)}{w_{ij}} = \\
&\quad \sum_{i=1}^{n} \sum_{j=1}^{K} w_{ij} \ln \frac{\dfrac{1}{(2\pi)^{d/2} |\Sigma_j|^{1/2}} \cdot \exp\left(-\frac{1}{2}(x_i - \mu_j)^\top \Sigma_j^{-1} (x_i - \mu_j)\right) \cdot \pi_j}{w_{ij}} = \\
&\quad \sum_{i=1}^{n} \sum_{j=1}^{K} w_{ij} \left[-\ln((2\pi)^{d/2} |\Sigma_j|^{1/2} w_{ij}) - \frac{1}{2}(x_i - \mu_j)^\top \Sigma_j^{-1} (x_i - \mu_j) + \ln \pi_j \right]
\end{aligned} \tag{387}$$

Ca o *consecinţă* a calculelor de mai sus, obiectivul nostru a devenit maximizarea expresiei (387). Pentru aceasta, vom proceda în maniera clasică: vom

[893] Vedeţi problema 1, pag. 981 de la capitolul *Schema algoritmică EM* din această culegere.

[894] De fapt, folosind notaţiile de la problema 1 de la capitolul *Schema algoritmică EM*, pentru orice distribuţie de probabilitate q peste variabilele neobservabile z, urmează că funcţia $F(q(z), \theta')$ — unde, F este definit prin relaţia (405), iar $\theta' \overset{not.}{=} (\pi', \mu', \Sigma')$ — este o margine inferioară pentru $\ln P(x|\theta')$, log-verosimilitatea variabilelor observabile x.

Însă dintre toate aceste distribuţii, cea mai bună — adică, cea pentru care se atinge $\max_{q(z)} F(q(z), \theta')$, este distribuţia $q_{\pi', \mu', \Sigma'}$ definită astfel: $q(z_i = j) \overset{not.}{=} q_{\pi', \mu', \Sigma'}(z_i = j) \overset{not.}{=} p(z_i = j | x_i; \pi', \mu', \Sigma') \overset{not.}{=} w_{ij}$. (Vedeţi demonstraţia de la problema 1.c de la capitolul *Schema algoritmică EM*.)

căuta soluţiile derivatelor parţiale de ordinul întâi ale acestei expresii în raport cu parametrii μ, Σ şi π.

Mai întâi vom calcula derivata parţială în raport unde μ_l, cu $l \in \{1, \ldots, K\}$.[895]

$$\begin{aligned}
&\frac{\partial}{\partial \mu_l} \sum_{i=1}^{n} \sum_{j=1}^{K} w_{ij} \left[-\ln \left((2\pi)^{d/2} |\Sigma_j|^{1/2} w_{ij} \right) - \frac{1}{2} (x_i - \mu_j)^\top \Sigma_j^{-1} (x_i - \mu_j) + \ln \pi_j \right] \\
&= -\frac{\partial}{\partial \mu_l} \sum_{i=1}^{n} \sum_{j=1}^{K} w_{ij} \frac{1}{2} (x_i - \mu_j)^\top \Sigma_j^{-1} (x_i - \mu_j) \\
&= -\frac{1}{2} \sum_{i=1}^{n} w_{il} \frac{\partial}{\partial \mu_l} (x_i - \mu_l)^\top \Sigma_l^{-1} (x_i - \mu_l) \qquad (388) \\
&= \frac{1}{2} \sum_{i=1}^{n} w_{il} \frac{\partial}{\partial \mu_l} \left(2\mu_l^\top \Sigma_l^{-1} x_i - \mu_l^\top \Sigma_l^{-1} \mu_l \right) \qquad (389) \\
&= \sum_{i=1}^{n} w_{il} \left(\Sigma_l^{-1} x_i - \Sigma_l^{-1} \mu_l \right). \qquad (390)
\end{aligned}$$

***Detalii de calcul*:**

O primă posibilitate este ca, pornind de la expresia (388) să se treacă la expresia (389) — folosind pe de o parte distributivitatea operaţiei de înmulţire a matricelor faţă de operaţia de adunare sau de scădere şi, pe de altă parte, formula $(1e)$ —, iar apoi de la expresia (389) să se treacă la expresia (390), ţinând cont de formulele $(5c)$ — sau, mai simplu, $(5a)$ — şi $(5b)$ din documentul *Matrix Identities* de Sam Roweis.

A doua posibilitate este ca, pornind de la expresia (388) să obţinem direct expresia (390), folosind formulele $(5g)$ şi $(1e)$ din documentul menţionat. Formula $(1e)$ se aplică matricei Σ_l^{-1}. Se ţine cont că Σ_l este matrice simetrică şi inversabilă.[896]

O a treia posibilitate este să se lucreze în mod clasic (nu derivând în raport cu un vector), şi anume scriind derivatele parţiale ale expresiei (387) în raport cu fiecare dintre componentele vectorului μ_l (şi scriind de asemenea pe componente vectorul x_i şi respectiv matricea Σ_l^{-1}).[897]

Egalând expresia (390) **cu vectorul-coloană 0 (format din K elemente, toate având valoarea $0 \in \mathbb{R}$), vom obţine regula de actualizare pentru μ_l:**

$$\mu_l = \frac{\sum_{i=1}^{n} w_{il} x_i}{\sum_{i=1}^{n} w_{il}}. \qquad (391)$$

***Detalii de calcul*:**

$$\sum_{i=1}^{n} w_{il} \left(\Sigma_l^{-1} x_i - \Sigma_l^{-1} \mu_l \right) = 0 \Leftrightarrow \sum_{i=1}^{n} w_{il} \Sigma_l^{-1} x_i = \sum_{i=1}^{n} w_{il} \Sigma_l^{-1} \mu_l$$

Înmulţind ultima egalitate la stânga cu matricea Σ_l, obţinem $\sum_{i=1}^{n} w_{il} x_i = \sum_{i=1}^{n} w_{il} \mu_l$. Este imediat că membrul drept al acestei ultime egalităţi se poate scrie ca $\left(\sum_{i=1}^{n} w_{il} \right) \mu_l$. (Observaţi că $\sum_{i=1}^{n} w_{il}$ este un număr real.) Prin urmare, $\mu_l = \frac{\sum_{i=1}^{n} w_{il} x_i}{\sum_{i=1}^{n} w_{il}}$.

[895] Atenţie: μ_l este vector (coloană) şi, în consecinţă, derivata parţială în raport cu μ_l va fi de asemenea un vector (coloană). Similar, derivata parţială în raport cu Σ_l va fi o matrice.

[896] Vedeţi problema 20 de la capitolul de *Fundamente*.

[897] În acelaşi mod se pot demonstra şi formulele $(5g)$, $(1e)$, $(5c)$ şi $(5b)$ menţionate mai sus (precum şi cele care vor fi utilizate mai jos).

Să deducem acum relaţiile de actualizare de la pasul pasul M pentru matricele Σ_j, unde $j = 1, \ldots, K$. Reţinând în expresia (387) doar termenii care depind de Σ_j, rezultă că trebuie să maximizăm (în raport cu Σ_j) expresia

$$\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=1}^{K} w_{ij}\left[\ln\frac{1}{|\Sigma_j|^{1/2}} - \frac{1}{2}(x_i-\mu_j)^\top \Sigma_j^{-1}(x_i-\mu_j)\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{K} w_{ij}\left[\frac{1}{2}\ln|\Sigma_j^{-1}| - \frac{1}{2}(x_i-\mu_j)^\top \Sigma_j^{-1}(x_i-\mu_j)\right] \\
&= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{K} w_{ij}\left[\ln|\Sigma_j^{-1}| - (x_i-\mu_j)^\top \Sigma_j^{-1}(x_i-\mu_j)\right]. \qquad (392)
\end{aligned}$$

***Detalii de calcul*:**

Pentru a obţine prima egalitate de mai sus, se foloseşte formula $(2b)$. Concret, $\frac{1}{|\Sigma_j|^{1/2}} = |\Sigma_j|^{-1/2} = (|\Sigma|^{-1})^{1/2} \overset{(2b)}{=} |\Sigma^{-1}|^{1/2}$.

În astfel de situaţii, pentru convenienţă, se foloseşte următorul „artificiu de calcul": în loc să se lucreze cu *matricea de covarianţă* Σ_j (presupusă a fi inversabilă), se preferă să se lucreze cu *matricea de precizie* $\Lambda_j \overset{not.}{=} \Sigma_j^{-1}$.[898] Aşadar, vom maximiza dublul expresei (392) în raport cu Λ_j. Derivând (în raport cu Λ_j), vom obţine:

$$\begin{aligned}
&\frac{\partial}{\partial\Lambda_j}\sum_{i=1}^{n} w_{ij}\left[\ln|\Lambda_j| - (x_i-\mu_j)^\top \Lambda_j(x_i-\mu_j)\right] \qquad (393)\\
&= \sum_{i=1}^{n} w_{ij}\frac{1}{|\Lambda_j|}\frac{\partial}{\partial\Lambda_j}|\Lambda_j| - \sum_{i=1}^{n} w_{ij}\frac{\partial}{\partial\Lambda_j}\left[(x_i-\mu_j)^\top \Lambda_j(x_i-\mu_j)\right] \qquad (394)\\
&= \sum_{i=1}^{n} w_{ij}\Lambda_j^{-1} - \sum_{i=1}^{n} w_{ij}(x_i-\mu_j)(x_i-\mu_j)^\top \qquad (395)\\
&= \Big(\sum_{i=1}^{n} w_{ij}\Big)\Lambda_j^{-1} - \sum_{i=1}^{n} w_{ij}(x_i-\mu_j)(x_i-\mu_j)^\top. \qquad (396)
\end{aligned}$$

***Detalii de calcul*:**

Expresia (394) se poate obţine din expresia (393) folosind pentru primul termen formula de derivare a funcţiilor compuse, iar expresia (395) se poate obţine din expresia (394) folosind pentru primul termen formula $(4a)$ (luând $A = B = I$, matricea identitate), iar pentru termenul al doilea formula $(5e)$. Alternativ, expresia (395) se poate obţine direct din expresia (393),[899] folosind pentru derivarea primului termen formula $(4b)$ şi pentru derivarea celui de-al doilea termen formula $(5e)$. În plus, la fiecare dintre aceste două variante de rezolvare, se ţine cont şi de formula $(1e)$.

[898] Aşa s-a procedat şi la problema 53 de la capitolul de *Fundamente*, unde am arătat cum anume se face estimarea parametrilor distribuţiei gaussiene multidimensionale.

[899] Remarcaţi că $\Lambda_j^\top = \Lambda_j$. Într-adevăr, întrucât $\Sigma_j = \Sigma_j^\top$, urmează că $\Lambda_j = \Sigma_j^{-1} = (\Sigma_j^\top)^{-1} \overset{(1e)}{=} (\Sigma_j^{-1})^\top = \Lambda_j^\top$.

Egalând expresia (396) cu 0 (matrice pătratică de dimensiune $K \times K$, ale cărei elemente au, toate, valoarea $0 \in \mathbb{R}$), vom obţine:

$$\Sigma_j = \Lambda_j^{-1} = \frac{\sum_{i=1}^n w_{ij}(x_i - \mu_j)(x_i - \mu_j)^\top}{\sum_{i=1}^n w_{ij}}. \tag{397}$$

În sfârşit, vom deriva regula de actualizare de la pasul M pentru parametrii π_j, unde $j = 1, \ldots, K$. Reţinând în expresia (387) doar acei termeni care depind de π_j, rezultă că va trebui să maximizăm

$$\sum_{i=1}^n \sum_{j=1}^K w_{ij} \ln \pi_j. \tag{398}$$

Să observăm însă — exact ca la rezolvarea problemei 18, cazul $K > 2$ — că soluţiile optime π_j trebuie să satisfacă o restricţie suplimentară, şi anume $\sum_{j=1}^K \pi_j = 1$, din moment ce $\pi_j \overset{not.}{=} p(z_i = j|\pi)$, pentru orice $j \in \{1, \ldots, K\}$ şi orice $i \in \{1, \ldots, n\}$. Vom proceda la fel şi aici, folosind *metoda multimplicatorilor lui Lagrange* pentru a identifica acele valori ale variabilelor π_j care maximizează expresia (398) satisfăcând în acelaşi timp restricţia $\sum_{j=1}^K \pi_j = 1$. Aşadar, vom căuta punctul de optim al polinomului lagrangean

$$\mathcal{L}(\pi, \lambda) = \sum_{i=1}^n \sum_{j=1}^K w_{ij} \ln \pi_j + \lambda\Big(\sum_{j=1}^K \pi_j - 1\Big),$$

unde $\lambda \in \mathbb{R}$ este aşa-numita variabilă („multiplicator“) Lagrange.[900] Derivatele parţiale ale polinomului $\mathcal{L}(\pi, \lambda)$ în raport cu π_j, pentru $j = 1, \ldots, K$ sunt:

$$\frac{\partial}{\partial \pi_j}\mathcal{L}(\pi, \lambda) = \sum_{i=1}^n \frac{w_{ij}}{\pi_j} + \lambda = \frac{1}{\pi_j}\left(\sum_{i=1}^n w_{ij}\right) + \lambda.$$

Egalând cu 0, rezultă:

$$\pi_j = -\frac{\sum_{i=1}^n w_{ij}}{\lambda}.$$

Impunând restricţia $\sum_j \pi_j = 1$, rezultă $-\lambda = \sum_{j=1}^K \sum_{i=1}^n w_{ij} = \sum_{i=1}^n \sum_{j=1}^K w_{ij} = \sum_{i=1}^n 1 = n$. (Aici am folosit faptul că $w_{ij} \overset{not.}{=} p(z_i = j|x_i; \mu', \Sigma')$ şi, din moment ce pentru fiecare $i = 1, \ldots, n$ (fixat) aceasta reprezintă o funcţie de probabilitate, rezultă că $\sum_j w_{ij} = 1$.) Aşadar, relaţia de actualizare de la pasul M pentru parametrul π_j va fi:

$$\pi_j = \frac{1}{n}\sum_{i=1}^n w_{ij}. \tag{399}$$

Desigur, $\pi_j \geq 0$, întrucât $w_{ij} \geq 0$ pentru orice $j \in \{1, \ldots, K\}$ şi orice $i \in \{1, \ldots, n\}$.[901]

[900] Ca şi la problema 18, vom observa că nu este nevoie să ne mai ocupăm şi de restricţiile $\pi_j \geq 0$ — care ar trebui adăugate în mod normal la formularea problemei de optimizare —, pentru că, aşa cum vom vedea în curând, soluţia pe care o vom găsi pur şi simplu derivând polinomul $\mathcal{L}(\pi)$ va satisface (în mod implicit) aceste restricţii.

[901] Conform notei de subsol 892, dacă valorile atribuite iniţial parametrilor π_j satisfac restricţia $\pi_j > 0$ pentru orice $j \in \{1, \ldots, K\}$, rezultă că $w_{ij} > 0$ pentru orice $i \in \{1, \ldots, n\}$, deci şi noile valori obţinute pentru π_j la pasul M (pentru $j = 1, \ldots, K$) vor fi tot strict pozitive.

***Observaţie* (2):** Acum cititorul ar trebui să compare relaţiile de actualizare de la pasul M[902] cu formulele de estimare directă (MLE) atunci când variabilele z_i sunt cunoscute / observabile[903]: ele sunt identice, cu excepţia faptului că în locul funcţiilor-indicator $1_{\{z_i=j\}}$ acum apar ponderile / mediile w_{ij}. Aşadar, formulele de la pasul M al acestui algoritm EM (deci, în cazul învăţării nesupervizate) sunt versiuni *ponderate* ale formulelor MLE (cazul supervizat). De asemenea, forma acestui algoritm EM ne aminteşte şi de algoritmul de clusterizare K-means: totul este similar, cu excepţia faptului că în loc de asignare "hard" a instanţelor x_i la clustere $c(i)$, avem acum asignări "soft" (w_{ij}). Ca şi K-means, algoritmul EM este sensibil la problema optimului local, deci rularea repetată, cu diferite valori iniţiale pentru unii parametri (sau, pentru toţi parametrii) poate fi folositoare. Este limpede că putem asocia algoritmului EM o *interpretare* foarte naturală, şi anume aceea a încercării repetate de a ghici necunoscutele z_i.

25. **(Algoritmii K-means şi EM pentru GMM în $\mathbb{R}^2$; o întrebare de ordin calitativ)**

CMU, 2010 spring, E. Xing, T. Mitchel, A. Singh, midterm, pr. 5.1

Considerăm setul de date de antrenament de mai jos şi doi algoritmi de clusterizare: K-means şi EM pentru modelare de mixturi de gaussiene (EM/GMM). Credeţi că aceşti algoritmi vor produce în final aceiaşi centroizi pentru clustere dacă pornesc de la aceleaşi valori iniţiale μ_1 şi μ_2? Explicaţi pe scurt.



Răspuns:

Nu. Diferenţa esenţială dintre cei doi algoritmi constă în faptul că algoritmul K-means foloseşte asignări de tip "hard", adică fiecare punct aparţine unui singur cluster, în timp ce algoritmul EM/GMM utilizează asignări de tip "soft", ceea ce înseamnă că fiecare punct este asignat cu o anumită probabilitate fiecărui cluster.

Aşadar, în cazul K-means, la fiecare nouă iteraţie poziţia fiecărui centroid este determinată de media punctelor asignate clusterui corespunzător, pe când în cazul algoritmului EM/GMM poziţia fiecărui centroid este o anumită medie ponderată obţinută din coodonatele tuturor punctelor. Ponderile reprezintă probabilitatea de generare a punctului de către gaussiana asociată respectivului centroid. Prin urmare, centroizii finali obţinuţi de EM/GMM vor fi (cel puţin sensibil) deplasaţi faţă de centroizii determinaţi de K-means.

26. **(Adevărat sau Fals?)**

a. ***Liviu Ciortuz***

Aplicând algoritmul de clusterizare EM pentru o mixtură de K distribuţii gaussiene obţinem acelaşi rezultat ca şi în cazul folosirii algoritmului K-means.

b. ***CMU, 2006 (10-701/15-781), final exam, pr. 1.e***

[902] Şi anume (399), (391) şi (397). Acestea coincid cu formulele (383), (384) şi (385) din ultima parte a enunţului.
[903] Vedeţi *Observaţia* (1) din enunţ şi / sau problema 53.a de la capitolul de *Fundamente*.

În afară de algoritmul EM, pentru a învăţa parametrii unei mixturi de distribuţii gaussiene se mai poate folosi şi metoda gradientului.

Răspuns:

a. Fals. Mai precis: posibil da, posibil nu; depinde de valorile iniţiale atribuite varianţelor distribuţiilor gaussiene (chiar dacă valorile iniţiale ale centroizilor / mediilor se corespund). Vedeţi problema 22.

b. Adevărat. Algoritmul EM este însă preferat, fiindcă uneori derivatele parţiale ale funcţiei de verosimilitate a datelor observabile în raport cu parametrii de estimat sunt dificil sau chiar imposibil de calculat.

7.2 Clusterizare — Probleme propuse

7.2.1 Clusterizare ierarhică

27. **(Clusterizare ierarhică aglomerativă, folosind similaritate de tip "single-", "complete-" şi "average-linkage")**

prelucrare de Liviu Ciortuz, după

■ • ○ ***CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, HW4, pr. 2.a***

Tabelul de mai jos reprezintă matricea de distanţe pentru [o mulţime formată din] şase obiecte.

	A	B	C	D	E	F
A	0					
B	0.12	0				
C	0.51	0.25	0			
D	0.84	0.16	0.14	0		
E	0.28	0.77	0.70	0.45	0	
F	0.34	0.61	0.93	0.20	0.67	0

a. Aplicaţi algoritmul de clusterizare ierarhică aglomerativă pe aceste date, folosind mai întâi similaritate *single-linkage* şi apoi similaritate *complete-linkage*. La fiecare pas al algoritmului, rescrieţi în mod corespunzător matricea de distanţe.[904] **La final, desenaţi dendrogramele rezultate.**

***Indicaţie*: Înălţimea corespunzătoare fiecărui cluster non-singleton (adică, a fiecărui nod intern) din dendrogramă va fi considerată ca fiind egală cu distanţa (i.e., conform măsurii de similaritate) dintre cele două sub-clustere constitutive.**

b. Dacă aţi lucrat corect, atunci cele două dendrograme obţinute la punctul a nu coincid [nici măcar] ca structură. Modificaţi două valori din matricea de distanţe dată mai sus, în aşa fel încât de data aceasta cele două dendrograme care se obţin să fie identice ca structură.

c. Procedaţi similar cu cerinţele de la punctul a, dar de această dată pentru *average-linkage*. La actualizarea matricei de distanţe (sau, de „proximitate") veţi ţine cont de formula:

$$\Delta(X \cup Y, Z) \overset{def.}{=} \frac{1}{(|X|+|Y|)\,|Z|} \sum_{x \in X \cup Y} \sum_{z \in Z} d(x,z)$$

$$\overset{calcul}{=} \frac{1}{|X|+|Y|}(|X|\,\Delta(X,Z) + |Y|\,\Delta(Y,Z)). \tag{400}$$

[904] De la o iteraţie la alta, se micşorează cu 1 numărul liniilor precum şi al coloanelor folosite. De exemplu, dacă la prima iteraţie se formează clusterul $\{A, B\}$, veţi înlocui liniile A şi B, precum şi coloanele A şi B cu o linie şi respectiv o coloană notată cu AB, în care se vor pune distanţele de la clusterul $\{A, B\}$ la fiecare dintre celelalte clustere reprezentate în matricea de proximitate la iteraţia respectivă.

unde X, Y şi Z sunt clustere disjuncte două câte două, iar notaţia $|X|$ desemnează cardinalul lui X (adică, numărul de elemente din X).

d. Demonstraţi formula enunţată la punctul c.

28. **(Clusterizare ierarhică aglomerativă, folosind similaritate de tip "single-linkage", "complete-linkage" şi "average-linkage"; aplicare pe date din $\mathbb{R}$)**

$*$ ***CMU, 2009 spring, Ziv Bar-Joseph, final exam., pr. 9.2***

Dorim să clusterizăm numerele de la 1 la 1024, folosind algoritmul de clusterizare ierarhică aglomerativă. Dacă la o iteraţie oarecare distanţele dintre două perechi de clustere au aceeaşi valoare, ne vom folosi de ordonarea numerică.[905]

Să se compare rezultatele obţinute cu trei variante ale funcţiei de similaritate discutate la curs — şi anume, "single-linkage", "complete-linkage" şi "average-linkage" —, specificându-se pentru fiecare dintre aceste variante care este numărul de elemente asignate fiecăruia dintre cele două clustere [care compun clusterul corespunzător nodului rădăcină] de la vârful dendrogramei rezultate.

29. **(Clusterizare ierarhică aglomerativă: aplicare pe date din $\mathbb{R}$)**

□ • ○ ***CMU, 2012 spring, Ziv Bar-Joseph, midterm, pr. 9.1***

Ne propunem să clusterizăm în manieră ierarhică aglomerativă instanţele $2^0, 2^1, 2^2, \ldots, 2^n$ (deci, în total, $n+1$ puncte), folosind distanţa euclidiană. Trasaţi dendrogramele (adică arborii de clusterizare ierarhică) care se obţin pentru fiecare dintre următoarele trei tipuri de funcţii de similaritate: single-, complete- şi average-linkage. Vom considera că fiecare nod neterminal din dendrogramă — adică rădăcina fiecărui cluster non-singleton —, are înălţimea egală cu valoarea măsurii de similaritate (adică, distanţa) dintre cele două clustere din care a fost obţinut.

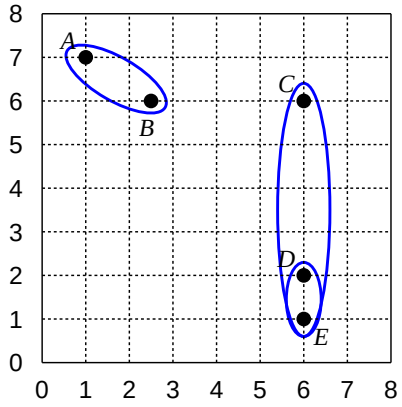
30. **(Clusterizare ierarhică aglomerativă: un exemplu simplu de aplicare, cu single-, complete- şi average-linkage, pe date din $\mathbb{R}^2$)**

prelucrare de Liviu Ciortuz, 2022, după
• ○ ***CMU, 2021 fall, Aarti Signh, Recitation 9, pr. 3***

Considerăm punctele $A(1,7)$, $B(2.5,6)$, $C(6,6)$, $D(6,2)$ şi $E(6,1)$ din planul euclidian. Pe acest dataset veţi aplica algoritmul de clusterizare ierarhică aglomerativă (i.e., bottom-up), folosind pe rând (separat) funcţiile de similaritate single-linkage, complete-linkage şi average-linkage.

Care dintre aceste funcţii de similaritate va conduce după executarea a trei iteraţii consecutive la ierarhia aplatizată prezentată în figura de mai jos? (*Observaţie*: Pentru simplitate, nu am mai desenat şi elipsa corespunzătoare clusterului care include toate punctele de clusterizat.)

[905] Vedeţi de exemplu cum s-a procedat la problema 1.



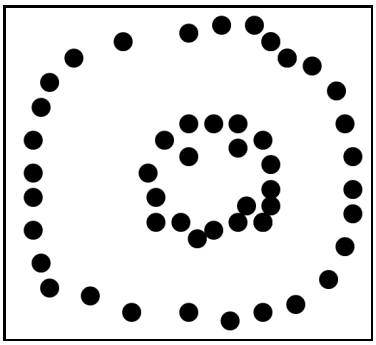
31. **(Clusterizare ierarhică aglomerativă: aplicare în manieră intuitivă pe date din $\mathbb{R}^2$)**

∗ ***CMU, 2010 fall, Ziv Bar-Joseph, midterm, pr. 8.b***

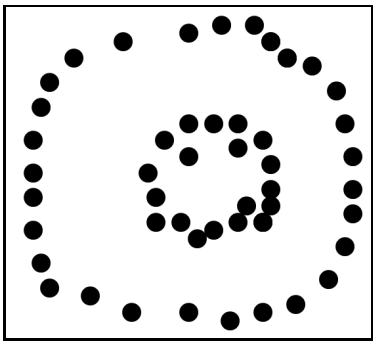
a. Folosind distanţa euclidiană, aplicaţi algoritmul de clusterizare ierarhică aglomerativă cu similaritate de tip "single-linkage" pe datele din figura alăturată.

Indicaţi pe desen care sunt instanţele care formează cele două clustere de la vârful dendrogramei.

***Observaţie*: Nu este necesar să construiţi efectiv dendrograma.**



b. Care este rezultatul dacă se foloseşte similaritate de tip "average-linkage"?

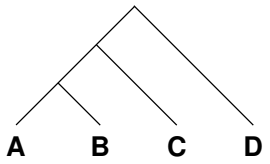


32. **(Clusterizare ierarhică aglomerativă: raţionamente calitative)**

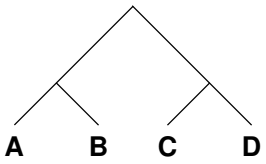
• ○ ***CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, midterm, pr. 9.AB***

A. În figurile de mai jos sunt reprezentate rezultatele clusterizării a patru puncte (A, B, C şi D) folosind o aceeaşi matrice de distanţe T şi două tipuri

diferite de măsuri de similaritate, după cum este indicat sub fiecare dintre cei doi arbori de clusterizare consideraţi.



Single-linkage



Average-linkage

În cele ce urmează, vom nota spre exemplu cu $T(A, B)$ distanţa dintre A şi B dată în matricea de intrare, şi cu

$$T'(\{A, B\}, \{C, D\}) = \frac{T(A, C) + T(A, D) + T(B, C) + T(B, D)}{4}$$

distanţa corespunzătoare măsurii de similaritate "average-linkage" (adică distanţa medie) dintre clusterele $\{A, B\}$ şi respectiv $\{C, D\}$. Ca de obicei, notăm cu $\min(X, Y)$ cea mai mică dintre valorile X şi Y.

Facem *presupunerea* că toate distanţele din matricea de intrare sunt diferite şi că în niciuna dintre inegalităţile de mai jos termenii care se compară nu sunt egali (aşadar, răspunsul este fie $>$ fie $<$).

La fiecare din cele trei puncte de mai jos, precizaţi dacă inegalitatea indicată are loc, nu are loc, sau este imposibil de precizat. Justificaţi în mod riguros alegerea făcută.

a. $\min(T(A, C), T(B, C)) > T(C, D)$.

☐ **adevărat** ☐ **fals** ☐ **imposibil de precizat**

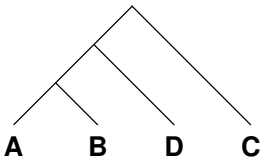
b. $T'(\{A, B\}, \{C\}) > T(C, D)$.

☐ **adevărat** ☐ **fals** ☐ **imposibil de precizat**

c. $T(A, D) > T(A, C)$.

☐ **adevărat** ☐ **fals** ☐ **imposibil de precizat**

B. Considerăm arborele de clusterizare din figura alăturată. Este oare posibil ca acest arbore să fi fost obţinut pornind de la matricea de distanţe T de la punctul A de mai sus folosind similaritate de tip "complete-linkage" (aşa cum se specifică în figură)?



Complete-linkage

33. **(Clusterizare ierarhică [aglomerativă]: o altă funcţie / măsură de similaritate între clustere: metrica / distanţa lui Ward)**

prelucrare făcută de Liviu Ciortuz, după CMU, 2010 fall, Aarti Singh, HW3, pr. 4.1

În acest exerciţiu veţi analiza o modalitate alternativă pentru a defini distanţa dintre două clustere disjuncte, care a fost propusă de către Joe H. Ward în 1963.[906] O vom numi *metrica lui Ward*.

Această metrică defineşte distanţa dintre două clustere disjuncte X şi Y (ambele finite şi incluse în $\mathbb{R}^d$ cu $d \in \mathbb{N}^*$) ca fiind dată de creşterea *sumei pătratelor* distanţelor de la fiecare instanţă x_i la *centroidul* clusterului la care ea este asignată, atunci când reunim cele două clustere ca să formăm un cluster nou. Din punct de vedere formal, vom scrie:

$$\Delta(X,Y) = \sum_{x_i \in X \cup Y} \|x_i - \mu_{X\cup Y}\|^2 - \sum_{x_i \in X} \|x_i - \mu_X\|^2 - \sum_{x_i \in Y} \|x_i - \mu_Y\|^2 \tag{401}$$

unde, spre exemplu, μ_X este centroidul („centrul de greutate“) clusterului X, iar x_i este o instanţă generică dintr-un cluster [oarecare, fixat]. Prin definiţie, aici vom considera $\mu_X = \frac{1}{n_X} \sum_{x_i \in X} x_i$, unde n_X este numărul de elemente din X. Mărimea $\Delta(X,Y)$ din formula (401) poate fi interpretată ca reprezentând *costul pentru combinarea* celor două clustere X şi Y într-un singur cluster.[907]

a. Aduceţi expresia din partea dreaptă a formulei (401) la o formă mai simplă. (Redactaţi toţi paşii intermediari.) *Sugestie*: Formula obţinută trebuie să fie doar în funcţie de numărul de elemente din cele două clustere (n_X şi respectiv n_Y) şi de distanţa $\|\mu_X - \mu_Y\|^2$ dintre centroizii acestor clustere, μ_X şi respectiv μ_Y.

b. Comentaţi metrica lui Ward. Ce credeţi că urmăreşte ea să realizeze? *Sugestie*: Varianta simplificată a formulei de mai sus vă va ajuta să răspundeţi la întrebarea care a fost pusă aici.

c. Presupuneţi că vi se dau două *perechi* de clustere, P_1 şi P_2, iar centroizii celor două clustere din P_1 sunt situaţi la o distanţă mai mare decât distanţa dintre centroizii clusterelor din perechea P_2. Oare, folosind metrica lui Ward, algoritmul de clusterizare aglomerativă va alege *întotdeauna* să „combine“ mai întâi cele două clustere din P_2 (fiindcă sunt situate la o 'distanţă' mai mică)? De ce (nu)? Justificaţi răspunsul dumneavoastră cu ajutorul unui exemplu simplu.

d. Elaboraţi o formulă pentru calculul lui $\Delta(X \cup Y, Z)$, folosind doar cardinalii n_X, n_Y, n_Z şi centroizii μ_X, μ_Y, μ_Z, presupunând că X, Y şi Z sunt clustere disjuncte două câte două.

e. La clusterizare, de obicei nu este uşor să decizi care este numărul potrivit de clustere în care trebuie grupate datele. Folosind metrica lui Ward în conjuncţie cu clusterizarea aglomerativă, puteţi concepe o euristică simplă care să vă ajute să alegeţi [bine] numărul de clustere, K?

[906] Ward, J. H., Jr., *Hierarchical Grouping to Optimize an Objective Function*, Journal of the American Statistical Association, 58 (1963), 236–244.

[907] Remarcaţi faptul că instanţele x_i fiind dintr-un spaţiu $\mathbb{R}^d$, urmează că $\mu_X \in \mathbb{R}^d$ (ca şi ceilalţi centroizi), iar $\|x\|^2 = x \cdot x$ pentru orice $x \in \mathbb{R}^d$, operatorul $\cdot$ desemnând produsul scalar al vectorilor în $\in \mathbb{R}^d$.

34. **(Clusterizare ierarhică aglomerativă: aplicare pe date din $\mathbb{R}^2$, folosind metrica lui Ward)**

Liviu Ciortuz, 2018, folosind datele de la Edinburgh, 2009 fall, C. Williams, V. Lavrenko, HW4, pr. 3

■ ○ ∗

La problema 33 am prezentat o funcţie de similaritate numită *metrica lui Ward*. Potrivit acestei metrici, distanţa dintre două clustere disjuncte X şi Y se defineşte astfel:

$$\Delta(X,Y) = \sum_{x_i \in X\cup Y} \|x_i - \mu_{X\cup Y}\|^2 - \sum_{x_i \in X} \|x_i - \mu_X\|^2 - \sum_{x_i \in Y} \|x_i - \mu_Y\|^2 \tag{402}$$

unde, spre exemplu, μ_X este centroidul [sau „centrul de greutate“ al] clusterului X, iar x_i este o instanţă generică dintr-un cluster [oarecare, fixat]. Prin definiţie, aici vom considera $\mu_X = \frac{1}{n_X}\sum_{x_i\in X} x_i$, unde n_X este numărul de elemente din X. (Similar sunt definiţi centroizii μ_Y şi $\mu_{X\cup Y}$.)

Se poate arăta (vedeţi tot problema 33) că

$$\Delta(X,Y) \quad = \quad \frac{n_X n_Y}{n_X + n_Y}||\mu_X - \mu_Y||^2. \tag{403}$$

***Observaţii*:**

1. Pentru perechi de clustere (X,Y) şi (X',Y') astfel încât $n_X = n_{X'}$ şi $n_Y = n_{Y'}$,[908] formula (403) arată că la clusterizare ierarhică folosind merica lui Ward [la o iteraţie oarecare] este „favorizată“ acea pereche pentru care centroizii (μ_X şi μ_Y, respectiv $\mu_{X'}$ şi $\mu_{Y'}$) sunt mai apropiaţi.

2. Invers, dacă $\|\mu_X - \mu_Y\| = \|\mu_{X'} - \mu_{Y'}\|$, atunci este favorizată perechea pentru care ponderea[909] (adică $\frac{n_X n_Y}{n_X + n_Y}$, respectiv $\frac{n_{X'} n_{Y'}}{n_{X'} + n_{Y'}}$) este mai mică.

Aceste două *observaţii* vă vor ajuta să simplificaţi / reduceţi foarte mult calculele pe care ar trebui să le faceţi pentru a rezolva următoarea cerinţă!

Aplicaţi algoritmul de clusterizare ierarhică aglomerativă pe setul de date de la problema 2,

$$A:(-4,-2), B:(-3,-2), C:(-2,-2), D:(-1,-2), E:(+1,-1)$$
$$F:(+1,+1), G:(+2,+3), H:(+3,+2), I:(+3,+4), J:(+4,+3)$$

utilizând [de această dată] metrica lui Ward.[910] Ca rezultat al clusterizării, veţi reprezenta *dendrograma* sub formă *aplatizată*, folosind elipse (şi indici) pentru a indica clusterele formate.

Coincide rezultatul obţinut aici cu vreunul din rezultatele de la problema 2?

***Precizare*: Dacă la o iteraţie a algoritmului de clusterizare distanţele (adică similarităţile) dintre două perechi de clustere au aceeaşi valoare, veţi considera că prioritatea la alcătuirea noului cluster este dictată de ordinea alfabetică.**

[908] Sau, mai general, atunci când $\frac{n_X n_Y}{n_X + n_Y} = \frac{n_{X'} n_{Y'}}{n_{X'} + n_{Y'}}$.

[909] Această pondere este jumătate din *media armonică* a cardinalilor n_X şi n_Y.

[910] Acest tip de clusterizare este cunoscut în literatura de specialitate şi sub numele de *clusterizare ierarhică bazată pe centroizi* (engl., centroid-based).

35. **(Clusterizare ierarhică aglomerativă: calcularea eficientă a matricei de „proximitate“)**

□ • ○ ***CMU, 2010 fall, Aarti Singh, HW3, pr. 4.2***

La curs, atunci când am prezentat algoritmul de clusterizare ierarhică aglomerativă, am arătat că este necesar să actualizăm matricea de distanţe (numită şi *matricea de proximitate*; engl., the closeness matrix) la fiecare iteraţie a algoritmului.[911] Calcularea distanţelor dintre clusterul nou-obţinut la iteraţia curentă (prin „contopirea“ clusterelor X şi Y) şi toate celelalte clustere se poate face, în mod *naiv*, folosind instanţele [individuale] din clustere.

a. Presupunem că dorim să obţinem / realizăm o procedură mai eficientă de actualizare a matricei de proximitate, folosind *doar* elementele din forma curentă a matricei de proximitate, adică fără să folosim instanţele [individuale] din clustere. Pentru care dintre următoarele tipuri de funcţii de similaritate putem să atingem acest obiectiv?

i. **single-linkage,**
ii. **complete-linkage,**
iii. **average-linkage,**
iv. **metrica lui Ward.**[912]

Prezentaţi raţionamentul dumneavoastră, până la nivel de detaliu.

b. Dacă la vreunul dintre cazurile *i–iv* nu este posibil să atingem obiectivul pe care ni l-am propus la punctul *a* — repetăm, folosind *doar* conţinutul curent al matricei de proximitate —, am putea totuşi să reuşim dacă (*în plus*) vom memora doar câteva informaţii adiţionale pentru fiecare cluster? Pentru fiecare din cazurile identificate [la acest punct] veţi specifica în mod clar ce informaţii adiţionale trebuie memorate.

36. **(Clusterizare ierarhică: particularizare pentru cazul când similaritatea se exprimă cu ajutorul funcţiei cosinus între instanţe „normalizate“)**

Liviu Ciortuz, 2016, pornind de la
□ ***Foundations of Statistical Natural Language Processing, C. Manning, H. Schütze, MIT Press, 1999, pag. 507-509***

Ştim că produsul scalar al doi vectori $x = (x_1, \ldots, x_d)$ şi $y = (y_1, \ldots, y_d)$ din $\mathbb{R}^d$ se defineşte astfel: $x \cdot y = \sum_{i=1}^{d} x_i y_i$. Dacă atât x cât şi y sunt nenuli, atunci definim $\cos(x, y) = \frac{x \cdot y}{||x||\ ||y||}$.[913] Dacă x şi y sunt vectori unitari, adică $||x|| = ||y|| = 1$, atunci definiţia de mai sus devine: $\cos(x, y) = x \cdot y$. Atunci când x şi y nu sunt unitari, îi putem „normaliza“, înmulţindu-i cu scalarii $1/||x||$ şi respectiv $1/||y||$.

În continuare vom presupune că folosim doar vectori unitari / normalizaţi.

Funcţia cos definită mai sus este folosită uneori în clusterizarea ierarhică, ca măsură de similaritate între instanţe, de exemplu la clusterizarea textelor. De

[911] Vedeţi, ca exemplu, rezolvarea problemei 27.
[912] Vedeţi problema 33.
[913] Se poate arăta că această definiţie extinde în mod natural definiţia funcţiei trigonometrice cosinus, aşa cum o ştim de la geometria plană din şcoală. Cosinusul este [măsurat în] funcţie de unghiul dintre cei doi vectori.

obicei, vectorii consideraţi într-un astfel de context sunt nenegativi (i.e., au toate componentele pozitive sau 0), aşa că atunci valoarea maximă a lui cos va fi 1 (şi anume, pentru cazul $x = y$), iar 0 va fi valoarea sa minimă (şi anume, pentru cazul când x şi y sunt vectori ortogonali).

***Observaţie*: Trebuie să ştim că, deşi uneori această funcţie este numită „distanţa cosinus", ea nu este de fapt o *măsură de distanţă*, fiindcă nu satisface condiţiile de nenegativitate, identitatea indiscernabililor şi inegalitatea triunghiului.**[914]

a. Folosind funcţia cos, definim *coeziunea* (internă) a unui cluster oarecare X astfel:[915]

$$Coh(X) = \frac{1}{C^2_{|X|}} \cdot \frac{1}{2} \cdot \sum_{x \neq y \in X} \cos(x, y),$$

adică exact *media similarităţii* instanţelor din clusterul X. În formula aceasta, am notat cu $|X|$ cardinalul clusterului X, adică numărul de instanţe din X.

Arătaţi că $Coh(X)$ poate fi exprimată folosind doar $|X|$ şi centrul de greutate (sau, *centroidul*) clusterului X. Acesta din urmă este notat cu μ_X şi este definit ca $\frac{1}{|X|} \sum_{x \in X} x$.

b. Date fiind două clustere oarecare disjuncte A şi B, particularizaţi expresia funcţiei de coeziune (Coh) care a fost definită la punctul precedent pentru clusterul $A \cup B$. Ulterior veţi exprima $Coh(A \cup B)$ folosind doar $|A|$, $|B|$, μ_A şi μ_B.

c. Arătaţi că la clusterizare ierarhică cu similaritate de tip "average-linkage" bazată pe măsura cos, date fiind două clustere oarecare disjuncte A şi B, „distanţa" dintre clusterul $A \cup B$ şi un cluster oarecare X disjunct de A şi de B se poate exprima folosind doar cardinalii $|A|$, $|B|$ şi $|X|$ şi centroizii μ_A, μ_B şi μ_X.

***Sugestie*: Ca o consecinţă ce decurge din definiţia similarităţii "average-linkage", această „distanţă" se poate calcula cu ajutorul coeziunilor $Coh((A \cup B) \cup X)$, $Coh(A \cup B)$ şi $Coh(X)$.**

***Observaţie*: Ca şi la problema 35.cd, va rezulta că la fiecare iteraţie a algoritmului de clusterizare aglomerativă, matricea de „proximitate" poate fi actualizată folosind doar conţinutul acestei matrice la iteraţia precedentă, cardinalii clusterelor ($n_X \overset{not.}{=} |X|$) şi centroizii lor ($\mu_x$).**

37. **(O legătură între clusterizarea ierarhică şi arborii de decizie)**

• ○ ***CMU, 2012 spring, Ziv Bar-Joseph, midterm, pr. 9.3***

Considerăm un set de instanţe etichetate. Ne propunem să folosim un *arbore de clusterizare ierarhică* (adică, o dendrogramă) pe post de *arbore de decizie*. Pentru aceasta, selectăm o *măsură de similaritate* de tip *linkage* şi construim arborele de clusterizare ierarhică bazat pe setul de date de antrenament furnizat, fără a ţine cont de etichetele instanţelor. Considerăm o instanţă oarecare

[914] Pentru formalizarea acestor proprietăţi, vedeţi problema 2 de la capitolul *Învăţare bazată pe memorare*.
[915] Prin C^2_n notăm ca de obicei numărul de combinări de n obiecte luate câte 2.

de test (engl., *test sample* sau *query sample*) x_q. Vom parcurge cu ea arborele pe care tocmai l-am construit, în felul următor.

La fiecare nod din arbore calculăm distanţele dintre [clusterul singleton format doar din instanţa] x_q şi fiecare dintre sub-clusterele care sunt descendenţii imediaţi ai nodului respectiv, folosind măsura de similaritate pe care am ales-o mai înainte, după care selectăm ramura corespunzătoare minimului dintre aceste distanţe. În acest fel, indiferent care ar fi instanţa de test considerată, putem să o propagăm (de sus în jos) în arborele de clusterizare ierarhică. La final, adică atunci când se ajunge la un nod frunză (care reprezintă un cluster singleton), asignăm lui x_q eticheta nodului respectiv.

Vă cerem să identificaţi o *măsură de similaritate* de tip *linkage* care are proprietatea că atunci când este folosită în maniera descrisă mai sus funcţionează exact ca un arbore de decizie care (la rândul lui) este echivalent cu un anumit algoritm de clasificare automată pe care l-am prezentat la curs. Care este acest clasificator? (Justificaţi într-o manieră suficient de detaliată.)

38. **(Clusterizare ierarhică bottom-up: implementare)**

Liviu Ciortuz, 2015

a. Elaboraţi — de preferinţă în limbajul de programare C/C++ — implementarea algoritmului de clusterizare ierarhică bottom-up.

Veţi considera că instanţele de clusterizat sunt puncte într-un spaţiu euclidian $\mathbb{R}^d$ (d fiind un număr natural, $d \geq 1$). Veţi presupune de asemenea că instanţele sunt înregistrate într-un fişier text, câte o instanţă $x_i = (x_{i_1}, x_{i_2}, \ldots, x_{i_d})$ pe fiecare linie, şi că între fiecare două atribute / coordonate succesive ale unei astfel de instanţe se află cel puţin un [caracter de tip] spaţiu. Numele acestui fişier text, eventual numărul de dimensiuni al spaţiului în care se lucrează (d), precum şi tipul funcţiei de similaritate (*single-linkage*, *complete-linkage*, *average-linkage* sau *metrica lui Ward*[916]) vor fi indicate ca argumente în linia de comandă a programului.

Ca punct de plecare puteţi considera pseudo-codul din *Foundations of Statistical Natural Processing*, C. Manning, H. Schütze, MIT Press, 2002, pag. 496, pe care însă îl veţi extinde cu calculul *înălţimilor* asociate nodurilor din dendrograma rezultantă.

Distanţele dintre instanţele date vor fi calculate şi apoi memorate într-o matrice.[917] Pentru a face în mod eficient actualizarea acestei matrice la fiecare iteraţie a algoritmului de clusterizare ierarhică bottom-up, veţi folosi rezultatele de la problema 35.

Înălţimea unui de nod oarecare din dendrogramă va fi dată de media tuturor distanţelor $d(x, y)$, unde x şi y sunt puncte în clusterul respectiv (vedeţi, ca exemplu, problema 1).[918]

[916] Vedeţi problema 33.

[917] Iniţial veţi lucra cu o matrice pătratică. Ulterior, pentru motive de economie de memorie, veţi putea înlocui această matrice cu una triunghiulară (intuitiv, aceasta din urmă reprezintă zona de desupra diagonalei principale a matricei pătratice, memorată sub o formă convenabilă).

[918] Coeziunea unui cluster se defineşte ca fiind inversul acestei medii.

Ca să „reprezentaţi“ în mod convenabil ieşirea programului puteţi folosi o notaţie bazată pe expresii cu paranteze imbricate.[919] Ulterior, veţi putea extinde eventual programul, cuplându-l cu o interfaţă grafică, atât pentru colectarea intrărilor (vedeţi de exemplu problema 31) cât şi pentru ieşiri (trasarea efectivă a dendrogramelor).

Vă sugerăm să testaţi programul pe datele din problemele 1, 2, 27 şi 28.

b. Extindeţi implementarea în aşa fel încât, dacă se solicită (printr-o anumită optiune la linia de comandă), să se livreze toate soluţiile posibile. Testaţi această variantă a programului dumneavoastră folosind datele de la problema 3.

39. **(Clusterizare ierarhică top-down: implementare)**

□ ***Liviu Ciortuz, 2016***

Implementaţi — de preferinţă în C/C++ — algoritmul de clusterizare ierarhică top-down prezentat în problema rezolvată 6. Pentru convenienţă, puteţi porni de la o implementare oarecare disponibilă pe web pentru algoritmul lui Kruskal sau algoritmul lui Prim pentru aflarea arborelui de cost minim (MST) dintr-un graf. Testaţi implementarea realizată de dumneavoastră pe datele de la problema pe care am menţionat-o anterior.

7.2.2 Algoritmul K-means

40. **(Algoritmul K-means: aplicare în $\mathbb{R}^2$)**

■ ○ ***T.U. Dresden, 2006 summer, S. Hölldobler, A. Grossmann, HW3***

Folosiţi algoritmul K-means şi distanţa euclidiană pentru a grupa următoarele 8 instanţe din $\mathbb{R}^2$ în 3 clustere:

$$A(2,10),\ B(2,5),\ C(8,4),\ D(5,8),\ E(7,5),\ F(6,4),\ G(1,2),\ H(4,9).$$

Se vor lua drept centroizi iniţiali punctele A, D şi G.

a. Rulaţi prima iteraţie a algoritmului K-means. Pe un grid de valori 10×10 veţi marca instanţele date, poziţiile centroizilor la începutul primei iteraţii şi componenţa fiecărui cluster la finalul acestei iteraţii. (Trasaţi mediatoarele segmentelor determinate de centroizi, ca *separatori* ai clusterelor.)

b. Câte iteraţii sunt necesare pentru ca algoritmul K-means să conveargă? Desenaţi pe câte un grid rezultatul rulării fiecărei iteraţii.

[919] De exemplu, pentru dendrograma obţinută la problema 1 se poate folosi notaţia simplă ("flat hierarchy"):

$$((x_1, x_2), ((x_3, ((x_4, x_5), x_6)), ((x_7, x_8), (x_9, x_{10}))))$$

sau, o variantă extinsă cu informaţii despre ordinea de formare a clusterelor şi înălţimile nodurilor (interne) corespunzătoare clusterelor în dendrogramă.

$$((x_1, x_2)^{0.2}_{C1}, ((x_3, ((x_4, x_5)^{0.1}_{C1}, x_6)^{0.2}_{C5})^{0.3(6)}_{C7}, ((x_7, x_8)^{0.1}_{C2}, (x_9, x_{10})^{0.1}_{C3})^{0.2(3)}_{C6})^{1.1}_{C8})^{1.77(3)}_{C9}.$$

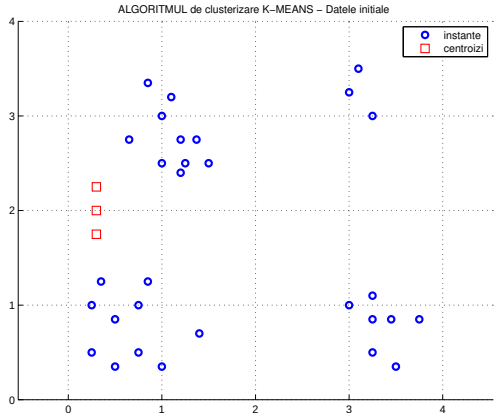
41. **(Algoritmul K-means: aplicare pe date din $\mathbb{R}^2$)**

$*$ ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW3, pr. 5***

Aplicaţi algoritmul K-means pe setul de date din imaginea următoare.

Cerculeţele reprezintă instanţele de clusterizat, iar pătrăţelele sunt centroizii iniţiali ai clusterelor. Pentru fiecare iteraţie a algoritmului desenaţi centroizii şi separatorii care definesc fiecare cluster. Folosiţi oricâte imagini aveţi nevoie până ajungeţi la convergenţă.

Coordonatele acestor instanţe, precum şi cele ale centroizilor vă sunt puse la dispoziţie în fişiere depuse pe site-ul acestei cărţi.[920]



***Observaţie*: La execuţia algoritmului se consideră că în cazul în care un centroid nu are puncte asignate lui, atunci el rămâne pe loc în iteraţia respectivă.**

42. **(Algoritmul K-means: aplicare în $\mathbb{R}$ şi $\mathbb{R}^2$; verificarea monotoniei „criteriului“ coeziunii intra-clustere (J))**

■ **(pt. punctul a)** □ $*$ ***Liviu Ciortuz, 2020, 2025***

În acest exerciţiu veţi folosi algoritmul K-means în varianta dată în enunţul exerciţiului 12.[921]

Vă readucem aminte definiţia aşa-numitului criteriu J, care este o măsură a coeziunii intra-clustere:

$$J(C^{(t)}, \mu^{(t)}) = \sum_{i=1}^{n} (x_i - \mu^{(t)}_{C^{(t)}(x_i)})^2,$$

unde $C^{(t)}$ este ansamblul clusterelor la momentul / iteraţia t, apoi $\mu^{(t)}$ desemnează ansamblul centroizilor la iteraţia t şi, în sfârşit, $\mu^{(t)}_{C^{(t)}(x_i)}$ este centroidul clusterului la care este asignată instanţa x_i la iteraţia t.

a. Fie următorul set de date din $\mathbb{R}$: -9, -8, -7, -6, -5, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9. Considerăm $\mu^{(0)}_1 = -20$ şi $\mu^{(0)}_2 = -10$.[922] **Demonstraţi în manieră *analitică* (NU numeric!) că pentru $t = 1$ avem**

$$J(C^{(t)}, \mu^{(t)}) \leq J(C^{(t-1)}, \mu^{(t-1)}).$$

b. Cerinţele de la acest punct sunt similare cu cele de la punctul precedent, însă de data aceasta veţi lucra pe următorul set de date din $\mathbb{R}^2$: $A(-1, 0)$,

[920] http://profs.info.uaic.ro/~ciortuz/ML.ex-book/res/CMU.2004f.TM+AM.HW3.pr5.cl.dat, http://profs.info.uaic.ro/~ciortuz/ML.ex-book/res/CMU.2004f.TM+AM.HW3.pr5.init.dat.

[921] *Atentie*! În eventualitatea că la o iteraţie oarecare a algoritmului K-means un cluster este vid, centroidul său rămâne pe loc la iteraţia respectivă.

[922] Acestea sunt datele de la exerciţiul 17.

$B(1,0)$, $C(0,1)$, $D(3,0)$, $E(3,1)$; **drept centroizi iniţiali veţi considera** $\mu_1^{(0)} = A$ **şi** $\mu_2^{(0)} = E$.[923]

c. Revenim acum la setul de date de la punctul a**. Definim**

$$\underline{J}_K \stackrel{def.}{=} \min_{C,\mu} J(C,\mu)$$

Precizaţi valorile pentru

i. $\underline{J}_2$

ii. $\underline{J}_{10}$, $\underline{J}_{11}$, ..., $\underline{J}_{15}$

iii. $\underline{J}_9$.

Veţi justifica riguros răspunsurile date pentru fiecare dintre cele trei cazuri de mai sus (i**,** ii **şi** iii**).**

43. **(Algoritmul K-means: convergenţă)**

∗ ***CMU, 2009 spring, Ziv Bar-Joseph, final exam, pr. 1.7***

Fie un set de date neetichetate şi o K-partiţie a acestui set de date, generată la sfârşitul unei iteraţii oarecare a algoritmului K-means. Este posibil ca algoritmul K-means să reviziteze această K-partiţie?

***Observaţie*: Noţiunea de K-partiţie a fost introdusă la problema 11. Aveţi grijă că noţiunea de *K-partiţie revizitată* de mai sus nu este aceeaşi cu cea de *K-partiţie stabilă* de la problema 11.**

44. **(Algoritmul K-means: discuţie asupra alegerii valorii lui K)**

• ○ ***CMU, 2010 fall, Aarti Singh, HW3, pr. 5.4***
CMU, 2015 spring, T. Mitchell, N. Balcan, HW7, pr. 1.1

Unul dintre dezavantajele algoritmului K-means este acela că trebuie specificată valoarea parametrului K. Ce părere aveţi despre următoarea *strategie*:

Se poate alege K în mod automat, încercând toate valorile lui posibile ($K = 1, \ldots, n$, unde n este numărul de instanţe de clusterizat), şi reţinând apoi acea valoare a lui K pentru care s-a obţinut cea mai mică valoare a criteriului „sumei celor mai mici pătrate", J_K.[924]

Justificaţi de ce această strategie este bună (sau rea).

45. **(Algoritmul K-means ca algoritm de optimizare: maximizarea aproximativă a distanţelor dintre centroizii clusterelor)**

■ • ○ ***CMU, 2010 fall, Aarti Singh, HW3, pr. 5.2***

În această problemă vom lucra cu o versiune a algoritmului K-means uşor modificată faţă de cea dată în enunţul problemei 12.

[923] Acestea sunt datele de la exerciţiul 7.
[924] Pentru definiţia riguroasă a acestui criteriu vedeţi problema 12.

Fie $X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ o mulţime de instanţe, iar K numărul de clustere cu care vom lucra. De data aceasta vom specifica asignările instanţelor la clustere folosind o *matrice-indicator* $\gamma \in \{0,1\}^{n\times K}$, cu $\gamma_{ij} = 1$ dacă şi numai dacă $\mathbf{x}_i$ aparţine clusterului j. Vom impune ca fiecare instanţă să aparţină câte unui singur cluster, deci $\sum_{j=1}^{K} \gamma_{ij} = 1$.

După cum s-a arătat la problema 12, algoritmul K-means „estimează" matricea γ făcând minimizarea criteriului (sau, a „măsurii de distorsiune") J, pe care, folosind matricea γ, îl rescriem sub forma

$$J(\gamma, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K) \;=\; \sum_{i=1}^{n}\sum_{j=1}^{K} \gamma_{ij}\|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2,$$

unde $\|\cdot\|$ desemnează norma vectorială L_2. Concret, algoritmul K-means alternează „estimarea" matricei γ cu re-calcularea centroizilor $\boldsymbol{\mu}_j$.

Acestea fiind spuse, putem da acum noua versiune a algoritmului K-means:[925]

- Se iniţializează în mod arbitrar centroizii $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K$ şi se ia $C = \{1, \ldots, K\}$.
- Atâta timp cât valoarea lui J descreşte în mod strict,[926] repetă:

Pasul 1: Calculează γ astfel:

$$\gamma_{ij} \;\leftarrow\; \begin{cases} 1, & \text{dacă } \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2 \le \|\mathbf{x}_i - \boldsymbol{\mu}_{j'}\|^2,\ \forall j' \in C, \\ 0, & \text{în caz contrar.} \end{cases}$$

În caz de egalitate, alege în mod arbitrar cărui cluster (dintre cele eligibile) să-i aparţină $\mathbf{x}_i$.

Pasul 2: Recalculează $\boldsymbol{\mu}_j$ folosind matricea γ actualizată:
Pentru fiecare $j \in C$, dacă $\sum_{i=1}^{n} \gamma_{ij} > 0$, asignează

$$\boldsymbol{\mu}_j \;\leftarrow\; \frac{\sum_{i=1}^{n} \gamma_{ij}\mathbf{x}_i}{\sum_{i=1}^{n} \gamma_{ij}}.$$

Altfel, menţine neschimbat centroidul μ_j.

Vom nota cu $\bar{\mathbf{x}}$ media instanţelor date şi vom considera următoarele trei cantităţi:[927]

$$\textbf{Variaţia totală: } T(X) = \frac{\sum_{i=1}^{n} \|\mathbf{x}_i - \bar{\mathbf{x}}\|^2}{n}$$

$$\textbf{Variaţia intra-clustere: } W_j(X) = \frac{\sum_{i=1}^{n} \gamma_{ij}\|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2}{\sum_{i=1}^{n} \gamma_{ij}} \textbf{ pentru } j = 1, \ldots, K$$

$$\textbf{Variaţia inter-clustere: } B(X) = \sum_{j=1}^{K} \left(\frac{\sum_{i=1}^{n} \gamma_{ij}}{n}\right) \|\boldsymbol{\mu}_j - \bar{\mathbf{x}}\|^2.$$

[925] Această versiune a algoritmului K-means poate servi pentru a introduce [o variantă simplă pentru] algoritmul EM/GMM — vedeţi pseudo-codul de la exerciţiul 15 — ca versiune probabilistă a algoritmului K-means. De asemenea, această versiune a algoritmului K-means poate servi pentru a demonstra o *proprietate* interesantă, şi anume, *convergenţa* [în anumite condiţii a] rezultatelor acestor doi algoritmi; vedeţi exerciţiul 65.

[926] Această condiţie de oprire este uşor diferită de cea formulată de Lloyd: opreşte execuţia algoritmului atunci când matricea γ nu se mai schimbă.

[927] Veţi observa că $T(X)$, prima dintre aceste cantităţi este o „măsură" a împrăştierii instanţelor $\mathbf{x}_i$ faţă de $\bar{\mathbf{x}}$, centrul de greutate al întregii mulţimi; ea nu depinde de μ sau de γ. Mai exact, $T(X)$ este media pătratelor distanţelor de la instanţele din mulţimea X la $\bar{\mathbf{x}}$. Similar, $W_j(X)$ este media pătratelor distanţelor de la instanţele din clusterul j la centroidul $\boldsymbol{\mu}_j$, care este totuna cu raportul dintre coeziunea clusterului j şi cardinalul acestui cluster. În fine, $B(X)$ este suma ponderată a pătratelor distanţelor de la centroizii $\boldsymbol{\mu}_j$ la $\bar{\mathbf{x}}$, centrul de greutate al mulţimii de instanţe X.

a. Care este relaţia dintre aceste trei cantităţi?

***Observaţie*: Veţi ţine cont că această relaţie poate să conţină un termen suplimentar care nu este menţionat mai sus.**

b. Folosind relaţia stabilită la punctul a, arătaţi că putem interpreta algoritmul K-means ca tinzând să minimizeze (şi anume descrescând, dar nu neapărat strict monoton) o medie ponderată a variaţiei intra-clustere în timp ce el tinde să maximizeze (crescând) însă doar în mod *aproximativ* variaţia inter-clustere.

46. **(Algoritmul K-means: aplicare pe date din $\mathbb{R}^3$ calcularea variaţiei / coeziunii intra- şi inter-clustere)**

prelucrare de Liviu Ciortuz, 2021, după
□ • · ***Andreas Wickert, Luis Sa-Couto, Machine Learning – A Journey to Deep Learning, 2021, pag. 370-385***

Fie următoarele instanţe de antrenament, neetichetate:

$$x_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix},\ x_2 = \begin{pmatrix} 8 \\ 8 \\ 4 \end{pmatrix},\ x_3 = \begin{pmatrix} 3 \\ 3 \\ 0 \end{pmatrix},\ x_4 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix},\ x_5 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix},\ x_6 = \begin{pmatrix} 3 \\ 2 \\ 1 \end{pmatrix}.$$

a. Aplicaţi algoritmul K-means cu $K = 2$ pe acest set de date, până ajungeţi la convergenţă. Pentru iniţializarea centroizilor, veţi lua primele K instanţe din setul de date furnizat.

b. Aplicaţi algoritmul K-means cu $K = 3$ pe acelaşi set de date, până ajungeţi la convergenţă. Pentru iniţializarea centroizilor, veţi lua primele K instanţe din setul de date furnizat.

c. Ce valoare a lui $K \in \{2, 3\}$ produce o clusterizare mai bună dacă se foloseşte drept criteriu de evaluare *variaţia / coeziunea intra-clustere* (adică, suma pătratelor distanţelor de la fiecare instanţă la centroidul cel mai apropiat)?

d. Ce valoare a lui $K \in \{2, 3\}$ produce o clusterizare mai bună dacă se foloseşte drept criteriu de evaluare *variaţia / coeziunea inter-clustere*? (Vedeţi definiţia de la problema 45.)

47. **(Clusterizare partiţională în $\mathbb{R}$: un algoritm de programare dinamică, cu complexitate $\mathcal{O}(Kn^2)$ pentru calculul minimului criteriului „celor mai mici pătrate“ (J))**

prelucrare de Liviu Ciortuz, după
□ • ○ ***CMU, 2015 spring, T. Mitchell, N. Balcan, HW7, pr. 1.5-8***

Ştim că, date fiind instanţele $x_1, \ldots, x_n \in \mathbb{R}^d$ cu $d \in \mathbb{N}^*$, algoritmul de clusterizare K-means urmăreşte să găsească centroizii a K clustere, şi anume $\mu_j \in \mathbb{R}^d, j \in \{1, \ldots, K\}$, astfel încât suma pătratelor distanţelor de la fiecare instanţă la cel mai apropiat centroid să fie minimă (vedeţi problema 12). Altfel

spus, K-means încearcă să găsească acei $\mu_1, \ldots, \mu_K$ care minimizează valoarea aşa-numitului criteriu al „celor mai mici pătrate“:

$$J \overset{def.}{=} \sum_{i=1}^{n} \min_{j \in \{1,\ldots,K\}} \|x_i - \mu_j\|^2,$$

Pentru a realiza aceasta, algoritmul K-means procedează iterativ, alternând asignarea fiecărei instanţe x_i ($i = 1, \ldots, n$) la cel mai apropiat centroid cu actualizarea centroizilor clusterelor (concret, μ_j va deveni media instanţelor asignate la clusterul j).

În general, găsirea minimului criteriului J pentru o valoare fixată a lui K este o problemă NP-dificilă (engl., NP-hard). Totuşi, se poate arăta că ea poate fi rezolvată în timp polinomial (în n şi K) dacă instanţele de clusterizat sunt într-un spaţiu unidimensional ($d = 1$). La acest exerciţiu ne vom concentra atenţia asupra acestui caz.

a. Să considerăm situaţia în care $K = 3$ şi ne sunt date 4 instanţe, $x_1 = 1, x_2 = 2, x_3 = 5$ şi $x_4 = 7$. Care este clusterizarea optimă pentru acest set de date? Cât este valoarea corespunzătoare pentru funcţia obiectiv J?

b. Am putea fi tentaţi să credem că în cazul $d = 1$ algoritmul K-means converge în mod cert la valoarea minimă (globală) a criteriului J. Considerând din nou instanţele date la punctul a, arătaţi că există o asignare suboptimală a lor la clustere, pe care algoritmul K-means n-o poate îmbunătăţi. (Indicaţi asignarea, arătaţi de ce este suboptimală şi explicaţi de ce anume ea nu va putea fi îmbunătăţită.)

c. Presupunem că sortăm instanţele noastre astfel încât $x_1 \leq x_2 \leq \cdots \leq x_n$. Demonstraţi că orice *asignare optimă* a acestor instanţe la clustere — în sensul că orice instanţă este asociată la cel mai apropiat centroid — are proprietatea că fiecare cluster nevid corespunde unui anumit „interval“ de instanţe. Adică, pentru fiecare cluster nevid j există $i_1, i_2 \in \{1, \ldots, n\}$, cu $i_1 \leq i_2$, astfel încât clusterul acesta constă din instanţele $\{x_{i_1}, x_{i_1+1}, \ldots, x_{i_2}\}$.[928]

d. Concepeţi un algoritm de clusterizare de tip programare dinamică având complexitatea $O(Kn^2)$, ca înlocuitor pentru algoritmul K-means în cazul unidimensional.

***Indicaţie*: Dat fiind rezultatul de la punctul c, ceea ce trebuie să „optimizăm“ / setăm sunt cele $K - 1$ graniţe / margini (engl., boundaries) ale clusterelor, marginea cu numărul de ordine i fiind cea mai mare instanţă din clusterul i.**

48. **(K-means pentru comprimarea imaginilor)**

• ○ *CMU, 2017 fall, Nina Balcan, HW5, pr. 3, question 12*

La curs am arătat că algoritmul K-means poate fi folosit pentru comprimarea imaginilor.

Presupunem că sunt necesari 24 de biţi pentru a memora valorile celor trei componente {R, G, B} în care se descompune culoarea fiecărui pixel. Câţi biţi sunt necesari pentru a memora o imagine care are N pixeli?

[928] Se poate observa imediat că de fapt această proprietate este satisfăcută mereu pe parcursul aplicării algoritmului K-means pe seturi de date din $\mathbb{R}$.

Vom presupune acum că folosim algoritmul K-means (cu K clustere) pentru a comprima imagini şi că identificăm culoarea pentru fiecare pixel folosind id-ul clusterului său (un număr din mulţimea $\{0, \ldots, K-1\}$). Câţi biţi sunt necesari acum pentru a memora imaginea [care are N pixeli]?

Care este raportul de compresie?

49. **(Algoritmul K-means: aplicare, folosind distanţa euclidiană, respectiv distanţa Manhattan / norma L_1 robusteţea algoritmului K-means la prezenţa outlier-elor)**

prelucrare de Liviu Ciortuz, după
• ◦ CMU, 2010 fall, Aarti Singh, HW3, pr. 5.3
CMU, 2014 spring, B. Poczos, A. Singh, HW3, pr. 1.4
CMU, 2014 spring, Seyoung Kim, HW3, pr. 1.1

Fie setul date alăturat (X), fiecare rând / linie reprezentând o instanţă.

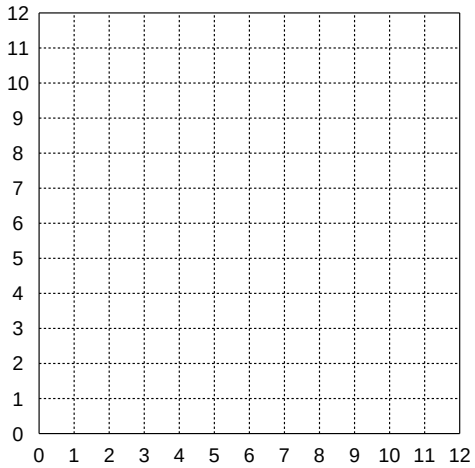
A	$(1,1)$
B	$(3,3)$
C	$(6,6)$
D	$(6,12)$
E	$(9,9)$
F	$(11,11)$
G	$(0,3)$
H	$(3,0)$
I	$(9,3)$

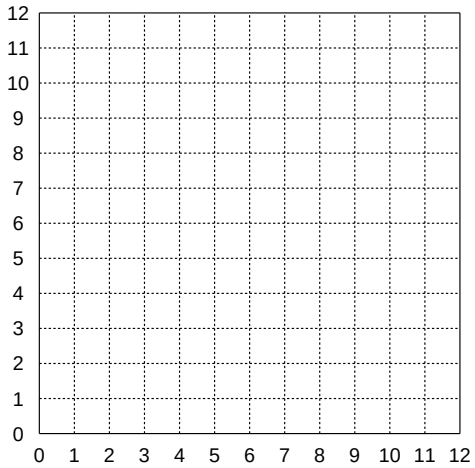
A. K-means folosind distanţa euclidiană

Aplicaţi algoritmul K-means pe acest set de date, folosind $K = 3$ şi distanţa euclidiană. La pasul de iniţializare nu veţi seta poziţiile centroizilor ci, în schimb, veţi iniţializa clusterele, după cum urmează:
$C1 : \{A, B, F\}$, $C2 : \{C, H, I\}$, $C3 : \{D, E, G\}$.
În consecinţă, veţi inversa paşii 1 şi 2 din ciclul iterativ al algoritmului K-means.

Folosiţi gridurile de mai jos; puteţi adăuga şi altele decă veţi considera că este necesar.

Puteţi proceda fie în *maniera analitică* (calculând distanţele de la instanţe la centroizi), fie trasând mediatoarele segmentelor care unesc centroizii.





B. K-means folosind distanţa Manhattan

În această parte a exerciţiului vom folosi distanţa Manhattan, desemnată prin

$\|\cdot\|_1$, ceea ce înseamnă că vom defini [ca nouă „măsură de distorsiune"] funcţia

$$J_1(\gamma, \boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_K) \;=\; \sum_{i=1}^{n}\sum_{k=1}^{K} \gamma_{ik} \|\mathbf{x}_i - \boldsymbol{\mu}_k\|_1.$$

Vă reamintim faptul că *distanţa Manhattan* (d_M) dintre un vector $x = (x_1, \ldots, x_p)$ şi un alt vector $y = (y_1, \ldots, y_p)$, ambii fiind din $\mathbb{R}^p$, este definită astfel:

$$d_M = \sum_{i=1}^{p} |x_i - y_i|.$$

Vom minimiza funcţia J_1 folosind următoarea variantă a algoritmului K-means

- **Iniţializează centroizii $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$.**
- **Atâta timp cât un anumit *criteriu de oprire* nu este încă îndeplinit (de exemplu, atâta timp cât valoarea lui J_1 încă scade), repetă următorii paşi:**
  1. **Calculează [variabilele-indicator] γ astfel:**

$$\gamma_{ik} \;\leftarrow\; \begin{cases} 1, & \|\mathbf{x}_i - \boldsymbol{\mu}_k\|_1 \le \|\mathbf{x}_i - \boldsymbol{\mu}_{k'}\|_1,\ \forall k' \in C, \\ 0, & \textbf{în caz contrar.} \end{cases}$$

     În cazul în care minimul se atinge pentru mai mulţi astfel de k, veţi asigna x_i la centroidul cu indicele cel mai mic pentru care se atinge distanţa minimală.
  2. **Recalculează $\boldsymbol{\mu}_k$ folosind variabilele γ care tocmai au fost actualizate: pentru orice $k = 1, \ldots, K$,**
     dacă $\sum_{i=1}^{n} \gamma_{ik} > 0$, redefineşte centroidul $\boldsymbol{\mu}_k \overset{not}{=} ((\boldsymbol{\mu}_k)_1, \ldots, (\boldsymbol{\mu}_k)_p)$ astfel:

$$(\boldsymbol{\mu}_k)_d \leftarrow \text{median}\,(\{(\mathbf{x}_i)_d \,|\, \gamma_{ik} = 1\}),\ \textbf{pentru}\ d = 1, \ldots, p, \qquad (404)$$

     unde funcţia median ( ) **este specificată în *Definiţia* de mai jos;**
     în caz contrar, lasă centroidul $\boldsymbol{\mu}_k$ neschimbat.

***Definiţie*: Fie o mulţime formată din n numere reale, $\{x_1, x_2, \ldots, x_n\}$.**
Dacă $x_1 \le x_2 \le \ldots \le x_n$, atunci

i. **pentru cazul în care n este impar, *valoarea mediană* a secvenţei $x_1, x_2, \ldots, x_n$ este $x_{\lceil n/2 \rceil}$, unde simbolul $\lceil z \rceil$ desemnează partea întreagă superioară a numărului z;**

ii. **pentru cazul în care n este par, *valoarea mediană* a secvenţei $x_1, x_2, \ldots, x_n$ este orice număr din intervalul $[x_{n/2}, x_{(n/2)+1}]$.**

***Proprietate*: Soluţia problemei de minimizare**

$$\min_{\mu} \sum_{i=1}^{n} |x_i - \mu|$$

este *valoarea mediană* pentru mulţimea $\{x_1, x_2, \ldots, x_n\}$.[929]

[929] În *consecinţă*, răspunsul la problema noastră nu este în mod neapărat unic!

a. Demonstraţia următoare fundamentează formula (404)**:**

$$\begin{aligned}
&\min_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_K} \sum_{i=1}^{n}\sum_{k=1}^{K} \gamma_{ik}\|\mathbf{x}_i - \boldsymbol{\mu}_k\|_1 \\
&= \min_{\boldsymbol{\mu}_1,\ldots,\boldsymbol{\mu}_K} \sum_{k=1}^{K}\sum_{i=1}^{n} \gamma_{ik}\|\mathbf{x}_i - \boldsymbol{\mu}_k\|_1 \\
&= \sum_{k=1}^{K} \min_{\boldsymbol{\mu}_k} \sum_{i=1}^{n} \gamma_{ik}\|\mathbf{x}_i - \boldsymbol{\mu}_k\|_1 \\
&= \sum_{k=1}^{K} \min_{\boldsymbol{\mu}_k} \sum_{i=1}^{n}\sum_{d=1}^{p} \gamma_{ik}|(\mathbf{x}_i)_d - (\boldsymbol{\mu}_k)_d| \\
&= \sum_{k=1}^{K} \min_{\boldsymbol{\mu}_k} \sum_{d=1}^{p}\sum_{i=1}^{n} \gamma_{ik}|(\mathbf{x}_i)_d - (\boldsymbol{\mu}_k)_d| \\
&= \sum_{k=1}^{K}\sum_{d=1}^{p} \min_{(\boldsymbol{\mu}_k)_d} \sum_{i=1}^{n} \gamma_{ik}|(\mathbf{x}_i)_d - (\boldsymbol{\mu}_k)_d|
\end{aligned}$$

Justificaţi fiecare pas al acestei demonstraţii.

b. Dacă setul de date de clusterizat conţine *outliere*, ce versiune a algoritmului K-means ar fi indicat să o folosiţi — aceasta (cu distanţa Manhattan / L_1) ori cea originală (cu distanţa euclidiană / L_2)? Justificaţi.

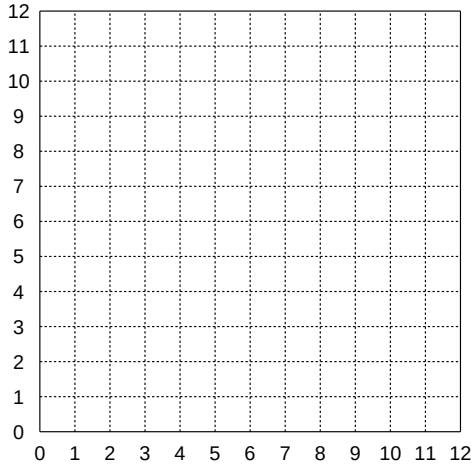
c. Aplicaţi algoritmul K-means pe acelaşi set de date ca la punctul A — tot cu $K = 3$ şi aceleaşi iniţializări ale clusterelor ca acolo —, folosind însă distanţa Manhattan în scrierea criteriului / funcţiei de „distorsiune", aşa cum s-a arătat în versiunea lui K-means de mai sus.

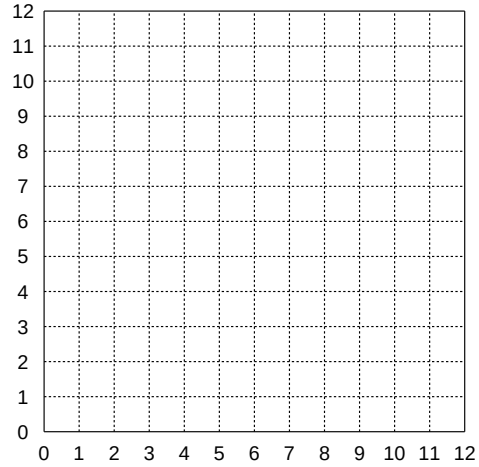
***Atenţie*: Iniţializarea de aici diferă de cea din pseudo-codul dat în enunţ; veţi proceda în consecinţă, adică veţi inversa paşii 1 şi 2 din ciclul iterativ al algoritmului K-means. Şi (din nou!) nu veţi iniţializa poziţiile centroizilor în mod arbitrar!**

Folosiţi gridurile următoare; puteţi adăuga şi altele decă veţi considera că este necesar.

***Observaţie* (1): Nu este nevoie să calculaţi valorile variabilelor-indicator γ_{ij}; în schimb, pur şi simplu asignaţi fiecare instanţă la cel mai apropiat centroid. Pentru aceasta va trebui să faceţi efectiv calculul distanţelor Manhattan; nu veţi trasa mediatoare!**

***Observaţie* (2): În situaţia în care va trebui să aplicaţi cazul ii al definiţiei pe care am dat-o mai sus pentru *valoarea mediană*, veţi considera că funcţia** median( ) **returnează mijlocul intervalului $[x_{n/2}, x_{(n/2)+1}]$.**





d. Demonstraţi *proprietatea* care a fost dată în enunţ [particularizată] pentru următoarele secvenţe de numere:

i. 1, 3, 11

ii. 0, 1, 3, 3

C. Comparaţi rezultatele de la punctele A şi B.c. Dacă ele sunt diferite, care este justificarea?

50. **(Algoritmul K-means++: comparaţie cu algoritmul K-means)**

Liviu Ciortuz, 2018, după
• ***CMU, 2017 fall, Nina Balcan, midterm, pr. 3.2***
CMU, 2012 fall, E. Xing, A. Singh, HW3, pr. 1

***Comentariu*:**

În exerciţiul de faţă vom considera că la faza de iniţializare algoritmul K-means alege centroizii în mod arbitrar dintre instanţele de clusterizat.

Dacă datele de clusterizat sunt [a priori] bine separate în K clustere, atunci este foarte posibil ca la finalul iniţializării să existe [măcar] un cluster din care algoritmul nu a selectat niciun punct. În astfel de situaţii, algoritmul K-means nu va produce clusterele dorite de noi.

În schimb, varianta K-means++ a algoritmului K-means, propusă de David Arthur şi Sergei Vassilvitskii în 2007, încearcă să selecteze pentru poziţiile iniţiale ale celor K centroizi instanţe care sunt [pe cât se poate] mai distanţate unele de altele. În acest fel, pot fi selectate, cu o probabilitate [destul de] mare, instanţe din toate clusterele.

***Formalizare*:**

K-means++ face iniţializarea centroizilor în maniera următoare:

i. **alege primul centroid, μ_1, în manieră uniform aleatorie dintre instanţele de clusterizat, $x_1, \ldots, x_n$. Cu alte cuvinte, alegem mai întâi un indice i în mod uniform aleatoriu din mulţimea $\{1, \ldots, n\}$ şi fixăm $\mu_1 = x_i$.**

ii. **pentru $j = 2, \ldots, K$:**

- **Pentru fiecare instanţă x_i, calculează distanţa D_i până la cel mai apropiat centroid ales / fixat la o iteraţie anterioară:**

$$D_i = \min_{j'=1,\ldots,j-1} ||x_i - \mu_{j'}||.$$

- **Alege centroidul μ_j în mod aleatoriu dintre instanţele $x_1, \ldots, x_n$, cu probabilitate direct proporţională cu $D_1^2, \ldots, D_n^2$. Altfel spus, alegem un indice i în mod aleatoriu din mulţimea $\{1, \ldots, n\}$ cu probabilităţi egale cu $\frac{D_1^2}{\sum_{i'=1}^n D_{i'}^2}$, ..., $\frac{D_n^2}{\sum_{i'=1}^n D_{i'}^2}$, şi fixăm $\mu_j = x_i$.**

iii. **Returnează $\mu \overset{not.}{=} (\mu_1, \ldots, \mu_K)$, setul de asignări ale poziţiilor iniţiale ale centroizilor clusterelor pentru algoritmul lui Lloyd (K-means).**

Vom ilustra acum diferenţa dintre K-means++ şi K-means [la iniţializare], folosind un set de date simplu, format din cinci puncte în planul euclidian bidimensional: $A(-1, 0)$, $B(1, 0)$, $C(0, 1)$, $D(3, 0)$ şi $E(3, 1)$.[930]

a. Presupunem că aplicăm algoritmul K-means cu $K = 2$, făcând iniţializarea centroizilor cu instanţe din mulţimea pe care tocmai am precizat-o. Presupunem că a fost deja ales centroidul $\mu_1 = A$. Dacă selecţia următorului centroid (μ_2) se face în manieră uniform aleatorie — aşa cum procedează îndeobşte algoritmul K-means — din mulţimea $\{B, C, D, E\}$, care este probabilitatea ca μ_2 să fie din submulţimea $\{B, C\}$? Justificaţi.

b. Aplicăm acum pe acelaşi set de date algoritmul K-means++, tot cu $K = 2$. Presupunem, ca şi la punctul a, că a fost selectat centroidul $\mu_1 = A$. Care este acum probabilitatea ca μ_2 să fie selectat din submulţimea $\{B, C\}$? Rezultă oare într-adevăr o îmbunătăţire semnificativă faţă de [iniţializarea făcută de] algoritmul K-means? Justificaţi riguros.

51. **(Algoritmul K-means: varianta „kernelizată“)**

prelucrare de Liviu Ciortuz, după

■ □ • ○ ***CMU, 2015 spring, T. Mitchell, N. Balcan, HW7, pr. 1.2-4***

Atunci când aplicăm algoritmul K-means folosind distanţa euclidiană, lucrăm în mod implicit cu *presupoziţia* că orice două clustere sunt liniar-separabile. Însă, evident, datele nu satisfac întotdeauna această presupoziţie. Un *exemplu* clasic este acela în care avem două clustere constituite din puncte situate în două zone / benzi circulare concentrice din planul euclidian ($\mathbb{R}^2$).

În general, pentru algoritmii de învăţare automată care determină *separatori liniari* este posibil să folosim funcţii-nucleu (engl., *kernel functions*, sau, pe scurt, *kernels*) ca să obţinem versiuni neliniare. Algoritmul K-means nu face excepţie.

Vă reamintim că există două *aspecte principale* ale problemelor rezolvate cu ajutorul algoritmilor „kernelizaţi“:

i. Soluţia unei astfel de probleme se exprimă ca o combinaţie liniară de „reprezentări“ /

[930] Aceste date sunt preluate de la problema 7.

„imagini“ ale instanţelor într-un aşa-numit „spaţiu de trăsături“;[931]

ii. Algoritmul foloseşte doar produsele scalare dintre [vectorii care reprezintă] imagini de instanţe, nu imaginile propriu-zise ($\Phi(x_i)$).[932]

În cele ce urmează, vom arăta — în manieră uşor simplificată[933] la punctele a, b şi c — că aceste două aspecte pot fi satisfăcute în cazul algoritmului K-means.

a. Fie γ_{ij} o *variabilă-indicator*, care este egală cu 1 dacă instanţa x_i este [în prezent] asignată la clusterul j şi 0 în caz contrar. Arătaţi că pentru orice j, centroidul clusterului j (notat cu μ_j, unde $j \in \{1, \ldots, K\}$) care se actualizează în corpul iterativ al algoritmului K-means poate fi calculat folosind o formulă de genul $\mu_j = \sum_{i=1}^{n} \alpha_{ij} x_i$, unde coeficienţii α_{ij} — pe care îi veţi determina — pot fi calculaţi în funcţie de [toate] variabilele-indicator γ.

b. Arătaţi că pentru două puncte oarecare x şi x' din $\mathbb{R}^d$, expresia $\|x - x'\|^2$ poate fi calculată folosind doar (combinaţii liniare de) produse scalare de elemente (în speţă x şi x') din $\mathbb{R}^d$.

c. Ţinând cont de rezultatele de la punctele a şi b, arătaţi că expresiile $\|x_i - \mu_j\|^2$, care sunt utilizate în corpul iterativ al algoritmului K-means, se pot calcula folosind doar (combinaţii liniare de) produse scalare dintre instanţele $x_1, \ldots, x_n$.

d. Considerând dată o funcţie de „mapare“ $\Phi : \mathbb{R}^d \to \mathbb{R}^m$ şi funcţia-nucleu corespunzătoare $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, cu $K(x, y) = \Phi(x) \cdot \Phi(y)$, scrieţi pseudo-codul algoritmului K-means kernel-izat.[934] Concret, veţi porni cu anumite puncte iniţiale ca centroizi. Apoi, în mod iterativ veţi folosi răspunsul de la punctul c ca să găsiţi cel mai apropiat centroid pentru fiecare instanţă şi veţi seta valorile variabilelor-indicator γ_{ij}, după care veţi face uz de răspunsul de la punctul a pentru a recalcula poziţiile centroizilor.

52. **(Comparaţie între clusterizarea ierarhică şi algoritmul K-means)**

• ○ ∗ ***CMU, 2012 fall, E. Xing, A. Singh, HW3, pr. 1.2.a***

Pornind de la o dendrogramă (arbore de clusterizare ierarhică) oarecare, putem defini o partiţionare a datelor în K clustere „tăind“ ramurele arborelui în dreptul unor anumite nivele, sub rădăcina arborelui.

De exemplu, pentru $K = 2$ putem defini două clustere pornind de la cei doi subarbori ai rădăcinii dendrogramei. Pentru $K = 4$, putem folosi subarborii nodurilor descendente din nodul-rădăcină, şi aşa mai departe. (Observaţi că dacă valoarea lui K nu este putere a lui 2, atunci ar trebui să definim un anumit criteriu pentru a stabili care este prioritatea în alegerea subarborilor.)

[931] Formal, date fiind instanţele $x_i \in \mathbb{R}^d$, cu $i = 1, \ldots, n$ şi $d \in \mathbb{N}^*$, se va lucra cu o funcţie de „mapare“ $\Phi : \mathbb{R}^d \to \mathbb{R}^m$, cu $m \gg d$, operatorul relaţional ' $\gg$ ' având semnificaţia „mult mai mare decât“. Imaginea lui x_i prin funcţia Φ este $\Phi(x_i)$. Spaţiul de trăsături este $\mathbb{R}^m$.

[932] Această *restricţie* se justifică prin faptul că se consideră doar funcţii de mapare Φ pentru care dacă definim $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ prin $K(x, y) = \Phi(x) \cdot \Phi(y)$, atunci K este calculabilă în mod eficient în $\mathbb{R}^d$. K se numeşte funcţie-nucleu. *Observaţie*: nu există nicio legătură între această funcţie K şi numărul K care este apare în algoritmul de clusterizare K-means.

[933] Şi anume, lucrând în $\mathbb{R}^d$ în loc de în $\mathbb{R}^m$.

[934] Această variantă a algoritmului K-means caută separatori liniari între clusterele ce se vor forma cu $\Phi(x_1), \ldots, \Phi(x_n) \in \mathbb{R}^m$, însă foloseşte efectiv doar instanţele $x_i \in \mathbb{R}^d$ (prin intermediul funcţiei-nucleu K) nu şi imaginile lor prin funcţia de mapare (Φ) în $\mathbb{R}^m$.

Folosind această procedură pentru a forma o partiţie (a setului de date de intrare) din rezultatul unui clustering de tip ierarhizat, care dintre cele trei măsuri de similaritate prezentate la curs — "single-linkage", "complete-linkage" şi "average-linkage" — va conduce cel mai probabil la formarea unor clustere foarte asemănătoare cu cele obţinute de către algoritmul K-means? (Presupunem că valoarea lui K este o putere a lui 2).

Vă readucem aminte că măsurile de similaritate "single-linkage", "complete-linkage" şi "average-linkage" dintre două mulţimi X şi Y dintr-un spaţiu dotat cu o măsură de distanţă sunt definite în raport cu minimul, maximul şi respectiv media distanţelor dintre perechile de puncte (x, y) cu $x \in X$ şi $y \in Y$.

53. **(Algoritmul K-means: un caz special; comparaţie cu algoritmi de clasificare)**

○ ∗ ***CMU, 2003 fall, T. Mitchell, A. Moore, HW7, pr. 1.b***

Pentru această problemă folosim algoritmul K-means, unde K este egal cu numărul de instanţe / puncte de clusterizat, iar fiecare cluster este format dintr-un singur punct. Considerând dată o anumită măsură de distanţă, vom putea clasifica o instanţă nouă (de test) asociind-o cu clusterul al cărui centroid se află cel mai aproape de instanţa respectivă. Atenţie: centroizii clusterelor nu se modifică în faza de test.

Cu ce metodă de clasificare automată este echivalent modelul descris aici?

54. **(Algoritmul K-means: Adevărat sau Fals?)**

● ○ ***CMU, 2021 fall, M. Gormley, H. Chai, Exam 3 practice problems, pr. 6.1***

a. Dat fiind un anumit set de date şi o valoare fixată pentru K, algoritmul K-means va produce întotdeauna acelaşi rezultat, dacă
– păstrăm aceleaşi iniţializări pentru centroizi;
– pentru acest set de date şi această valoare fixată pentru K nu apar niciodată situaţii de „indecizie" (engl., tie) la asignarea instanţelor la centroizi. (Adică, la pasul de reasignare a instanţelor la centroizi, nicio instanţă x_i nu se află la distanţă minimală (şi egală) faţă de doi centroizi diferiţi, c_j şi $c_{j'}$.)

b. Algoritmul K-means converge pe orice set de date la optimul global al criteriului J (adică, echivalent, la optimul global al coeziunii intra-clustere). Veţi considera valoarea lui K ca fiind oarecare, dar fixată.

c. Este posibil ca la două iteraţii consecutive ale algoritmului K-means, asignările instanţelor la clustere să nu se schimbe deloc.

d. Algoritmul K-means nu este deloc sensibil la prezenţa outlier-elor în setul de date de clusterizat.

e. K de la algoritmul K-Means şi k de la algoritmul k-NN au aceeaşi semnificaţie.

f. Care credeţi că este deosebirea esenţială (sau, cea mai importantă) dintre algoritmii K-Means şi k-NN?

55. **(Algoritmul K-means: implementare)**

□ ***Liviu Ciortuz, 2016***

Implementaţi — de preferinţă în C/C++ — algoritmul K-means. Ca şi la problema 38, veţi considera că instanţele de clusterizat sunt puncte într-un spaţiu euclidian $\mathbb{R}^d$ şi că ele sunt înregistrate într-un fişier în format CSV (Comma Separated Value). Numele acestui fişier şi eventual d vor fi indicate ca argumente în linia de comandă a programului.

Testaţi implementarea pe datele de la problema 40. Ulterior, puteţi extinde implementarea cu funcţii de reprezentare grafică, ca la rezolvarea problemelor 7, 8, 9, 10 şi 41. O altă extensie utilă este calculul valorii criteriului J la fiecare iteraţie, aşa cum s-a arătat la problemele 12 şi 45.

Încă o direcţie posibilă pentru extinderea acestei implementări se referă la varianta kernel-izată a algoritmului K-means, al cărei obiectiv l-a constituit problema 51. Implementaţi şi apoi testaţi această variantă pe setul de date corespunzător conceptului XOR, folosind funcţia-nucleu polinomială de ordinul al doilea

$$K(x, x') = (x \cdot x' + 1)^2 \; \forall x, x' \in \mathbb{R}^2.$$

7.2.3 Algoritmul EM pentru modele de mixturi gaussiene

56. **(EM/GMM, cazul unidimensional: aplicarea manuală a unei iteraţii, pentru o mixtură de tipul următor: μ_1, μ_2 liberi, $\pi_1 = \pi_2$, $\sigma_1 = \sigma_2 = 1/\sqrt{2}$)**

prelucrare de Liviu Ciortuz, 2021, după

□ • · ***CMU, 2015 spring, Roni Rosenfeld,***

"Numerical example of one EM iteration over a Mixture of Gaussians"

Acesta este un mini-exemplu numeric pentru executarea unei singure iteraţii a algoritmului EM aplicat la problema estimării mediilor a două distribuţii gaussiene care formează o mixtură.

Deviaţiile standard ale celor două gaussiene sunt $\sigma_1 = \sigma_2 = 1/\sqrt{2}$.
Probabilităţile a priori corespunzătoare acestor gaussiene sunt $\pi_1 = \pi_2 = 0.5$.
Vom nota cu j indexul ale cărui valori identifică aceste gaussiene, cu i indexul pentru instanţe, şi cu t indexul pentru iteraţiile algoritmului EM.

Vom iniţializa mediile celor două gaussiene cu nişte valori convenabile (pentru uşurinţa calculelor): $\mu_1^{(0)} = 3$, $\mu_2^{(0)} = 6$.

Vom considera că variabila z_{ij} ia valoarea 1 dacă instanţa x_i a fost generată de către gaussiana j, şi 0 în cazul contrar. Variabilele z_{ij} sunt latente / „ascunse" / „neobservabile".

***Remember*:**[935]

[935] Modul în care se deduc regulile de actualizare corespunzătoare celor doi paşi din ciclul repetitiv al algoritmului EM specific acestui caz este prezentat în multe cărţi de specialitate (de exemplu, în cartea *Machine Learning* a profesorului Tom Mitchell, la secţiunea 6.12.1). În prezenta culegere de exerciţii, demonstraţia aceasta constituie obiectul problemei 15.

Verosimilitatea unei instanţe „observabile“ se exprimă astfel:

$$L(x_i|\mu^{(t)}) = \sum_j \pi_j \mathcal{N}(x_i|\mu^{(t)}, \sigma_j^2).$$

Pasul E este:

$$E[z_{ij}|\mu^{(t)}] = \frac{\pi_j \mathcal{N}(x_i|\mu_j^{(t)}, \sigma_j^2)}{\sum_{j'} \pi_{j'} \mathcal{N}(x_i|\mu_{j'}^{(t)}, \sigma_{j'}^2)}.$$

Pasul M este:

$$\mu_j^{(t+1)} = \frac{\sum_i E[z_{ij}|\mu_j^{(t)}] \cdot x_i}{\sum_i E[z_{ij}|\mu^{(t)}]}.$$

a. Completaţi următorul tabel şi apoi calculaţi valorile celor două medii ($\mu \overset{not.}{=} (\mu_1, \mu_2)$) care maximizează verosimilitatea datelor furnizate, adică:

$$\mu_{MLE} = \underset{\mu_1, \mu_2}{\operatorname{argmax}} \prod_i L(x_i|\mu^{(t)}).$$

i	1	2	3
x_i	2	4	7
$\mathcal{N}(x_i\|\mu_1^{(0)}, \sigma_1^2)$			
$\mathcal{N}(x_i\|\mu_2^{(0)}, \sigma_2^2)$			
$L(x_i\|\mu_1^{(0)}, \mu_2^{(0)})$			
$E[z_{i1}\|x_i, \mu_1^{(0)}, \mu_2^{(0)})$			
$E[z_{i2}\|x_i, \mu_1^{(0)}, \mu_2^{(0)})$			

$$\mu_1^{(1)} = \ldots \approx \ldots$$

$$\mu_2^{(1)} = \ldots \approx \ldots$$

b. Care este valoarea verosimilităţii $L(x_1, x_2, x_3|\mu_1^{(1)}, \mu_2^{(1)})$? Comparaţi-o cu verosimilitatea $L(x_1, x_2, x_3|\mu_1^{(0)}, \mu_2^{(0)})$. Rezultatul pe care l-aţi obţinut este în acord cu proprietatea teoretică (referitoare la convergenţa algoritmului EM) care a fost prezentată curs? (Vedeţi problema 2 de la capitolul *Schema algortmică EM*.)

57. **(EM/GMM, cazul unidimensional, varianta generală: executarea manuală a pasului M)**

□ • ***N. de Freitas, preluat de Kevin Murphy, în ex. 11.7 din "Machine Learning, A Probabilistic Perspective", MIT Press, 2012***

În acest exerciţiu urmărim să clusterizăm setul de date $x = \{1, 10, 20\}$ din $\mathbb{R}$, folosind algoritmul EM pentru o mixtură de două distribuţii gaussiene (EM/GMM). Presupunem că rezultatul execuţiei pasului E este cel dat în următoarea matrice:

$$\begin{bmatrix} 1 & 0 \\ 0.4 & 0.6 \\ 0 & 1 \end{bmatrix},$$

în care elementul generic $p_{i,j}$ reprezintă probabilitatea ca „observaţia" x_i să fi fost generată de către gaussiana j.

Vi se cere să executaţi pasul M al algoritmului EM. Veţi scrie mai întâi formulele pentru actualizarea parametrilor mixturii formate din cele două gaussiene (vedeţi rezolvarea problemei 18), fără să trebuiască să faceţi demonstraţia lor; trebuie doar să aplicaţi aceste formule pe setul de date indicat mai sus. Elaboraţi toate detaliile necesare.

a. Scrieţi expresia aşa-numitei *funcţii auxiliare* (Q) care se optimizează la acest pas. Vă readucem aminte că funcţia auxiliară este media funcţiei de log-verosimilitate a *datelor complete* (adică, cele *observabile* (x_i) şi cele *neobservabile* (z_{ij})).

b. Care vor fi noile valori ale probabilităţilor de selecţie π_1 şi π_2 după execuţia pasului M? Dar ale mediilor μ_1 şi μ_2? Dar ale varianţelor σ_1^2 şi σ_2^2?

58. **(Algoritmul EM/GMM, cazul unidimensional, varianta generală: aplicare pe un set de date din $\mathbb{R}$)**

• ○ ***MIT, 2016 spring, T. Jaakkola, R. Barzilay, HW5, pr. 5***

Fie mixtura de două distribuţii gaussiene $P(x|\theta) = p_1\mathcal{N}(x|\mu_1, \sigma_1^2) + p_2\mathcal{N}(x|\mu_2, \sigma_2^2)$, care are parametrii $\theta = (p_1, p_2, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2)$, adică mediile, varianţele şi ponderile (probabilităţile de selecţie) pentru fiecare dintre cele două gaussiene. Valorile iniţiale $\theta^{(0)}$ asignate parametrilor mixturii sunt: $p_1 = p_2 = 0.5$, $\mu_1 = 6$, $\mu_2 = 7$, $\sigma_1 = 1$ şi $\sigma_2 = 2$.

Avem următoarele instanţe ale lui x: $x_1 = 0$, $x_2 = 1$, $x_3 = 5$, $x_4 = 6$ şi $x_5 = 7$.

a. Scrieţi expresia funcţiei de log-verosimilitate, $\ell(x|\theta)$, pe care urmărim să o maximizăm. De ce este convenabil să folosim algoritmul EM [comparativ, de exemplu, cu metoda gradientului]? Furnizează oare *întotdeauna* acest algoritm soluţia optimă?

b. Referindu-ne la prima execuţie a pasului E, care sunt instanţele x_i care vor fi cel mai probabil (dar nu în totalitate) asignate celei de-a doua gaussiene? Altfel spus, care sunt punctele x_i pentru care $P(z_i = 2|x_i, \theta^{(0)}) > P(z_i = 1|x_i, \theta^{(0)})$?

c. Referindu-ne acum la prima execuţie a pasului M, în ce direcţie se vor deplasa mediile celor două gaussiene? Justificaţi riguros.

d. Tot în legătură cu prima execuţie a pasului M, varianţele σ_1^2 şi σ_2^2 vor creşte sau vor descreşte? Justificaţi riguros.

e. Folosind o implementare a algoritmului EM/GMM specifică pentru cazul de faţă, faceţi o rulare pe datele acestei probleme. Cât sunt valorile mediilor celor două gaussiene atunci când algoritmul ajunge la convergenţă? Care dintre varianţele celor două gaussiene obţinute la final este mai mare?

59. **(Algoritmul EM/GMM, cazul unidimensional, varianta generală: întrebări calitative)**

□ • ○ ***MIT, 2006 fall, Tommi Jaakkola, final exam, pr. 3***

Estimăm parametrii unei mixturi de două distribuţii gaussiene unidimensionale, aplicând versiunea generală pentru algoritmul EM/GMM (vedeţi pr. 18) pe setul de date $S = \{0, 1, 5, 6, 7\}$. Funcţia de densitate de probabilitate (engl., probability density function, p.d.f.) pentru această mixtură de distribuţii este definită (pentru $\forall x \in \mathbb{R}$) astfel:

$$P(x|z, \theta) = P(z = 1) \cdot \mathcal{N}(x|\mu_1, \sigma_1^2) + P(z = 2) \cdot \mathcal{N}(x|\mu_2, \sigma_2^2) \text{ cu } \theta \overset{not.}{=} (\mu_1, \mu_2, \sigma_1, \sigma_2).$$

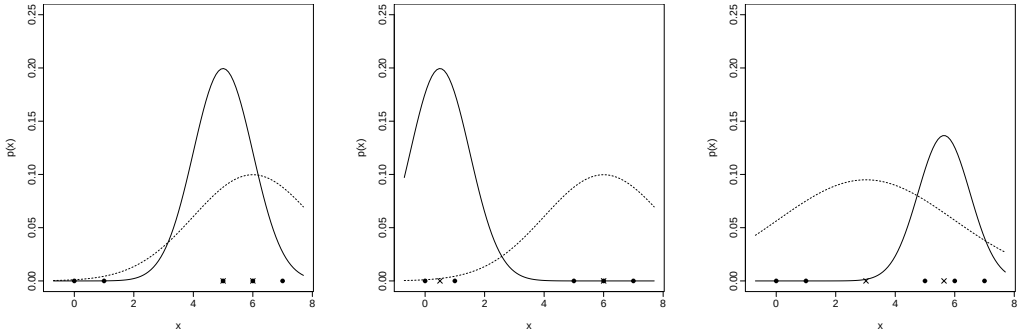
Orice student de la acest curs ar trebui să ştie să rezolve o astfel de problemă de estimare de parametri. Ei bine, un student (dintre cei mai neatenţi / zăpăciţi), a încurcat ordinea figurilor care ilustrează grafic „actualizările“ (engl., updates) făcute de către algoritmul EM/GMM pe dataset-ul S. Mai mult, este posibil ca el să fi strecurat din greşeală o figură care nu aparţine acestui „experiment“.

***Comentariu*:**

În cele ce urmează, vă vom cere să indicaţi care dintre figurile de mai jos reprezintă iteraţii

/ „actualizări“ *successive* executate de algoritmul EM/GMM pe dataset-ul S. Veţi explica de ce anume ordinea pe care aţi indicat-o dumneavoastră are sens din punctul de vedere al modului cum lucrează algoritmul EM/GMM.

În toate aceste figuri se reprezintă grafic cu o linie continuă produsul $P(z=1)\cdot\mathcal{N}(x|\mu_1,\sigma_1^2)$ văzut ca funcţie de x, iar cu o linie punctată produsul $P(z=2)\cdot\mathcal{N}(x|\mu_2,\sigma_2^2)$.



a.
i. **Clasificaţi cu ′Adevărat′ sau ′Fals′ afirmaţia următoare:**
În acest model de mixtură, putem identifica asignarea a posteriori cea mai probabilă (engl., *the most likely posterior assignment*), adică acel j pentru care se maximizează probabilitatea $P(j|x)$, comparând valorile produselor $P(z=1)\cdot\mathcal{N}(x|\mu_1,\sigma_1^2)$ şi $P(z=2)\cdot\mathcal{N}(x|\mu_2,\sigma_2^2)$.

ii. **Justificaţi apoi în mod riguros răspunsul dumneavoastră.**

b. Asignaţi două dintre figurile prezentate mai sus la iteraţii succesive (şi corecte) ale algoritmului EM/GMM. Justificaţi succint alegerea pe care aţi făcut-o.

c. Explicaţi [în mod succint] cum anume modelul de mixtură pe care l-aţi ales pentru ′iteraţia 2′ decurge din modelul de mixtură pe care l-aţi ales pentru ′iteraţia 1′.

60. **(EM/GMM, cazul unidimensional: efectuarea manuală a unei iteraţii, pentru o mixtură de tipul $\mu_1=\mu_2$ (liber), $\pi_1=\pi_2,\sigma_1=1,\sigma_2=2$)**

• · ***U. Toronto, Radford Neal, “Statistical Methods for Machine Learning and Data Mining” course, 2014 spring, Practice problems, set 2, pr. 4***

Considerăm o mixtură de două distribuţii gaussiene unidimensionale, pentru care funcţia de densitate de probabilitate (p.d.f.) are pentru o „observaţie“ oarecare x expresia

$$\frac{1}{2}\mathcal{N}(x|\mu,1)+\frac{1}{2}\mathcal{N}(x|\mu,2^2).$$

În această formulă, $\mathcal{N}(x|\mu,\sigma^2)$ desemnează, pentru x, valoarea densităţii distribuţiei normale unidimensionale de medie μ şi varianţă σ^2. Remarcaţi faptul că în acest model de mixtură probabilităţile de mixare / selecţie sunt egale,

apoi că mediile celor două componente sunt egale şi, de asemenea, că deviaţiile standard ale celor două componente au valorile fixate 1 şi respectiv 2. Modelul acesta de mixtură are un singur parametru, μ.

Presupunem că dorim să estimăm valoarea parametrului μ prin metoda maximizării verosimilităţii, folosind algoritmul EM. Răspundeţi la următoarele întrebări privitoare la modul în care operează paşii E şi M ai acestui algoritm, atunci când considerăm cele trei instanţe / „observaţii" de mai jos:

$$4.0,\ 4.6,\ 2.0.$$

Vă punem la dispoziţie un tabel cu anumite valori ale funcţiei de densitate pentru distribuţia normală standard, de care veţi avea probabil nevoie în rezolvarea acestui exerciţiu:

x	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$\mathcal{N}(x\|0,1)$	0.40	0.40	0.39	0.38	0.37	0.35	0.33	0.31	0.29	0.27	0.24
x		1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2.0
$\mathcal{N}(x\|0,1)$		0.22	0.19	0.17	0.15	0.13	0.11	0.09	0.08	0.07	0.05

a. Găsiţi valoarea probabilităţilor condiţionate calculate la pasul E, presupunând că la execuţia precedentului pas M estimarea obţinută pentru parametrul modelului a fost $\mu = 4$ (iar $\sigma_1 = 1$ şi $\sigma_2 = 2$, totdeauna). Remarcaţi faptul că, întrucât probabilităţile condiţionate pentru cele două componente ale mixturii trebuie să se sumeze la valoarea 1, este suficient să se calculeze $p_{i1} \stackrel{not.}{=} P(componenta_1|x_i)$ pentru $i = 1, 2, 3$.

***Atenţie*! Nu este suficient să faceţi schimbarea de variabilă necesară pentru „standardizare", şi anume $x \leftarrow \frac{x-\mu}{\sigma}$. Trebuie să folosiţi formula de legătură între o distribuţie gaussiană unidimensională oarecare şi cea standard:**

$$\mathcal{N}(x|\mu, \sigma^2) = \frac{1}{\sigma}\mathcal{N}(\frac{x-\mu}{\sigma}|0, 1).$$

În prealabil veţi demonstra această formulă.

b. Folosind probabilităţile pe care le-aţi calculat la punctul a, găsiţi estimarea pentru parametrul μ care va fi obţinută la următoarea execuţie a pasului M. Vă reamintim că pasul M maximizează media (engl., expected value) log-probabilităţii comune pentru instanţele x_1, x_2 şi x_3 şi variabilele-indicator [pentru componentele mixturii], care sunt latente. Media aceasta se calculează în raport cu distribuţia de probabilitate condiţionată a variabilelor-indicator [pentru componentele mixturii], care a fost determinată / calculată la pasul E al aceleiaşi iteraţii.

***Sugestie*: Nu este necesar ca în prealabil să faceţi elaborarea formulelor algoritmului EM pentru acest caz specific de mixtură de distribuţii gaussiene unidimensionale. Este suficient să lucraţi cu funcţia „auxiliară" Q corespunzătoare iteraţiei respective.**

c. Calculaţi log-verosimilitatea datelor „observabile" atunci când $\mu = 4$. Cum este valoarea acestei expresii (mai mică, egală, sau mai mare?) faţă de valoarea log-verosimilităţii aceloraşi date „observabile" atunci când μ ia valoarea calculată la finalul pasului M descris la punctul b?

61. **(EM/GMM, cazul unidimensional, cu parametrii μ_1 şi π_1 liberi, $\mu_2 = 3/2\mu_1$, $\pi_2 = 1 - \pi_1$, $\sigma_1 = \sigma_2 = 1$; elaborare + implementare + rulare)**

• ○ ***U. Toronto, Radford Neal, "Statistical Computation" course, 2003 spring, HW3, pr. 2***

Să presupunem că un ornitolog[936] face un studiu referitor la când anume păsările femele se întorc la cuiburile lor după ce fac „expediţii" / zboruri de strângere de hrană pentru puişorii lor. Pentru toate păsările femele studiate, ornitologul cunoaşte momentele de timp când puii lor ies din ouă, iar toate măsurătorile de timp făcute de către ornitolog sunt relative la aceste momente.

Se crede că, după ce au scos puii din ouă, păsările se întorc din „expediţiile" / zborurile de strângere a hranei la intervale [aproximativ] regulate de timp, θ, 2θ, 3θ, etc., însă cu o anumită variaţie aleatoare, despre care vom presupune că urmează o distribuţie normală / gaussiană de medie 0 şi varianţă 1. Se presupune că păsările acţionează în mod independent unele de altele.

Pentru a evita să deranjeze păsările la momente critice, ornitologul nu a „observat" / notat timpul de întoarcere din prima „expediţie", pentru nicio pasăre. În schimb, pentru fiecare pasăre, ornitologul a „observat" / notat timpul de întoarcere fie de la a doua fie de la a treia „expediţie". (Păsările sunt foarte dificil de zărit, aşa că anumite întoarceri nu au fost observate.) Momentele de timp când au avut loc aceste întoarceri constituie datele noastre, $X_1, \ldots, X_n$. Din nefericire, ornitologul nu cunoaşte, în legătură cu fiecare instanţă X_i, dacă ea reprezintă timpul de întoarcere corespunzător celei de-a doua „expediţii" ori a celei de-a treia. Probabilitatea de a observa o a doua întoarcere versus o a treia nu este cunoscută.

Datele pot fi modelate cu ajutorul unei mixturi, ale cărei componente sunt distribuţiile $\mathcal{N}(2\theta, 1)$ şi $\mathcal{N}(3\theta, 1)$, cu probabilităţile de mixare / selecţie p şi respectiv $1-p$. Cu alte cuvinte, expresia funcţiei de densitate de probabilitate pentru o „observaţie" x este următoarea:

$$f(x) = p \cdot \frac{1}{\sqrt{2\pi}} \cdot \exp(-(x-2\theta)^2/2) + (1-p) \cdot \frac{1}{\sqrt{2\pi}} \cdot \exp(-(x-3\theta)^2/2).$$

Considerând datele $X_1, \ldots, X_n$, va trebui să găsiţi estimările de verosimilitate maximă pentru θ şi p, folosind un algoritm EM pe care îl veţi deduce dumneavoastră. Variabilele neobservabile ar trebui să fie variabilele-indicator relativ la care întoarcere — a doua ori a treia — se referă fiecare „observaţie" / instanţă X_i.

a. Deduceţi formulele de actualizare care se folosesc la paşii E şi M ai algoritmului EM corespunzător acestei probleme.

b. Elaboraţi o implementare a algoritmului EM pe care l-aţi dezvoltat şi testaţi-l pe datele pe care le furnizăm pe site-ul web asociat acestei cărţi,[937] precum şi pe alte date pe care le consideraţi potrivite. Programul dumneavoastră ar trebui să execute un număr de iteraţii pe care-l veţi specifica (nu

[936] Ornitologia este o ramură a zoologiei, care studiază păsările.

[937] https://profs.info.uaic.ro/∼ciortuz/ML.ex-book/implementation-exercises/UToronto.2003s.HW3.pr2.EM-for-a-particular-uni-varGMM.data+R-code+sol/data2.

trebuie să detectaţi în mod automat când anume se realizează convergenţa). De asemenea, ar trebui ca programul să aibă o opţiune de depanare ("debug") la selectarea căreia să se imprime, la fiecare iteraţie, valorile parametrilor şi ale log-verosimilităţii. (Aduceţi-vă aminte că log-verosimilitatea datelor observabile nu trebuie să descrească de la o iteraţie la alta.) Programul trebuie să primească la intrare datele „observabile", precum şi valorile iniţiale ale parametrilor.

Vă recomandăm să evaluaţi cât de bine s-a comportat algoritmul dumneavoastră. În particular, precizaţi dacă [au apărut indicii că] există maxime locale multiple pe suprafaţa funcţiei de verosimilitate.

62. **(EM/GMM, cazul unidimensional: estimarea parametrilor μ şi π pentru o mixtură de distribuţii gaussiene unidimensionale, presupunând $\sigma_1^2 = \sigma_2^2$)**

prelucrare de Liviu Ciortuz, după
* ***CMU, 2010 fall, Aarti Singh, HW4, pr. 2.1-2***

În această problemă, vom rezolva problema mixturii a două distribuţii gaussiene pentru o variantă ceva mai generală decât cea care este prezentată în cartea *Machine Learning* a lui Tom Mitchell, pag. 193 (şi preluată de noi în problema 15), unde se consideră că $\pi_1 = \pi_2 = 1/2$ şi $\sigma_1^2 = \sigma_2^2$. Aici parametrii π_1 şi π_2 sunt liberi, deci vom estima atât mediile μ cât şi probabilităţile de selecţie π, pe când acolo se estimau doar mediile μ. Vom vedea şi de data aceasta — ca şi în problema 15 / cartea lui T. Mitchell – că necunoaşterea valorilor varianţelor $\sigma_1^2 = \sigma_2^2$ nu va impieta asupra calculelor.

Vă reamintim faptul că un model probabilist de tip mixtură este de fapt o funcţie densitate de probabilitate care a fost obţinută prin extragerea fiecărei instanţe X conform uneia din două distribuţii posibile, $P(X|Z = 1)$ sau $P(X|Z = 2)$. Z este o variabilă aleatoare neobservabilă / „ascunsă“ care este definită peste două clase. Pur şi simplu, Z indică pentru fiecare instanţă X care anume dintre cele două distribuţii considerate a generat-o.

Vom presupune că $P(Z)$ este o distribuţie de tip Bernoulli şi că fiecare funcţie (densitate) de probabilitate condiţionată $P(X|Z = z)$ este o distribuţie gaussiană 1-dimensională cu varianţa σ^2. Aşadar, funcţia (densitate) de probabilitate marginală este:

$$P(X = x) = \sum_{z \in \{1,2\}} P(X = x|Z = z) \cdot P(Z = z),$$

iar ca funcţie de parametrii μ şi π ea se rescrie astfel:

$$P(X = x \mid \mu, \pi) = \sum_{z \in \{1,2\}} \pi_z \cdot \frac{1}{\sqrt{2\pi}\sigma} \cdot e^{-\frac{(x-\mu_z)^2}{2\sigma^2}}.$$

Parametrii acestui model sunt $\mu \overset{not.}{=} (\mu_1, \mu_2)$ şi $\pi \overset{not.}{=} (\pi_1, \pi_2)$, unde μ_z este media gaussienei care corespunde clasei z, iar $\pi_z = P(Z = z)$ este probabilitatea de a extrage / genera o instanţă din clasa z. (Reţineţi faptul că $\pi_1 + \pi_2 = 1$.) Dorim să folosim algoritmul EM pentru a estima aceşti parametri din setul de date independent generate $\{x_i\}_{i=1}^m$, unde $x_i \in \mathbb{R}$.

a. Vom nota cu p_{iz} probabilitatea ca instanţa i să fie extrasă din clasa z (adică, $p_{iz} = P(Z = z|X = x_i, \mu, \pi)$). În cadrul iteraţiei t a algoritmului EM, la pasul E se calculează probabilităţile p_{iz} pentru toate perechile de valori posibile ale indicilor i, z, folosind valorile parametrilor care au fost calculate la iteraţia precedentă, şi anume $\mu^{(t-1)} \overset{not.}{=} (\mu_1^{(t-1)}, \mu_2^{(t-1)})$ şi $\pi^{(t-1)} \overset{not.}{=} (\pi_1^{(t-1)}, \pi_2^{(t-1)})$. Scrieţi / deduceţi expresia de calcul pentru probabilităţile p_{iz} în funcţie de aceşti parametri.

b. La pasul M de la iteraţia t a algoritmului EM, se tratează cantităţile p_{iz} ca fiind nişte count-uri fracţionale pentru valorile neobservabile z şi se actualizează / estimează valorile parametrilor μ şi π ca şi când punctul (x_i, z) ar fi fost generat / observat de p_{iz} ori. Deduceţi regula de actualizare a parametrilor $\mu^{(t)}$ şi $\pi^{(t)}$ în funcţie de probabilităţile p_{iz}.

c. Scrieţi pseudo-codul algoritmului EM care rezolvă acest tip de mixturi de distribuţii gaussiene.

63. **(EM/GMM, cazul unidimensional: efectuarea manuală a pasului M (precum şi a următorului pas E), pentru o mixtură de un tip particular, parametrii liberi fiind π_1, μ_1, μ_2)**

prelucrare de Liviu Ciortuz, 2021, după
□ • · ***U. Toronto, Radford Neal, "Statistical Methods for Machine Learning and Data Mining" course, 2014 spring, Practice problems, set 2, pr. 3***

Să zicem că dorim să antrenăm (engl., to fit) un model de tip mixtură de distribuţii gaussiene unidimensionale, folosind $K = 2$ componente. Avem $n = 5$ instanţe, ale căror valori (notate cu x_i, cu $i = 1, \ldots, 5$) sunt:

$$5,\ 15,\ 25,\ 30,\ 40.$$

Folosim algoritmul EM pentru a găsi estimările de verosimilitate maximă (engl., maximum likeihood estimates) pentru parametrii modelului, şi anume: probabilităţile de selecţie (sau, mixare) ale celor două componente (vom nota aceste probabilităţi cu π_1 şi π_2) şi mediile acestor componente (notate cu μ_1 şi respectiv μ_2). Deviaţiile standard ale celor două componente sunt considerate fixate şi au valoarea 10.

Presupunem că în timpul execuţiei algoritmului EM, la o anumită iteraţie t, la pasul E au fost determinate mediile variabilelor neobservabile Z_{ij} — aceste medii se mai numesc *responsabilităţi*; engl., responsibilities — corespunzătoare celor două componente, pentru fiecare dintre cele cinci instanţe, după cum urmează:

$p_{i1}^{(t)}$	0.2	0.2	0.8	0.9	0.9
$p_{i2}^{(t)}$	0.8	0.8	0.2	0.1	0.1

a. Deduceţi expresia funcţiei „auxiliare" $Q(\pi_1, \mu_1, \mu_2 | \pi_1^{(t)}, \mu_1^{(t)}, \mu_2^{(t)})$ care urmează să fie optimizată la execuţia pasului M al respectivei iteraţii (t) a algoritmului EM.

b. Ce valori vor fi atribuite parametrilor π_1, π_2, μ_1 şi μ_2 la execuţia respectivului pas M?

***Indicaţie*: Veţi optimiza *direct* funcţia „auxiliară" Q pe care aţi determinat-o la punctul a. [În cazul în care nu aţi rezolvat punctul a sau nu ştiţi cum se optimizează în mod *direct* respectiva funcţie Q — dar doar în acest caz! —, puteţi folosiţi *formule* de genul celor pe care le-am predat la curs. Rezultatele trebuie să fie aceleaşi, în ambele variante!]**

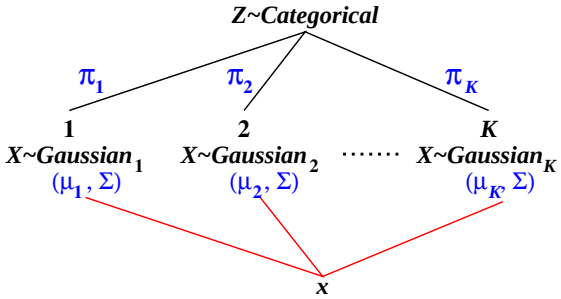
c. i. Deduceţi expresiile necesare pentru a calcula $p_{i1}^{(t+1)}$ şi $p_{i2}^{(t+1)}$, mediile variabilelor neobservabile Z_{ij} corespunzătoare următorului pas E al algoritmului EM. ii. Calculaţi aceste probabilităţi pentru $i = 1, \ldots, 5$.

64. **(EM/GMM, cazul multidimensional: deducerea regulii de actualizare pentru Σ, atunci când $\Sigma_j = \Sigma$ pt. $j = 1, \ldots, K$)**

• ○ ***CMU, 2010 fall, Aarti Singh, HW4, pr. 1.1-2***

Un model de mixturi de distribuţii gaussiene (engl., Gaussian mixture model, GMM) este o familie de distribuţii ale căror densităţi de probabilitate (engl., probability density functions, p.d.f.) au forma următoare:

$$gmm(\mathbf{x}) \;=\; \sum_{j=1}^{K} \pi_j \, \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_j, \Sigma_j),$$



unde prin $\mathcal{N}(\cdot|\boldsymbol{\mu}, \Sigma)$ am notat funcţia de *densitate* a distribuţiei gaussiene de medie $\boldsymbol{\mu}$ şi matrice de covarianţă Σ,[938] iar $\pi_1, \ldots, \pi_K$ sunt ponderile mixturii, care satisfac restricţiile $\sum_{j=1}^{K} \pi_j = 1$ şi $\pi_j \geq 0$ pentru $j \in \{1, \ldots, K\}$.

Pseudo-codul algoritmului EM (Expectation-Maximization) pentru învăţarea [valorilor parametrilor] unui GMM pornind de la un set de instanţe $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ poate fi schiţat astfel:[939]

- Iniţializează $\boldsymbol{\mu}_j$, Σ_j şi π_j, pentru $j \in \{1, \ldots, K\}$.
- Repetă următorii doi paşi până la „convergenţă“:

  Pasul E:

$$E[z_{ij}] = P\Big(\mathbf{x}_i \in \text{cluster } j \;\Big|\; \mathbf{x}_i,\; \{(\pi_{j'}, \boldsymbol{\mu}_{j'}, \Sigma_{j'})\}_{j'=1}^{K}\Big),$$

  Pasul M:[940]

$$\{(\pi_j, \boldsymbol{\mu}_j, \Sigma_j)\}_{j=1}^{K} \;\leftarrow\; \arg\max \sum_{i=1}^{n} \sum_{j=1}^{K} E[z_{ij}] \Big(\ln \mathcal{N}(\mathbf{x}_i|\boldsymbol{\mu}_j, \Sigma_j) + \ln \pi_j \Big)$$

Considerăm o variantă simplificată de GMM, în care toate componentele mixturii au o aceeaşi matrice de covarianţă, adică $\Sigma_j = \Sigma$ pentru $j = 1, \ldots, K$. Deduceţi regulile de actualizare pentru matricea Σ_j, de la pasul M. (Răspunsul dumneavoastră poate folosi valoarea lui $\boldsymbol{\mu}_j$ deja actualizată la prezentul pas M.)

***Indicaţie*: Vă recomandăm să faceţi apel la următoarele formule de derivare a matricelor în raport cu vectori / matrice de variabile:**[941]

[938] $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \dfrac{1}{\sqrt{(2\pi)^d}\,\sqrt{|\Sigma|}} \exp\left(-\dfrac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{x}-\boldsymbol{\mu})\right)$, unde d este dimensiunea spaţiului ($\mathbb{R}^d$) în care se lucrează.

[939] Această procedură este uşor simplificată în raport cu formularea algoritmului EM/GMM folosită (în mod implicit) la problema 24.

[940] Pentru justificarea formulei date aici la Pasul M, a se vedea problema 24 de la acest capitol: termenul $-\sum_i \sum_j E[z_{ij}] \ln w_{ij} = -\sum_i \sum_j w_{ij} \ln w_{ij}$ din formula (387) de la pag. 922 a fost lăsat deoparte aici, întrucât el reprezintă o entropie (vedeţi problema 1.c de la capitolul *Schema algoritmică EM* din prezenta culegere) şi] nu depinde de π_j, μ_j şi Σ_j.

[941] A se vedea *Matrix indentities*, Sam Roweis, New York University, June 1999.

$$|A|^{-1} = |A^{-1}| \tag{1e}$$

$$\frac{\partial \ln |\mathbf{X}|}{\partial \mathbf{X}} = (\mathbf{X}^{-1})^\top = (\mathbf{X}^\top)^{-1} \tag{4b}$$

$$\frac{\partial a^\top \mathbf{X} b}{\partial \mathbf{X}} = ab^\top \tag{5c}$$

unde operatorul $\top$ desemnează transpunerea matricelor / vectorilor.[942]

65. **(EM/GMM, cazul multidimensional; o proprietate importantă: atunci când $\Sigma_j = \sigma^2 I$, $\forall j$ şi $\sigma^2 \to 0$, algoritmul EM/GMM tinde să se comporte asemenea algoritmului K-means)**

■ □ • ○ ***CMU, 2008 fall, Eric Xing, HW4, pr. 2.2***

Vă readucem aminte că, date fiind instanţele $\mathbf{x}_1, \ldots, \mathbf{x}_n$, algoritmul K-means le va grupa în K clustere minimizând criteriul de „distorsiune“ (sau: coeziune intra-clustere) $J_K = \sum_{i=1}^n \sum_{j=1}^K \gamma_{ij} \|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2$, unde $\boldsymbol{\mu}_j$ este centroidul clusterului j, iar $\gamma_{ij} = 1$ dacă $\mathbf{x}_i$ aparţine clusterului j şi $\gamma_{ij} = 0$ în caz contrar. În acest exerciţiu vom folosi pentru algoritmul K-means următoarea procedură:[943]

- Iniţializează cu valori arbitrare ($\boldsymbol{\mu}_j$) centroizii clusterelor ($j = 1, \ldots, K$);
- Iterează până se ajunge la „convergenţă“:

  – pentru fiecare punct $\mathbf{x}_i$ ($i = 1, \ldots, n$), revizuieşte asignarea sa la clustere: $\gamma_{ij} = 1$ dacă $j = \arg\min_{j'} \|\mathbf{x}_i - \boldsymbol{\mu}_{j'}\|^2$ şi $\gamma_{ij} = 0$ în caz contrar;

  – actualizează poziţiile centroizilor clusterelor: $\boldsymbol{\mu}_j = \frac{\sum_{i=1}^n \gamma_{ij} \mathbf{x}_i}{\sum_{i=1}^n \gamma_{ij}}$ pentru $j = 1, \ldots, K$.

Considerăm acum o variantă de GMM în care toate componentele mixturii au matricea de covarianţă $\sigma^2 I$, unde $\sigma^2 > 0$, iar I este matricea identitate. Presupunem că ni se dă un set de instanţe $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ şi că aplicăm algoritmul EM pentru a estima ponderile (π_j) şi mediile ($\boldsymbol{\mu}_j$) acestei mixturi şi pentru a obţine pentru fiecare instanţă $\mathbf{x}_i$ probabilităţile de apartenenţă la fiecare cluster j.[944]

Vom presupune că următoarele proprietăţi sunt adevărate / satisfăcute pe parcursul întregii execuţii a algoritmului EM:

- **Există $\varepsilon > 0$ astfel încât $\pi_j \geq \varepsilon, \forall j \in \{1, \ldots, K\}$ — adică ponderile mixturii sunt menţinute la (o anumită) distanţă faţă de 0 — pe parcursul tuturor iteraţiilor.**
- **Pe parcursul tuturor iteraţiilor,**

$$\|\mathbf{x}_i - \boldsymbol{\mu}_j\|^2 \neq \|\mathbf{x}_i - \boldsymbol{\mu}_{j'}\|^2 \quad \forall\, i \in \{1, \ldots, n\},\ j \neq j'.$$

[942] Ca şi la alte probleme de acest gen, vom considera că vectorii $\mathbf{x}$ şi $\boldsymbol{\mu}_j$ din $\mathbb{R}^d$ sunt vectori-coloană.

[943] Această procedură este uşor simplificată în raport cu formularea de la problema 45.

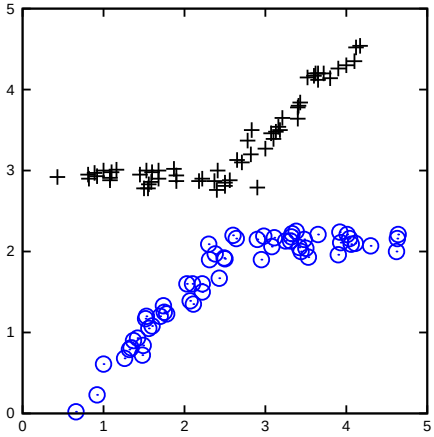
[944] Aşadar, σ^2 este considerat fixat, iar π_j şi μ_j sunt liberi. Pentru derivarea completă a algoritmului EM/GMM corespunzător situaţiei în care $\Sigma_j = \sigma^2 I$, $\forall j$, dar σ^2, $\boldsymbol{\mu}_j$ şi π_j sunt toţi parametri variabili / liberi, a se vedea problema 20.

Arătaţi că atunci când $\sigma^2 \to 0$, expresiile calculate la pasul E al algoritmului EM tind la valorile calculate de regula de actualizare (engl., update rule) pentru variabilele-indicator γ_{ij} din cadrul algoritmului K-means (adică $p(z_{ij} = 1|x_i) \to \gamma_{ij}$), deci asignarea "soft" devine "hard".

66. **(Algoritmul EM/GMM, cazul bidimensional; chestiuni de ordin calitativ, discutate pe date din $\mathbb{R}^2$)**

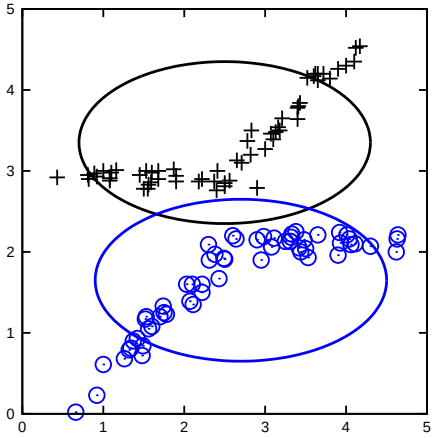
∗ ***CMU, 2006 fall, E. Xing, T. Mitchell, final exam, pr. 4***

Profesorul de la cursul de învăţare automată a rugat pe trei dintre studenţii săi (Alin, Bogdan şi Cezar) să folosească GMM (Gaussian Mixture Models) pe setul de date ilustrat în figura alăturată. Exemplele etichetate cu + sunt pozitive, iar cele etichetate cu ∘ sunt negative.[945]

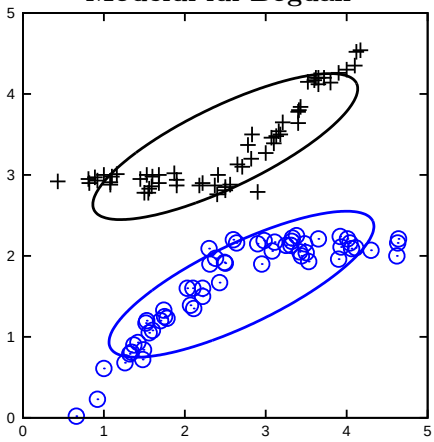


a. Alin şi Bogdan au decis să folosească o distribuţie gaussiană pentru exemplele pozitive şi una pentru exemplele negative, obţinând modelele următoare:

Modelul lui Alin



Modelul lui Bogdan



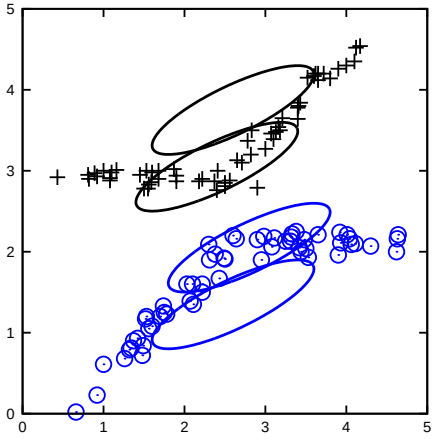
Dintre acestea, ce model preferi pentru acest set de date? Ce anume determină diferenţa dintre ele?

b. Cezar a decis să utilizeze [LC: o mixtură de] două distribuţii gaussiene pentru exemplele pozitive şi [LC: o altă mixtură de] două distribuţii gaussiene pentru cele negative. El a utilizat algoritmul EM pentru estimarea parametrilor, şi de asemenea, a încercat diferite iniţializări. În coloana din stânga sunt reprezentate 3 iniţializări diferite, iar în coloana din dreapta sunt reprezentate

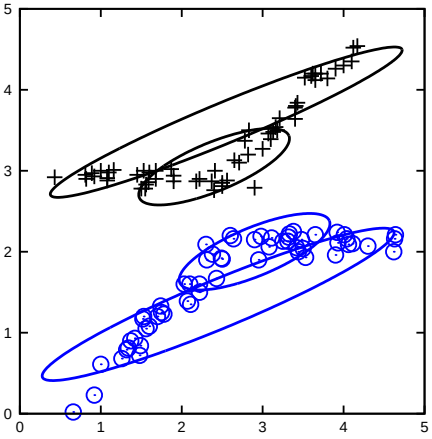
[945] Coordonatele acestor instanţe vă sunt puse la dispoziţie într-un fişier depus pe site-ul acestei cărţi: http://profs.info.uaic.ro/∼ciortuz/ML.ex-book/res/CMU.2006f.EX+TM.final.pr4.em.m.

cele 3 modele obţinute după prima iteraţie. Precizaţi corespondenţa dintre modelele din cele două coloane.

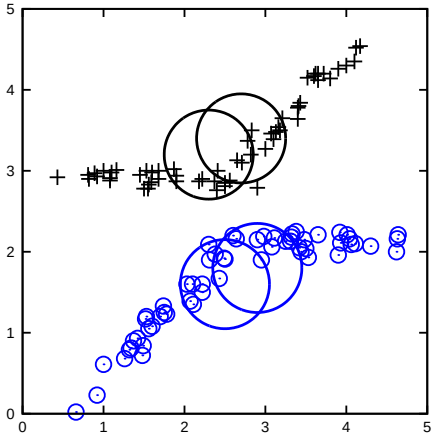
I_1 **- Iniţializare**



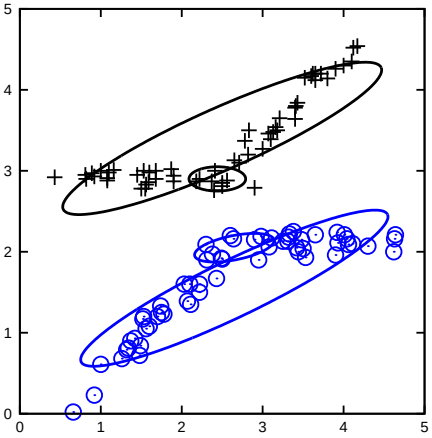
M_1 **- Model după prima iteraţie**



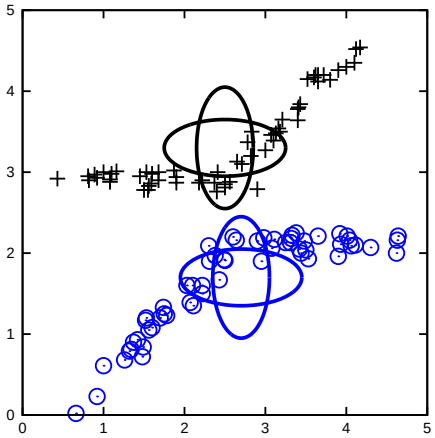
I_2 **- Iniţializare**



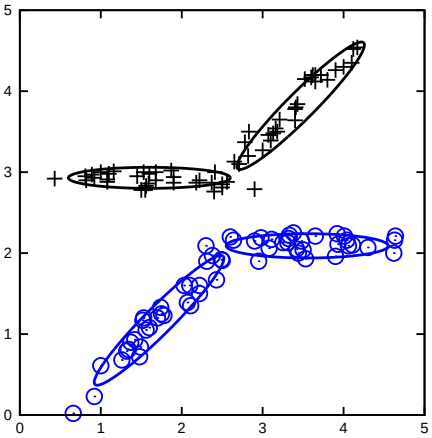
M_2 **- Model după prima iteraţie**



I_3 **- Iniţializare**



M_3 **- Model după prima iteraţie**



67. **(Comparaţii privind (in)adecvarea algoritmilor de clusterizare K-means şi EM/GMM pe diverse tipuri de seturi de date din $\mathbb{R}^2$)**

• ○ ***CMU, 2010 fall, Ziv Bar-Joseph, HW4, pr. 2.2-4***

a. Schiţaţi în planul euclidian un set de date pentru care o mixtură de distribuţii gaussiene sferice (i.e., pentru care matricea de covarianţă este matricea identitate înmulţită cu un număr pozitiv) poate să modeleze bine datele, însă algoritmul K-means nu poate.

b. Schiţaţi (tot în planul euclidian) un set de date pentru care o mixtură de distribuţii gaussiene diagonale (i.e., pentru care matricea de covarianţă poate avea valori diferite de zero doar pe prima diagonală) poate să modeleze bine datele, dar algoritmul K-means şi o mixtură de distribuţii gaussiene sferice nu pot.

c. Schiţaţi (tot în planul euclidian) un set de date pentru care o mixtură de distribuţii gaussiene având matricele de covarianţă nerestricţionate pot să modeleze bine datele, dar algoritmul K-means şi o mixtură de distribuţii gaussiene diagonale nu pot.

***Cerinţă suplimentară*: În fiecare dintre cazurile a, b şi c, veţi multiplica desenele de câte ori este nevoie în aşa fel încât să puteţi pune în evidenţă *graniţele de separare / decizie* induse de către fiecare dintre algoritmii menţionaţi în enunţ.**

68. **(Comparaţii privind (in)adecvarea algoritmilor de clusterizare ierarhică folosind diverse măsuri de similaritate, K-means şi EM/GMM, pe diverse tipuri de seturi de date din $\mathbb{R}^2$)**

• ○ ***CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, HW4, pr. 2.b***

Pentru fiecare dintre figurile următoare, precizaţi care anume dintre metodele de clusterizare listate mai jos va / vor produce rezultatele indicate, considerând $K = 2$. Alegeţi de fiecare dată cea mai probabilă metodă (sau, cele mai probabile metode) şi explicaţi pe scurt de ce ea va (respectiv, ele vor) lucra mai bine decât celelalte metode pe datele respective.

Iată lista de metode de clusterizare pe care le veţi considera:

– clusterizare ierarhică cu similaritate *single-linkage*

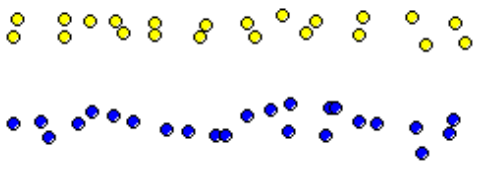
– clusterizare ierarhică cu similaritate *complete-linkage*

– clusterizare ierarhică cu similaritate *average-linkage*

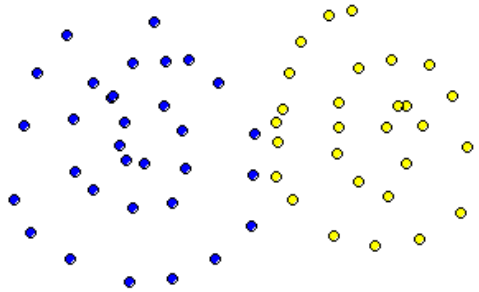
– K-means

– EM/GMM (fără a face vreo presupunere particulară în legătură cu matricea de covarianţă).

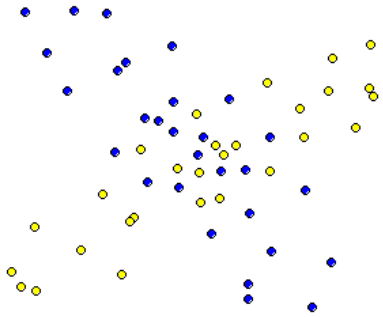
a.



b.



c.



69. **(K-means vs. EM/GMM, cazul multidimensional (π, μ, Σ))**

• ○ ***CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, midterm, pr. 8***

Redăm mai jos pseudo-codul algoritmului EM — de fapt, doar partea sa iterativă, conţinând cei doi paşi, E şi M — pentru clusterizare prin modelare de mixturi de gaussiene (GMM), cazul multidimensional.[946] Îţi cerem să faci schimbările minimale(!) necesare pentru a-l transforma într-un pseudo-cod corespunzător buclei principale a algoritmului de clusterizare K-means. Rescrie paşii care trebuie modificaţi, folosind spaţiul lăsat disponibil sub fiecare pas. Dacă un pas nu necesită schimbări, scrie „*Nicio modificare*“ în spaţiul disponibil sub pasul respectiv. Dacă un anumit pas nu este necesar, scrie „*Elimină acest pas*“ în spaţiul disponibil sub pasul respectiv.

***Pasul E*:**
Calculează probabilitatea $w_{ij}^{(t)}$ de asignare a instanţei x_i la clusterul [corespunzător gaussienei] j, ţinând cont de valorile actuale ale parametrilor $\pi^{(t)}$, $\mu^{(t)}$, $\Sigma^{(t)}$, unde t identifică iteraţia curentă a algoritmului EM.

$$w_{ij}^{(t)} = p(z_i = j | x_i, \pi^{(t)}, \mu^{(t)}, \Sigma^{(t)}) \text{ pentru } i \in \{1, \ldots, n\} \text{ şi } j \in \{1, \ldots, K\}$$

***Pasul M*:**
A. Re-calculează probabilităţile a priori pentru fiecare cluster / componentă a mixturii:

$$\pi_j^{(t+1)} = \sum_{i=1}^{n} \frac{w_{ij}^{(t)}}{n} \text{ pentru } j \in \{1, \ldots, K\}$$

B. Re-calculează [centroizii clusterelor, reprezentaţi de] mediile distribuţiilor gaussiene care „modelează“ clusterele:

[946] Pentru versiunea completă a acestui algoritm, vedeţi problema 24.

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^n w_{ij}^{(t)}\, x_i}{\sum_{i=1}^n w_{ij}^{(t)}} \textbf{ pentru } j \in \{1, \ldots, K\}$$

C. Re-calculează varianţele distribuţiilor gaussiene care „modelează“ clusterele:

$$\Sigma_j^{(t+1)} = \frac{\sum_{i=1}^n \left\{ w_{ij}^{(t)} (x_i - \mu_j^{(t+1)})(x_i - \mu_j^{(t+1)})^\top \right\}}{\sum_{i=1}^n w_{ij}^{(t)}} \textbf{ pentru } j \in \{1, \ldots, K\}$$

70. **(EM/GMM, cazul multidimensional, cazul când $\Sigma_k = \sigma_k^2 I$ şi probabilităţile de selecţie π_k sunt cunoscute / fixate: legătura dintre *metoda gradientului ascendent* pentru maximizarea funcţiei de log-verosimiltate a datelor observabile şi o versiune particulară a algoritmului EM)**

□ • ○ ***CMU, 2010 fall, Aarti Singh, HW4, pr. 1.3***

În acest exerciţiu vom explora conexiunile dintre algoritmul EM şi gradientul ascendent [ca metode de identificare a maximului funcţiei de verosimilitate a datelor].

Considerăm un model de mixtură de gaussiene (GMM) în care $\Sigma_k = \sigma_k^2 I$ pentru $k = 1, \ldots, K$, deci „clopotele“ corespunzătoare acestor K distribuţii sunt cu „deschidere“ / secţiune circulară, însă mărimile acestor „deschideri“ sunt în general diferite. Pe lângă aceasta, vom presupune că π_k, ponderile (engl., weights) distribuţiilor din mixtură, sunt cunoscute. Expresia funcţiei de log-verosimilitate este următoarea:

$$l\left(\{\boldsymbol{\mu}_k, \sigma_k^2\}_{k=1}^K\right) = \ln \prod_{i=1}^n \left(\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \sigma_k^2 I) \right) = \sum_{i=1}^n \ln \left(\sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, \sigma_k^2 I) \right).$$

Formulăm mai jos un algoritm de identificare a maximului funcţiei de log-verosimilitate, bazat pe metoda gradientului ascendent:

- **Iniţializează parametrii $\boldsymbol{\mu}_k$ şi σ_k^2, pentru $k \in \{1, \ldots, K\}$ cu valori alese în mod aleatoriu ($\boldsymbol{\mu}_k \in \mathbb{R}$ şi $\sigma_k^2 \in \mathbb{R}^+$). Setează t, contorul de iteraţie, la valoarea 1.**
- **Repetă paşii următori până când se ajunge la convergenţă:**
  - **Pentru $k = 1, \ldots, K$,**

$$\boldsymbol{\mu}_k^{(t+1)} \leftarrow \boldsymbol{\mu}_k^{(t)} + \eta_k^{(t)} \frac{\partial}{\partial \boldsymbol{\mu}_k} l\left(\{\boldsymbol{\mu}_k^{(t)}, (\sigma_k^2)^{(t)}\}_{k=1}^K\right),$$

unde $\eta_k^{(t)} > 0$ desemnează o rată de învăţare;

– **Pentru** $k = 1, \ldots, K$**,**

$$(\sigma_k^2)^{(t+1)} \leftarrow (\sigma_k^2)^{(t)} + s_k^{(t)} \frac{\partial}{\partial \sigma_k^2} l\left(\{\boldsymbol{\mu}_k^{(t+1)}, (\sigma_k^2)^{(t)}\}_{k=1}^K\right),$$

unde $s_k^{(t)} > 0$ **desemnează de asemenea o rată de învăţare;**

– **Incrementează contorul de iteraţie:** $t \leftarrow t + 1$**.**

Demonstraţi că dacă alegem în mod convenabil mărimile ratelor de învăţare (engl., step sizes) $\eta_k^{(t)}$ **şi** $s_k^{(t)}$**, acest algoritm bazat pe metoda gradientului ascendent este echivalent cu următorul algoritm de tip EM (care este însă uşor modificat în raport cu algoritmii de tip EM pe care i-am întâlnit până acum):**

- **Iniţializează în mod aleatoriu parametrii** $\boldsymbol{\mu}_k \in \mathbb{R}$ **şi** $\sigma_k^2 \in \mathbb{R}^+$**, pentru** $k \in \{1, \ldots, K\}$**. Setează** t **la valoarea** 1**.**
- **Repetă paşii următori până când se ajunge la convergenţă:**

  Pas E:

$$\tilde{z}_{ik}^{(t+0.5)} \leftarrow \textbf{Prob}\Big(\mathbf{x}_i \in \textbf{ cluster } k \;\Big|\; \{(\boldsymbol{\mu}_j^{(t)}, (\sigma_j^2)^{(t)})\}_{j=1}^K, \mathbf{x}_i\Big);$$

  Pas M:

$$\{\boldsymbol{\mu}_k^{(t+1)}\}_{k=1}^K \leftarrow \arg\max_{\{\boldsymbol{\mu}_k\}_{k=1}^K} \sum_{i=1}^n \sum_{k=1}^K \tilde{z}_{ik}^{(t+0.5)} \Big(\ln \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k, (\sigma_k^2)^{(t)} I) + \ln \pi_k\Big);$$

  Pas E:

$$\tilde{z}_{ik}^{(t+1)} \leftarrow \textbf{Prob}\Big(\mathbf{x}_i \in \textbf{ cluster } k \;\Big|\; \{(\boldsymbol{\mu}_j^{(t+1)}, (\sigma_j^2)^{(t)})\}_{j=1}^K, \mathbf{x}_i\Big);$$

  Pas M:

$$\{(\sigma_k^2)^{(t+1)}\}_{k=1}^K \leftarrow \arg\max_{\{\sigma_k^2\}_{k=1}^K} \sum_{i=1}^n \sum_{k=1}^K \tilde{z}_{ik}^{(t+1)} \Big(\ln \mathcal{N}(\mathbf{x}_i | \boldsymbol{\mu}_k^{(t+1)}, \sigma_k^2 I) + \ln \pi_k\Big);$$

  Incrementează contorul de iteraţie: $t \leftarrow t + 1$**.**

Aţi observat, desigur, că principala modificare (în raport cu ceilalţi algoritmi de tip EM) este faptul că a fost inserat un al doilea pas E între pasul M (i.e., aplicarea regulii de actualizare) pentru mediile $\boldsymbol{\mu}_k$ **şi pasul M pentru** σ_k^2**.**

***Sugestie*: Mărimile alese pentru ratele de învăţare** $\eta_k^{(t)}$ **şi** $s_k^{(t)}$ **trebuie să fie puse în corespondenţă cu cei doi paşi E.**

71. **(Algoritmul EM semi-supervizat: cazul mixturilor gaussiene multidimensionale; concepţie, implementare şi aplicare)**

□ • ***Stanford, 2020 summer, Andrew Ng, HW3, pr. 4.b-f***

În această problemă ne vom referi la modelul de mixturi de distribuţii gaussiene (engl., Gaussian Mixture Model, GMM), la care vom adapta algoritmul

EM semi-supervizat (vedeţi problema 22 de la capitolul *Schema algoritmică EM*).

Vom considera un „scenariu" în care datele sunt generate de către $k \in \mathbb{N}^*$ distribuţii gaussiene, având atât mediile $\mu_j \in \mathbb{R}^d$ cât şi matricele de covarianţă $\Sigma_j \in \mathbb{S}_+^d$ necunoscute, cu $j \in \{1, \ldots, k\}$. Avem n instanţe $x^{(i)} \in \mathbb{R}^d$, cu $i \in \{1, \ldots, n\}$, iar fiecărei instanţe îi corespunde o variabilă latentă (sau: ascunsă, necunoscută) $z^{(i)} \in \{1, \ldots, k\}$, care indică ce distribuţie a generat instanţa $x^{(i)}$. În mod concret, $z^{(i)} \sim$ Categorial(π), astfel încât $\sum_{j=1}^{k} \pi_j = 1$ şi $\pi_j \geq 0$ pentru orice j, iar $(x^{(i)}|z^{(i)}) \sim \mathcal{N}(\mu_{z^{(i)}}, \Sigma_{z^{(i)}})$ sunt independente şi identic distribuite (i.i.d.). Prin urmare, μ, Σ şi π sunt parametrii modelului.

De asemenea, mai avem $\tilde{n}$ instanţe $\tilde{x}^{(i)} \in \mathbb{R}^d$, cu $i \in \{1, \ldots, \tilde{n}\}$, cărora le sunt asociate variabilele *observabile* $\tilde{z}^{(i)} \in \{1, \ldots, k\}$, fiecare $\tilde{z}^{(i)}$ indicând distribuţia care a generat instanţa $\tilde{x}^{(i)}$. Remarcaţi faptul că $\tilde{z}^{(i)}$ sunt constante ale căror valori sunt cunoscute (în contrast cu $z^{(i)}$, care sunt variabile *aleatoare* având valori necunoscute). Ca şi mai înainte, presupunem că $(\tilde{x}^{(i)}|\tilde{z}^{(i)}) \sim \mathcal{N}(\mu_{\tilde{z}^{(i)}}, \Sigma_{\tilde{z}^{(i)}})$ sunt i.i.d.

Ca să rezumăm, aici considerăm $n + \tilde{n}$ instanţe de antrenament, dintre care n sunt instanţe neetichetate $x^{(i)}$, cărora le sunt asociate variabilele neobservabile $z^{(i)}$, iar $\tilde{n}$ sunt instanţe etichetate $\tilde{x}^{(i)}$, cărora le sunt asociate variabilele observabile $\tilde{z}^{(i)}$. Algoritmul EM tradiţional a fost conceput să ia ca input doar n instanţe neetichetate, iar apoi să înveţe parametrii modelului, μ, Σ şi π. *Obiectivul* acestui exerciţiu este să adaptaţi algoritmul EM semi-supervizat la modelul de mixturi de distribuţii gaussiene (GMM), luând în considerare — pe lângă cele n instanţe neetichetate — şi cele $\tilde{n}$ instanţe etichetate, şi elaborând regulile de actualizare specifice paşilor E şi M din acest algoritm de învăţare semi-supervizată pentru GMM.

a. [Pasul E semi-supervizat]
Indicaţi în mod clar care sunt variabilele latente care trebuie să fie re-estimate la pasul E. Deduceţi regula de actualizare de la pasul E pentru re-estimarea tuturor variabilelor latente pe care le-aţi indicat. În scrierea acestei regului de actualizare veţi putea să folosiţi doar x, z, μ, Σ, π şi constante universale.

b. [Pasul M semi-supervizat]
Indicaţi în mod clar care sunt parametrii care trebuie să fie re-estimaţi la pasul M. Deduceţi regulile de actualizare de la pasul M pentru re-estimarea tuturor parametrilor pe care i-aţi indicat. Mai precis, deduceţi formulele analitice (engl., closed form expressions) pentru regulile de actualizare ale parametrilor $\mu^{(t+1)}$, $\Sigma^{(t+1)}$ şi $\pi^{(t+1)}$, folosind funcţia obiectiv semi-supervizată.

c. [Implementarea algoritmului EM/GMM clasic (nesupervizat)]
La acest punct veţi lucra doar cu cele n instanţe neetichetate. Urmăriţi instrucţiunile din fişierul `src/semi_supervised_em/gmm.py` [furnizat în directorul corespunzător acestei probleme pe site-ul prezentei culegeri[947]] pentru a implementa algoritmul EM tradiţional, iar apoi rulaţi-l pe setul de instanţe neetichetate, până se ajunge la convergenţă.

Rulaţi algoritmul de trei ori şi folosiţi funcţia de reprezentare grafică furnizată pentru a construi o diagramă (engl., scatter plot) în care se pun în

[947] https://profs.info.uaic.ro/ ciortuz/ML.ex-book/implementation-exercises/Stanford.2020summer.AndrewNg.HW3.pr.4.semi-supervised-EM4GMM.DATA+CODE/

evidenţă asignările instanţelor la clustere. (Veţi face câte un grafic pentru fiecare rulare.) Graficele trebuie să indice asignările instanţelor la clustere cu ajutorul culorilor asociate clusterelor. (Clusterul la care este asociată o instanţă oarecare este acela pentru care se obţine cea mai mare probabilitate [de apartenenţă] la pasul E al ultimei iteraţii.)

d. [Implementarea algoritmului EM/GMM semi-supervizat]
Acum veţi folosi atât instanţele neetichetate cât şi instanţele etichetate (în total, $n + \tilde{n}$ instanţe), cu câte 5 instanţele etichetate la fiecare cluster. Vă punem la dispoziţie [tot pe site-ul acestei culegeri] schiţa un program pentru împărţirea setului de antrenament în matricele `x` şi `x_tilde` corespunzătoare instanţelor neetichetate şi respectiv instanţelor etichetate. Adăugaţi codul dumneavoastră la `src/semi_supervised_em/gmm.py` pentru a implementa algoritmul EM modificat, iar apoi rulaţi-l pe pe setul de instanţe de antrenament, până se ajunge la convergenţă. (Veţi face câte un grafic pentru fiecare rulare, aşa cum aţi procedat şi la punctul precedent.)

e. [Comparaţie între EM/GMM nesupervizat şi EM/GMM semi-supervizat]
Descrieţi succint diferenţele pe care l-aţi observat între algoritmul EM nesupervizat şi algoritmul EM semi-supervizat, referindu-vă la următoarele chestiuni:

i. **Numărul de iteraţii necesare pentru a se ajunge la convergenţă.**

ii. **Stabilitate (adică, cât de mult se schimbă asignările instanţelor în funcţie de diferitele iniţializări aleatoare?).**

iii. **Calitatea per ansamblu a asignărilor instanţelor la clustere.**

***Observaţie*: Setul de date de antrenament a fost eşantionat dintr-o mixtură de trei distribuţii gaussiene cu varianţă mică, precum şi o a patra distribuţie având varianţă mare. Această informaţie ar trebui să vă ajute la evaluarea calităţii per ansamblu a asignărilor pe care le produc cei doi algoritmi.**

72. (O legătură între clasificatorul Bayes Naiv gaussian şi algoritmul EM/GMM [pentru rezolvarea unei mixturi de distribuţii gaussiene multidimensionale satisfăcând presupoziţia de independenţă condiţională de tip Bayes Naiv]; o variantă semi-supervizată a algoritmului EM/GMM)

□ • ○ ***CMU, 2010 spring, T. Mitchell, E. Xing, A. Singh, midterm, pr. 5.3***

În acest exerciţiu vom analiza relaţia dintre un tip particular de mixturi de distribuţii gaussiene (engl., Gaussian Mixture Models, GMMs) multidimensionale şi clasificatorul Bayes Naiv gaussian (engl., Gaussian Naive Bayes, GNB). Vom arăta mai întâi că ambii algoritmi folosesc în esenţă acelaşi model probabilist.

Pentru simplitate, în cele ce urmează vom presupune că *mixturile* cu care vom lucra sunt constituite din doar două [componente] gaussiene.[948] Vom nota cu μ_0, μ_1, σ_0^2 şi σ_1^2 mediile şi varianţele celor două gaussiene şi vom considera că

[948] Veţi putea constata singuri că rezultatele care vor fi obţinute aici se pot extinde uşor la un număr oarecare (supra-unitar) de gaussiene.

proporţiile de mixare corespunzătoare celor două componente ale mixturii sunt π_0 şi respectiv $1-\pi_0$. De asemenea, vom folosi simbolul θ pentru a ne referi la întregul set de parametri $(\mu_0, \mu_1, \sigma_0^2, \sigma_1^2, \pi_0)$ care definesc modelul nostru de mixtură:

$$p(x) = \pi_0 \mathcal{N}(\mu_0, \sigma_0^2 I) + (1-\pi_0)\mathcal{N}(\mu_1, \sigma_1^2 I).$$

Pe de altă parte, *clasificatorul GNB*, ştim, face presupunerea că probabilitatea condiţionată $p(Y|X)$ satisface relaţia

$$p(Y|X) = \frac{p(Y)\prod_{k=1}^{d} p(X^k|Y)}{p(X)},$$

iar fiecare atribut X^k, la condiţionare cu [o valoare oarecare a lui] Y, este guvernat de o distribuţie gaussiană. În contextul exerciţiului nostru, este natural să considerăm că variabila aleatoare Y este de tip Bernoulli, cu $P(Y = 0) = \pi_0$ şi (din nou, pentru simplitate) că toate atributele au aceeaşi varianţă, deci

$$P(X^k|Y = j) \sim \mathcal{N}(\mu_{jk}, \sigma^2).$$

Remarcaţi faptul că ambii algoritmi specificaţi mai sus (GNB şi EM/GMM) folosesc acest model probabilist „generativ". Cu alte cuvinte, ambele modele presupun că generăm instanţe alegând mai întâi o valoare pentru Y (în funcţie de parametrul π_0), iar apoi extragem / generăm un X conform distribuţiei gaussiene condiţionate / determinate de valoarea lui Y. Singura *diferenţă* este că antrenăm clasificatorul GNB folosind instanţe etichetate pentru care care valoarea lui Y este cunoscută, pe când în cazul [execuţiei] algoritmului EM/GMM presupunem că valorile lui Y sunt necunoscute.

a. [Se ştie că] la antrenarea clasificatorului GNB se alege setul de parametri θ care maximizează verosimilitatea datelor:

$$\operatorname*{argmax}_{\theta} \prod_{i=1}^{n} p(x_i, y_i|\theta),$$

unde indicele i a fost folosit pentru a desemna exemplul de antrenament [LC: cu numărul de ordine] i. Vă cerem să scrieţi expresia funcţiei obiectiv pe care urmăreşte să o maximizeze celălalt algoritm, EM/GMM, atunci când învaţă parametrii aceluiaşi model, însă fără a cunoaşte valorile [pentru] y_i.

b. Scrieţi regulile de actualizare pentru paşii E şi M din algoritmul EM standard pentru rezolvarea mixturii de gaussiene.

c. Clasificatorul GNB este antrenat folosind exemple / instanţe etichetate, în vreme ce algoritmul EM/GMM foloseşte instanţe neetichetate. Presupunem că avem un set de date de antrenament în care avem instanţe de ambele tipuri. Cunoaştem valorile etichetei y pentru instanţele $x_1, x_2, \ldots, x_m$, în vreme ce instanţele $x_{m+1}, \ldots, x_{m+n}$ nu au valori cunoscute pentru y. Vă cerem să adaptaţi algoritmul EM/GMM pentru acest tip de date. Pentru aceasta, mai întâi veţi scrie funcţia obiectiv pe care o maximizează algoritmul EM/GMM modificat (pe care tocmai l-aţi obţinut). În expresia pe care o veţi scrie, va trebui să distingeţi între exemplele de antrenament pentru care y este cunoscut şi cele pentru care y este necunoscut.

d. Formulaţi paşii E şi M corespunzători acestui nou algoritm EM/GMM.[949]

[949] Vedeţi problema 22 de la capitolul *Schema algoritmică EM.*

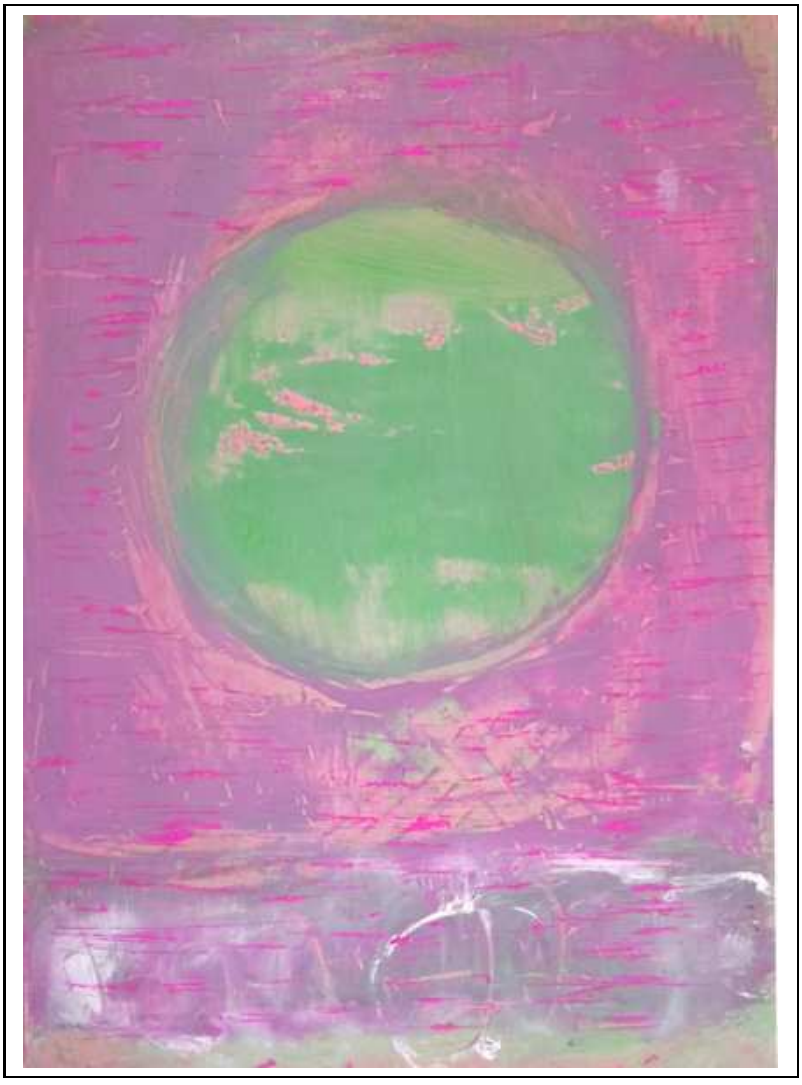
73. **(Algoritmul EM/GMM: implementare)**

□ ***Liviu Ciortuz, 2016***

Se consideră numărul natural $d \geq 1$, precum şi o mulţime formată din n instanţe ($X = \{x_1, \ldots, x_n\}$) din spaţiul $\mathbb{R}^d$. Obiectivul acestei probleme este să implementaţi în manieră orietată pe obiecte (pentru aceasta, vă sugerăm să folosiţi limbajul C++) algoritmul EM pentru a clusteriza instanţele din mulţimea X în K clustere, folosind o mixtură de distribuţii gaussiene de forma $\sum_{k=1}^{K} \pi_k \, \mathcal{N}(x|\mu_k, \Sigma_k)$.

***Cerinţe*:**

1. Veţi implementa câte o funcţie / „metodă“ pentru fiecare componentă a algoritmului EM: pasul de iniţializare, pasul E, pasul M şi condiţia de oprire. Adiţional, veţi implementa şi o „metodă“ pentru calculul funcţiei de verosimilitate a datelor „observabile“ X în raport cu parametrii modelului probabilist asociat. În ce priveşte condiţia de oprire, veţi crea posibilitatea ca utilizatorul să poată să aleagă (printr-o opţiune la linia de comandă) între două variante: fie să se efectueze un număr (prestabilit) de iteraţii, fie să se testeze diferenţa dintre verosimilitatea datelor X la iteraţia t şi verosimilitatea datelor X la iteraţia $t-1$ în raport cu un prag (prestabilit) ε.
2. Veţi trata separat — folosind supra-scrierea „metodelor“ precizate la punctul anterior — cazul distribuţiilor unidimensionale ($d = 1$) faţă de cazul distribuţiilor multidimensionale ($d > 1$). Pentru primul caz veţi implementa ambele variante ale algoritmului EM/GMM care apar în enunţul problemei 17.bc (şi veţi face testarea pe datele de la problema aceea, precum şi pe cele de la problema 16). Pentru a doilea caz veţi implementa algoritmul care a fost dedus la problema 24 (pentru testare, folosiţi datele de la problemele 21, 22, 23, 25 şi 66).
3. Veţi implementa — tot prin supra-scrierea „metodelor“ — încă două variante ale algoritmului EM/GMM cazul multidimensional, conform problemelor 20 şi 64. (Pentru testare, puteţi folosi datele de la problemele indicate la punctul precedent.)



© *M. Romanică*

8 Schema algoritmică EM

Sumar

Noţiuni preliminare

– **distribuţii probabiliste: vedeţi secţiunea *Distribuţii probabiliste uzuale* de la capitolul de *Fundamente*;**

– **estimarea parametrilor distribuţiilor probabiliste în sensul verosimilităţii maxime (MLE) şi respectiv în sensul probabilităţii maxime a posteriori (MAP): vedeţi secţiunea *Estimarea parametrilor unor distribuţii probabiliste* de la capitolul de *Fundamente*;**

– **mixturi de distribuţii probabiliste: vedeţi ex. 29, ex. 113 şi ex. 114 de la capitolul de *Fundamente*, ex. 5, ex. 7.AB şi ex. 27 de la prezentul capitol;**[950]

– **metoda maximizării alternante pe coordonate (engl., *coordinate ascent*) pentru rezolvarea unor probleme de optimizare: ex. 1;**[951]

– ***metoda multiplicatorilor lui Lagrange* pentru rezolvarea unor probleme de optimizare cu restricţii: vedeţi secţiunea *Metode de optimizare în învăţarea automată* de la capitolul de *Fundamente*, precum şi ex. 10, ex. 12, ex. 14, ex. 28, ex. 11 şi ex. 35 de la prezentul capitol.**

Schema algoritmică EM

– **pseudo-cod: *Machine Learning*, Tom Mitchell, 1997, pag. 194-195;**

– **fundamentare teoretică:**

(P0) ex. 1: pentru funcţia de log-verosimilitate a datelor observabile, $\ln P(X|\theta)$**, există o margine inferioară,** $F(q, \theta)$**; algoritmul EM face maximizarea acestei margini inferioare aplicând *metoda* [iterativă a] *creşterii alternative pe coordonate* (engl., coordinate ascent);**

(P1) ex. 2: monotonia valorilor funcţiei de log-verosimilitate a datelor observabile, care sunt calculate la iteraţii succesive ale lui EM, $\ln P(X|\theta^{(t)})$**;**

(P1′) nu se garantează găsirea optimului global al funcţiei de log-verosimilitate a datelor observabile, $\log P(X|\theta)$**, ci a unui optim local (dacă** $\ln P(X|\theta^{(t)})$ **este funcţie continuă);**

ex. 3: MAP EM – algoritmul EM pentru *estimarea* nu în sens MLE (cum este cazul adeseori), ci *în sens MAP*; pentru exemplificare, vedeţi ex. 32.B;[952]
ex. 22: algoritmul EM semi-supervizat (particularizare pentru cazul mixturilor de distribuţii Bernoulli: ex. 38);[953]
ex. 23: "hard" EM – algoritmul EM cu asignare "hard" a instanţelor la clustere;
ex. 24: algoritmul EM generalizat (engl., Generalized EM, GEM).

[950] Pentru mixturi de distribuţii gaussiene vedeţi secţiunea *Algoritmul EM pentru modele de mixturi gaussiene* de la capitolul de *Clusterizare*.

[951] Vedeţi, de asemenea, utilizarea aceleiaşi metode de optimizare (eventual pentru minimizare în loc de maximizare) în cazul altor algoritmi de învăţare automată: pentru algoritmul AdaBoost, ex. 22, ex. 28 şi ex. 29 de la capitolul de *Arbori de decizie*; pentru algoritmul K-means, ex. 12 de la capitolul de *Clusterizare*; în sfârşit, pentru algoritmul SMO, ex. 22 de la capitolul de *Maşini cu vectori-suport*.

[952] O altă exemplificare: MAP EM pentru mixturi de distribuţii [de acelaşi tip] din familia exponenţială: CMU, 2015 spring, Alex Smola, HW9, pr. 1.

[953] Pentru cazul mixturilor de distribuţii gaussiene, vedeţi ex. 71 de la capitolul *Clusterizare*.

EM pentru modelarea de mixturi de distribuţii probabiliste

– **mixturi de distribuţii *Bernoulli*: ex. 5, ex. 7, ex. 6 şi ex. 27;**

mixturi de *vectori* de distribuţii *Bernoulli*, cu presupunerea de independenţă condiţională a atributelor de intrare în raport cu atributul de ieşire (eticheta): ex. 8, ex. 9;

o versiune particulară — clusterizare în interiorul claselor —, cu *aplicare* la recunoaşterea cifrelor scrise de mână: ex. 29;

mixturi de distribuţii *categoriale*: ex. 10 şi ex. 28;

***aplicaţii*: [EM pentru] *dezambiguizarea semantică* a cuvintelor dintr-un document (engl., word sense disambiguation): ex. 12 şi respectiv pentru clusterizare de documente (engl., topic model): ex. 30;**

mixturi de *vectori* de distribuţii *categoriale*, cu presupunerea de independenţă condiţională a atributelor de intrare în raport cu atributul de ieşire (eticheta): ex. 11 (algoritmul *Bayes Naiv nesupervizat*, cu *asignare "soft"* a instanţelor la clustere);

mixturi de [vectori de] distribuţii *Poisson*: ex. 34, ex. 35;

mixturi de distribuţii *Gamma*: ex. 36;

– **EM pentru *estimarea probabilităţii de selecţie* a unei componente din cadrul unei mixturi (i.e., combinaţie liniară) de două distribuţii probabiliste oarecare: ex. 18.**

EM pentru estimarea parametrilor unor distribuţii probabiliste

– **EM pentru estimarea parametrilor unor distribuţii *binomiale*: ex. 31;**[954]

– **EM pentru estimarea parametrilor unor distribuţii *multinomiale* (care se definesc cu ajutorul uneia sau mai multor distribuţii *categoriale*): ex. 13, ex. 14, ex. 32, ex. 33.**

EM pentru estimarea parametrilor unei distribuţii, atunci când o parte din date lipesc

– **cazul distribuţiei *Poisson*: ex. 17.**

EM pentru estimarea parametrilor unei *sume* de două distribuţii[955]

– **cazul distribuţiilor *exponenţiale*: ex. 15;**

– **cazul distribuţiilor *gaussiene*: ex. 16.**

[954] În acest exerciţiu, apar trei distribuţii binomiale, dintre care una se defineşte cu ajutorul unei mixturi de două distribuţii Bernoulli.

[955] Exlicaţie: ne referim aici la aplicarea algoritmului EM pentru estimarea parametrilor a două distribuţii probabiliste atunci când se dau instanţe care sunt generate prin însumarea unor perechi de valori generate de cele două distribuţii;

Alte instanţe ale schemei algoritmice EM

– **algoritmul EM pentru învăţare supervizată; cazul mixturilor de regresori liniari: ex. 37.**

Alte probleme

– **chestiuni metodologice (relativ la iniţializarea parametrilor): ex. 39;**

– **probleme recapitulative (A/F): ex. 4, ex. 19, ex. 20, ex. 21 şi ex. 40.**

Analiza generală a algoritmilor EM

– **ca algoritm de *învăţare statistică*:**
algoritmul EM poate fi văzut ca o metodă de *estimare* a parametrilor (engl., parameter fitting) în sensul *verosimilităţii maxime* (engl., maximum likelihood estimation, MLE);[956]

– **ca algoritm *per se*:**
***algoritm iterativ*: pleacă de la o soluţie (instanţiere pentru parametri) aleasă eventual în mod arbitrar / aleatoriu şi o „îmbunătăţeşte“ la fiecare iteraţie;**

(P2) rezultatele algoritmului EM depind (ca şi la K-means) de valorile atribuite parametrilor la iniţializare;

(P3) anumite valori atribuite iniţial parametrilor algoritmului EM pot provoca rularea la infinit a algoritmului, fără ca [la pasul M] valorile parametrilor să se modifice de la o iteraţie la alta: ex. 6.f;

– **ca algoritm de *optimizare*:**

în *esenţă* / *rezumat*, metoda de maximizare a funcţiei de *log-verosimilitate a datelor observabile* $\log P(X|\theta)$ **este maximizarea la fiecare iteraţie** t **a unei funcţii auxiliare** Q_t**, care constituie o margine inferioară a lui** $\log P(X|\theta)$**, şi anume media funcţiei de *log-verosimilitate a datelor complete* în raport cu distribuţia de probabilitate a *variabilelor neobservabile* la iteraţia** t**;**

mai precis, la fiecare iteraţie t **se calculează funcţia „auxiliară“** $Q_t(\theta|\theta^{(t)})$**, care reprezintă media funcţiei de log-verosimilitate a datelor „complete“ (cele „observabile“ plus cele „neobservabile“), unde** $\theta^{(0)}$**, constând din valorile iniţiale ale parametrilor mixturii (**θ**), se alege în mod arbitrar, iar apoi** $\theta^{(t+1)} = \operatorname{argmax}_\theta Q_t(\theta|\theta^{(t)})$**;**

media reprezentată de funcţia Q_t **se calculează în funcţie de distribuţiile condiţionale ale variabilelor „neobservabile“** Z **în raport cu datele observabile** X **şi cu** $\theta^{(t)}$**;**

– **ca algoritm de *învăţare automată*:**
algoritmul EM este o metodă de identificare / învăţare de ipoteze ML (Maximum Likelihood); vedeţi capitolul / secţiunea 6.4 din cartea *Machine Learning*;
învăţare în prezenţa unor variabile aleatoare neobservabile(!);
[urmată eventual de] „generalizare“: o instanţă nouă x **se asociază clusterului (i.e., distribuţiei)** j **pentru care se atinge** $\max_{j'} P(X = x|h_{j'})P(h_{j'})$**.**

[956] Acesta este cazul general. În cazuri particulare, acest principiu poate fi înlocuit cu *principiul probabilităţii maxime a posteriori* (engl., maximum a posteriori probability, MAP). Vedeţi problema 3.

8.1 Schema algoritmică EM — Probleme rezolvate

***Observaţie importantă*: Pe tot parcursul acestui capitol, atunci când baza funcţiei** log **nu este specificată, o vom considera în mod implicit ca fiind supraunitară.**[957]

8.1.1 Fundamente teoretice

1. **(Algoritmul EM, fundamentare teoretică: pasul E [şi pasul M])**

prelucrare de Liviu Ciortuz, după
■ ***CMU, 2008 fall, Eric Xing, HW4, pr. 1.1-3***

Algoritmul EM (Expectation-Maximization) este unul dintre cele mai importante procedee din învăţarea automată. El permite crearea unor modele probabiliste care pe de o parte depind de un set de parametri θ iar pe de altă parte includ pe lângă variabilele obişnuite („observabile“ sau „vizibile“) x şi variabile necunoscute („neobservabile“, „ascunse“ sau „latente“) z.[958] **În general, în astfel de situaţii / modele, nu se poate face în manieră analitică (adică, prin optimizare directă) o estimare a parametrilor modelului (θ), în aşa fel încât să se garanteze obţinerea maximului verosimilităţii datelor observabile x.**[959] **În schimb, algoritmul EM procedează în manieră iterativă, constituind astfel o modalitate foarte convenabilă de estimare a parametrilor θ.**

Definim log-verosimilitatea datelor *complete* (observabile, x, şi neobservabile, z) ca fiind $\log P(x, z \mid \theta)$, iar log-verosimilitatea datelor observabile ca fiind $\log P(x \mid \theta)$.

a. Log-verosimilitatea datelor *observabile* (x) se poate exprima în funcţie de datele neobservabile (z), astfel:[960]

$$\ell(\theta) \stackrel{not.}{=} \log P(x \mid \theta) = \log \Big(\sum_z P(x, z \mid \theta) \Big)$$

[957] În secţiunea *Estimarea parametrilor unor distribuţii probabiliste* de la capitolul de *Fundamente*, precum şi în secţiunea *Algoritmul EM pentru modele de mixturi gaussiene* de la capitolul de *Clusterizare*, am folosit ca bază a logaritmului numărul e. Întrucât şi acolo şi aici ne interesează în general doar(!) să obţinem maximul unor funcţii de log-verosimilitate, alegerea bazei nu contează (atâta timp cât este un număr supraunitar). Totuşi, va trebui să observaţi că la punctul b şi c al problemei 1, precum şi la problema 2, intervin în calcule atât verosimilităţi cât şi noţiuni legate de teoria informaţiei (entropii, cross-entropii şi etropii relative). La secţiunea *Elemente de teoria informaţiei* de la capitolul de *Fundamente*, am folosit ca bază a logaritmului numărul 2. Aşadar, în astfel de situaţii va trebui să avem grijă cum „armonizăm“ între ele enunţurile şi relaţiile.

[958] Un astfel de model sunt *mixturile de distribuţii probabiliste*. Pentru exemple de mixturi de distribuţii Bernoulli şi respectiv distribuţii categoriale, vedeţi problemele 114, 113 şi 29 de la capitolul de *Fundamente*. Rezolvarea mixturilor de distribuţii gaussiene folosind algoritmul EM a constituit subiectul unei întregi secţiuni din capitolul de *Clusterizare*.

[959] Atunci când funcţia de log-verosimilitate este derivabilă şi toate variabilele sunt cunoscute / observabile, se poate calcula maximul acestei funcţii fie în mod direct, calculând rădăcinile derivatelor parţiale, fie aplicând o metodă de aproximare, aşa cum este metoda gradientului. În cazul în care o parte din variabile sunt necunoscute / neobservabile, aflarea soluţiilor derivatelor parţiale ale funcţiei de log-verosimilitate necesită adeseori aplicarea unor metode de calcul numeric sofisticate. Algoritmul EM, care este foarte uşor de implementat, ne oferă o cale alternativă, foarte utilă în astfel de cazuri.

[960] Cititorul trebuie să reţină că x, vectorul de date observabile, este fixat (dat), în vreme ce z, vectorul de date neobservabile, este liber (variabil).

În continuare vom nota cu q o funcţie / distribuţie de probabilitate definită peste variabilele ascunse / neobservabile z.

Folosiţi *inegalitatea lui Jensen*[961] pentru a demonstra că are loc următoarea inegalitate:

$$\log P(x \mid \theta) \geq \sum_z q(z) \log \left(\frac{P(x, z \mid \theta)}{q(z)} \right)$$

pentru orice set de date observabile x (fixat), pentru orice valoare a parametrului θ şi pentru orice distribuţie probabilistă q definită peste variabilele neobservabile z.

***Observaţie* (1):** Semnificaţia acestei inegalităţi este următoarea:
Funcţia

$$F(q, \theta) \stackrel{def.}{=} \sum_z q(z) \log \left(\frac{P(x, z \mid \theta)}{q(z)} \right) \tag{405}$$

constituie o margine inferioară pentru funcţia de log-verosimilitate a datelor observabile / incomplete, $\ell(\theta) \stackrel{not.}{=} \log P(x \mid \theta)$.[962] Remarcaţi faptul că F este o funcţie de două variabile, iar prima variabilă nu este de tip numeric (cum este θ), ci este de tip funcţional.[963] Mai mult, se observă că expresia funcţiei F este de fapt (similară cu) o medie, $E_{q(z)} \left[\log \frac{P(x, z \mid \theta)}{q(z)} \right]$, atunci când x, q şi parametrul θ se consideră fixaţi, iar z este lăsat să varieze.

b. Conform definiţiei divergenţei Kullback-Leibler (numită şi entropia relativă), care a fost dată la problema 65 de la capitolul de *Fundamente*, putem scrie:[964]

$$KL(q(z) \,||\, P(z \mid x, \theta)) = -\sum_z q(z) \log \left(\frac{P(z \mid x, \theta)}{q(z)} \right) \tag{406}$$

Arătaţi că — dacă folosim numărul 2 ca bază a logaritmului — avem:

$$\log P(x \mid \theta) = F(q(z), \theta) + KL(q(z) \,||\, P(z \mid x, \theta)).$$

***Observaţie* (2):** Semnificaţia egalităţii care trebuie demonstrată la acest punct este foarte interesantă: diferenţa dintre funcţia obiectiv $\ell(\theta) \stackrel{not.}{=} \log P(x \mid \theta)$ şi marginea sa inferioară $F(q(z), \theta)$ — a se vedea punctul a — este $KL(q(z) \,||\, P(z \mid x, \theta))$. Aceasta este divergenţa Kullback-Leibler dintre distribuţia (arbitrar) considerată $q(z)$ şi distribuţia condiţională a variabilei ascunse z în raport cu variabila observabilă x. Tocmai pe această chestiune se va „construi“ punctul final, şi cel mai important, al problemei noastre. Însă înainte de aceasta, este foarte util să formulăm încă o observaţie.

[961] Vedeţi problema 79 de la capitolul de *Fundamente*, şi în special punctul c al acestei probleme.

[962] În inegalitatea $\ell(\theta) \geq F(q, \theta)$, deşi poate părea surpinzător, funcţia ℓ are un singur argument, în vreme ce funcţia F are două argumente. Diversele aspecte (foarte interesante!) ale relaţiei dintre cele două funcţii vor fi „aduse la lumină“ în cele ce urmează.

[963] În continuare, pentru a aduce mereu aminte cititorului că distribuţia q se referă la datele neobservabile z, vom folosi notaţia $q(z)$ în loc de q. În consecinţă, în cele ce urmează, în funcţie de context, $q(z)$ va desemna fie la distribuţia q, fie la valoarea acestei distribuţii pentru o valoare oarecare [a variabilei neobservabile] z. (Suntem conştienţi de faptul că această lejeră ambiguitate poate induce în eroare cititorul neexperimentat.)

[964] Veţi observa că în relaţia (405) apare $P(x, z \mid \theta)$, iar în relaţia (406) apare $P(z \mid x, \theta)$.

***Observaţie* (3):** Ideile de bază ale algoritmului EM sunt două:
1. În loc să calculeze maximul funcţiei de log-verosimilitate $\log P(x \mid \theta)$ în raport cu θ, algoritmul EM va maximiza marginea sa inferioară, $F(q(z), \theta)$, în raport cu ambele argumente, $q(z)$ şi θ.
2. Pentru a căuta maximul (de fapt, un maxim local al) marginii inferioare $F(q(z), \theta)$, algoritmul EM aplică metoda *creşterii pe coordonate* (engl., coordinate ascent): după ce iniţial se fixează $\theta^{(0)}$ eventual aleatoriu, se maximizează *iterativ* funcţia $F(q(z), \theta)$, în mod *alternativ*: mai întâi în raport cu distribuţia $q(z)$ şi apoi în raport cu parametrul θ.

$$\text{Pasul E:} \quad q^{(t)}(z) = \underset{q(z)}{\operatorname{argmax}} F(q(z), \theta^{(t)})$$
$$\text{Pasul M:} \quad \theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} F(q^{(t)}(z), \theta)$$

c. Fie $\theta^{(t)}$ valoarea obţinută pentru parametrul / parametrii θ la iteraţia t a algoritmului EM. Considerând această valoare fixată, arătaţi că maximul lui F în raport cu argumentul / distribuţia $q(z)$ este atins pentru distribuţia $P(z \mid x, \theta^{(t)})$, iar — dacă baza logaritmului este 2 — valoarea maximului se poate exprima astfel:

$$\max_{q(z)} F(q(z), \theta^{(t)}) = E_{P(z|x,\theta^{(t)})}[\log P(x, z \mid \theta^{(t)})] + H(P(z \mid x, \theta^{(t)}))$$

Aşadar, atunci când se consideră fixat cel de-al doilea argument al funcţiei F, valoarea maximă a acestei funcţii (F) în raport cu primul său argument, $q(z)$, este suma a doi termeni: media log-verosimilităţii datelor complete (observabile şi neobservabile) în raport cu distribuţia posterioară a variabilelor neobservabile z (primul termen) şi entropia acestei distribuţii posterioare (al doilea termen).

Răspuns:

a. În contextul teoriei probabilităţilor, inegalitatea lui Jensen este exprimată astfel: dacă X este o variabilă aleatoare iar φ este o funcţie convexă, atunci $\varphi(E[X]) \leq E[\varphi(X)]$. Dacă φ este funcţie concavă, inegalitatea lui Jensen devine $\varphi(E[X]) \geq E[\varphi(X)]$.[965]

În cazul nostru, folosim funcţia $\log$ cu bază supraunitară, care este o funcţie concavă, deci aplicând inegalitatea lui Jensen obţinem: $\log(E[X]) \geq E[\log(X)]$.

Log-verosimilitatea datelor observabile este:

$$\begin{aligned} \log P(x \mid \theta) &= \log\Big(\sum_z P(x, z \mid \theta)\Big) = \log\Big(\sum_z q(z)\frac{P(x, z \mid \theta)}{q(z)}\Big) \\ &\overset{def.}{=} \log\left(E_{q(z)}\left[\frac{P(x, z \mid \theta)}{q(z)}\right]\right) \end{aligned}$$

De aici, conform inegalităţii lui Jensen — concret, înlocuind X din inegalitatea menţionată cu $\dfrac{P(x, z \mid \theta)}{q(z)}$ —, rezultă:

$$\log P(x \mid \theta) \geq E_{q(z)}\left[\log \frac{P(x, z \mid \theta)}{q(z)}\right] \overset{def.}{=} \sum_z q(z) \log \frac{P(x, z \mid \theta)}{q(z)},$$

[965] Vedeţi problema 79.c de la capitolul de *Fundamente*.

adică tocmai ceea ce trebuia să demonstrăm.

b. Pentru a demonstra egalitatea cerută, vom pleca de la definiţia dată în enunţ pentru funcţia $F(q(z), \theta)$. Apoi, în expresia din definiţie vom înlocui $P(x, z \mid \theta)$ cu $P(z \mid x, \theta) \cdot P(x \mid \theta)$ — conform regulii de înmulţire a probabilităţilor — şi vom obţine:

$$\begin{aligned} F(q(z), \theta) &\stackrel{def.}{=} \sum_z q(z) \log \left(\frac{P(x, z \mid \theta)}{q(z)} \right) \\ &= \sum_z q(z) \log \left(\frac{P(z \mid x, \theta) \cdot P(x \mid \theta)}{q(z)} \right) \\ &= \sum_z q(z) \left[\log \frac{P(z \mid x, \theta)}{q(z)} + \log P(x \mid \theta) \right] \\ &= \sum_z q(z) \log \left(\frac{P(z \mid x, \theta)}{q(z)} \right) + \sum_z q(z) \log P(x \mid \theta) \\ &= -KL(q(z) \mid\mid P(z \mid x, \theta)) + \log P(x \mid \theta) \cdot \underbrace{\sum_z q(z)}_{=1} \end{aligned}$$

Rezultă că $\log P(x \mid \theta) = F(q(z), \theta) + KL(q(z) \mid\mid P(z \mid x, \theta))$.

***Observaţie* (4):** Conform proprietăţii $KL(p \mid\mid q) \geq 0$ pentru $\forall p, q$, care a fost demonstrată şi de noi la exerciţiul 65 de la capitolul de *Fundamente*, rezultă că $KL(q(z) \mid\mid P(z \mid x, \theta)) \geq 0$. Aşadar, din egalitatea care tocmai a fost demonstrată la punctul b obţinem (din nou!, după rezultatul de la punctul a) că $F(q(z), \theta)$ este o margine inferioară pentru log-verosimilitatea datelor observabile, $\ell(\theta) \stackrel{not.}{=} \log P(x \mid \theta)$.

c. Trebuie să maximizăm $F(q(z), \theta^{(t)})$ — marginea inferioară a log-verosimilităţii datelor observabile x — în raport cu distribuţia $q(z)$.

Pe de o parte, rezultatul de la punctul a ne spune că $F(q(z), \theta) \leq \log P(x \mid \theta)$, pentru orice valoare a lui θ; în particular, pentru $\theta^{(t)}$ avem

$$\log P(x \mid \theta^{(t)}) \geq F(q(z), \theta^{(t)})$$

Pe de altă parte, dacă în egalitatea demonstrată la punctul b se înlocuieşte θ cu $\theta^{(t)}$, rezultă:

$$\log P(x \mid \theta^{(t)}) = F(q(z), \theta^{(t)}) + KL(q(z) || P(z \mid x, \theta^{(t)}))$$

În fine, dacă alegem $q(z) = P(z \mid x, \theta^{(t)})$, atunci termenul $KL(q(z) || P(z \mid x, \theta^{(t)}))$ din dreapta egalităţii de mai sus devine zero (a se vedea punctul a de la acelaşi exerciţiu 65 de la capitolul de *Fundamente*). Aşadar, valoarea $\max_{q(z)} F(q(z), \theta^{(t)})$ se obţine pentru distribuţia $q(z) = P(z \mid x, \theta^{(t)})$.

Acum vom calcula această valoare maximă:

$$\begin{aligned} \log P(x \mid \theta^{(t)}) &= \max_{q(z)} F(q(z), \theta^{(t)}) = F(P(z \mid x, \theta^{(t)}), \theta^{(t)}) \\ &\stackrel{def.\ F}{=} \sum_z P(z \mid x, \theta^{(t)}) \log \left(\frac{P(x, z \mid \theta^{(t)})}{P(z \mid x, \theta^{(t)})} \right) \\ &= E_{P(z|x,\theta^{(t)})} \left[\log \frac{P(x, z \mid \theta^{(t)})}{P(z \mid x, \theta^{(t)})} \right] \\ &= E_{P(z|x,\theta^{(t)})} \left[\log P(x, z | \theta^{(t)}) - \log P(z | x, \theta^{(t)}) \right] \end{aligned}$$

$$
\begin{aligned}
&= E_{P(z|x,\theta^{(t)})}\left[\log P(x,z|\theta^{(t)})\right] - E_{P(z|x,\theta^{(t)})}\left[\log P(z|x,\theta^{(t)})\right] \\
&= E_{P(z|x,\theta^{(t)})}\left[\log P(x,z|\theta^{(t)})\right] + H[P(z|x,\theta^{(t)})] \\
&= Q(\theta^{(t)} \mid \theta^{(t)}) + H[P(z \mid x,\theta^{(t)})],
\end{aligned}
$$

unde $Q(\theta \mid \theta^{(t)}) \stackrel{not.}{=} E_{P(z|x,\theta^{(t)})}[\log P(x,z \mid \theta)]$. Aşadar, am obţinut rezultatul care a fost cerut în enunţ.

***Observaţie* (5): [Fundamentarea pasului M]**
Notând

$$G_t(\theta) \stackrel{def.}{=} F(P(z \mid x,\theta^{(t)}),\theta),$$

din calculul de mai sus rezultă că

$$\log P(x \mid \theta^{(t)}) = G_t(\theta^{(t)}) = Q(\theta^{(t)} \mid \theta^{(t)}) + H(P(z \mid x,\theta^{(t)})).$$

Se poate demonstra uşor — procedând similar cu calculul de mai sus — egalitatea

$$G_t(\theta) = Q(\theta \mid \theta^{(t)}) + H[P(z \mid x,\theta^{(t)})]$$

Observând că termenul $H[P(z \mid x,\theta^{(t)})]$ din această ultimă egalitate nu depinde de θ, rezultă imediat că

$$\operatorname*{argmax}_{\theta} G_t(\theta) = \operatorname*{argmax}_{\theta} Q(\theta \mid \theta^{(t)})$$

În consecinţă,

$$\theta^{(t+1)} \stackrel{def.}{=} \operatorname*{argmax}_{\theta} \underbrace{F(P(z \mid x,\theta^{(t)}),\theta)}_{G_t(\theta)} = \operatorname*{argmax}_{\theta} Q(\theta \mid \theta^{(t)})$$

Ultima egalitate de mai sus este responsabilă pentru următoarea reformulare (cea uzuală!) a corpului iterativ al algoritmului EM:

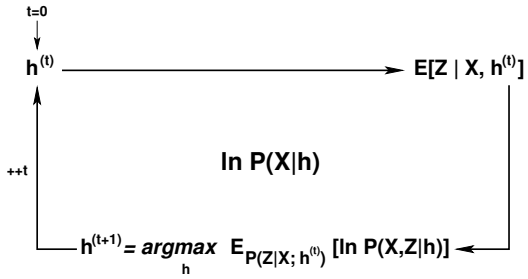
Pasul E′: **calculează** $Q(\theta \mid \theta^{(t)}) = E_{P(z|x,\theta^{(t)})}[\log P(x,z \mid \theta)]$
Pasul M′: **calculează** $\theta^{(t+1)} = \operatorname*{argmax}_{\theta} Q(\theta \mid \theta^{(t)})$

Aşadar, pasul E ("expectation") al algoritmul EM va consta în calcularea funcţiei auxiliare $Q(\theta \mid \theta^{(t)})$. Pentru calculul expresiei acestei funcţii (care este o medie, vedeţi definiţia ei de mai sus) se folosesc mediile variabilelor neobservabile, $E[z_i \mid x,\theta^{(t)}]$. Adeseori, în practică, la pasul E al algoritmului EM se face doar(!) calcularea mediilor acestor variabile neobservabile.[966] În continuare, la pasul M al aceluiaşi algoritm se vor calcula parametrii $\theta^{(t+1)} = \operatorname{argmax}_{\theta} Q(\theta \mid \theta^{(t)})$, de obicei ca rădăcini ale derivatelor parţiale ale funcţiei $Q(\theta \mid \theta^{(t)})$, atunci când aceste derivate există.

***Observaţie* (6): Reprezentarea grafică următoare sintetizează modul în care funcţionează algoritmul EM (sau, de fapt, mai general, schema algoritmică**

[966] În cazul în care $E[z_i \mid x,\theta^{(t)}]$ reprezintă totodată un parametru al distribuţiei probabiliste care guvernează datele observabile — de exemplu, factorul de combinare a unor distribuţii în cadrul unei mixturi, care este adeseori parametrul unei distribuţii de tip Bernoulli —, calculul lui $E[z_i \mid x,\theta^{(t)}]$ este în sine o procedură de *estimare* a acestui parametru. Alţi parametri (de exemplu μ şi σ^2 pentru distribuţii gaussiene) pot fi determinaţi în funcţie de mediile acestor variabile neobservabile. Aşa se explică de ce termenul "Estimation" este folosit de către unii autori în locul termenului "Expectation" în denumirea algoritmului EM.

EM), conform rezultatelor demonstrate la acest exerciţiu. În această imagine, $h^{(t)}$ corespunde lui $\theta^{(t)}$ din exerciţiul nostru.



2. **(Algoritmul EM: monotonie / convergenţă)**

prelucrare de Liviu Ciortuz, 2012, după
en.wikipedia.org/wiki/Expectation-maximization

La problema precedentă (1) am văzut că în loc să urmărească în mod direct maximizarea funcţiei de log-verosimilitate a datelor observabile, adică $\ell(\theta) \stackrel{def.}{=} \log P(x \mid \theta)$, unde baza logaritmului (nespecificată) este considerată supraunitară, algoritmul EM procedează în mod iterativ, optimizând la pasul M al fiecărei iteraţii (t) o funcţie „auxiliară" $Q(\theta \mid \theta^{(t)}) \stackrel{def.}{=} E_{P(z|x,\theta^{(t)})}[\log P(x, z \mid \theta)]$, reprezentând media log-verosimilităţii datelor complete (observabile şi neobservabile) în raport cu distribuţia a posteriori $P(z \mid x, \theta^{(t)})$.

Vom considera iteraţiile $t = 0, 1, \ldots$ şi $\theta^{(t+1)} = \operatorname{argmax}_\theta Q(\theta \mid \theta^{(t)})$, cu $\theta^{(0)}$ ales în mod arbitrar.

În acest exerciţiu veţi demonstra că pentru orice t fixat (arbitrar) şi pentru orice θ astfel încât $Q(\theta \mid \theta^{(t)}) \geq Q(\theta^{(t)} \mid \theta^{(t)})$ are loc inegalitatea:

$$\log P(x \mid \theta) - \log P(x \mid \theta^{(t)}) \geq Q(\theta \mid \theta^{(t)}) - Q(\theta^{(t)} \mid \theta^{(t)}) \tag{407}$$

***Observaţie* (1):** Această inegalitate are o semnificaţie pe cât de simplă pe atât de importantă: orice îmbunătăţire a valorii funcţiei $Q(\theta \mid \theta^{(t)})$ conduce la o îmbunătăţire cel puţin la fel de mare a valorii funcţiei obiectiv, $\ell(\theta) \stackrel{def.}{=} \log P(x \mid \theta)$.

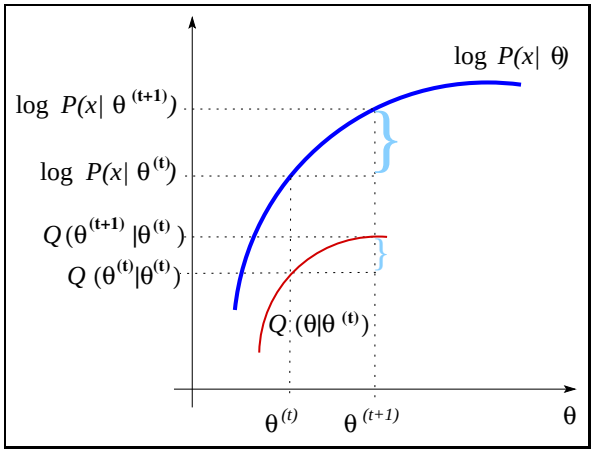
***Observaţie* (2):**
Dacă în inegalitatea (407) se înlocuieşte θ cu

$$\theta^{(t+1)} = \operatorname*{argmax}_\theta Q(\theta \mid \theta^{(t)}),$$

va rezulta

$$\log P(x \mid \theta^{(t+1)}) \geq \log P(x \mid \theta^{(t)}).$$

Altfel spus, la fiecare iteraţie a algoritmului EM, odată cu trecerea de la $\theta^{(t)}$ la $\theta^{(t+1)}$, valoarea funcţiei de log-verosimiliate a datelor observabile, $\ell(\theta) \stackrel{def.}{=} \log P(x|\theta)$ creşte sau, în cel mai rău caz, rămâne pe loc.



În final, vom avea $\ell(\theta^{(0)}) \leq \ell(\theta^{(t)}) \leq \ell(\theta^{(t+1)}) \leq \ldots$. Şirul acesta (monoton) este mărginit superior de 0 (vedeţi definiţia lui ℓ), deci converge la o anumită valoare ℓ^*. În anumite cazuri / condiţii, această valoare este un maxim (în general, local) al funcţiei de log-verosimilitate a datelor observabile.[967]

***Observaţie* (3):** Conform aceleiaşi inegalităţi (407), la pasul M de la iteraţia t a algoritmului EM, este suficient ca în loc să se ia $\theta^{(t+1)} = \text{argmax}_\theta\, Q(\theta \mid \theta^{(t)})$, să se aleagă $\theta^{(t+1)}$ astfel încât $Q(\theta^{(t+1)} \mid \theta^{(t)}) > Q(\theta^{(t)} \mid \theta^{(t)})$. Aceasta constituie *versiunea „generalizată“* a algoritmului EM.

Răspuns:

$P(x, z \mid \theta) = P(z \mid x, \theta) \cdot P(x \mid \theta)$**, conform regulii de înmulţire a probabilităţilor. Prin logaritmarea acestei egalităţi, rezultă:**

$$\log P(x \mid \theta) = \log P(x, z \mid \theta) - \log P(z \mid x, \theta)$$

Ca să obţinem expresia funcţiei „auxiliare“ Q **pe care o optimizează algoritmul EM la pasul M al iteraţiei** t**, vom înmulţi ambii membri ai egalităţii precedente cu** $P(z \mid x, \theta^{(t)})$ **şi apoi vom suma după toate valorile posibile ale lui** z**:**

$$\sum_z P(z \mid x, \theta^{(t)}) \cdot \log P(x \mid \theta) =$$

$$\sum_z P(z \mid x, \theta^{(t)}) \cdot \log P(x, z \mid \theta) - \sum_z P(z \mid x, \theta^{(t)}) \cdot \log P(z \mid x, \theta) \qquad (408)$$

Membrul stâng al acestei egalităţi poate fi rescris astfel:

$$\sum_z P(z \mid x, \theta^{(t)}) \cdot \log P(x \mid \theta) = \log P(x \mid \theta) \cdot \underbrace{\sum_z P(z \mid x, \theta^{(t)})}_{1} = \log P(x \mid \theta)$$

În ceea ce priveşte membrul drept al aceleiaşi egalităţi (408), întrucât termenul al doilea, şi anume $-\sum_z P(z \mid x, \theta^{(t)}) \cdot \log P(z \mid x, \theta)$ **reprezintă o cross-entropie (vedeţi problema 63 de la capitolul de *Fundamente*), o vom nota cu** $CH(\theta \mid \theta^{(t)})$**. Aşadar, egalitatea (408) devine:**

$$\log P(x \mid \theta) = Q(\theta \mid \theta^{(t)}) + CH(\theta \mid \theta^{(t)}) \qquad (409)$$

Această egalitate este valabilă pentru toate valorile posibile ale parametrului θ**. În particular pentru** $\theta = \theta^{(t)}$**, vom avea:**

$$\log P(x \mid \theta^{(t)}) = Q(\theta^{(t)} \mid \theta^{(t)}) + CH(\theta^{(t)} \mid \theta^{(t)}) \qquad (410)$$

Scăzând membru cu membru ultimele două egalităţi, obţinem:

$$\log P(x \mid \theta) - \log P(x \mid \theta^{(t)}) = Q(\theta \mid \theta^{(t)}) - Q(\theta^{(t)} \mid \theta^{(t)}) + CH(\theta \mid \theta^{(t)}) - CH(\theta^{(t)} \mid \theta^{(t)})$$

Conform inegalităţii lui Gibbs — a se vedea problema 64 de la capitolul de *Fundamente* —,[968] **avem** $CH(\theta \mid \theta^{(t)}) \geq CH(\theta^{(t)} \mid \theta^{(t)})$**, deci în final rezultă:**

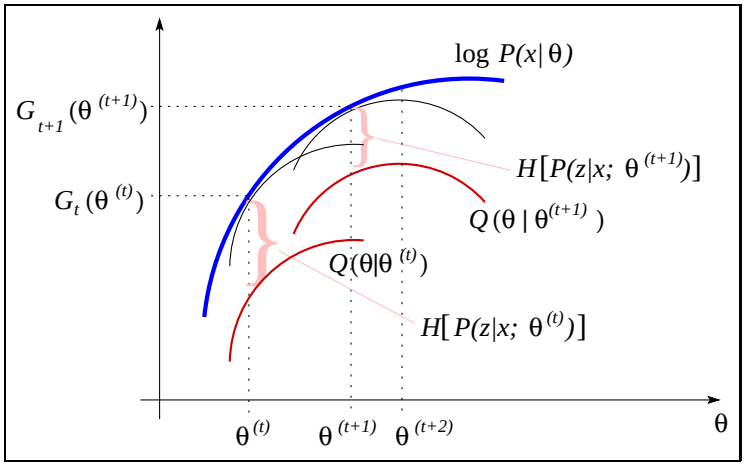
$$\log P(x \mid \theta) - \log P(x \mid \theta^{(t)}) \geq Q(\theta \mid \theta^{(t)}) - Q(\theta^{(t)} \mid \theta^{(t)})$$

[967] Vedeţi capitolul *The EM Algorithm*, de Shu Kay Ng et al., în *Handbook of Computational Statistics*, Springer, 2004, pag. 139.

[968] Folosind noţiunea de divergenţă Kullback-Leibler, inegalitatea aceasta se rescrie sub forma $KL(p \,||\, q) \geq 0$. În notaţiile / termenii din rezolvarea problemei noastre, această inegalitate se scrie $CH(p\|q) \geq CH(p\|p) = H(p)$.

Aceasta este chiar inegalitatea pe care trebuia s-o demonstrăm.

***Observaţie* (4):**
Cu ajutorul figurii alăturate, putem să ilustrăm şi totodată să sumarizăm în mod grafic rezultatele pe care le-am demonstrat la aceste ultime două probleme (1 şi 2).



***Observaţie* (5): [Proprietăţi ale algoritmului EM[969]]**
Algoritmul EM este util datorită următoarelor *avantaje*: simplitate conceptuală, posibilitatea estimării datelor neobservabile, uşurinţa implementării, precum şi faptul că la fiecare iteraţie se îmbunătăţesc valorile parametrului / parametrilor θ. Rata de convergenţă la primele câteva iteraţii ale algoritmului EM este în general foarte bună, însă ea poate deveni foarte mică pe măsură ce ne apropiem de punctul de optim local. În general, algoritmul EM dă rezultate bune atunci când proporţia datelor neobservabile este mică, iar dimensiunea datelor (adică, numărul de atribute ale instanţelor) nu este prea mare. EM poate necesita executarea multor iteraţii, iar în cazul dimensiunilor mari pasul E poate deveni foarte lent.

3. **(Algoritmul EM pentru estimarea parametrilor în sens MAP: fundamentare teoretică)**

□ • ○ ***Stanford, 2016 fall. A. Ng, J. Duchi, HW4, pr. 1***

Algoritmul EM, aşa cum l-am folosit până la acest exerciţiu, a fost conceput pentru a rezolva [unele] probleme de *estimare de parametri în sensul verosimilităţii maxime* (engl., maximum likelihood estimation, MLE). În astfel de probleme se urmăreşte să se maximizeze o expresie de forma

$$\prod_{i=1}^{n} p(x_i|\theta) = \prod_{i=1}^{n} \Big(\sum_{z_i} p(x_i, z_i|\theta) \Big), \tag{411}$$

unde

θ **este setul de parametri pentru distribuţia probabilistă** p**,**

x_i **(cu** $i = 1, \ldots, n$**) sunt variabile aleatoare observabile,**

z_i **(cu** $i = 1, \ldots, n$**) sunt variabile aleatoare neobservabile / latente.**

[969] Cf. documentului *The EM Algorithm*, Ajit Singh, CMU, November 20, 2005. Vedeţi de asemenea secţiunea 5.2.5 din capitolul *The EM Algorithm*, de Shu Kay Ng et al., din volumul *Handbook of Computational Statistics*, Springer, 2004.

În acest exerciţiu presupunem că lucrăm într-un *cadru bayesian*, adică dorim să găsim *estimarea în sensul probabilităţii maxime a posteriori* (engl., maximum a posteriori probability, MAP) pentru parametrii θ. Aceasta revine la a maximiza o expresie de forma următoare:[970]

$$\left(\prod_{i=1}^{n} p(x_i|\theta)\right) \cdot p(\theta) = \left(\prod_{i=1}^{n} \left(\sum_{z_i} p(x_i, z_i|\theta)\right)\right) \cdot p(\theta),$$

unde $p(\theta)$ desemnează o distribuţie de probabilitate a priori definită peste valorile parameterilor θ.

Generalizaţi algoritmul EM astfel încât să realizeze estimări în sens MAP. Veţi presupune că ambele distribuţii — $p(x, z|\theta)$ şi $p(\theta)$ — sunt concave în raport cu parametrul θ. Aceasta va implica faptul că pasul M al algoritmului EM este *realizabil* (engl., tractable) atunci când se cere doar să se maximizeze o combinaţie liniară de aceste cantităţi ($p(x, z|\theta)$ şi $p(\theta)$).[971]

Asiguraţi-vă că pasul M din formularea pe care o veţi adopta pentru noul algoritm EM este într-adevăr *realizabil*. De asemenea, demonstraţi că valorile expresiei $(\prod_{i=1}^{n} p(x_i|\theta)) \cdot p(\theta)$, văzută ca funcţie de θ, nu descresc — adică, fie cresc, fie rămân pe loc — de la o iteraţie la alta a noului algorithm.

Răspuns:

Vom deduce regulile de actualizare de la cei doi paşi ai algoritmului EM într-un mod similar celui în care am procedat la problema 1 pentru estimare în sensul verosimilităţii maxime (MLE). La fiecare iteraţie (t), creşterea este monotonă, din aceleaşi motive ca şi la problema 1: la pasul E calculăm [o] cea mai bună *margine inferioară* (engl., lower bound) a log-probabilităţii a posteriori pentru datele observabile, în raport cu $\theta^{(t)}$, care reprezintă valorile curente ale parametrilor θ, iar la pasul M alegem pentru θ o nouă valoare, $\theta^{(t+1)}$, care este optimă în raport cu această margine inferioară.

Pornind de la expresia (411) din enunţ, vom identifica marginea inferioară despre care tocmai am vorbit:

$$\log\left(\left(\prod_{i=1}^{n} p(x_i|\theta)\right) \cdot p(\theta)\right) = \log p(\theta) + \sum_{i=1}^{n} \log p(x_i|\theta)$$

$$= \log p(\theta) + \sum_{i=1}^{n} \log \sum_{z_i} p(x_i, z_i|\theta) = \log p(\theta) + \sum_{i=1}^{n} \log \sum_{z_i} q(z_i) \cdot \frac{p(x_i, z_i|\theta)}{q(z_i)}$$

[970] Justificarea este aceasta: log-probabilitatea a posteriori pentru datele observabile este

$$\log p(\theta|x_1, \ldots, x_n) \overset{F.\ Bayes}{=} \log \frac{p(x_1, \ldots, x_n|\theta) \cdot p(\theta)}{p(x_1, \ldots, x_n)} = \log(p(x_1, \ldots, x_n|\theta) \cdot p(\theta)) - \log p(x_1, \ldots, x_n)$$

$$\overset{i.i.d.}{=} \log\left(\left(\prod_{i=1}^{n} p(x_i|\theta)\right) \cdot p(\theta)\right) - \log p(x_1, \ldots, x_n).$$

Întrucât cel de-al doilea termen nu depinde de θ, rezultă

$$\operatorname*{argmax}_{\theta} \log p(\theta|x_1, \ldots, x_n) = \operatorname*{argmax}_{\theta} \log\left(\left(\prod_{i=1}^{n} p(x_i|\theta)\right) \cdot p(\theta)\right).$$

[971] Grosso-modo, aceasta revine la a presupune că estimarea în sens MAP este *realizabilă* atunci când variabilele x, z sunt în totalitate observabile, exact cum procedam în cadrul „frecvenţionist“ (adică, non-bayesian).

$$\geq \quad \log p(\theta) + \sum_{i=1}^{n} \sum_{z_i} q(z_i) \cdot \log \frac{p(x_i, z_i|\theta)}{q(z_i)}. \tag{412}$$

La ultimul pas de mai sus am aplicat inegalitatea lui Jensen.[972] **Ca şi la problema 1, inegalitatea** (412) **are loc cu egalitate atunci când la *pasul E* (de la iteraţia** t**) luăm**

$$q(z_i) = p(z_i|x_i; \theta^{(t)}).$$

Pentru *pasul M*, trebuie să maximizăm marginea inferioară (vedeţi relaţia (412)**), adică vom calcula**

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}} \left(\log p(\theta) + \sum_{i=1}^{n} \sum_{z_i} q(z_i) \log \frac{p(x_i, z_i|\theta)}{q(z_i)} \right).$$

Pasul M este *realizabil*, întrucât ceea ce se cere aici este doar să maximizăm o combinaţie liniară de termenii $\log p(\theta)$ **şi** $\log p(x, z|\theta)$**, care, conform ipotezei din enunţ, sunt concavi.**

În sfârşit, nu este deloc dificil să demonstrăm faptul că produsul $(\prod_{i=1}^{n} p(x_i|\theta)) \cdot p(\theta)$**, văzut ca o funcţie de argumentul** θ**, creşte monoton de la o iteraţie la alta a algoritmului EM. Practic, tot ceea ce trebuie făcut este să extindem uşor demonstraţia de la problema 2. Într-adevăr, notând**

$$Q'(\theta \mid \theta^{(t)}) = \log p(\theta) + \underbrace{\sum_{i=1}^{n} \sum_{z_i} q(z_i) \log \frac{p(x_i, z_i|\theta)}{q(z_i)}}_{Q(\theta|\theta^{(t)})},$$

relaţiile (409) **şi** (410) **devin**

$$\log P(x \mid \theta) + \log P(\theta) = Q(\theta \mid \theta^{(t)}) + CH(\theta \mid \theta^{(t)}) + \log P(\theta)$$
$$\log P(x \mid \theta^{(t)}) + \log P(\theta^{(t)}) = Q(\theta^{(t)} \mid \theta^{(t)}) + CH(\theta^{(t)} \mid \theta^{(t)}) + \log P(\theta^{(t)}).$$

Scăzându-le membru cu membru, obţinem:

$$\begin{aligned} &\log P(x \mid \theta) + \log P(\theta) - (\log P(x \mid \theta^{(t)}) + \log P(\theta^{(t)})) \\ &= Q(\theta \mid \theta^{(t)}) + \log P(\theta) - (Q(\theta^{(t)} \mid \theta^{(t)}) + \log P(\theta^{(t)})) + CH(\theta \mid \theta^{(t)}) - CH(\theta^{(t)} \mid \theta^{(t)}) \\ &= Q'(\theta \mid \theta^{(t)}) - Q'(\theta^{(t)} \mid \theta^{(t)}) + \underbrace{CH(\theta \mid \theta^{(t)}) - CH(\theta^{(t)} \mid \theta^{(t)})}_{\geq 0}. \end{aligned}$$

Concluzia este că în mod garantat valoarea funcţiei de probabilitate a posteriori a datelor observabile nu descreşte, ci fie creşte fie rămâne pe loc, la fiecare iteraţie a acestui algoritm de tip EM pentru estimare de parametri în sens MAP.

4. **(Adevărat sau Fals?)**

□ • ○ ***CMU, 2011 fall, Eric Xing, HW2, pr. 3.1.ac***

a. Algoritmul EM maximizează log-verosimilitatea datelor complete (observabile şi neobservabile).

[972] Vedeţi problema 79.c de la capitolul de *Fundamente*.

b. Funcţia optimizată de către algoritmul EM este o margine inferioară (engl., lower bound) pentru log-verosimilitatea datelor complete.

Răspuns:

a. Fals. Algoritmul EM aplică o metodă de optimizare (şi anume *coordinate-ascent*) pentru a găsi un maxim (în general, un maxim local) al funcţiei $F(q, \theta)$,[973] care este media log-verosimilităţii datelor complete (observabile şi neobservabile) şi, în acelaşi timp, o margine inferioară pentru log-verosimilitatea datelor incomplete / observabile x. (A se vedea problema 1 de la acest capitol.)

b. Fals. Funcţia F este o margine inferioară pentru log-verosimilitatea datelor incomplete / observabile.

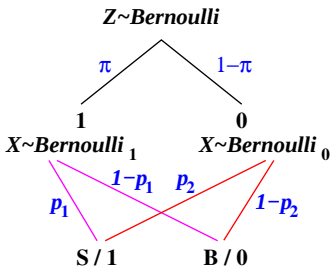
8.1.2 Mixturi de distribuţii Bernoulli / categoriale

5. **(O mixtură de distribuţii Bernoulli; estimarea unui parametru cu ajutorul algoritmului EM, folosind distribuţia binomială[974])**

formulare de Liviu Ciortuz, pornind de la CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, HW1, pr. 2.b

Presupunem că avem două monede: una este perfectă, având deci probabilitatea de apariţie a feţei cu stema $p_1 = 1/2$, iar cealaltă monedă este măsluită / imperfectă, având probabilitatea de apariţie a stemei $p_2 = 1/3$.

Facem 100 de aruncări după cum urmează. De fiecare dată alegem una dintre cele două monede. Cu o probabilitate necunoscută π, alegem moneda perfectă, iar cu probabilitatea $1-\pi$ alegem moneda măsluită. Aruncăm moneda aleasă şi notăm / reţinem rezultatul aruncării, dar nu notăm şi informaţia referiotoare la ce monedă am folosit. Constatăm la final că din totalul celor 100 de aruncări am obţinut de 40 de ori stema şi de 60 de ori banul.



***Observaţie* (1): Pe lângă cele trei metode prezentate la pr. 45 de la capitolul de *Fundamente*, pentru estimarea în sensul verosimilităţii maxime (MLE) a parametrului π al mixturii din enunţ se poate folosi şi algoritmul EM.**

a. Elaboraţi varianta / „instanţa“ algoritmului EM pentru problema de faţă, stabilind mai întâi cine sunt variabilele observabile şi respectiv variabilele neobservabile. Scrieţi expresia funcţiei care trebuie optimizată — funcţia de log-verosimilitate a datelor „observabile“ — şi expresia funcţiei auxiliare $Q(\pi|\pi^{(t)})$, care este media log-verosimilităţii datelor „complete“ (adică, „observabile“ şi

[973] Această funcţie se mai numeşte „funcţia de energie liberă“ (engl., free-energy functional).
[974] Pentru estimarea aceluiaşi parametru folosind metoda analitică, metoda gradientului şi metoda lui Newton, vedeţi ex. 45 de la capitolul *Fundamente*.

„neobservabile“) la iteraţia t, iar în final deduceţi formulele de actualizare pentru cei doi paşi (E şi M) din corpul iterativ al algoritmului.

b. Implementaţi algoritmul EM pe care l-aţi obţinut la punctul a. Comparaţi numărul de iteraţii [precum şi timpul necesar] pentru a se ajunge la convergenţă pentru cei trei algoritmi: algoritmul EM, metoda lui Newton şi metoda gradientului ascendent. Veţi folosi în toate cele trei cazuri aceeaşi valoare iniţială $\pi^{(0)} = 0.1$, precum şi aceeaşi condiţie de oprire, $|\pi^{(t+1)} - \pi^{(t)}| < 10^{-4}$.

Răspuns:

a. Asociind fiecărei „observaţii“ din cele 100 despre care este vorba în enunţ o variabilă aleatoare $X_i \sim Bernoulli(q)$ şi notând $X = (X_1, \ldots, X_{100})$, funcţia de log-verosimilitate a datelor „observabile“ va fi scrisă astfel:

$$\ell(\pi) = \ln P(X|\pi) = \ln \Big(C_{100}^{40} \underbrace{P(X_i = 1)}_{q}^{40} \underbrace{P(X_i = 0)}_{1-q}^{60} \Big) = \ln \big(C_{100}^{40} q^{40} (1-q)^{60} \big),$$

unde

$$\begin{aligned} q &\overset{not.}{=} P(X = 1) \overset{F.P.T.}{=} P(X = 1, Z = 1) + P(X = 1, Z = 0) \\ &= P(X = 1|Z = 1) \cdot P(Z = 1) + P(X = 1|Z = 0) \cdot P(Z = 0) \\ &= \frac{1}{2} \cdot \pi + \frac{1}{3} \cdot (1 - \pi) = \frac{1}{3} + \frac{\pi}{6}. \end{aligned}$$

Notând cu $\tilde{z}_1$ numărul (neobservabil) de apariţii ale stemei pentru moneda 1 (din totalul celor $n_S = 40$ de aruncări) şi cu $\tilde{z}_0$ numărul (neobservabil) de apariţii ale banului pentru moneda 1 (din totalul celor $n_B = 60$ de aruncări), log-verosimilitatea datelor „complete“ va putea fi scrisă astfel:[975]

$$\begin{aligned} \ell_{compl}(\pi) &\overset{def.}{=} \ln P(X, Z|\pi) \overset{indep.}{=} \sum_{i=1}^{100} \ln \big(P(X_i|Z_i, \pi) \cdot P(Z_i|\pi) \big) \qquad (413) \\ &= \tilde{z}_1 \ln(\pi p_1) + (n_S - \tilde{z}_1) \ln((1-\pi) p_2) \\ &\quad + \tilde{z}_0 \ln(\pi(1-p_1)) + (n_B - \tilde{z}_0) \ln((1-\pi)(1-p_2)) \\ &= \tilde{z}_1 (\ln \pi + \ln p_1) + (n_S - \tilde{z}_1)(\ln(1-\pi) + \ln p_2) \\ &\quad + \tilde{z}_0 (\ln \pi + \ln(1-p_1)) + (n_B - \tilde{z}_0)(\ln(1-\pi) + \ln(1-p_2)). \end{aligned}$$

***Observaţie* (2): În relaţia (413) am considerat variabilele aleatoare (neobservabile) Z_i luând valoarea 1 dacă „observaţia“ $X_i = x_i$ a fost realizată folosind moneda 1, şi respectiv valoarea 0 dacă s-a folosit moneda 2.**

Vom calcula funcţia „auxiliară“ $Q(\pi|\pi^{(t)})$, folosind proprietatea de liniaritate a mediilor (vedeţi pr. 9.a de la capitolul de *Fundamente*):

$$\begin{aligned} Q(\pi|\pi^{(t)}) &\overset{def.}{=} E[\ell_{compl}(\pi)] \\ &= E[\tilde{z}_1](\ln \pi + \ln p_1) + (n_S - E[\tilde{z}_1])(\ln(1-\pi) + \ln p_2) \\ &\quad + E[\tilde{z}_0](\ln \pi + \ln(1-p_1)) + (n_B - E[\tilde{z}_0])(\ln(1-\pi) + \ln(1-p_2)) \\ &= \underbrace{c}_{const.} + (E[\tilde{z}_1] + E[\tilde{z}_0]) \ln \pi + (\underbrace{n_S + n_B}_{100} - E[\tilde{z}_1] - E[\tilde{z}_0]) \ln(1-\pi). \qquad (414) \end{aligned}$$

[975] Putem considera n_S şi n_B ca fiind datele „observabile“.

Acum vom determina regulile de „actualizare“ de la cei doi paşi (Pasul E şi respectiv Pasul M) ai algoritmului EM:

<u>Pasul E:</u> Mediile $E[\tilde{z}_1]$ şi $E[\tilde{z}_0]$ se calculează ţinând cont că $\tilde{z}_1 \sim Binomial(40, q_1^{(t)})$ şi $\tilde{z}_0 \sim Binomial(60, q_0^{(t)})$, unde

$$q_1^{(t)} \overset{not.}{=} P(Z_i = 1 | X_i = 1, \pi^{(t)}) \overset{F.B., F.P.T.}{=} \frac{\pi^{(t)} p_1}{\pi^{(t)} p_1 + (1 - \pi^{(t)}) p_2}$$

$$q_0^{(t)} \overset{not.}{=} P(Z_i = 1 | X_i = 0, \pi^{(t)}) \overset{F.B., F.P.T.}{=} \frac{\pi^{(t)}(1 - p_1)}{\pi^{(t)}(1 - p_1) + (1 - \pi^{(t)})(1 - p_2)},$$

ceea ce implică[976]

$$E[\tilde{z}_1] = n_S \, q_1^{(t)} \text{ şi } E[\tilde{z}_0] = n_B \, q_0^{(t)}, \tag{415}$$

unde $n_S = 40$ şi $n_B = 60$.

<u>Pasul M:</u> Pentru a determina regula de „actualizare“ de la acest pas, vom calcula derivata parţială a funcţiei „auxiliare“ $Q(\pi|\pi^{(t)})$ în raport cu π şi apoi o vom egala cu 0:

$$\frac{\partial Q}{\partial \pi}(\pi|\pi^{(t)}) \overset{(414)}{=} \frac{E[\tilde{z}_1] + E[\tilde{z}_0]}{\pi} - \frac{n_S + n_B - E[\tilde{z}_1] - E[\tilde{z}_0]}{1 - \pi}$$

$$\frac{\partial Q}{\partial \pi}(\pi) = 0 \Leftrightarrow (1 - \cancel{\pi})(E[\tilde{z}_1] + E[\tilde{z}_0]) = \pi(n_S + n_B - \cancel{(E[\tilde{z}_1] + E[\tilde{z}_0])})$$

$$\Rightarrow \pi = \frac{E[\tilde{z}_1] + E[\tilde{z}_0]}{n_S + n_B}. \tag{416}$$

Se poate constata uşor că derivata a doua a funcţiei $Q(\pi|\pi^{(t)})$ în raport cu π este negativă pe tot domeniul ei de definiţie, ceea ce implică faptul că soluţia primei derivate (care a fost calculată mai sus) este punct de maxim pentru $Q(\pi|\pi^{(t)})$.

***Observaţi* semnificaţia rezultatului (416): $E[n_S]+E[n_B]$ reprezintă câte aruncări au fost făcute (în medie) cu moneda 1 (la iteraţia t), deci este „natural“ ca π să devină (pentru iteraţia $t+1$) raportul dintre $E[n_S] + E[n_B]$ şi numărul total de aruncări, $n_S + n_B$.**

Concluzionând, regula de actualizare pentru Pasul M al algoritmului EM pentru rezolvarea problemei de estimare a parametrului π al distribuţiei probabiliste descrise în enunţul problemei este următoarea:

$$\pi^{(t+1)} = \frac{40 q_1^{(t)} + 60 q_0^{(t)}}{100} = \frac{2 q_1^{(t)} + 3 q_0^{(t)}}{5}.$$

b. Pseudo-codul algoritmului EM pentru rezolvarea problemei de tip MLE din enunţ este următorul:

Iniţializare:
  atribuie o valoare arbitrară $\pi^{(0)}$ în intervalul $(0, 1)$ pentru parametrul π;
Corpul iterativ:
  pentru $t = 0, \ldots, T-1$ (cu T fixat în avans)
    (sau: până când $|\pi^{(t)} - \pi^{(t+1)}| < \varepsilon$, cu ε fixat în avans) execută

[976] Pentru formula de calcul a mediei distribuţiei binomiale, vedeţi pr. 25.b de la capitolul de *Fundamente*.

Pasul E:

$$q_1^{(t)} = \frac{\pi^{(t)}\, p_1}{\pi^{(t)}\, p_1 + (1-\pi^{(t)})\, p_2};$$

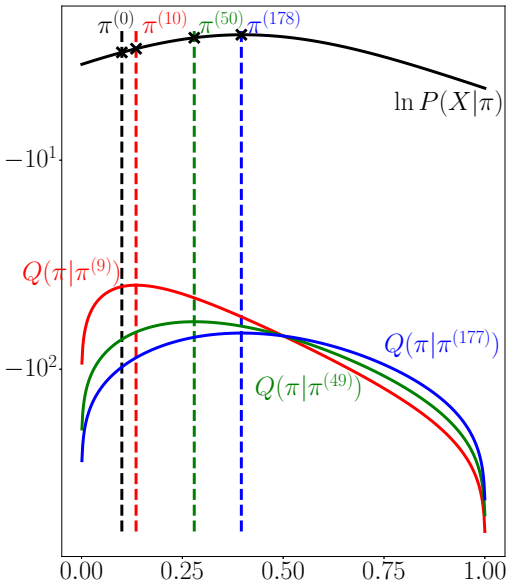
$$q_0^{(t)} = \frac{\pi^{(t)}(1-p_1)}{\pi^{(t)}(1-p_1) + (1-\pi^{(t)})(1-p_2)};$$

Pasul M:

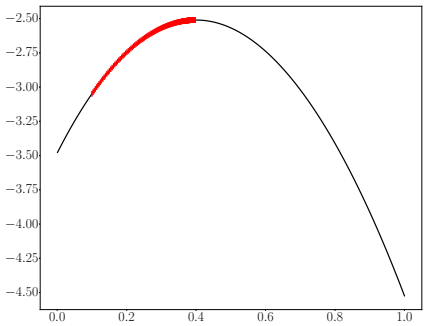
$$\pi^{(t+1)} = \frac{2q_1^{(t)} + 3q_0^{(t)}}{5};$$

Returnează $\pi^{(t+1)}$;

Graficul alăturat a fost făcut de către doctorandul Andi Munteanu. El pune în evidenţă funcţia de log-verosimilitate a datelor „observabile“, precum şi trei funcţii „auxiliare“, corespunzătoare iteraţiilor $t = 9$, $t = 49$, şi respectiv $t = 177$. Se observă convergenţa către valoarea optimă a funcţiei de log-verosimilitate a datelor „observabile“.



Graficul alăturat a fost făcut tot de către Andi Munteanu. El pune în evidenţă evoluţia algoritmului EM. Pe axa Ox a fost reprezentată valoarea probabilităţii π, iar pe axa Oy funcţia de verosimilitate. Punctul de plecare a fost $\pi^{(0)} = 0.1$. Numărul de iteraţii necesare pentru a ajunge la convergenţă (pentru $\varepsilon = 10^{-4}$) a fost 178. Timpul necesar pentru executarea acestor iteraţii a fost de 0.02597 secunde.



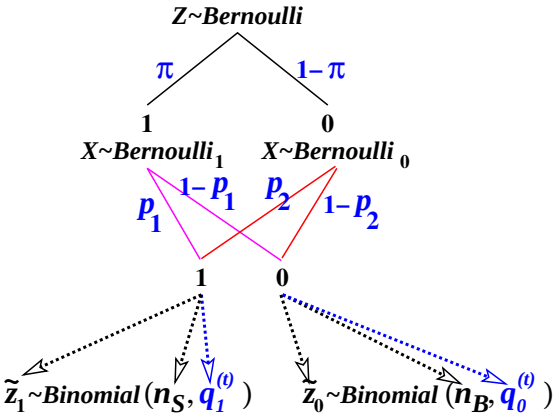
***Observaţie* (3): Pentru o comparaţie între numărul de iteraţii şi timpul necesar pentru a se ajunge la convergenţă pentru metoda gradientului şi metoda lui Newton, vedeţi pr. 45 de la capitolul *Fundamente*.**

***Observaţie* (4): Este posibil să construim un algoritm EM care să rezolve aceeaşi *problemă* (adică mixtura de distribuţii Bernoulli din enunţ), însă fără a face apel la variabilele $\tilde{z}_1$ şi $\tilde{z}_0$ care urmează distribuţii binomiale. (Aşa se lucrează la problema 6, unde se consideră însă că toţi cei trei parametri ai mixturii, şi anume π, p_1 şi p_2 sunt liberi.)**

***Observaţie importantă*: Remarcaţi următorul *fapt interesant*: Modul de estimare a parametrului / parametrilor unei mixturi de distribuţii Bernoulli cu ajutorul algoritmului EM care a fost prezentat în această problemă se bazează**

pe relaţiile (415). **Vă reamintim că** $n_H = 40$ **şi** $n_T = 60$ **sunt datele / variabilele „observabile“, iar** $\tilde{z}_1$ **şi** $\tilde{z}_0$ **sunt variabilele neobservabile.**

Ilustrăm această importantă „dependenţă“ între cele două *modele probabiliste* în figura alăturată (pentru a facilita reţinerea ei de către cititor!).

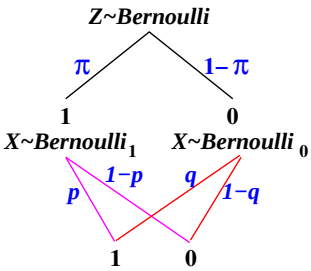


Această idee / modalitate va fi folosită şi la rezolvarea problemelor 25, 26, 27 şi 31 şi, de asemenea, într-o anumită măsură, la problemele 13, 14, 17, 32 şi 33.

6. **(Algoritmul EM: estimarea parametrilor unei mixturi de două distribuţii Bernoulli (cazul general))**

■ • ***CMU, 2008 fall, Eric Xing, HW4, pr. 1.4-7***
CMU, “The EM Algorithm,” Ajit Singh, November 20, 2005

Să presupunem că avem două monede. La aruncarea primei monede se obţine faţa ‘stemă’ cu probabilitatea p**, în vreme ce la aruncarea celei de-a doua monede se obţine ‘stemă’ cu probabilitatea** q**. Mai presupunem că se efectuează** n **aruncări, iar la fiecare aruncare se alege prima monedă cu probabilitatea** π**, iar moneda a doua cu probabilitatea** $1-\pi$**. Rezultatul fiecărei aruncări** i **este** $x_i \in \{0,1\}$**, notaţia aceasta din urmă codificând mulţimea ordonată** $\{T,H\} = \{\text{‘}tail\text{’}, \text{‘}head\text{’}\} = \{\text{‘}ban\text{’}, \text{‘}stemă\text{’}\}$**.**



$X \sim \pi Bernoulli(p) + (1-\pi) Bernoulli(q)$.

Jocul pe care ţi-l propunem este următorul:

Noi îţi furnizăm doar rezultatul celor n **aruncări, adică** $x = \{x_1, x_2, \ldots, x_n\}$**, fără a-ţi spune ce monedă am folosit pentru fiecare aruncare. Sarcina ta este următoarea: folosind algoritmul EM şi dispunând de datele „observabile“** x**, va trebui să estimezi valorile [de verosimilitate maximă] pentru parametrii probabilişti** p**,** q **şi** π**. Vom desemna ansamblul acestor parametri prin** θ**.**

Pentru a calcula aceste estimări, se va considera $z = \{z_1, z_2, \ldots, z_n\}$**, cu** $z_i \in \{0,1\}$ **variabilă „ascunsă“ indicând moneda utilizată la aruncarea** i**. Dacă, de exemplu, avem** $z_2 = 1$**, aceasta înseamnă că la aruncarea a doua a fost folosită prima monedă.**

a. Arată că $E[z_i \mid x_i, \theta] = P(z_i = 1 \mid x_i, \theta)$**.**

b. Foloseşte regula lui Bayes pentru a calcula $P(z_i = 1 \mid x_i, \theta)$ în funcţie de x_i, z_i, p, q şi π.

c. Calculează log-verosimilitatea datelor „complete", $\log P(x, z \mid \theta)$, ca funcţie de variabilele x_i şi z_i (pentru $i = 1, \ldots, n$) şi parametrii p, q şi π.

d. *Pasul E*: Arată că media log-verosimilităţii datelor complete $Q(\theta \mid \theta^{(t)}) \stackrel{not.}{=} E_{P(z|x,\theta^{(t)})}[\log P(x, z \mid \theta)]$ este dată de expresia:

$$\begin{aligned} Q(\theta \mid \theta^{(t)}) &= \sum_{i=1}^{n} E[z_i \mid x_i, \theta^{(t)}] \cdot (\log \pi + x_i \log p + (1 - x_i) \log(1 - p)) + \\ &\quad + \left(1 - E[z_i \mid x_i, \theta^{(t)}]\right) \cdot (\log(1 - \pi) + x_i \log q + (1 - x_i) \log(1 - q)), \end{aligned}$$

unde $\theta^{(t)} \stackrel{def.}{=} \{p^{(t)}, q^{(t)}, \pi^{(t)}\}$ desemnează valorile celor trei parametri la iteraţia t a algoritmului EM.

e. *Pasul M*: Elaborează formulele de calcul prin care la finalul iteraţiei t a algoritmului EM se obţin noile valori pentru parametri, $p^{(t+1)}$, $q^{(t+1)}$ şi $\pi^{(t+1)}$.

f. Scrie pseudo-codul algoritmului EM pentru rezolvarea acestui model de mixtură şi apoi execută manual o iteraţie pe inputul $x = \{1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1\}$, cu valorile iniţiale $1/3$, $2/3$ şi $1/2$ pentru parametrii p, q şi respectiv π. Ce se poate observa în privinţa convergenţei algoritmului EM?

Răspuns:

a. Este uşor de calculat $E[z_i \mid x_i, \theta]$, utilizând definiţia mediei unei variabile aleatoare. (Ştim că variabila z_i ia valori în mulţimea $\{0, 1\}$.)

$$E[z_i \mid x_i, \theta] = \sum_{z \in \{0,1\}} z_i P(z_i \mid x_i, \theta) = 0 \cdot P(z_i = 0 \mid x_i, \theta) + 1 \cdot P(z_i = 1 \mid x_i, \theta)$$
$$\Rightarrow E[z_i \mid x_i, \theta] = P(z_i = 1 \mid x_i, \theta).$$

b. Folosind regula lui Bayes şi probabilităţile din enunţ, obţinem:

$$\begin{aligned} &P(z_i = 1 \mid x_i, \theta) \\ &= \frac{P(x_i \mid z_i = 1, \theta) P(z_i = 1 \mid \theta)}{P(x_i \mid z_i = 1, \theta) P(z_i = 1 \mid \theta) + P(x_i \mid z_i = 0, \theta) P(z_i = 0 \mid \theta)} \\ &= \frac{p^{x_i} \cdot (1 - p)^{1 - x_i} \cdot \pi}{p^{x_i} \cdot (1 - p)^{1 - x_i} \cdot \pi + q^{x_i} \cdot (1 - q)^{1 - x_i} \cdot (1 - \pi)}. \end{aligned}$$

La ultima egalitate am ţinut cont că $x_i \in \{0, 1\}$. Observaţi că produsul

$$p^{x_i}(1 - p)^{1 - x_i} \tag{417}$$

exprimă simultan ambele cazuri posibile pentru $P(x_i | z_i = 1)$, adică $P(x_i = 1 | z_i = 1) = p$ şi $P(x_i = 0 | z_i = 1) = 1 - p$.[977] Similar pentru $q^{x_i}(1 - q)^{1 - x_i}$.

[977] Acesta este *artificiul exponenţierii* (engl., the exponentiation trick).

c. Pentru a exprima log-verosimilitatea datelor complete, ţinem cont de independenţa celor n aruncări:[978]

$$\begin{aligned}
\log P(x, z \mid \theta) &\overset{i.i.d.}{=} \log \prod_{i=1}^{n} P(x_i, z_i \mid \theta) = \log \prod_{i=1}^{n} P(x_i \mid z_i, \theta) \cdot P(z_i \mid \theta) \\
&= \log \prod_{i=1}^{n} \left(p^{x_i}(1-p)^{1-x_i}\pi\right)^{z_i} \left(q^{x_i}(1-q)^{1-x_i}(1-\pi)\right)^{1-z_i} \\
&= \sum_{i=1}^{n} \log \left(\left(p^{x_i}(1-p)^{1-x_i}\pi\right)^{z_i} \left(q^{x_i}(1-q)^{1-x_i}(1-\pi)\right)^{1-z_i} \right) \\
&= \sum_{i=1}^{n} \left[z_i \log \left(p^{x_i}(1-p)^{1-x_i}\pi\right) + (1-z_i) \log \left(q^{x_i}(1-q)^{1-x_i}(1-\pi)\right) \right].
\end{aligned}$$

La a treia egalitate de mai sus am ţinut cont că $z_i \in \{0, 1\}$, iar valoarea $z_i = 1$ desemnează prima monedă în vreme ce valoarea $z_i = 0$ desemnează cea de-a doua monedă.

d. Pentru a calcula media log-verosimilităţii datelor complete în raport cu distribuţia a posteriori $P(z \mid x, \theta^{(t)})$, vom ţine cont de rezultatul de la punctul c şi de proprietatea de liniaritate a mediilor.

$$\begin{aligned}
Q(\theta \mid \theta^{(t)}) &\overset{not.}{=} E_{P(z|x,\theta^{(t)})}[\log P(x, z \mid \theta)] \\
&= E_{P(z|x,\theta^{(t)})} \Big[\sum_{i=1}^{n} \big[z_i \log \left(p^{x_i}(1-p)^{1-x_i}\pi\right) + \\
&\qquad (1-z_i) \log \left(q^{x_i}(1-q)^{1-x_i}(1-\pi)\right) \big] \Big] \\
&= \sum_{i=1}^{n} \Big[E[z_i \mid x_i, \theta^{(t)}] \cdot \log \left(p^{x_i}(1-p)^{1-x_i}\right) \pi + \\
&\qquad + \left(1 - E[z_i \mid x_i, \theta^{(t)}]\right) \cdot \log \left(q^{x_i}(1-q)^{1-x_i}(1-\pi)\right) \Big] \\
&= \sum_{i=1}^{n} \Big[E[z_i \mid x_i, \theta^{(t)}] \cdot (\log \pi + x_i \log p + (1-x_i) \log(1-p)) + \\
&\qquad + \left(1 - E[z_i \mid x_i, \theta^{(t)}]\right) \cdot (\log(1-\pi) + x_i \log q + (1-x_i) \log(1-q)) \Big].
\end{aligned}$$

e. Pentru a calcula relaţiile de actualizare pentru $p^{(t+1)}$, $q^{(t+1)}$ şi $\pi^{(t+1)}$, se maximizează funcţia $Q(\theta \mid \theta^{(t)})$, care reprezintă media log-verosimilităţii datelor complete, în raport cu parametrii $\theta = (p, q, \pi)$. Mai exact, se folosesc derivatele parţiale de ordinul întâi în funcţie de p, q şi respectiv π. Pentru a spori claritatea calculelor de mai jos, vom nota cu $\mu_i^{(t)}$ media $E[z_i \mid x_i, \theta^{(t)}]$, conform calculelor de la punctele a şi b:

$$\mu_i^{(t)} = E[z_i \mid x_i, \theta^{(t)}] \overset{a,b}{=} \frac{(p^{(t)})^{x_i} \cdot (1-p^{(t)})^{1-x_i} \cdot \pi^{(t)}}{(p^{(t)})^{x_i} \cdot (1-p^{(t)})^{1-x_i} \cdot \pi^{(t)} + (q^{(t)})^{x_i} \cdot (1-q^{(t)})^{1-x_i} \cdot (1-\pi^{(t)})}.$$

[978] Din punct de vedere metodologic, pentru a calcula $P(x, z|\theta)$ ne putem inspira de la punctul precedent: observăm că la numitorul fracţiei care ne dă valoarea lui $P(z_i|x_i, \theta)$ avem $P(x_i, z_i = 1|\theta)$ şi $P(x_i, z_i = 0|\theta)$. Pornind de la aceste două expresii, ne vom propune să le exprimăm în mod unitar, adică sub forma unei singure expresii, folosind tot *artificiul exponenţierii*.

Cu această notaţie, funcţia $Q(\theta|\theta^{(t)})$, a cărei expresie a fost calculată la punctul d devine:

$$Q(\theta \mid \theta^{(t)}) = \sum_{i=1}^{n} \Big[\mu_i^{(t)} \left(\log \pi + x_i \log p + (1 - x_i) \log(1 - p)\right) + \\ + \left(1 - \mu_i^{(t)}\right) \cdot \left(\log(1 - \pi) + x_i \log q + (1 - x_i) \log(1 - q)\right)\Big].$$

Regula de *actualizare* (engl., update) pentru parametrul p se obţine astfel:

$$\frac{\partial Q(\theta \mid \theta^{(t)})}{\partial p} = 0 \Leftrightarrow \sum_{i=1}^{n} \mu_i^{(t)} \left(\frac{x_i}{p} - \frac{1 - x_i}{1 - p}\right) = 0$$

$$\Leftrightarrow \frac{1}{p} \sum_{i=1}^{n} \mu_i^{(t)} x_i = \frac{1}{1 - p} \sum_{i=1}^{n} \mu_i^{(t)} (1 - x_i) \Leftrightarrow (1 - p) \sum_{i=1}^{n} \mu_i^{(t)} x_i = p \sum_{i=1}^{n} \mu_i^{(t)} (1 - x_i)$$

$$\Leftrightarrow \sum_{i=1}^{n} \mu_i^{(t)} x_i = p \left(\sum_{i=1}^{n} \mu_i^{(t)} (1 - x_i) + \sum_{i=1}^{n} \mu_i^{(t)} x_i\right) \Leftrightarrow \sum_{i=1}^{n} \mu_i^{(t)} x_i = p \sum_{i=1}^{n} \mu_i^{(t)}$$

$$\Rightarrow p^{(t+1)} = \frac{\sum_{i=1}^{n} \mu_i^{(t)} x_i}{\sum_{i=1}^{n} \mu_i^{(t)}} \in [0, 1]. \tag{418}$$

Regula de actualizare pentru parametrul q este:

$$\frac{\partial Q(\theta \mid \theta^{(t)})}{\partial q} = 0 \Leftrightarrow \sum_{i=1}^{n} (1 - \mu_i^{(t)}) \left(\frac{x_i}{q} - \frac{1 - x_i}{1 - q}\right) = 0.$$

Se observă similaritatea cu derivata parţială în raport cu parametrul p, care a fost calculată mai sus. Aşadar, rezultă:

$$q^{(t+1)} = \frac{\sum_{i=1}^{n} (1 - \mu_i^{(t)}) x_i}{\sum_{i=1}^{n} (1 - \mu_i^{(t)})} \in [0, 1]. \tag{419}$$

Regula de actualizare pentru parametrul π se obţine astfel:

$$\frac{\partial Q(\theta \mid \theta^{(t)})}{\partial \pi} = 0 \Leftrightarrow \sum_{i=1}^{n} \left(\frac{\mu_i^{(t)}}{\pi} - \frac{1 - \mu_i^{(t)}}{1 - \pi}\right) = 0$$

$$\Leftrightarrow \frac{1}{\pi} \sum_{i=1}^{n} \mu_i^{(t)} = \frac{1}{1 - \pi} \sum_{i=1}^{n} (1 - \mu_i^{(t)}) \Leftrightarrow (1 - \pi) \sum_{i=1}^{n} \mu_i^{(t)} = \pi \sum_{i=1}^{n} (1 - \mu_i^{(t)})$$

$$\Leftrightarrow \sum_{i=1}^{n} \mu_i^{(t)} = \pi \left(\sum_{i=1}^{n} (1 - \mu_i^{(t)}) + \sum_{i=1}^{n} \mu_i^{(t)}\right) \Leftrightarrow \sum_{i=1}^{n} \mu_i^{(t)} = \pi \sum_{i=1}^{n} 1 \Leftrightarrow \sum_{i=1}^{n} \mu_i^{(t)} = n\pi$$

$$\Rightarrow \pi^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} \mu_i^{(t)} \in [0, 1]. \tag{420}$$

Se observă că toate cele trei derivate parţiale de ordin secund ale funcţiei auxiliare $Q(\theta|\theta^{(t)})$ (şi anume $\frac{\partial^2}{\partial p^2}$, $\frac{\partial^2}{\partial q^2}$ şi $\frac{\partial^2}{\partial \pi^2}$) au doar valori negative pe domeniile lor de definiţie. De asemenea, matricea hessiană (adică, matricea derivatelor parţiale de ordin secund) pentru funcţia Q este diagonală şi — datorită proprietăţii precedente — este negativ definită. În consecinţă, cele trei soluţii

aflate mai sus ($p^{(t+1)}$, $q^{(t+1)}$ şi $\pi^{(t+1)}$) corespund punctului de maxim al funcţiei „auxiliare“ $Q(\theta|\theta^{(t)})$.

***Observaţie* (1): Se constată relativ uşor că formulele** (418), (419) **şi** (420) **sunt variante probabiliste pentru formulele care ne-ar da estimările de verosimilitate maximă (MLE) pentru parametrii p, q şi π în cazul în care valorile variabilelor z_i ar fi cunoscute.**

f. Folosind rezultatele de la punctele b şi e, putem scrie imediat pseudo-codul algoritmului EM pentru rezolvare de mixturi de distribuţii Bernoulli.

Iniţializare:
  atribuie valori arbitrare ($\pi^{(0)}, p^{(0)}, q^{(0)}$ în intervalul $(0,1)$)
    pentru parametrii π, p şi respectiv q;
Corpul iterativ:
  pentru $t = 0, \ldots, T-1$ (cu T fixat în avans)
    (sau: până când log-verosimilitatea datelor observabile nu mai creşte semnificativ),
    (sau: până când $|\pi^{(t)} - \pi^{(t+1)}| < \varepsilon$, $|p^{(t)} - p^{(t+1)}| < \varepsilon$, $|q^{(t)} - q^{(t+1)}| < \varepsilon$,
     cu ε fixat în avans)
  execută
    Pasul E: pentru $i = 1, \ldots, n$, calculează

$$\mu_i^{(t)} = \frac{(p^{(t)})^{x_i} \cdot (1-p^{(t)})^{1-x_i} \cdot \pi^{(t)}}{(p^{(t)})^{x_i} \cdot (1-p^{(t)})^{1-x_i} \cdot \pi^{(t)} + (q^{(t)})^{x_i} \cdot (1-q^{(t)})^{1-x_i} \cdot (1-\pi^{(t)})};$$

    Pasul M:

$$p^{(t+1)} = \frac{\sum_{i=1}^n \mu_i^{(t)} x_i}{\sum_{i=1}^n \mu_i^{(t)}},\ q^{(t+1)} = \frac{\sum_{i=1}^n (1-\mu_i^{(t)}) x_i}{\sum_{i=1}^n (1-\mu_i^{(t)})},\ \pi^{(t+1)} = \frac{1}{n} \sum_{i=1}^n \mu_i^{(t)};$$

Returnează $\pi^{(T)}, p^{(T)}, q^{(T)}$;

La execuţia primei iteraţii a algoritmului EM pe datele $x = \{$ 1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1$\}$, făcând calculele se constată uşor că la pasul E obţinem

$$\mu_1^{(1)} = \mu_2^{(1)} = \mu_4^{(1)} = \mu_7^{(1)} = \mu_{11}^{(1)} = \mu_{12}^{(1)} = \frac{1}{3} \text{ şi } \mu_3^{(1)} = \mu_5^{(1)} = \mu_6^{(1)} = \mu_8^{(1)} = \mu_9^{(1)} = \mu_{10}^{(1)} = \frac{2}{3},$$

iar la pasul M

$$p^{(1)} = \frac{1}{3} = p^{(0)},\ q^{(1)} = \frac{2}{3} = q^{(0)} \text{ şi } \pi^{(1)} = \frac{1}{2} = \pi^{(0)}.$$

Aşadar, iniţializând parametrii p, q şi π cu valorile $p^{(0)}$, $q^{(0)}$ şi respectiv $\pi^{(0)}$, algoritmul EM va bucla!

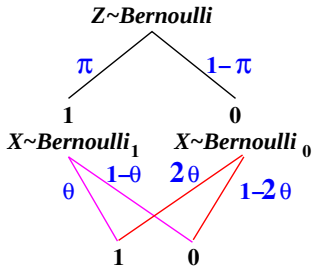
***Observaţie* (2): Se pot elabora relativ uşor două variante ale algoritmului EM care a fost dedus în această problemă. Prima dintre ele rezolvă aceeaşi mixtură de distribuţii Bernoulli din enunţ folosind însă două variabile binomiale, $\tilde{z}_1$ şi $\tilde{z}_0$ (similare cu cele de la rezolvarea problemei 5). Această variantă este mai eficientă din punct de vedere computaţional decât varianta prezentată în acest exerciţiu, fiindcă la pasul E se vor calcula doar două medii în loc de n medii. Cea de-a doua variantă extinde mixtura la K componente de tip Bernoulli. (Vedeţi problema 25.)**

7. **(Mixturi de distribuţii Bernoulli, cu şi fără variabile neobservabile: un *exemplu* de rezolvare, folosind un set simplu de date; deducerea regulilor de actualizare pentru două variante ale algoritmului EM; aplicare)**

■ □ ***L. Ciortuz, S. Ciobanu, 2020, folosind datele de la CMU, 2005 fall, T. Mitchell, A. Moore, midterm, pr. 1.3***

***Comentariu*:**
[**Estimarea parametrului unei *mixturi de distribuţii Bernoulli*, în sens MLE**]

În secţiunile A şi B, care urmează mai jos, vom face o „revizitare“ a problemei 42 de la capitolul de *Fundamente* — ea oferă un exemplu simplu de estimare a unui parametru probabilist în sensul verosimilităţii maxime —, analizând acum datele *în mod explicit* din perspectiva unei *mixturi de distribuţii Bernoulli*,[979] şi anume



$$X \sim \pi Bernoulli(\theta) + (1-\pi) Bernoulli(2\theta), \qquad (421)$$

sau, altfel spus,[980]

$$(X|Z=1) \sim Bernoulli(\theta) \text{ şi } (X|Z=0) \sim Bernoulli(2\theta), \text{ unde } Z \sim Bernoulli(\pi),$$

cu $\pi \in (0,1)$ fixat. Este util să precizăm (şi să comentăm) următoarele două situaţii posibile, în funcţie de cum este variabila Z, *observabilă* sau *neobservabilă*.

A. [MLE, când toate datele sunt observabile]

Considerând variabila Z *observabilă*, putem scrie nu doar una ci două funcţii de log-verosimilitate şi apoi putem (de fapt, în consecinţă, trebuie) să arătăm care este relaţia dintre aceste două funcţii.

	Moneda	Rezultat
i	Z_i	X_i
1	1	1 (stemă)
2	0	0 (ban)
3	0	0 (ban)
4	0	0 (ban)
5	0	1 (stemă)

Funcţia de *log-verosimilitate a datelor comune* („*completă*“) este următoarea:

$$\ell_{compl}(\theta) \overset{not.}{=} \ln P(X, Z|\theta) \overset{indep.}{=} \ln \prod_{i=1}^{5} P(X_i, Z_i|\theta) =$$

$$\ln \prod_{i=1}^{5} P(X_i|Z_i,\theta) \cdot P(Z_i|\theta) = \sum_{i=1}^{5} [\ln P(X_i|Z_i,\theta) + \ln P(Z_i|\theta)] =$$

$$\ln P(X_1=1|Z_1=1,\theta) + \ln P(Z_1=1|\theta) + \ln P(X_2=0|Z_2=0,\theta) + \ln P(Z_2=0|\theta) +$$
$$\ln P(X_3=0|Z_3=0,\theta) + \ln P(Z_3=0|\theta) + \ln P(X_4=0|Z_4=0,\theta) + \ln P(Z_4=0|\theta) +$$
$$\ln P(X_5=1|Z_5=0,\theta) + \ln P(Z_5=0|\theta) =$$
$$\ln\theta + \ln\pi + 3(\ln(1-2\theta) + \ln(1-\pi)) + \ln(2\theta) + \ln(1-\pi) =$$
$$\ln\theta + 3\ln(1-2\theta) + \ln(2\theta) + \ln\pi + 4\ln(1-\pi).$$

[979] Problema 29 de la capitolul de *Fundamente* exemplifică noţiunea de *mixtură de distribuţii categoriale*, iar problemele 113 şi 114 de la acelaşi capitol exemplifică noţiunea de *mixtură de distribuţii Bernoulli*.

[980] Veţi oberva că, pentru a obţine o formă mai convenabilă la calculele care urmează, moneda a doua a fost asociată / desemnată cu $Z = 0$.

Funcţia de *log-verosimilitate a datelor condiţionate* este:

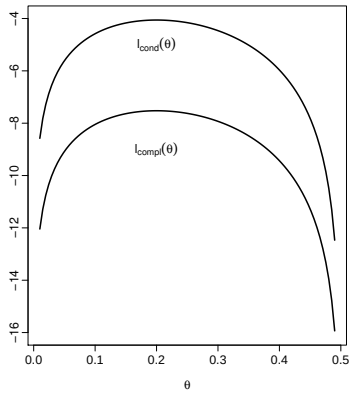
$$\ell_{cond}(\theta) \stackrel{not.}{=} \ln P(X|Z,\theta) \stackrel{indep.}{=} \ln \prod_{i=1}^{5} P(X_i|Z_i,\theta) = \ln\theta + 3\ln(1-2\theta) + \ln(2\theta).$$

identică cu funcţia de log-verosimilitate care a fost calculată la punctul a al problemei 42 de la capitolul de *Fundamente*.

Presupunând că π este fixat (şi nu depinde de θ), rezultă imediat că

$$\operatorname*{argmax}_{\theta\in(0,1/2)} \ell_{compl}(\theta) = \operatorname*{argmax}_{\theta\in(0,1/2)} \ell_{cond}(\theta).$$

pentru că $\ell_{compl}(\theta) = \ell_{cond}(\theta) + const$.
Prin urmare, în condiţiile presupunerii de mai sus, $\hat{\theta}_{MLE}$, care este estimarea de verosimilitate maximă a parametrului θ poate fi calculat folosind oricare dintre cele două funcţii de log-verosimilitate indicate mai sus.
Graficul alăturat a fost realizat pentru $\pi = 1/2$.



B. [MLE, în manieră analitică, în prezenţa unor date neobservabile[981]]

În cazul când variabila Z este *neobservabilă* (sau *latentă*, sau încă, *ascunsă*), vom opera cu log-verosimilitatea datelor „complete“ şi cu log-verosimilitatea datelor „observabile“ (sau, „incomplete“). (*Observaţie*: Veţi constata la secţiunea C că *şi* algoritmul EM operează cu aceste două funcţii de log-verosimilitate.)

	Moneda	**Rezultat**
i	Z_i	X_i
1	**?**	1 **(stemă)**
2	**?**	0 **(ban)**
3	**?**	0 **(ban)**
4	**?**	0 **(ban)**
5	**?**	1 **(stemă)**

Pentru prima dintre aceste două funcţii de log-verosimilitate, deşi definiţia rămâne aceeaşi (ca mai sus), exprimarea concretă diferă, întrucât valorile variabilelor Z_i nu mai sunt observabile.[982]

$$\begin{aligned}
\ell_{compl}(\theta) &\stackrel{not.}{=} \ln P(X=(x_1\ldots,x_5), Z=(z_1,\ldots,z_5)|\theta) \\
&\stackrel{indep.}{=} \ln\prod_{i=1}^{5} P(X_i=x_i, Z_i=z_i|\theta) = \sum_{i=1}^{5} \ln[P(X_i=x_i|Z_i=z_i,\theta)\cdot P(Z_i=z_i|\theta)] \\
&= \ln((\theta\pi)^{z_1}(2\theta(1-\pi))^{1-z_1}) + \ln(((1-\theta)\pi)^{z_2}((1-2\theta)(1-\pi))^{1-z_2}) + \\
&\quad \ln(((1-\theta)\pi)^{z_3}((1-2\theta)(1-\pi))^{1-z_3}) + \ln(((1-\theta)\pi)^{z_4}((1-2\theta)(1-\pi))^{1-z_4}) + \\
&\quad \ln((\theta\pi)^{z_5}(2\theta(1-\pi))^{1-z_5}) \\
&= z_1\ln(\theta\pi) + (1-z_1)\ln(2\theta(1-\pi)) + \\
&\quad z_2\ln((1-\theta)\pi) + (1-z_2)\ln((1-2\theta)(1-\pi)) + \\
&\quad z_3\ln((1-\theta)\pi) + (1-z_3)\ln((1-2\theta)(1-\pi)) + \\
&\quad z_4\ln((1-\theta)\pi) + (1-z_4)\ln((1-2\theta)(1-\pi)) + \\
&\quad z_5\ln(\theta\pi) + (1-z_5)\ln(2\theta(1-\pi)). \qquad (422)
\end{aligned}$$

[981] Ca şi la secţiunea A, se va considera că valoarea lui π este fixată în intervalul $(0,1)$.
[982] *Observaţie*: La cea de-a patra egalitate din deducerea expresiei pentru $\ell_{compl}(\theta)$ am folosit un mod de scriere compactă a p.m.f. pentru distribuţia Bernoulli. Este vorba despre *artificiul ridicării la putere* (engl., the exponentiation trick).

Pe lângă faptul că această scriere a funcţiei de log-verosimilitate a datelor „complete“ este *generală* în raport cu diferite valori posibile ale variabilelor Z_i, observăm că ea mai are un avantaj: ea ne permite să scriem într-un mod foarte convenabil *media* funcţiei de log-verosimilitate a datelor „complete“, aplicând proprietatea de liniaritate a mediilor,[983] adică pur şi simplu înlocuind fiecare z_i din ultima expresie de mai sus cu $E[Z_i]$.[984]

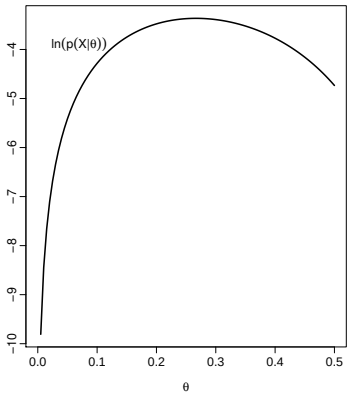
În ce priveşte log-verosimilitatea datelor „observabile“ (sau, „incomplete“) X_i, atunci când variabilele Z_i sunt *neobservabile*, putem scrie:

$$\ell_{obs}(\theta) \stackrel{def.}{=} \ln P(X|\theta) = \ln \sum_{Z} P(X, Z|\theta),$$

înţelegând prin simbolul $\sum_Z$ că se însumează probabilităţile $P(X, Z|\theta)$ obţinute pentru toate asignările posibile ale variabilelor Z. Concret, pentru exemplul de mai sus (când, precizăm din nou, variabilele Z_i sunt neobservabile), log-verosimilitatea datelor observabile se calculează astfel:

$$\begin{aligned}
\ell_{obs}(\theta) &\stackrel{def.}{=} \ln P(X|\theta) \stackrel{indep.}{=} \ln \prod_{i=1}^{5} P(X_i|\theta) = \sum_{i=1}^{5} \ln P(X_i|\theta) \\
&= \sum_{i=1}^{5} \ln \Big(\sum_{Z_i \in \{0,1\}} P(X_i, Z_i|\theta) \Big) = \sum_{i=1}^{5} \ln \Big(\sum_{Z_i \in \{0,1\}} P(X_i|Z_i, \theta) \cdot P(Z_i|\theta) \Big) \\
&= 2\ln(\pi\theta + (1-\pi)2\theta) + 3\ln(\pi(1-\theta) + (1-\pi)(1-2\theta)).
\end{aligned}$$

Valoarea parametrului θ pentru care se atinge minimul acestei funcţii se obţine rezolvând ecuaţia $\ell'_{obs}(\theta) = 0$, care depinde de valoarea care a fost atribuită lui π. De *exemplu*, pentru $\pi = 1/2$ graficul funcţiei $\ell_{obs}(\theta)$ este prezentat în figura alăturată şi vom avea:



$$\ell'_{obs}(\theta) = 0 \Leftrightarrow \frac{2}{\theta} = \frac{9}{2-3\theta} \Leftrightarrow 4 - 6\theta = 9\theta \Leftrightarrow$$
$$\theta = \frac{4}{15} = 0.2(6).$$

Se observă că această valoare aparţine intervalului $(0, 1/2)$, deci este validă.

Similar, pentru $\pi = 1/5$, care este chiar probabilitatea (estimată în sensul verosimilităţii maxime) de alegere a monedei 1 în experimentul aleator din prima parte a acestei probleme, se obţine soluţia $\theta = 2/9$. Remarcăm faptul că ea *diferă*(!) uşor de $\hat{\theta}_{MLE} = 1/5$, estimarea de verosimilitate maximă care a fost obţinută atunci când s-a considerat că variabilele Z_i sunt observabile.

***Observaţie*: Pentru aceste date — adică, pentru această mixtură de distribuţii Bernoulli — am putut calcula în mod direct (folosind o *formulă analitică*) maximul funcţiei de log-verosimilitate a datelor „observabile“. Însă pentru foarte multe probleme de rezolvare de mixturi de distribuţii probabiliste nu există o astfel de posibilitate. *Alternativa* este să folosim un algoritm de optimizare**

[983] Vedeţi problema 9 de la capitolul de *Fundamente*.

[984] Vă vom preciza / indica la momentul corespunzător care anume este distribuţia de probabilitate în raport cu care vom calcula această medie.

[iterativă], precum algoritmul EM (sau algoritmul gradientului ascendent dacă funcţia ℓ_{obs} este derivabilă sau, încă, metoda lui Newton dacă funcţia ℓ_{obs} este dublu derivabilă; vedeţi pr. 80 de la capitolul de *Fundamente*).

C. [MLE, în prezenţa unor date neobservabile, folosind algoritmul EM]

Elaboraţi regulile de actualizare pentru paşii E şi M ai algoritmului EM şi apoi aplicaţi-l la rezolvarea unei de mixturi de distribuţii Bernoulli,[985] conform datelor din secţiunea B.

Vă cerem să elaboraţi *două variante* ale acestui algoritm EM:

a. La acest punct veţi considera că probabilitatea de selecţie π este un număr fixat (în intervalul $(0,\,1)$); aşadar, în acest caz singurul parametru (liber) al algoritmului EM este θ. Vă cerem să executaţi primele două iteraţii ale acestui algoritm, pentru cazul când $\pi = 1/2$, atribuind lui θ valoarea iniţială $\theta^{(0)} = 0.45$. Veţi reprezenta grafic (folosind în acest scop R, Matlab sau limbajul dumneavoastră preferat) atât funcţia de log-verosimilitate a datelor observabile, $\ln P(x_1, \ldots, x_5|\theta)$, cât şi funcţiile auxiliare $Q(\theta|\theta^{(0)})$ şi $Q(\theta|\theta^{(1)})$.[986] Marcaţi pe grafic punctele de maxim ale funcţiilor auxiliare.

b. De adata aceasta veţi considera că probabilitatea de selecţie π este — şi ea — parametru liber (de asemenea în intervalul $(0,\,1)$); aşadar, în acest caz, parametrii algoritmului EM sunt θ şi π.

Răspuns:

Conform enunţului, vom lucra cu mixtura de distribuţii Bernoulli

$$X \sim \pi Bernoulli\,(\theta) + (1-\pi) Bernoulli\,(2\theta), \text{ cu } \theta \in (0,\,1/2).$$

Altfel spus, considerând $Z \sim Bernoulli\,(\pi)$ cu $\pi \in (0,\,1)$, vom lucra cu $X|(Z = 1) \sim Bernoulli\,(\theta)$ şi $X|(Z = 0) \sim Bernoulli\,(2\theta)$.

a. La acest punct *probabilitatea de selecţie* π este considerată *fixată*.

La *pasul E* calculăm mai întâi media variabilelor neobservabile Z_i în raport cu datele observabile x_i şi cu valoarea curentă a parametrului θ, adică $\theta^{(t)}$.

$$\begin{aligned} E[z_i|x_i,\theta^{(t)}] &= P(z_i = 1|x_i,\theta^{(t)}) \\ &\overset{F.\ Bayes}{=} \frac{\pi\,(\theta^{(t)})^{x_i}(1-\theta^{(t)})^{1-x_i}}{\pi\,(\theta^{(t)})^{x_i}(1-\theta^{(t)})^{1-x_i} + (1-\pi)(2\theta^{(t)})^{x_i}(1-2\theta^{(t)})^{1-x_i}} \overset{not.}{=} \mu_i^{(t)} \end{aligned} \quad (423)$$

Apoi calculăm log-verosimilitatea datelor complete, $\ell_{compl}(\theta)$, similar cu modul în care am procedat la relaţia (422).

$$\begin{aligned} \ell_{compl}(\theta) &\overset{not.}{=} \ln P(X, Z|\theta) \\ &= Z_1 \ln(\theta\pi) + (1-Z_1)\ln(2\theta(1-\pi)) + \\ &\quad Z_2 \ln((1-\theta)\pi) + (1-Z_2)\ln((1-2\theta)(1-\pi)) + \\ &\quad Z_3 \ln((1-\theta)\pi) + (1-Z_3)\ln((1-2\theta)(1-\pi)) + \\ &\quad Z_4 \ln((1-\theta)\pi) + (1-Z_4)\ln((1-2\theta)(1-\pi)) + \\ &\quad Z_5 \ln(\theta\pi) + (1-Z_5)\ln(2\theta(1-\pi)). \end{aligned}$$

[985] Puteţi lucra folosind formularea schemei algoritmice EM din *Observaţia* (5) de la problema 1. Alternativ, puteţi adapta algoritmul EM general pentru rezolvarea de mixturi de distribuţii Bernoulli, al cărui pseudo-cod este dat la finalul soluţiei de la problema 6; în principiu, pentru ceea ce este necesar aici, veţi ignora regulile de actualizare pentru parametrii q şi π de acolo.

[986] Aceste două funcţii auxiliare sunt mediile unor funcţii de log-verosimilitate pentru datele „complete“ (adică, atât pentru datele „observabile“ cât şi pentru datele „neobservabile“).

În sfârşit, vom calcula *funcţia auxiliară* $Q(\theta|\theta^{(t)})$, folosind relaţia pe care tocmai am obţinut-o şi proprietatea de liniaritate a mediei (vedeţi pr. 9.a de la capitolul de *Fundamente*).

$$\begin{aligned}
Q(\theta|\theta^{(t)}) &\overset{not.}{=} E[\ell_{compl}(\theta)] \\
&= E[Z_1]\ln(\theta\pi) + (1-E[Z_1])\ln(2\theta(1-\pi)) + \\
&\quad E[Z_2]\ln((1-\theta)\pi) + (1-E[Z_2])\ln((1-2\theta)(1-\pi)) + \\
&\quad E[Z_3]\ln((1-\theta)\pi) + (1-E[Z_3])\ln((1-2\theta)(1-\pi)) + \\
&\quad E[Z_4]\ln((1-\theta)\pi) + (1-E[Z_4])\ln((1-2\theta)(1-\pi)) + \\
&\quad E[Z_5]\ln(\theta\pi) + (1-E[Z_5])\ln(2\theta(1-\pi)) \qquad (424) \\
&\overset{\pi\ const.}{=} const. + E[Z_1]\ln\theta + (1-E[Z_1])\ln\theta + \\
&\quad E[Z_2]\ln(1-\theta) + (1-E[Z_2])\ln(1-2\theta) + \\
&\quad E[Z_3]\ln(1-\theta) + (1-E[Z_3])\ln(1-2\theta) + \\
&\quad E[Z_4]\ln(1-\theta) + (1-E[Z_4])\ln(1-2\theta) + \\
&\quad E[Z_5]\ln\theta + (1-E[Z_5])\ln\theta \\
&= const. + 2\ln\theta + \\
&\quad (E[Z_2]+E[Z_3]+E[Z_4])\ln(1-\theta) + (3-(E[Z_2]+E[Z_3]+E[Z_4]))\ln(1-2\theta).
\end{aligned}$$

La *pasul M*, trebuie să calculăm noua valoare a parametrului θ:

$$\theta^{(t+1)} \overset{def.}{=} \underset{\theta\in(0,1/2)}{\operatorname{argmax}}\, Q(\theta|\theta^{(t)}).$$

Pentru aceasta, vom calcula derivata funcţiei $Q(\theta|\theta^{(t)})$ în raport cu θ şi o vom egala cu 0:

$$\begin{aligned}
\frac{\partial}{\partial\theta}Q(\theta|\theta^{(t)}) = 0 \Leftrightarrow \frac{2}{\theta} &= \frac{E[Z_2]+E[Z_3]+E[Z_4]}{1-\theta} + \frac{2[3-(E[Z_2]+E[Z_3]+E[Z_4])]}{1-2\theta} \\
\Leftrightarrow 2(1-\theta)(1-2\theta) &= (E[Z_2]+E[Z_3]+E[Z_4])\theta(1-2\theta) + \\
&\quad 2(3-(E[Z_2]+E[Z_3]+E[Z_4]))\theta(1-\theta) \\
\Leftrightarrow 2(1-\theta)(1-2\theta) &= (\mu_2^{(t)}+\mu_3^{(t)}+\mu_4^{(t)})\theta(1-2\theta) + 2(3-(\mu_2^{(t)}+\mu_3^{(t)}+\mu_4^{(t)}))\theta(1-\theta) \\
\Leftrightarrow a\theta^2+b\theta+c=0,\ &\textbf{unde } a=10,\ b=-12+\mu_2^{(t)}+\mu_3^{(t)}+\mu_4^{(t)},\ c=2\in\mathbb{R}. \qquad (425)
\end{aligned}$$

Remarcaţi faptul că

$$\frac{\partial^2}{\partial\theta^2}Q(\theta|\theta^{(t)}) = -\frac{2}{\theta^2} - \frac{\overbrace{\mu_2^{(t)}+\mu_3^{(t)}+\mu_4^{(t)}}^{\geq 0}}{(1-\theta)^2} - \frac{\overbrace{4(3-(\mu_2^{(t)}+\mu_3^{(t)}+\mu_4^{(t)}))}^{\geq 0}}{(1-2\theta)^2} < 0.$$

Aşadar, $Q(\theta|\theta^{(t)})$ este o funcţie strict concavă şi, prin urmare, ea admite un punct unic de maxim în intervalul $(0,1/2)$.

Întrucât $\mu_i^{(t)} \in [0,1]$ conform relaţiei (423), rezultă că $b \in [-12,-9]$, deci discriminantul ecuaţiei de gradul al doilea din relaţia (425) este $\Delta = b^2-4ac = b^2-80 \geq 1 > 0$. Prin urmare, rădăcinile acestei ecuaţii (notate cu θ_1 şi θ_2) sunt din mulţimea numerelor reale. Folosind cunoştinţe de analiză matematică din liceu, se poate demonstra uşor că θ_1 şi θ_2 sunt pozitive.[987] Mai mult, se poate

[987] Fiindcă $S = -\frac{b}{a} > 0$ şi $P = \frac{c}{a} > 0$, unde $S \overset{not.}{=} \theta_1+\theta_2$ şi $P \overset{not.}{=} \theta_1\theta_2$.

arăta că una dintre aceste rădăcini este în intervalul $(0, 1/2)$, iar cealaltă rădăcină este mai mare decât $1/2$ sau egală cu $1/2$. În consecinţă, această ecuaţie de gradul al doilea are o singură rădăcină în intervalul $(0, 1/2)$, şi anume cea mai mică dintre θ_1 şi θ_2.[988]

Sintetizând, putem scrie:

$$\theta^{(t+1)} = \frac{-b - \sqrt{b^2 - 4ac}}{2a}. \tag{426}$$

Regulile de actualizare pentru această [primă] versiune a algoritmului EM sunt deci relaţiile (423) şi (426). Aşadar, putem scrie acum pseudo-codul algoritmului EM pentru rezolvarea acestei mixturi de distribuţii Bernoulli.

Iniţializare:
  atribuie o valoare arbitrară $\theta^{(0)}$ în intervalul $(0, 1/2)$ pentru parametrul θ;
Corpul iterativ:
  pentru $t = 0, \ldots, T-1$ (cu T fixat în avans)
    (sau: până când $|\theta^{(t)} - \theta^{(t+1)}| < \varepsilon$, cu ε fixat în avans) execută
    Pasul E:

$$\mu_i^{(t)} \overset{not.}{=} \frac{\pi\,(\theta^{(t)})^{x_i}(1-\theta^{(t)})^{1-x_i}}{\pi\,(\theta^{(t)})^{x_i}(1-\theta^{(t)})^{1-x_i} + (1-\pi)(2\theta^{(t)})^{x_i}(1-2\theta^{(t)})^{1-x_i}};$$

    Pasul M:

$$\theta^{(t+1)} = \frac{-b - \sqrt{b^2 - 4ac}}{2a}, \text{ cu}$$

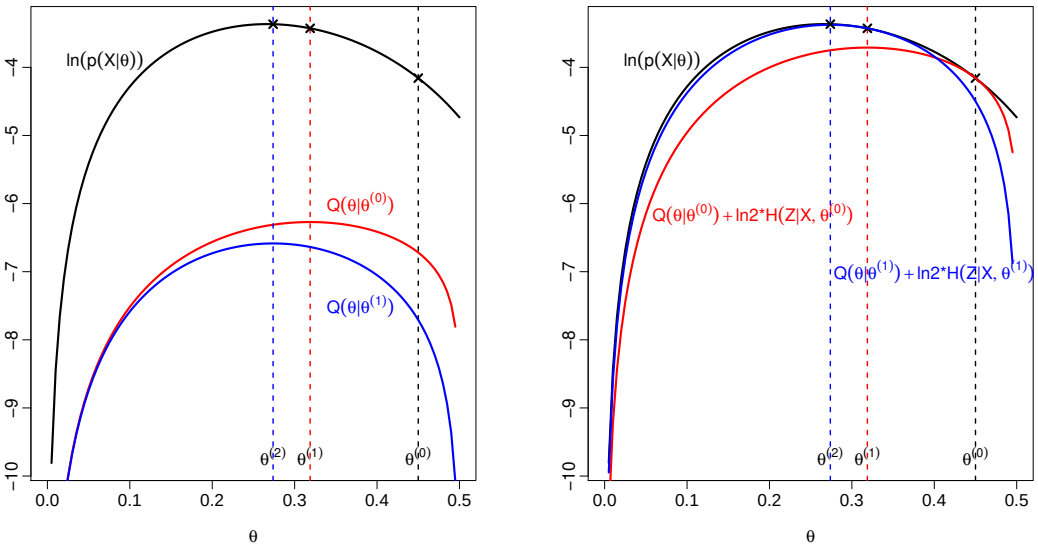
$$a = 10,\ b = -12 + \mu_2^{(t)} + \mu_3^{(t)} + \mu_4^{(t)} \text{ şi } c = 2;$$

Returnează $\theta^{(t+1)}$;

Graficul cerut în enunţ — pentru primele două iteraţii ale acestui algoritm, pentru cazul când $\pi = 1/2$ şi atribuind lui θ valoarea iniţială $\theta^{(0)} = 0.45$ – este redat mai jos, în partea stângă. În graficul din partea dreaptă am adăugat la [fiecare dintre] funcţiile auxiliare entropia distribuţiei a posteriori a variabilelor neobservabile, conform proprietăţii (410) din problema 2.[989]

[988] Fiindcă se dovedeşte imediat că $P' \leq 0$, unde $P' \overset{not.}{=} \left(\theta_1 - \frac{1}{2}\right) \cdot \left(\theta_2 - \frac{1}{2}\right)$.

[989] Pe aceste două grafice se observă uşor „tinderea“ valorilor $\theta^{(t)}$ către abscisa punctului de maxim al funcţiei de log-verosimilitate a datelor „observabile“, $\ln P(X|\theta)$.



Menţionăm că am obţinut $\theta^{(1)} = 0.3187987$ şi $\theta^{(2)} = 0.2738492$, iar valorile log-verosimilităţii datelor observabile ℓ pentru aceste valori ale lui θ sunt -3.426862 şi respectiv -3.366261 (pentru $\theta^{(0)}$, ea a fost -4.157875). Pentru $t \geq 5$, se obţine $\theta^{(t)} \approx 0.266 \approx 0.2(6)$, cât a fost calculat la secţiunea B!

***Observaţie importantă*:**

Acum dispunem de un *model probabilist* (de tip MLE), mai exact o mixtură de distribuţii Bernoulli, pentru datele din tabelul dat la începutul enunţului secţiunii B. Concret, acest model este cel obţinut înlocuind în formula (421) parametrul θ cu $\theta^{(2)}$ şi, bineînţeles, π cu $1/2$.

Ce putem face cu acest model?

Putem formula „cunoştinţe“ noi, în locul semnelor de întrebare din tabelul menţionat. Anume, pentru fiecare x_i, cu $i = 1, \ldots, 5$, putem spune care este probabilitatea ca — în modelul obţinut — x_i să fi fost generat de către prima distribuţie Bernoulli (adică, prima monedă) din mixtura (421) şi, complementar, care este probabilitatea ca x_i să fi fost generat de către a doua distribuţie Bernoulli (adică, a doua monedă) din mixtura respectivă.

Aşadar, pentru x_1 şi x_5 (ambele fiind 1, adică 'stemă'), avem:

$$\begin{aligned} P(Z_i = 1|X_i = 1, \theta^{(2)}) &= E(Z_i|X_i = 1, \theta^{(2)}) = \mu_i^{(2)} \\ &= \frac{P(X_i = 1|Z_i = 1, \theta^{(2)}) \cdot \pi}{P(X_i = 1|Z_i = 1, \theta^{(2)}) \cdot \pi + P(X_i = 1|Z_i = 0, \theta^{(2)}) \cdot (1 - \pi)} = \frac{\theta^{(2)} \cdot \frac{1}{2}}{\theta^{(2)} \cdot \frac{1}{2} + 2\theta^{(2)} \cdot \frac{1}{2}} \\ &= \frac{1}{3}. \end{aligned}$$

Aceasta este probabilitatea ca x_1 şi respectiv x_5 să fie în clusterul (de aruncări / „observaţii“) corespunzător primei monede.

Similar, pentru x_2, x_3 şi x_4 (toate fiind 0, adică 'ban'), avem:

$$\begin{aligned} P(Z_i = 1|X_i = 0, \theta^{(2)}) &= E(Z_i|X_i = 0, \theta^{(2)}) = \mu_i^{(2)} \\ &= \frac{P(X_i = 0|Z_i = 1, \theta^{(2)}) \cdot \pi}{P(X_i = 0|Z_i = 1, \theta^{(2)}) \cdot \pi + P(X_i = 0|Z_i = 0, \theta^{(2)}) \cdot (1 - \pi)} \end{aligned}$$

$$= \frac{(1-\theta^{(2)})\cdot\frac{1}{2}}{(1-\theta^{(2)})\cdot\frac{1}{2}+(1-2\theta^{(2)})\cdot\frac{1}{2}} = \frac{1-\theta^{(2)}}{2-3\theta^{(2)}} \approx 0.6162.$$

Aceasta este probabilitatea ca x_2, x_3 şi respectiv x_4 să fie în clusterul (de „observaţii") corespunzător primei monede.

Sumarizând, în tabelul de la începutul acestei secţiuni putem înlocui acum coloana Z_i (în care apar doar semnele '?') cu două coloane care corespund probabilităţilor $P(Z_i = 1|X_i = x_i, \theta^{(2)})$ şi $P(Z_i = 0|X_i = x_i, \theta^{(2)})$. Aceste probabilităţi cuantifică numeric apartenenţa "soft" a instanţelor x_i la clusterul reprezentat de prima monedă şi respectiv la clusterul reprezentat de cea de-a doua monedă.

i	$P(Z_i = 1\|X_i, \theta^{(2)})$	$P(Z_i = 0\|X_i, \theta^{(2)})$	X_i
1	1/3	2/3	1 **(stemă)**
2	0.6162	0.3838	0 **(ban)**
3	0.6162	0.3838	0 **(ban)**
4	0.6162	0.3838	0 **(ban)**
5	1/3	2/3	1 **(stemă)**

Dacă, în schimb, dorim să facem o asignare "hard" a instanţelor $x_1, \ldots, x_5$ la aceste două clustere, atunci vom proceda folosind *regula de decizie* de la clasificarea Bayesiană şi vom obţine că primul cluster va fi format din instanţele x_2, x_3 şi x_4, iar al doilea cluster va fi format din instanţele x_1 şi x_5.

b. Aşa cum s-a specificat în enunţ, aici vom elabora — în vederea aplicării pe datele noastre — varianta de algoritm EM pentru rezolvarea unei mixturi de două distribuţii Bernoulli pentru cazul când parametrul π, care reprezintă *probabilitatea de selecţie* (a primei componente dintre cele două ale mixturii), este lăsat liber în intervalul $(0, 1)$. Similar cu modul în care am procedat la punctul a, vom elabora calculele necesare pentru obţinerea *formulelor de actualizare* care vor fi folosite la cei doi paşi ai algoritmului EM, în noile condiţii.

Pasul E:
Calculăm media variabilelor neobservabile z_i în raport cu x_i şi $\theta^{(t)}$.

$$E[z_i|x_i, \theta^{(t)}, \pi^{(t)}] = P(z_i = 1|x_i, \theta^{(t)}, \pi^{(t)})$$
$$\overset{F.\ Bayes}{=} \frac{\pi^{(t)}\,(\theta^{(t)})^{x_i}(1-\theta^{(t)})^{1-x_i}}{\pi^{(t)}\,(\theta^{(t)})^{x_i}(1-\theta^{(t)})^{1-x_i} + (1-\pi^{(t)})(2\theta^{(t)})^{x_i}(1-2\theta^{(t)})^{1-x_i}} \overset{not.}{=} \mu_i^{(t)} \quad (427)$$

şi apoi funcţia auxiliară $Q(\theta, \pi|\theta^{(t)}, \pi^{(t)})$, folosind proprietatea de liniaritate a mediei şi obţinând o relaţie similară cu relaţia (424):[990]

$$\begin{aligned} Q(\theta, \pi|\theta^{(t)}, \pi^{(t)}) &\overset{not.}{=} E[\ell_{compl}(\theta, \pi)] \\ &= E[Z_1]\ln(\theta\pi) + (1-E[Z_1])\ln(2\theta(1-\pi)) + \\ &\quad E[Z_2]\ln((1-\theta)\pi) + (1-E[Z_2])\ln((1-2\theta)(1-\pi)) + \\ &\quad E[Z_3]\ln((1-\theta)\pi) + (1-E[Z_3])\ln((1-2\theta)(1-\pi)) + \\ &\quad E[Z_4]\ln((1-\theta)\pi) + (1-E[Z_4])\ln((1-2\theta)(1-\pi)) + \\ &\quad E[Z_5]\ln(\theta\pi) + (1-E[Z_5])\ln(2\theta(1-\pi)). \end{aligned}$$

[990] Veţi observa că mediile variabilelor neobservabile, $E[Z_i]$, se calculează aici în funcţie de $\theta^{(t)}$ şi $\pi^{(t)}$, în vreme ce la relaţia (424) ele se calculau doar în funcţie de $\theta^{(t)}$.

Pasul M:

$\theta^{(t+1)}$ va fi calculat folosind aceeaşi regulă de actualizare ca la punctul a, iar în ce priveşte $\pi^{(t+1)}$, reamintim că $\pi^{(t+1)} \stackrel{def.}{=} \text{argmax}_{\pi\in(0,1)} Q(\theta,\pi|\theta^{(t)},\pi^{(t)})$, deci îl vom putea calcula egalând cu 0 derivata parţială a lui $Q(\theta,\pi|\theta^{(t)},\pi^{(t)})$ în raport cu π.

$$\frac{\partial}{\partial\pi}Q(\theta,\pi|\theta^{(t)},\pi^{(t)})=0 \Leftrightarrow$$

$$\frac{\partial}{\partial\pi}\big[const.+(E[Z_1]+E[Z_2]+E[Z_3]+E[Z_4]+E[Z_5])\ln\pi+$$

$$(5-(E[Z_1]+E[Z_2]+E[Z_3]+E[Z_4]+E[Z_5]))\ln(1-\pi)\big]=0 \Leftrightarrow$$

$$\frac{E[Z_1]+E[Z_2]+E[Z_3]+E[Z_4]+E[Z_5]}{\pi}=\frac{5-(E[Z_1]+E[Z_2]+E[Z_3]+E[Z_4]+E[Z_5])}{1-\pi}$$

$$\Leftrightarrow E[Z_1]+E[Z_2]+E[Z_3]+E[Z_4]+E[Z_5]=5\pi$$

$$\Leftrightarrow \pi=\frac{1}{5}(E[Z_1]+E[Z_2]+E[Z_3]+E[Z_4]+E[Z_5]).$$

Aşadar,

$$\pi^{(t+1)}=\frac{1}{5}(\mu_1^{(t)}+\mu_2^{(t)}+\mu_3^{(t)}+\mu_4^{(t)}+\mu_5^{(t)})\in[0,1]. \tag{428}$$

Remarcaţi faptul că $Q(\theta,\pi|\theta^{(t)},\pi^{(t)})$ este funcţie concavă în raport cu parametrul π.[991] Remarcaţi de asemenea că formula (428) este varianta probabilistă a formulei care ne-ar da estimarea de verosimilitate maximă (MLE) pentru π în cazul în care toate datele ar fi observabile.

Regulile de actualizare pentru această nouă versiune (şi anume, a doua) a algoritmului EM sunt relaţiile (427) pentru pasul E, şi (426) şi (428) pentru pasul M.[992] Aşadar, putem scrie acum pseudo-codul acestei noi variante a algoritmului EM pentru rezolvarea mixturii de distribuţii Bernoulli din enunţ.

Iniţializare:
  atribuie o valoare arbitrară $\theta^{(0)}$ în intervalul $(0,1/2)$ pentru parametrul θ şi
  o valoare arbitrară $\pi^{(0)}$ în intervalul $(0,1)$ pentru parametrul π;
Corpul iterativ:
  pentru $t=0,\ldots,T-1$ (cu T fixat în avans)
    (sau: până când $|\theta^{(t)}-\theta^{(t+1)}|<\varepsilon$, cu ε fixat în avans) execută
    Pasul E:

$$\mu_i^{(t)} \stackrel{not.}{=} \frac{\pi^{(t)}\,(\theta^{(t)})^{x_i}(1-\theta^{(t)})^{1-x_i}}{\pi^{(t)}\,(\theta^{(t)})^{x_i}(1-\theta^{(t)})^{1-x_i}+(1-\pi^{(t)})(2\theta^{(t)})^{x_i}(1-2\theta^{(t)})^{1-x_i}};$$

    Pasul M:

$$\theta^{(t+1)}=\frac{-b-\sqrt{b^2-4ac}}{2a}, \text{ cu}$$

$$a=10,\ b=-12+\mu_2^{(t)}+\mu_3^{(t)}+\mu_4^{(t)} \text{ şi } c=2;$$

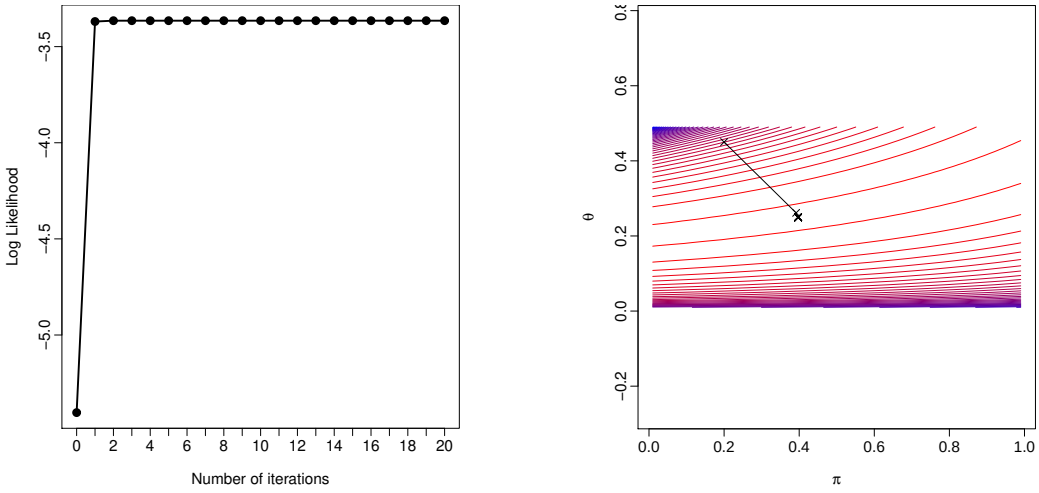
$$\pi^{(t+1)}=\frac{1}{5}(\mu_1^{(t)}+\mu_2^{(t)}+\mu_3^{(t)}+\mu_4^{(t)}+\mu_5^{(t)})\in[0,1];$$

Returnează $\theta^{(t+1)}$ şi $\pi^{(t+1)}$;

[991] De fapt, mai mult, este uşor de arătat că matricea hessiană a lui $Q(\theta,\pi|\theta^{(t)},\pi^{(t)})$ este negativ definită, deci Q admite un singur punct de maxim.

[992] Atenţie, în legătură cu regula de actualizare (426) de folosit aici: ea a fost dedusă din ecuaţia (425), în care apar mediile $\mu_i^{(t)}$. Pentru prezenta versiune a algoritmului EM, aceste medii sunt calculate cu relaţia (427).

Graficele de mai jos reprezintă evoluţia valorilor funcţiei de log-verosimilitate a datelor observabile în timpul primelor 20 de iteraţii ale acestei versiuni de algoritm EM, folosind aceleaşi date şi aceleaşi iniţializări ca la punctul precendent şi, în plus, $\pi^{(0)} = 0.2$. În graficul din partea stângă se observă o creştere substanţială a valorilor acestei funcţii chiar de la prima iteraţie, după care, în câteva iteraţii, se atinge un platou. Graficul din partea dreaptă foloseşte curbe de izocontur pentru a arăta evoluţia parametrilor θ şi π în cursul execuţiei algoritmului EM.



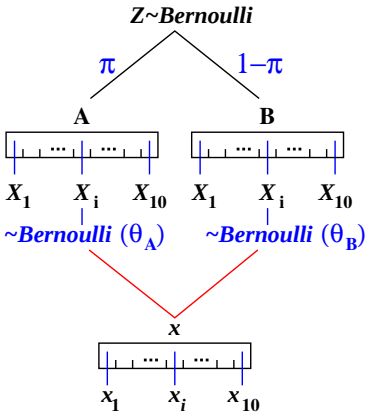
8. **(Un *exemplu* de rezolvare a unei mixturi de doi vectori de variabile Bernoulli independente şi [respectiv] identic distribuite**
A. când toate variabilele sunt observabile: MLE, în manieră directă;
B. când variabilele de selecţie sunt neobservabile: MLE, folosind algoritmul EM)

prelucrare de Liviu Ciortuz, 2015, după
■ □ ***"What is the expectation maximization algorithm?", Chuong B. Do, Serafim Batzoglou, Nature Biotechnology, vol. 26, no. 8, 2008, pag. 897-899***

Fie următorul experiment probabilist:

Dispunem de două monede, A şi B. Efectuăm 5 serii de operaţiuni de tipul următor:

- **alegem în mod aleatoriu una dintre monedele A şi B, cu probabilitate egală (1/2);**
- **aruncăm de 10 ori moneda care tocmai a fost aleasă (Z) şi notăm rezultatul, sumarizat ca număr (X) de feţe 'head' (rom., 'stemă') obţinute în urma aruncării monedei respective.**[993]



[993] Observaţi că dacă în loc de 10 aruncări am considera o singură aruncare, atunci am fi practic în condiţiile

***Observaţie* (1): Facem *presupunerea* că am reţinut succesiunea [tuturor] rezultatelor obţinute la aruncările celor două monede. Precizarea *de facto* a acestei succesiuni nu este esenţială. Dacă nu s-ar face această *presupunere*, ar trebui să lucrăm cu *distribuţia binomială*.**

A. La acest punct vom considera că s-a obţinut următorul rezultat pentru experimentul nostru:

i	Z_i	X_i
1	B	$5H$ $(5T)$
2	A	$9H$ $(1T)$
3	A	$8H$ $(2T)$
4	B	$4H$ $(6T)$
5	A	$7H$ $(3T)$

Semnificaţia variabilelor aleatoare Z_i şi X_i pentru $i = 1, \ldots, 5$ din tabelul de mai sus este imediată.

***i.* Calculaţi $\hat{\theta}_A$ şi $\hat{\theta}_B$, probabilităţile de apariţie a feţei 'head' pentru cele două monede, folosind definiţia clasică pentru probabilitatea evenimentelor aleatoare, şi anume raportul dintre numărul de cazuri favorabile şi numărul de cazuri posibile, relativ la întregul experiment.**

***ii.* Calculaţi $L_1(\theta_A, \theta_B) \stackrel{not.}{=} P(X, Z \mid \theta_A, \theta_B)$, funcţia de verosimilitate a datelor $X \stackrel{not.}{=} < X_1, \ldots, X_5 >$ şi $Z \stackrel{not.}{=} < Z_1, \ldots, Z_5 >$ în raport cu parametrii θ_A şi respectiv θ_B ai distribuţiilor (Bernoulli) care modelează aruncarea celor două monede.**

***iii.* Calculaţi $\hat{\theta}_A \stackrel{not.}{=} \operatorname{argmax}_{\theta_A} \log L_1(\theta_A, \theta_B)$ şi $\hat{\theta}_B \stackrel{not.}{=} \operatorname{argmax}_{\theta_B} \log L_1(\theta_A, \theta_B)$ cu ajutorul derivatelor parţiale de ordinul întâi. Baza logaritmului, lăsată aici nespecificată, se va considera supraunitară (de exemplu 2, e sau 10).**

***Observaţie* (2): Lucrând corect, veţi obţine acelaşi rezultat ca la punctul *i*.**

B. La acest punct se va relua experimentul de la punctul A, însă de data aceasta vom considera că valorile variabilelor Z_i nu sunt cunoscute.

i	Z_i	X_i
1	?	$5H$ $(5T)$
2	?	$9H$ $(1T)$
3	?	$8H$ $(2T)$
4	?	$4H$ $(6T)$
5	?	$7H$ $(3T)$

***iv.* Pentru convenienţă, în locul variabilelor „neobservabile" Z_i pentru $i = 1, \ldots, 5$ vom considera variabilele-indicator (de asemenea neobservabile) $Z_{i,A}$, $Z_{i,B} \in \{0, 1\}$, cu $Z_{i,A} = 1_{\{Z_i = A\}}$ şi $Z_{i,A} + Z_{i,B} = 1$.**[994]

Folosind teorema lui Bayes, calculaţi mediile variabilelor neobservabile $Z_{i,A}$ şi $Z_{i,B}$ condiţionate de variabilele observabile X_i. Veţi considera că parametrii

problemei 6, unde am prezentat algoritmul EM pentru mixturi de două distribuţii Bernoulli.
Mixtura de aici poate fi exprimată în mod analitic astfel:

$$X = (x_1, \ldots, x_{10}) \sim \pi \cdot Bernoulli\,(x_1; \theta_A) \cdot \ldots \cdot Bernoulli\,(x_{10}; \theta_A) + (1-\pi) \cdot Bernoulli\,(x_1; \theta_B) \cdot \ldots \cdot Bernoulli\,(x_{10}; \theta_B).$$

LC: Mulţumesc lui Sebastian Ciobanu pentru această observaţie.

[994] Evident, întrucât variabilele Z_i sunt aleatoare, rezultă că şi variabilele $Z_{i,A}$ şi $Z_{i,B}$ sunt aleatoare.

acestor distribuţii (Bernoulli) care modelează aruncarea monedelor A şi B au valorile iniţiale $\theta_A^{(0)} = 0.6$ şi respectiv $\theta_B^{(0)} = 0.5$.

Aşadar, se cer: $E[Z_{i,A} \mid X_i, \theta_A^{(0)} = 0.6, \theta_B^{(0)} = 0.5]$ şi $E[Z_{i,B} \mid X_i, \theta_A^{(0)} = 0.6, \theta_B^{(0)} = 0.5]$ pentru $i = 1, \ldots, 5$. Ca şi mai sus, probabilităţile a priori $P(Z_{i,A} = 1)$ şi $P(Z_{i,B} = 1)$ se vor considera egale cu $1/2$.

***v.* Calculaţi media funcţiei de log-verosimilitate a datelor „complete", X şi Z:**

$$L_2(\theta_A, \theta_B) \stackrel{def.}{=} E_{P(Z|X,\theta^{(0)})}[\log P(X, Z \mid \theta)],$$

unde $\theta = (\theta_A, \theta_B)$ şi $\theta^{(0)} = (\theta_A^{(0)}, \theta_B^{(0)})$. Semnificaţia notaţiei de mai sus este următoarea: funcţia $L_2(\theta_A, \theta_B)$ este o medie a variabilei aleatoare reprezentată de log-verosimilitatea datelor complete (observabile şi, respectiv, neobservabile), iar această medie se calculează în raport cu distribuţia probabilistă condiţională a datelor neobservabile, $P(Z \mid X, \theta^{(0)})$.

***Observaţie* (3): La elaborarea calculului, veţi folosi mai întâi proprietatea de liniaritate a mediilor variabilelor aleatoare, şi apoi rezultatele de la punctul *iv*.**

***vi.* Calculaţi $\theta_A^{(1)} \stackrel{not.}{=} \text{argmax}_{\theta_A} L_2(\theta_A, \theta_B)$ şi $\theta_B^{(1)} \stackrel{not.}{=} \text{argmax}_{\theta_B} L_2(\theta_A, \theta_B)$.**

C. Formalizaţi paşii E şi M ai algoritmului EM pentru estimarea parametrilor θ_A şi θ_B în condiţiile de la punctul B.[995]

D. Scrieţi expresia funcţiei de log-verosimilitate a datelor observabile,

$$\ln P(X|\theta_A, \theta_B) \text{ unde } X \stackrel{not.}{=} (X_1, \ldots, X_5).$$

Calculaţi valorile acestei funcţii la finalul pasului de iniţializare din algoritmul EM şi apoi la finalul execuţiei primei iteraţii. Arătaţi că

$$P(X|\theta_A^{(0)}, \theta_B^{(0)}) \leq P(X|\theta_A^{(1)}, \theta_B^{(1)}),$$

ceea ce ilustrează proprietatea care a fost demonstrată la problema 2, şi anume că de la o iteraţie la alta algoritmul EM creşte (sau cel puţin păstrează neschimbată) valoarea funcţiei de log-verosimilitate a datelor observabile.

Răspuns:

A. Acesta este un experiment probabilist în care toate variabilele sunt observabile.

***i.* Analizând datele din tabelul din enunţ, rezultă imediat $\hat{\theta}_A = \dfrac{24}{24+6} = 0.8$ şi $\hat{\theta}_B = \dfrac{9}{9+11} = 0.45$. Este de remarcat faptul că termenii 6 şi respectiv 11 de la numitorii acestor fracţii reprezintă numărul de feţe 'tail' (rom., 'ban') care au fost obţinute la aruncarea monedei A şi respectiv B: $6T = 1T + 2T + 3T$, $11T = 5T + 6T$.**

***Observaţie* (4): Dacă în locul variabilelor binare $Z_i \in \{A, B\}$ pentru $i = 1, \ldots, 5$ introducem în mod natural variabilele-indicator $Z_{i,A} \in \{0, 1\}$ şi $Z_{i,B} \in \{0, 1\}$ tot**

[995] Puteţi lucra folosind formularea schemei algoritmice EM din *Observaţia* (5) de la problema 1.

pentru $i = 1, \ldots, 5$, definite prin $Z_{i,A} = 1$ iff $Z_i = A$ şi $Z_{i,B} = 1$ iff $Z_i = B$, atunci procesările necesare pentru calculul probabilităţilor / parametrilor $\hat{\theta}_A$ şi $\hat{\theta}_B$ pot fi prezentate în mod sintetizat ca în tabelul de mai jos.[996]

i	$Z_{i,A}$	$Z_{i,B}$	X_i
1	0	1	$5H$
2	1	0	$9H$
3	1	0	$8H$
4	0	1	$4H$
5	1	0	$7H$

$\Rightarrow$

$X_i \cdot Z_{i,A}$	$X_i \cdot Z_{i,B}$
$0H \ \ (0T)$	$5H \ \ (5T)$
$9H \ \ (1T)$	$0H \ \ (0T)$
$8H \ \ (2T)$	$0H \ \ (0T)$
$0H \ \ (0T)$	$4H \ \ (6T)$
$7H \ \ (3T)$	$0H \ \ (0T)$
$\sum_{i=1}^5 X_i \cdot Z_{i,A} = 24H$	$\sum_{i=1}^5 X_i \cdot Z_{i,B} = 9H$
$\sum_{i=1}^5 (10 - X_i) \cdot Z_{i,A} = 6T$	$\sum_{i=1}^5 (10 - X_i) \cdot Z_{i,B} = 11T$

$$\Rightarrow \begin{cases} \hat{\theta}_A = \dfrac{24}{24+6} = 0.8 \\ \hat{\theta}_B = \dfrac{9}{9+11} = 0.45 \end{cases}$$

ii. **Calculul verosimilităţii datelor X şi Z:**

$$\begin{aligned} L_1(\theta_A, \theta_B) &\overset{def.}{=} P(X, Z_A, Z_B \mid \theta_A, \theta_B) = \prod_{i=1}^{5} P(X_i, Z_{i,A}, Z_{i,B} \mid \theta_A, \theta_B) \\ &\overset{indep.}{=} \prod_{i=1}^{5} P(X_i \mid Z_{i,A}, Z_{i,B}, \theta_A, \theta_B) \cdot P(Z_{i,A}, Z_{i,B} \mid \theta_A, \theta_B) \\ &= P(X_1 \mid Z_{B,1} = 1, \theta_B) \cdot 1/2 \cdot P(X_2 \mid Z_{A,2} = 1, \theta_A) \cdot 1/2 \cdot \\ &\quad P(X_3 \mid Z_{A,3} = 1, \theta_A) \cdot 1/2 \cdot P(X_4 \mid Z_{B,4} = 1, \theta_B) \cdot 1/2 \cdot \\ &\quad P(X_5 \mid Z_{A,5} = 1, \theta_A) \cdot 1/2 \\ &= \theta_B^5 (1-\theta_B)^5 \cdot \theta_A^9 (1-\theta_A) \cdot \theta_A^8 (1-\theta_A)^2 \cdot \theta_B^4 (1-\theta_B)^6 \cdot \theta_A^7 (1-\theta_A)^3 \cdot \frac{1}{2^5} \\ &= \frac{1}{2^5} \, \theta_A^{24} (1-\theta_A)^6 \, \theta_B^9 (1-\theta_B)^{11} \end{aligned}$$

iii. **Funcţia de log-verosimilitate a datelor complete se exprimă astfel:**

$$\log L_1(\theta_A, \theta_B) = -5 \log 2 + 24 \log \theta_A + 6 \log(1-\theta_A) + 9 \log \theta_B + 11 \log(1-\theta_B)$$

Prin urmare, maximul acestei funcţii în raport cu parametrul θ_A se calculează astfel:

$$\begin{aligned} &\frac{\partial \log L_1(\theta_A, \theta_B)}{\partial \theta_A} = 0 \quad \Leftrightarrow \quad \frac{\partial}{\partial \theta_A} [24 \log \theta_A + 6 \log(1-\theta_A)] = 0 \\ &\Leftrightarrow \frac{24}{\theta_A} - \frac{6}{1-\theta_A} = 0 \quad \Leftrightarrow \quad \frac{4}{\theta_A} = \frac{1}{1-\theta_A} \Leftrightarrow 4 - 4\,\theta_A = \theta_A \Leftrightarrow \hat{\theta}_A = 0.8 \end{aligned}$$

Similar, se face calculul şi pentru $\dfrac{\partial \log L_1(\theta_A, \theta_B)}{\partial \theta_B}$ şi se obţine $\hat{\theta}_B = 0.45$.[997] **Cele două valori obţinute, $\hat{\theta}_A$ şi $\hat{\theta}_B$ reprezintă estimarea de verosimilitate maximă a**

[996] Prezentăm acest „artificiu“ ca pregătire pentru rezolvarea (ulterioară a) punctului B al prezentei probleme.
[997] Se verifică uşor faptul că într-adevăr rădăcinile derivatelor parţiale de ordinul întâi pentru funcţia de log-

probabilităţilor de apariţie a feţei 'head' / stemă pentru moneda A şi respectiv moneda B.

***Observaţii*:**
5. Am arătat pe acest caz particular că metoda de calculare a probabilităţilor ($\hat{\theta}_A$ şi $\hat{\theta}_B$) direct din datele observate (aşa cum o ştim din liceu) corespunde de fapt metodei de estimare în sensul verosimităţii maxime (MLE).
6. La punctul B vom arăta cum anume se poate face estimarea aceloraşi parametri θ_A şi θ_B în cazul în care o parte din date, şi anume variabilele Z_i (pentru $i = 1, \ldots, 5$) sunt neobservabile.

B. Algoritmul EM ne permite să facem în mod iterativ estimarea parametrilor θ_A şi θ_B în funcţie de valorile variabilelor observabile, X_i, şi de valorile iniţiale atribuite parametrilor (în cazul nostru, $\theta_A^{(0)} = 0.6$ şi $\theta_B^{(0)} = 0.5$).

***Notă*:**
Vom sintetiza calculele de la prima iteraţie a algoritmului EM — care vor fi detaliate la punctele *iv*, *v* şi *vi* de mai jos — sub forma următoare, care seamănă (dar şi diferă!) într-o anumită măsură de tabelele de la punctul A:

i	$Z_{i,A}$	$Z_{i,B}$	X_i	$\overset{E}{\Longrightarrow}$	$E[Z_{i,A}]$	$E[Z_{i,B}]$	$\overset{M}{\Longrightarrow}$	$X_i \cdot E[Z_{i,A}]$	$X_i \cdot E[Z_{i,B}]$
1	?	?	$5H$		$0.45H$	$0.55H$		$2.2H$ $(2.2T)$	$2.8H$ $(2.8T)$
2	?	?	$9H$		$0.80H$	$0.20H$		$7.2H$ $(0.8T)$	$1.8H$ $(0.2T)$
3	?	?	$8H$		$0.73H$	$0.27H$		$5.9H$ $(1.5T)$	$2.1H$ $(0.5T)$
4	?	?	$4H$		$0.35H$	$0.65H$		$1.4H$ $(2.1T)$	$2.6H$ $(3.9T)$
5	?	?	$7H$		$0.65H$	$0.35H$		$4.5H$ $(1.9T)$	$2.5H$ $(1.1T)$

$$\Rightarrow \begin{cases} \sum_{i=1}^{5} X_i \cdot E[Z_{i,A}] = 21.3H \\ \sum_{i=1}^{5} (10 - X_i) \cdot E[Z_{i,A}] = 8.7T \\ \sum_{i=1}^{5} X_i \cdot E[Z_{i,B}] = 11.7H \\ \sum_{i=1}^{5} (10 - X_i) \cdot E[Z_{i,B}] = 8.3T \end{cases} \Rightarrow \begin{cases} \hat{\theta}_A^{(1)} = \dfrac{21.3}{21.3 + 8.7} \approx 0.71 \\ \hat{\theta}_B^{(1)} = \dfrac{11.7}{11.7 + 8.3} \approx 0.58 \end{cases}$$

Vom arăta că
- într-adevăr, este posibilă calcularea mediilor variabilelor neobservabile $Z_{i,A}$ şi $Z_{i,B}$, condiţionate de variabilele observabile X_i şi în funcţie de valorile asignate iniţial pentru parametrii θ_A şi θ_B;
- faţă de tabloul de sinteză de la punctul precedent, când toate variabilele erau observabile şi se calculau produsele $X_i \cdot Z_{i,A}$ şi $X_i \cdot Z_{i,B}$, aici se înlocuiesc variabilele $Z_{i,A}$ şi $Z_{i,B}$ cu mediile $E[Z_{i,A}]$ şi $E[Z_{i,B}]$ în produsele respective. De fapt, în loc să se calculeze $\sum_i X_i \cdot Z_{i,A}$ se calculează media $E[\sum_i X_i \cdot Z_{i,A}]$, şi similar pentru B.

***Observaţie importantă* (7): Pentru simplitate, în cele de mai sus (inclusiv în tabelele precedente), prin $E[Z_{i,A}]$ am notat $E[Z_{i,A} \mid X_i, \theta^{(0)}]$, iar prin $E[Z_{i,B}]$ am notat $E[Z_{i,B} \mid X_i, \theta^{(0)}]$, unde $\theta^{(0)} \overset{not.}{=} (\theta_A^{(0)}, \theta_B^{(0)})$.**

***iv.* Întrucât variabilele $Z_{i,A}$ au valori booleene (0 sau 1), rezultă că**

$$E[Z_{i,A} \,|\, X_i, \theta^{(0)}] = 0 \cdot P(Z_{i,A} = 0 \,|\, X_i, \theta^{(0)}) + 1 \cdot P(Z_{i,A} = 1 \,|\, X_i, \theta^{(0)}) = P(Z_{i,A} = 1 \,|\, X_i, \theta^{(0)})$$

verosimilitate reprezintă puncte de maxim. Pentru aceasta, se arată că matricea hessiană pentru funcţia L_1 este negativ definită. Într-adevăr, derivatele de ordinul al doilea $\dfrac{\partial^2}{\partial \theta_A^2} L_1(\theta_A, \theta_A)$ şi $\dfrac{\partial^2}{\partial \theta_B^2} L_1(\theta_A, \theta_A)$ au valori negative pe tot domeniul de definiţie, în vreme ce $\dfrac{\partial^2}{\partial \theta_A \partial \theta_B} L_1(\theta_A, \theta_B) = 0$ şi $\dfrac{\partial^2}{\partial \theta_B \theta_A} L_1(\theta_A, \theta_B) = 0$.

Probabilităţile $P(Z_{i,A} = 1 \mid X, \theta^{(0)}) = P(Z_{i,A} = 1 \mid X_i, \theta^{(0)})$, pentru $i = 1, \ldots, 5$ se pot calcula folosind teorema lui Bayes:

$$\begin{aligned} & P(Z_{i,A} = 1 \mid X_i, \theta^{(0)}) \\ = & \frac{P(X_i \,|\, Z_{i,A} = 1, \theta^{(0)}) \cdot P(Z_{i,A} = 1 | \theta^{(0)})}{P(X_i \,|\, Z_{i,A} = 1, \theta^{(0)}) \cdot P(Z_{i,A} = 1 | \theta^{(0)}) + P(X_i \,|\, Z_{i,A} = 0, \theta^{(0)}) \cdot P(Z_{i,A} = 0 | \theta^{(0)})} \\ = & \frac{P(X_i \mid Z_{i,A} = 1, \theta_A^{(0)})}{P(X_i \mid Z_{i,A} = 1, \theta_A^{(0)}) + P(X_i \mid Z_{i,B} = 1, \theta_B^{(0)})} \end{aligned}$$

S-a ţinut cont că $P(Z_{i,A} = 1 \mid \theta^{(0)}) = P(Z_{i,B} = 1 \mid \theta^{(0)}) = 1/2$ (a se vedea enunţul). De exemplu, pentru $i = 1$ vom avea:

$$\begin{aligned} E[Z_{A,1} \mid X_1, \theta^{(0)}] &= \frac{0.6^5(1-0.6)^5}{0.6^5(1-0.6)^5 + 0.5^5(1-0.5)^5} = \frac{0.24^5}{0.24^5 + 0.25^5} = \frac{1}{1 + \left(\frac{25}{24}\right)^5} \\ &\approx 0.45 \end{aligned}$$

Similar cu $E[Z_{A,1} \mid X_1, \theta^{(0)}]$ se calculează şi celelalte medii $E[Z_{i,A} \mid X_i, \theta^{(0)}]$ pentru $i = 2, \ldots, 5$ şi $E[Z_{i,B} \mid X_i, \theta^{(0)}]$ pentru $i = 1, \ldots, 5$.[998] **Am înregistrat aceste valori / medii în tabelul din mijloc din cadrul seriei formate din cele trei tabele din *Nota* de mai sus.**

v. **Media funcţiei de log-verosimilitate a datelor complete, $L_2(\theta_A, \theta_B)$, se calculează astfel:**

$$\begin{aligned} L_2(\theta_A, \theta_B) &\overset{def.}{=} E_{P(Z|X,\theta^{(0)})}[\log P(X, Z \mid \theta)] \\ &\overset{indep.}{=} E_{P(Z|X,\theta^{(0)})}[\log \prod_{i=1}^{5} P(X_i, Z_{i,A}, Z_{i,B} \mid \theta_A, \theta_B)] \\ &\overset{reg.\ de\ mult.}{=} E_{P(Z|X,\theta^{(0)})}[\log \prod_{i=1}^{5} P(X_i \mid Z_{i,A}, Z_{i,B}; \theta_A, \theta_B) \cdot P(Z_{i,A}, Z_{i,B} \mid \theta_A, \theta_B)] \end{aligned}$$

În continuare, omiţând din nou distribuţia probabilistă în raport cu care se calculează media aceasta întrucât ea poate fi subînţeleasă, vom scrie:

$$\begin{aligned} & L_2(\theta_A, \theta_B) = \\ & = E\Big[\log \prod_{i=1}^{5} (\theta_A^{Z_{i,A}})^{X_i} \cdot [(1-\theta_A)^{Z_{i,A}}]^{10-X_i} \cdot (\theta_B^{Z_{i,B}})^{X_i} \cdot [(1-\theta_B)^{Z_{i,B}}]^{10-X_i} \cdot \frac{1}{2} \Big] \\ & = E\Big[\sum_{i=1}^{5} [X_i \cdot Z_{i,A} \cdot \log \theta_A + (10 - X_i) \cdot Z_{i,A} \cdot \log(1-\theta_A) + \\ & \qquad X_i \cdot Z_{i,B} \cdot \log \theta_B + (10 - X_i) \cdot Z_{i,B} \cdot \log(1-\theta_B) - \log 2] \Big] \\ & \overset{lin.\ med.}{=} \sum_{i=1}^{5} [X_i \cdot E[Z_{i,A}] \cdot \log \theta_A + (10 - X_i) \cdot E[Z_{i,A}] \cdot \log(1-\theta_A) + \\ & \qquad X_i \cdot E[Z_{i,B}] \cdot \log \theta_B + (10 - X_i) \cdot E[Z_{i,B}] \cdot \log(1-\theta_B) - \log 2] \end{aligned}$$

[998] Se poate ţine cont că, de îndată ce s-a calculat $E[Z_{i,A} \mid X_i, \theta^{(0)}]$, se poate obţine imediat şi $E[Z_{i,B} \mid X_i, \theta^{(0)}] = 1 - E[Z_{i,A} \mid X_i, \theta^{(0)}]$, fiindcă $Z_{i,A} + Z_{i,B} = 1$.

$$= \sum_{i=1}^{5} \log[\theta_A^{X_i \cdot E[Z_{i,A}]} \cdot (1-\theta_A)^{(10-X_i)\cdot E[Z_{i,A}]} \cdot \theta_B^{X_i \cdot E[Z_{i,B}]} \cdot (1-\theta_B)^{(10-X_i)\cdot E[Z_{i,B}]} \cdot \frac{1}{2}]$$

$$= \log(\theta_A^{2.2} \cdot (1-\theta_A)^{2.2} \cdot \theta_B^{2.8} \cdot (1-\theta_B)^{2.8} \cdot \ldots \cdot \theta_A^{4.5} \cdot (1-\theta_A)^{1.9} \cdot \theta_B^{2.5} \cdot (1-\theta_B)^{1.1} \cdot \frac{1}{2^5}).$$

La ultima egalitate de mai sus, cantităţile fracţionare provin din calculele simple $X_1 \cdot E[Z_{1,A} \mid X, \theta] \approx 2.2$, $X_1 \cdot E[Z_{1,B} \mid X, \theta] \approx 2.8$, ..., $X_5 \cdot E[Z_{1,A} \mid X, \theta] \approx 4.8$, $X_5 \cdot E[Z_{1,B} \mid X, \theta] \approx 2.5$ (a se vedea tabelele din cadrul *Notei* de mai sus).

***Observaţie* (8): Comparând funcţia de [log-]verosimilitate de la partea A (vedeţi punctul *ii.*) cu cea de aici, se constată că**

– $X_i \cdot (Z_{i.A} + Z_{i,B})$ **de acolo — mai precis:** $X_i \cdot Z_{i.A}$ **sau** $X_i \cdot Z_{i,B}$**, în funcţie de valoarea lui** Z_i **— devine aici** $X_i \cdot E[Z_{i.A} + Z_{i,B}] = X_i \cdot (E[Z_{i.A}] + E[Z_{i,B}])$**,**

– $\theta_A^{X_i \cdot Z_{i,A}} \cdot \theta_B^{X_i \cdot Z_{i,B}}$ **— adică** $\theta_A^{X_i}$ **sau** $\theta_B^{X_i}$**, în funcţie de valoarea lui** Z_i **— se înlocuieşte cu** $\theta_A^{X_i \cdot E[Z_{i,A}]} \cdot \theta_B^{X_i \cdot E[Z_{i,B}]}$**.**

– $(1-\theta_A)^{(10-X_i)\cdot Z_{i,A}} \cdot (1-\theta_B)^{(10-X_i)\cdot Z_{i,B}}$ **— adică** $(1-\theta_A)^{10-X_i}$ **sau** $(1-\theta_B)^{10-X_i}$**, în funcţie de valoarea lui** Z_i **— se înlocuieşte cu** $(1-\theta_A)^{(10-X_i)\cdot E[Z_{i,A}]} \cdot (1-\theta_B)^{(10-X_i)\cdot E[Z_{i,B}]}$**.**

Aceste înlocuiri / corespondenţe, deşi dau acum o *perspectivă intuitivă* asupra acestei instanţe particulare a schemei algoritmice EM — iar aceasta se va regăsi şi în cazul altor instanţe ale aceleiaşi scheme algoritmice — se datorează în mod riguros proprietăţii de liniaritate a mediilor.

vi. **Valorile parametrilor θ_A şi θ_B pentru care se atinge maximul mediei funcţiei de log-verosimilitate a datelor complete se obţin cu ajutorul derivatelor parţiale de ordinul întâi:**[999]

$$\frac{\partial L_2(\theta_A, \theta_B)}{\partial \theta_A} = 0$$
$$\Rightarrow \frac{\partial}{\partial \theta_A}(2.2 \log \theta_A + 2.2 \log(1-\theta_A) + \ldots + 4.5 \log \theta_A + 1.9 \log(1-\theta_A)) = 0$$
$$\Rightarrow \frac{2.2}{\theta_A} - \frac{2.2}{1-\theta_A} + \ldots + \frac{4.5}{\theta_A} - \frac{1.9}{1-\theta_A} = 0 \Rightarrow \ldots \Rightarrow \theta_A^{(1)} \approx 0.71.$$

Similar, vom obţine $\theta_B^{(1)} \approx 0.58$.

C. Formulele care se folosesc în cadrul algoritmului EM pentru rezolvarea problemei date (adică estimarea parametrilor θ_A şi θ_B când variabilele Z_i sunt neobservabile) se deduc astfel:

[999] Este imediat că derivatele de ordinul al doilea $\frac{\partial^2}{\partial \theta_A^2} L_2(\theta_A, \theta_A)$ şi $\frac{\partial^2}{\partial \theta_B^2} L_2(\theta_A, \theta_A)$ au valori negative pe tot domeniul de definiţie, în vreme ce $\frac{\partial^2}{\partial \theta_A \partial \theta_B} L_2(\theta_A, \theta_A) = 0$ şi $\frac{\partial^2}{\partial \theta_B \theta_A} L_2(\theta_A, \theta_A) = 0$.

***Pasul E*:**

$$
\begin{aligned}
E[Z_{i,A} \mid X_i, \theta] &= P(Z_{i,A} = 1 \mid X_i, \theta) = P(Z_{i,A} = 1 \mid X_i, \theta) \\
&= \frac{P(X_i \mid Z_{i,A} = 1; \theta) \cdot \overbrace{P(Z_{i,A} = 1 \mid \theta)}^{1/2}}{P(X_i \mid Z_{i,A} = 1; \theta) \cdot P(Z_{i,A} = 1 \mid \theta) + P(X_i \mid Z_{i,B} = 1; \theta) \cdot \underbrace{P(Z_{i,B} = 1 \mid \theta)}_{1/2}} \\
&= \frac{P(X_i \mid Z_{i,A} = 1; \theta)}{P(X_i \mid Z_{i,A} = 1; \theta) + P(X_i \mid Z_{i,B} = 1; \theta)} \\
&= \frac{\theta_A^{X_i}(1-\theta_A)^{10-X_i}}{\theta_A^{X_i}(1-\theta_A)^{10-X_i} + \theta_B^{X_i}(1-\theta_B)^{10-X_i}}
\end{aligned}
$$

Procedând în mod similar, vom obţine:

$$
E[Z_{i,B} \mid X_i, \theta] = \frac{\theta_B^{X_i}(1-\theta_B)^{10-X_i}}{\theta_A^{X_i}(1-\theta_A)^{10-X_i} + \theta_B^{X_i}(1-\theta_B)^{10-X_i}}
$$

Notând cu

- x_i **valoarea variabilei** X_i**,**
- $\theta_A^{(t)}$ **şi respectiv** $\theta_B^{(t)}$ **estimările parametrilor** θ_A **şi** θ_B **la iteraţia** t **a algoritmului EM,**
- $p_{i,A}^{(t+1)}$ **şi respectiv** $p_{i,B}^{(t+1)}$**, mediile** $E[Z_{i,A} \mid X_i, \theta_A^{(t)}]$ **şi** $E[Z_{i,B} \mid X_i, \theta_B^{(t)}]$**,**

vom avea:

$$
\begin{aligned}
p_{i,A}^{(t+1)} &= \frac{(\theta_A^{(t)})^{x_i}(1-\theta_A^{(t)})^{10-x_i}}{(\theta_A^{(t)})^{x_i}(1-\theta_A^{(t)})^{10-x_i} + (\theta_B^{(t)})^{x_i}(1-\theta_B^{(t)})^{10-x_i}} \\
p_{i,B}^{(t+1)} &= \frac{(\theta_B^{(t)})^{x_i}(1-\theta_B^{(t)})^{10-x_i}}{(\theta_A^{(t)})^{x_i}(1-\theta_A^{(t)})^{10-x_i} + (\theta_B^{(t)})^{x_i}(1-\theta_B^{(t)})^{10-x_i}}
\end{aligned}
$$

***Pasul M*: Ca şi mai înainte, în formulele de mai jos vom nota** $E[Z_{i,A} \mid X_i, \theta^{(t)}]$ **cu** $E[Z_{i,A}]$ **şi** $E[Z_{i,B} \mid X_i, \theta^{(t)}]$ **cu** $E[Z_{i,B}]$**. Cu aceste notaţii, procedând similar cu calculul de la partea B, punctul** v**, vom avea:**

$$
L_2(\theta_A, \theta_B) = \log \prod_{i=1}^{5} \theta_A^{x_i E[Z_{i,A}]}(1-\theta_A)^{(10-x_i)E[Z_{i,A}]}\theta_B^{x_i E[Z_{i,B}]}(1-\theta_B)^{(10-x_i)E[Z_{i,B}]}
$$

Prin urmare,

$$
\begin{aligned}
&\frac{\partial}{\partial \theta_A} L_2(\theta_A, \theta_B) = 0 \Rightarrow \frac{1}{\theta_A}\sum_{i=1}^{5} x_i E[Z_{i,A}] = \frac{1}{1-\theta_A}\sum_{i=1}^{5}(10-x_i)E[Z_{i,A}] \\
\Rightarrow (1-\theta_A)\sum_{i=1}^{5} x_i E[Z_{i,A}] &= \theta_A \sum_{i=1}^{5}(10-x_i)E[Z_{i,A}] \Rightarrow \sum_{i=1}^{5} x_i E[Z_{i,A}] = 10\,\theta_A \sum_{i=1}^{5} E[Z_{i,A}] \\
&\Rightarrow \theta_A = \frac{\sum_{i=1}^{5} x_i\, E[Z_{i,A}]}{10\sum_{i=1}^{5} E[Z_{i,A}]} \quad \textbf{şi, similar, } \theta_B = \frac{\sum_{i=1}^{5} x_i\, E[Z_{i,B}]}{10\sum_{i=1}^{5} E[Z_{i,B}]}
\end{aligned}
$$

Aşadar, la pasul M al algoritmului EM vom avea:

$$\theta_A^{(t+1)} = \frac{\sum_{i=1}^5 \frac{x_i}{10}\, p_{i,A}^{(t+1)}}{\sum_{i=1}^5 p_{i,A}^{(t+1)}} \qquad \theta_B^{(t+1)} = \frac{\sum_{i=1}^5 \frac{x_i}{10}\, p_{i,B}^{(t+1)}}{\sum_{i=1}^5 p_{i,B}^{(t+1)}}$$

***Observaţie* (9): Implementând algoritmul EM cu relaţiile obţinute pentru pasul E şi pasul M, după execuţia a 10 iteraţii se vor obţine valorile $\theta_A^{(10)} \approx 0.80$ şi $\theta_B^{(10)} \approx 0.52$. Este interesant de observat că estimarea obţinută pentru parametrul θ_A este acum la acelaşi nivel cu cea obţinută prin metoda verosimilităţii maxime (MLE) în cazul observării tuturor variabilelor (0.80, vedeţi rezolvarea de la partea A, punctul i), iar estimarea obţinută pentru parametrul θ_B a coborât de la valoarea 0.58 care a fost obţinută la prima iteraţie a algoritmului EM la o valoare (0.52) care este considerabil mai apropiată de estimarea prin metoda MLE (0.45).**

D. Funcţia de verosimilitate a datelor observabile se calculează astfel:

$$\begin{aligned}
P(X|\theta_A,\theta_B) = P(X_1,\ldots,X_5|\theta_A,\theta_B) &\overset{indep.}{=} \prod_{i=1}^5 P(X_i|\theta_A,\theta_B) \\
&\overset{prob.\ marg.}{=} \prod_{i=1}^5 [\sum_{Z_i\in\{A,B\}} P(X_i,Z_i|\theta_A,\theta_B)] \\
&= \prod_{i=1}^5 [P(X_i,Z_i=A|\theta_A,\theta_B) + P(X_i,Z_i=B|\theta_A,\theta_B)] \\
&= \prod_{i=1}^5 [P(X_i|Z_i=A;\theta_A,\theta_B)\cdot \underbrace{P(Z_i=A|\theta_A,\theta_B)}_{1/2} + \\
&\qquad P(X_i|Z_i=B;\theta_A,\theta_B)\cdot \underbrace{P(Z_i=B|\theta_A,\theta_B)}_{1/2}] \\
&= \frac{1}{2^5}\prod_{i=1}^5 [P(X_i|Z_i=A;\theta_A,\theta_B) + P(X_i|Z_i=B;\theta_A,\theta_B)].
\end{aligned}$$

În consecinţă, funcţia de log-verosimilitate a datelor observabile este:

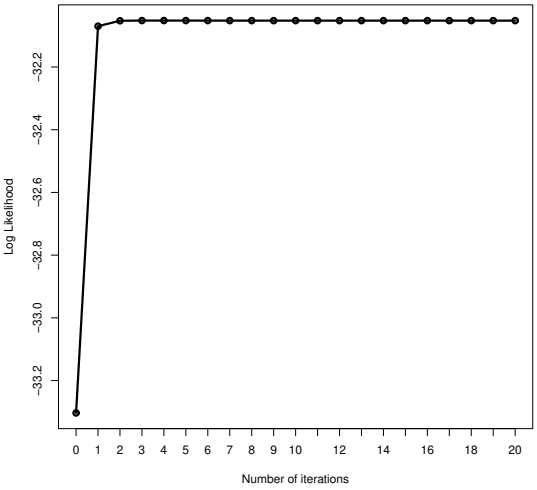
$$\begin{aligned}
&\ln P(X|\theta_A,\theta_B) \\
&= \left(\sum_{i=1}^5 \ln[P(X_i|Z_i=A;\theta_A,\theta_B) + P(X_i|Z_i=B;\theta_A,\theta_B)]\right) - 5\ln 2 \\
&= \ln[\theta_A^5(1-\theta_A)^5 + \theta_B^5(1-\theta_B)^5] + \ln[\theta_A^9(1-\theta_A) + \theta_B^9(1-\theta_B)] + \\
&\quad \ln[\theta_A^8(1-\theta_A)^2 + \theta_B^8(1-\theta_B)^2] + \ln[\theta_A^4(1-\theta_A)^6 + \theta_B^4(1-\theta_B)^6] + \\
&\quad \ln[\theta_A^7(1-\theta_A)^3 + \theta_B^7(1-\theta_B)^3] - 5\ln 2.
\end{aligned}$$

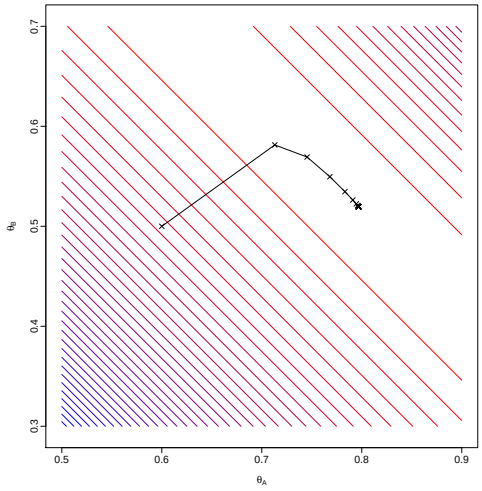
Folosind în această expresie valorile $\theta_A^{(0)} = 0.6$, $\theta_B^{(0)} = 0.5$ şi respectiv $\theta_A^{(1)} = 0.71$, $\theta_B^{(1)} = 0.58$, după ce facem calculele obţinem:

$$P(X|\theta_A^{(0)},\theta_B^{(0)}) = -33.094 \text{ şi } P(X|\theta_A^{(1)},\theta_B^{(1)}) = -31.870.$$

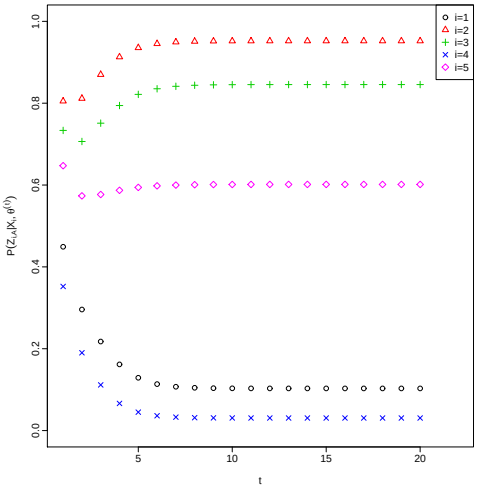
Prin urmare se verifică relaţia $P(X|\theta_A^{(0)}, \theta_B^{(0)}) \leq P(X|\theta_A^{(1)}, \theta_B^{(1)})$.

***Observaţie* (10): În graficul de mai jos, partea stângă sunt reprezentate valorile funcţiei de log-verosimilitate a datelor observabile care au fost obţinute (folosind o implementare a algoritmului EM) la iniţializare şi apoi la finalul fiecăreia din primele 20 iteraţii. În graficul din partea dreaptă avem reprezentarea sub formă de curbe de izocontur a valorilor log-verosimilităţii în funcţie de cei doi parametri, θ_A şi θ_B. Pe acest al doilea grafic a fost adăugat un „drum“ care pune în evidenţă succesiunea de valori pentru perechile $(\theta_A^{(t)}, \theta_B^{(t)})$ de-a lungul iteraţiilor executate de algoritmul EM.**[1000]





Este interesant de asemenea să observăm care este evoluţia valorilor mediilor variabilelor-indicator $Z_{i,A}$ şi $Z_{i,B}$ (adică, a probabilităţilor a posteriori de apartenenţă la clustere a instanţelor X_i). În graficul alăturat prezentăm valorile $p_{i,A}^{(t)} \overset{not.}{=} E[Z_{i,A} \mid X_i, \theta_A^{(t-1)}]$, pentru $i = 1, \ldots, 5$. (Evident, $p_{i,B}^{(t)} = 1 - p_{i,A}^{(t)}$.) Se observă că valorile acestea converg relativ repede şi, dacă la convergenţă se face trecerea la apartenenţă de tip “hard” (adică se aplică o regulă de decizie de tipul celei folosite de algoritmul Bayes Naiv), atunci se regăsesc exact etichetele care au fost „şterse“ la începutul secţiunii B, adică (în ordine) B, A, A, B, A!



[1000] Implementarea acestui algoritm EM, precum şi graficele au fost realizate de către Sebastian Ciobanu.

9. **(Algoritmul EM pentru rezolvarea unei mixturi de vectori de variabile Bernoulli independente; aplicare la clusterizarea cifrelor scrise de mână)**

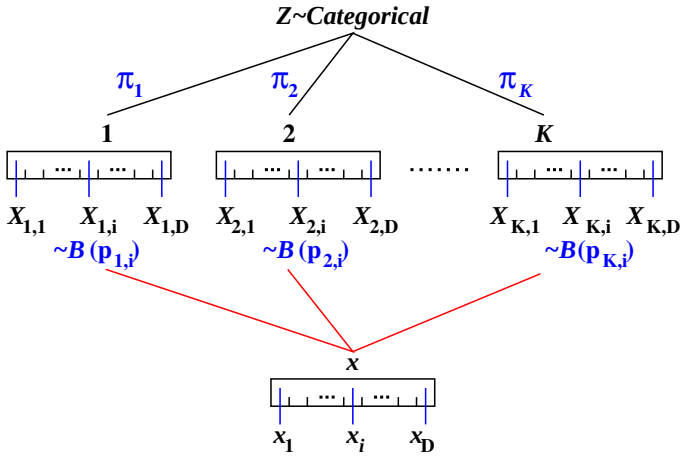
□ • ○ ***CMU, 2015 fall, A. Smola, B. Poczos, HW2, pr. 1***

În acest exerciţiu, veţi crea un algoritm de tip EM (expectation-maximization) care să clusterizeze imagini alb-negru. Input-urile $x^{(i)}$ pot fi văzute ca vectori de valori binare corespunzătoare culorilor alb şi negru. Obiectivul este să clusterizăm aceste imagini în [mai multe] grupuri. Pentru a rezolva această problemă, veţi folosi un model de tip mixtură de vectori de distribuţii Bernoulli independente.[1001]

A. Mixtura de distribuţii Bernoulli

a. Considerăm un vector de variabile aleatoare binare, $x \in \{0,1\}^D$. Presupunem că fiecare variabilă x_d (unde $d \in \{1,\ldots d\}$) urmează o distribuţie $Bernoulli(p_d)$, deci $P(x_d = 1) = p_d$. Fie $p \in (0,1)^D$ vectorul rezultat de parametri Bernoulli. Scrieţi expresia probabilităţii $P(x|p)$, considerând că variabilele x_d sunt independente între ele.

b. Să presupunem acum că avem o mixtură de K vectori de distribuţii Bernoulli, iar fiecare imagine din setul de date de clusterizat, văzută ca un vector $x^{(i)} \in (0,1)^D$, este generată folosind un vector de variabile Bernoulli independente (din cele K componente ale mixturii), având parametrul $p^{(k)} \overset{not.}{=} (p_1^{(k)}, \ldots, p_D^{(k)})$.



Presupunem de asemenea că dispunem de o distribuţie aleatoare categorială $\pi \overset{not.}{=} (\pi_1, \ldots, \pi_K)$, cu π_k indicând probabilitatea de selecţie a setului de parametri Bernoulli $p^{(k)}$, pentru $k = 1, \ldots, K$.[1002]

Folosind formula probabiltăţii totale, scrieţi expresia probabilităţii $P(x^{(i)}|p,\pi)$, unde cu p notăm (de acum încolo) ansamblul de vectori de distribuţii Bernoulli $(p^{(1)}, \ldots, p^{(K)})$.

c. Presupunem că avem inputul $X = \{x^{(1)}, \ldots, x^{(1)}\}$. Folosind rezultatele de la punctele anterioare, scrieţi expresia log-verosimilităţii datelor X, adică $\ln P(X|\pi, p)$.

[1001] Spre deosebire de modelul probabilist de la problema 8, aici variabilele Bernoulli dintr-un [acelaşi] vector nu vor mai fi identic distribuite.

[1002] Mixtura aceasta poate fi exprimată în mod analitic astfel:

$$X = (x_1, \ldots, x_D) \sim \pi_1 \cdot Bernoulli(x_1|p_{1,1}) \cdot \ldots \cdot Bernoulli(x_D|p_{1,D}) + \ldots + \pi_K \cdot Bernoulli(x_1|p_{K,1}) \cdot \ldots \cdot Bernoulli(x_D|p_{K,D}).$$

B. Pasul de estimare (engl., expectation step)

d. Acum vom introduce variabilele latente pentru algoritmul EM. Fie $z^{(i)} \in \{0,1\}^K$ un vector indicator, astfel încât $z_k^{(i)} = 1$ dacă vectorul $x^{(i)}$ a fost generat de vectorul de distribuţii $Bernoulli(p^{(k)})$, şi 0 în caz contrar. Fie $Z = \{z^{(1)}, \ldots, z^{(n)}\}$. Cât este probabilitatea $P(z^{(i)}|\pi)$? Dar $P(x^{(i)}|z^{(i)}, p, \pi)$?

***Indicaţie*: Folosiţi artificiul ridicării la putere (engl., exponentiation trick), ţinând cont de faptul că $z_k^{(i)} \in \{0,1\}$.**

e. Folosind cele două probabilităţi calculate la punctul d, scrieţi expresia verosimilităţii datelor complete, $P(X, Z|\pi, p)$.

f. Fie $\mu(z_k^{(i)}) \overset{not.}{=} E[z_k^{(i)}|x^{(i)}, \pi, p]$. Demonstraţi că

$$\mu(z_k^{(i)}) = \frac{\pi_k \prod_{d=1}^{D} (p_d^{(k)})^{x_d^{(i)}} (1 - p_d^{(k)})^{1 - x_d^{(i)}}}{\sum_{j=1}^{K} \pi_j \prod_{d=1}^{D} (p_d^{(j)})^{x_d^{(i)}} (1 - p_d^{(j)})^{1 - x_d^{(i)}}}.$$

Fie $\bar{p}$ şi $\bar{\pi}$ noile valori ale parametrilor pe care dorim să le obţinem prin maximizare, iar p şi π valorile lor de la iteraţia precedentă. Folosind aceste notaţii, arătaţi că funcţia „auxiliară", care este necesară pentru pasul M al algoritmului de estimare-maximizare, are expresia următoare:

$$\begin{aligned} & E[\ln P(X, Z|\bar{p}, \bar{\pi})|X, p, \pi] \\ &= \sum_{i=1}^{N} \sum_{k=1}^{K} \mu(z_k^{(i)}) \left[\ln \bar{\pi}_k + \sum_{d=1}^{D} \left(x_d^{(i)} \ln \bar{p}_d^{(k)} + (1 - x_d^{(i)}) \ln(1 - \bar{p}_d^{(k)}) \right) \right]. \end{aligned}$$

C. Pasul de maximizare

g. Acum trebuie să maximizăm expresia funcţiei „auxiliare" care a fost dedusă la punctul f, în raport cu $\bar{\pi}$ şi $\bar{p}$. Mai întâi, arătaţi că valoarea $\bar{p}$ care maximizează funcţia „auxiliară" este

$$\bar{p}^{(k)} = \frac{\sum_{i=1}^{n} \mu(z_k^{(i)})\, x^{(i)}}{N_k},$$

unde $N_k = \sum_{i=1}^{n} \mu(z_k^{(i)})$.

h. Demonstraţi că valoarea $\bar{\pi}$ care maximizează funcţia „auxiliară" este:

$$\bar{\pi}_k = \frac{N_k}{\sum_{k'} N_{k'}}.$$

***Sugestie*: Dat fiind faptul că este vorba de rezolvarea unei probleme de optimizare cu restricţii, puteţi apela la metoda multiplicatorilor Lagrange.**[1003]

D. Folosiţi algoritmul care a fost dedus mai sus pentru a clusteriza imagini ale cifrelor scrise de mână. Veţi utiliza setul de date MNIST.[1004] **Fiecare input este o versiune simplificată (engl., flattened) a unei imagini cu 28×28 de pixeli, iar fiecare pixel este asociat unui număr binar, care corespunde culorilor alb sau negru.**

Câteva sugestii:

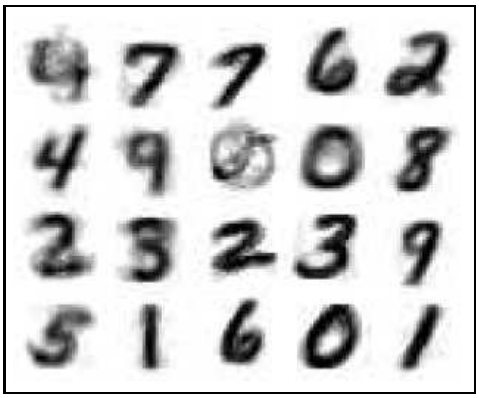
[1003] Alternativ, puteţi folosi notaţia de la familia de distribuţii exponenţiale. Vedeţi problema 41 de la capitolul de *Fundamente*.

[1004] Vedeţi https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/CMU.2015f.ASmola+BPoczos.HW2.pr2.EM4BMM.handwritten-digit-reco.data+code+sol/

- **Pentru a a sigura stablilitatea calculelor numerice, folosiţi operatorul** $\ln$**. În mod particular, fiţi atenţi la calcularea mediilor** $\mu(z_k^{(i)})$**.**
- **Va fi necesar să evitaţi ca** π **şi** p **să primească valoarea** 0**; altfel veţi obţine** $\ln 0 = -\infty$**. Pentru „netezirea" (engl., smoothing) acestor variabile, folosiţi câte o distribuţie a priori Dirichlet, de parametru** α_1 **şi respectiv** α_2**:**

$$\bar{p}^{(k)} \quad = \quad \frac{\sum_{i=1}^{N} \mu(z_k^{(i)})\, x^{(i)} + \alpha_1}{N_k + \alpha_1 D} \text{ şi } \bar{\pi}_k = \frac{N_k + \alpha_2}{\sum_{k'} N_{k'} + \alpha_2 K}.$$

- **Iniţializaţi parametrii** p **în mod aleatoriu, eşantionând cf. distribuţiei** $\mathit{Uniform}\,(0,1)$ **şi normalizând fiecare** $p^{(k)}$ **pentru a avea suma egală cu unitatea, iar** $\pi_k = 1/k$**.**
- **Rulând algoritmul pe setul de date MNIST cu** $K = 20$ **(clustere) şi** $\alpha_1 = \alpha_2 = 10^{-8}$ **timp de 20 de iteraţii, ar trebui să obţineţi rezultate similare cu următoarele:**



Pentru fiecare cluster, plasaţi probabilităţile $p^{(d)}$ **într-o matrice** 28×28 **şi imprimaţi în nuanţe de gri imaginile care rezultă.**[1005] **Ce observaţi? Explicaţi. Folosind implementarea algoritmului dumneavoastră, clusterizaţi datele. Utilizând etichetele date în fişierul** yTrain**, precizaţi câte cifre [unice] sunt în fiecare cluster. Sunt clustere în care există exact [câte] o cifră?**

Soluţie:

a. Întrucât componentele vectorului x **independente între ele, rezultă că** $P(x|p) = \prod_{d=1}^{D} P(x_d|p_d) = \prod_{d=1}^{D} [p_d^{(x_d)}\,(1-p_d)^{(1-x_d)}]$**. Am folosit „artificiul exponenţierii".**

b. Pentru fiecare $k \in \{1, \ldots, K\}$ **considerăm** A_k **evenimentul ca vectorul** $x = x^{(i)}$ **să fi fost generat de distribuţia** $\mathit{Bernoulli}\,(p^{(k)})$**. Atunci,**

$$P(x|p,\pi) = \sum_k P(x, A_k|p,\pi) = \sum_k P(x|A_k,p,\pi)\, P(A_k,p,\pi) = \sum_k P(x|p^{(k)})\pi_k.$$

[1005] Vedeţi https://www.gnu.org/software/octave/doc/interpreter/Representing-Images.html pentru cum anume poate fi imprimată o astfel de matrice, sau folosiţi funcţia furnizată show_clusters(p, a, b) care va imprima mixturile din p pe un grid $a \times b$.

c. Ţinând cont de independenţa producerii vectorilor $x^{(i)}$ pentru $i = 1, \ldots, n$, rezultă

$$\ln L(p, \pi) \stackrel{def.}{=} \ln P(X|p, \pi) = \sum_{i=1}^{n} \ln P(x^{(i)}|p, \pi) = \sum_{i=1}^{n} \ln(\sum_{k} P(x^{(i)}|p^{(k)})\pi_k).$$

d. Ca şi la punctul b, pentru $k \in \{1, \ldots, K\}$ considerăm A_k evenimentul ca vectorul $x = x^{(i)}$ să fi fost generat de distribuţia $Bernoulli(p^{(k)})$. Atunci, ţinând cont de faptul că $z_k^{(i)} \in \{0, 1\}$, putem scrie:

$$P(z^{(i)}|\pi) = \prod_{k=1}^{K} \pi_k^{z_k^{(i)}}$$

$$P(x^{(i)}|z^{(i)}, p, \pi) \stackrel{indep.}{=} \prod_{k=1}^{K} \left[P(x^{(i)}|z^{(i)}, p, \pi, A_k^{(i)})\right]^{z_k^{(i)}} = \prod_{k=1}^{K} \left[P(x^{(i)}|p^{(k)})\right]^{z_k^{(i)}}.$$

e. Verosimilitatea datelor „complete“ este:

$$\begin{aligned} P(X, Z|\pi, p) &= \prod_{i=1}^{n} P(x^{(i)}, z^{(i)}|p, \pi) = \prod_{i=1}^{n} P(x^{(i)}|z^{(i)}, p, \pi)\, P(z^{(i)}|\pi) \\ &= \prod_{i=1}^{n} \left[\prod_{k=1}^{K} [P(x^{(i)}|\pi^{(k)})]^{z_k^{(i)}}\right] \left[\prod_{k=1}^{K} \pi_k^{z_k^{(i)}}\right]. \end{aligned}$$

f. Pentru a calcula mediile $E[z_k^{(i)}|x^{(i)}, \pi, p]$, vom ţine cont mai întâi de faptul că variabilele $z_k^{(i)}$ sunt variabile-indicator şi apoi vom folosi formula lui Bayes.

$$\begin{aligned} \mu(z_k^{(i)}) &\stackrel{not.}{=} E[z_k^{(i)}|x^{(i)}, \pi, p] \\ &= P(z_k^{(i)} = 1|x^{(i)}, \pi, p) \\ &\stackrel{F.\ Bayes}{=} \frac{P(x^{(i)} = 1|z_k^{(i)} = 1, \pi, p)\, P(z_k^{(i)} = 1|\pi, p)}{\sum_{j=1}^{K} P(x^{(i)} = 1|z_j^{(i)} = 1, \pi, p)\, P(z_j^{(i)} = 1|\pi, p)} \\ &= \frac{\pi_k \prod_{d=1}^{D} (p_d^{(k)})^{x_d^{(i)}} (1 - p_d^{(k)})^{1 - x_d^{(i)}}}{\sum_{j=1}^{K} \pi_j \prod_{d=1}^{D} (p_d^{(j)})^{x_d^{(i)}} (1 - p_d^{(j)})^{1 - x_d^{(i)}}}. \end{aligned}$$

Folosind rezultatul de la punctul e, putem calcula expresia log-verosimilităţii datelor „complete“:

$$\begin{aligned} \ln P(X, Z|\bar{\pi}, \bar{p}) &= \sum_{i=1}^{n} \Big[\sum_{k=1}^{K} z_k^{(i)} \ln \big[P(x^{(i)}|\bar{p}^{(k)})\big]\Big] + \Big[\sum_{k=1}^{K} z_k^{(i)} \ln \bar{\pi}_k\Big] \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} z_k^{(i)} \big[\ln P(x^{(i)}|\bar{p}^{(k)}) + \ln \bar{\pi}_k\big] \\ &\stackrel{a.}{=} \sum_{i=1}^{n} \sum_{k=1}^{K} z_k^{(i)} \left[\ln \bar{\pi}_k + \ln \prod_{d=1}^{D} (\bar{p}_d^{(k)})^{x_d^{(i)}} (1 - \bar{p}_d^{(k)})^{1 - x_d^{(i)}}\right] \\ &= \sum_{i=1}^{n} \sum_{k=1}^{K} z_k^{(i)} \left[\ln \bar{\pi}_k + \sum_{d=1}^{D} \Big(x_d^{(i)} \ln \bar{p}_d^{(k)} + (1 - x_d^{(i)}) \ln(1 - \bar{p}_d^{(k)})\Big)\right]. \end{aligned}$$

Aplicând acestei log-verosimilităţi operatorul medie (E), apoi ţinând cont de proprietatea de liniaritate a mediei şi, în final, înlocuind $E[z_k^{(i)}]$ cu $\mu(z_k^{(i)})$, vom ajunge la soluţia indicată în enunţ.

g. Vom egala cu 0 derivata parţială a funcţei „auxiliare“ în raport cu $\bar{p}_d^{(k)}$:

$$\frac{\partial}{\partial \bar{p}_d^{(k)}} E[\ln P(X, Z|\bar{\pi}, \bar{p})] = \sum_{i=1}^{n} \mu(z_k^{(i)}) \left[\frac{x_d^{(i)}}{\bar{p}_d^{(k)}} - \frac{1 - x_d^{(i)}}{1 - \bar{p}_d^{(k)}} \right] = 0.$$

Aducând la acelaşi numitor, rezultă:

$$\sum_{i=1}^{n} \mu(z_k^{(i)}) \left[x_d^{(i)}(1 - \bar{p}_d^{(k)}) - (1 - x_d^{(i)})\bar{p}_d^{(k)} \right] = \sum_{i=1}^{n} \mu(z_k^{(i)}) \left[x_d^{(i)} - \bar{p}_d^{(k)} \right] = 0.$$

Scoţându-l pe $\bar{p}_d^{(k)}$ din această ecuaţie, vom obţine:

$$\bar{p}_d^{(k)} = \frac{\sum_{i=1}^{n} \mu(z_k^{(i)})\, x_d^{(i)}}{\sum_{i=1}^{n} \mu(z_k^{(i)})} = \frac{\sum_{i=1}^{n} \mu(z_k^{(i)})\, x_d^{(i)}}{N_k}.$$

Se poate demonstra relativ uşor că valorile $\bar{p}_d^{(k)}$ sunt corespunzătoare unui (unic) punct de maxim al funcţiei „auxiliare“.

h. Este suficient să minimizăm $-\sum_{i=1}^{n} \sum_{k=1}^{K} \mu(z_k^{(i)}) \ln \bar{\pi}_k$, fiindcă restul termenilor din expresia funcţiei „auxiliare“ nu depind de $\bar{\pi}$. Pentru a ne asigura că $\bar{\pi}$ corespunde unei distribuţii probabiliste, trebuie să impunem restricţia $\sum_k \bar{\pi}_k = 1$. Fie λ variabila Lagrange corespunzătoare acestei restricţii. Lagrangeanul generalizat va avea expresia următoare:

$$L(\bar{\pi}, \lambda) = -\sum_{i=1}^{n} \sum_{k=1}^{K} \mu(z_k^{(i)}) \ln \bar{\pi}_k + \lambda \Big(\sum_{k=1}^{K} \bar{\pi}_k - 1 \Big).$$

Calculând derivata parţială în raport cu $\bar{\pi}_k$ şi egalând-o apoi cu 0, vom obţine:

$$\frac{\partial}{\partial \bar{\pi}_k} L(\bar{\pi}, \lambda) = -\sum_{i=1}^{n} \frac{\mu(z_k^{(i)})}{\bar{\pi}_k} + \lambda = 0.$$

Din această ecuaţie vom obţine pentru $\bar{\pi}_k$ valoarea următoare:

$$\bar{\pi}_k = \frac{\sum_{i=1}^{n} \mu(z_k^{(i)})}{\lambda} = \frac{N_k}{\lambda}.$$

Întrucât $\sum_{k=1}^{K} \bar{\pi}_k = 1$, rezultă imediat că $\lambda = \sum_{k=1}^{K} N_k$ şi, înlocuid această valoare în expresia de mai sus pentru $\bar{\pi}_k$, obţinem

$$\bar{\pi}_k = \frac{N_k}{\sum_{k'} N_{k'}}.$$

Putem scrie acum pseudo-codul algoritmului EM pentru rezolvarea acestei mixturi de vectori de distribuţii Bernoulli independente.

Inițializare:
atribuie o valori arbitrare în intervalul $(0, 1)$ pentru parametrii $\pi_1, \ldots, \pi_K$ și $p^{(1)}, \ldots, p^{(K)}$;
Corpul iterativ:
pentru $t = 0, \ldots, T-1$ (cu T fixat în avans)
(sau: până când log-verosimilitatea datelor observabile nu mai crește semnificativ),
(sau...)
execută
Pasul E: pentru $i = 1, \ldots, n$ și $k = 1, \ldots, K$ calculează

$$\mu(z_k^{(i)}) = \frac{\pi_k \prod_{d=1}^{D} (p_d^{(k)})^{x_d^{(i)}} (1 - p_d^{(k)})^{1 - x_d^{(i)}}}{\sum_{j=1}^{K} \pi_j \prod_{d=1}^{D} (p_d^{(j)})^{x_d^{(i)}} (1 - p_d^{(j)})^{1 - x_d^{(i)}}};$$

Pasul M: calculează

$$\bar{p}^{(k)} = \frac{\sum_{i=1}^{n} \mu(z_k^{(i)})\, x^{(i)}}{N_k}, \text{ cu } N_k = \sum_{i=1}^{n} \mu(z_k^{(i)}), \text{ pentru } k = 1, \ldots, K;$$

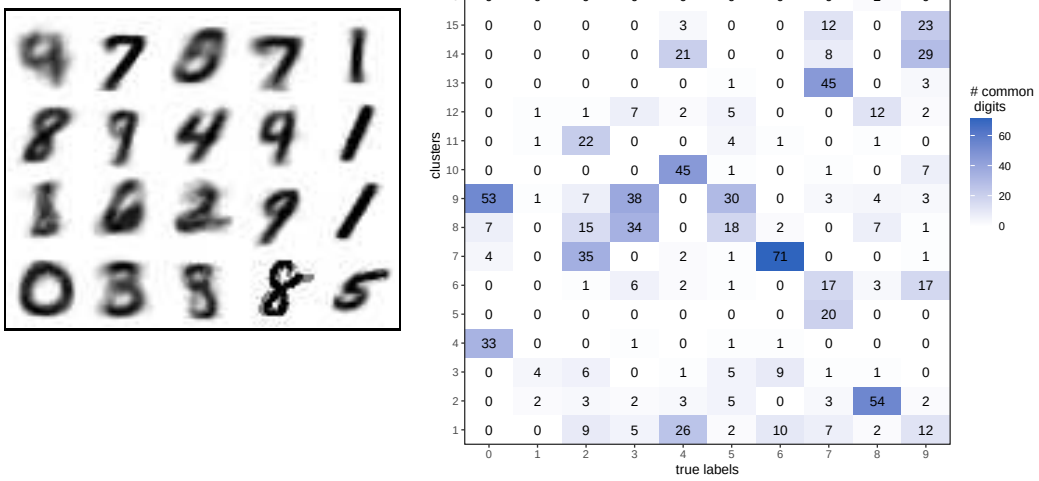
$$\bar{\pi}_k = \frac{N_k}{\sum_{k'} N_{k'}}, \text{ pentru } k = 1, \ldots, K;$$

$$\pi_k \leftarrow \bar{\pi}_k, \text{ pentru } k = 1, \ldots, K;$$

$$p^{(k)} \leftarrow \bar{p}^{(k)}, \text{ pentru } k = 1, \ldots, K;$$

Returnează $\pi_1, \ldots, \pi_K$ și $p^{(1)}, \ldots, p^{(K)}$;

D. După executarea a 100 de iteraţii, rezultatul clusterizării este cel ilustrat în figura de mai jos, partea stângă. (În această figură, imaginile corespunzătoare clusterelor sunt înşiruite pe coloane.) În partea dreaptă este redată *matricea de confuzie*.[1006]



Listăm aici în ordinea clusterelor, numărul de cifre [unice] prezente în fiecare cluster: 8, 8, 7, 4, 1, 7, 6, 7, 8, 4, 4, 7, 3, 3, 3, 1, 1, 1, 1, 2. Clusterele care conţin câte o singură cifră sunt clusterele 5, 16, 17, 18 şi 19. Clusterele 17, 18 şi 19 conţin toate doar cifra 1 şi corespund unor modalităţi uşor diferite de

[1006] Graficele au fost realizate de către studentul Andi Munteanu (2022).

ortografiere a acestei cifre. Similar, clusterele 5 şi 13 corespund unor modalităţi diferite de ortografiere a cifrei 7. Listăm acum, în ordinea clusterelor, cifra care are cele mai multe apariţii în interiorul clusterului respectiv: 4, 8, 6, 0, 7, 7 (şi 9), 6, 3, 0, 4, 2, 8, 7, 9, 9, 8, 1, 1, 1, 5. Cele mai „confuze" par a fi clusterele 1, 3, 6, 9, 12 şi 14. Pentru celelalte clustere se distinge destul de bine care este cifra care corespunde cel mai bine clusterului respectiv.

10. **(Algoritmul EM pentru estimarea parametrilor unei mixturi de două distribuţii probabiliste categoriale; elaborare / ilustrare într-un un caz particular[1007])**

prelucrare de Alina Munteanu şi Liviu Ciortuz, după CMU, 2009 spring, Ziv Bar-Joseph, HW5, pr. 1

Un student ia în fiecare dimineaţă autobuzul ca să vină la universitate. Dacă studentul ia autobuzul 71C, probabilitatea de a găsi un scaun liber lângă geam este μ_{11}, probabilitatea de a găsi un scaun liber lângă interval este μ_{12}, iar probabilitatea ca să călătorească în picioare este μ_{13}, unde $\mu_{11}+\mu_{12}+\mu_{13}=1$. În schimb, dacă studentul ia autobuzul 500, probabilităţile corespunzătoare sunt μ_{21}, μ_{22}, μ_{23}, cu $\mu_{21}+\mu_{22}+\mu_{23}=1$. Probabilitatea de a lua autobuzul 71C este β_1, iar probabilitatea de a lua autobuzul 500 este β_2, cu $\beta_1+\beta_2=1$.

În timpul a n deplasări pe care le-a făcut la universitate, studentul şi-a notat de fiecare dată *poziţia* p_i pe care a ocupat-o în autobuz — pe un scaun de lângă geam (1), pe un scaun de lângă interval (2) şi respectiv în picioare (3) — văzută ca valoare a variabilei aleatoare Pos_i, dar a omis să-şi noteze şi *tipul* autobuzului $B_i \in \{1,2\}$, corespunzând numerelor 71C respectiv 500.

În acest exerciţiu vi se va cere să folosiţi algoritmul EM pentru a estima parametrii $\theta =< \mu_{11}, \mu_{12}, \mu_{13}, \mu_{21}, \mu_{22}, \mu_{23}, \beta_1, \beta_2 >$ ai acestui model.

a. Exprimaţi $\ln P(Pos, B \mid \theta)$, adică log-verosimilitatea datelor complete — observabile (Pos_i) şi, respectiv, neobservabile (B_i) — în funcţie de θ.

b. Aplicând formula lui Bayes, calculaţi probabilităţile condiţionate ale variabilelor neobservabile (B_i) în raport cu variabilele observabile (Pos_i) şi cu $\theta^{(t)}$, care reprezintă valorile parametrilor θ la iteraţia t:

$$P(B_i \mid Pos_i, \theta^{(t)})$$

c. Calculaţi expresia funcţiei „auxiliare"

$$Q(\theta \mid \theta^{(t)}) \stackrel{def.}{=} E_{P(B|Pos,\theta^{(t)})}[\ln P(Pos, B \mid \theta)]$$

care reprezintă media log-verosimilităţii datelor complete ($\ln P(Pos, B \mid \theta)$) în raport cu funcţia de probabilitate condiţională $P(B \mid Pos, \theta^{(t)})$.

d. Calculaţi $\theta^{(t+1)} = (\mu_{11}^{(t+1)}, \mu_{12}^{(t+1)}, \mu_{13}^{(t+1)}, \mu_{21}^{(t+1)}, \mu_{22}^{(t+1)}, \mu_{23}^{(t+1)}, \beta_1^{(t+1)}, \beta_2^{(t+1)})$. Acestea sunt valorile parametrilor pentru care se atinge maximul expresiei $Q(\theta \mid \theta^{(t)})$:

$$\theta^{(t+1)} = \arg\max_{\theta} Q(\theta \mid \theta^{(t)}).$$

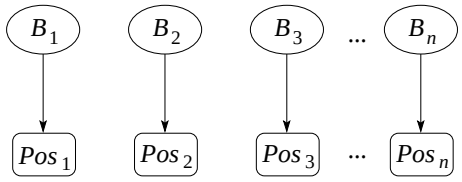
[1007] Pentru varianta generală a algoritmului EM pentru rezolvare de mixturi de K distribuţii categoriale, vedeţi problema 28.

***Indicaţie*: Aceste valori pot fi calculate cu ajutorul metodei multiplicatorilor Lagrange. Metoda aceasta va fi aplicată problemei de optimizare constând din**
– funcţia obiectiv $Q(\theta \mid \theta^{(t)})$
– restricţiile $\mu_{11} + \mu_{12} + \mu_{13} = 1$, $\mu_{21} + \mu_{22} + \mu_{23} = 1$ şi $\beta_1 + \beta_2 = 1$, cu $\mu_{ij} \geq 0$ şi $\beta_k \geq 0$.

e. Presupunem că, spre deosebire de situaţia din enunţul de mai sus, studentul a înregistrat numărul autobuzului B_i, dar nu a înregistrat poziţia P_i pe care a avut-o în autobuz. Este oare posibil ca şi în această variantă a problemei să folosim algoritmul EM pentru a calcula parametrii modelului? Justificaţi.

Răspuns:

a. Cele n deplasări făcute de student cu autobuzul sunt [considerate] independente între ele. Prin urmare, nu există nicio condiţionare între diversele variabile B_i. Mai departe, de valoarea acestor variabile ascunse / neobservabile depinde *poziţia* liberă observată în autobuz Pos_i. Aşadar, putem asocia acestei probleme *modelul grafic* următor:



Probabilitatea comună a datelor complete (observabile şi neobservabile) în funcţie de θ este:

$$P(Pos, B \mid \theta) \overset{indep.}{=} \prod_{i=1}^{n} P(Pos_i, B_i \mid \theta)$$

Aplicând regula de înmulţire a probbailităţilor, obţinem:

$$P(Pos, B \mid \theta) = \prod_{i=1}^{n} P(Pos_i \mid B_i, \theta) \cdot P(B_i \mid \theta)$$

Pentru $i \in \{1, \ldots, n\}, j \in \{1, 2\}$ şi $k \in \{1, 2, 3\}$ considerăm următoarele *variabile-indicator*:

$$b_{ij} = \begin{cases} 1 & \text{dacă } B_i = j \\ 0 & \text{altfel} \end{cases} \qquad p_{ik} = \begin{cases} 1 & \text{dacă } Pos_i = k \\ 0 & \text{altfel} \end{cases}$$

Cu alte cuvinte b_{ij} este 1 dacă în ziua i studentul a mers cu autobuzul j, şi 0 în caz contrar, cu convenţia că valoarea $j = 1$ corespunde autobuzului 71C, iar $j = 2$ autobuzului 500. Similar, p_{ik} este 1 dacă în ziua i studentul a mers într-o poziţie de tip k, unde $k = 1$ înseamnă pe scaun lângă geam, $k = 2$ – pe scaun lângă interval, iar $k = 3$ – în picioare.

Cu aceste notaţii, putem exprima (în manieră compactă) verosimilitatea datelor complete astfel:

$$\begin{aligned} P(Pos, B \mid \theta) &= \prod_{i=1}^{n} P(B_i \mid \theta) \cdot P(Pos_i \mid B_i, \theta) = \prod_{i=1}^{n} \Big(\prod_{j=1}^{2} \beta_j^{b_{ij}} \prod_{k=1}^{3} (\mu_{jk})^{b_{ij}\, p_{ik}} \Big) \\ &= \prod_{i=1}^{n} \prod_{j=1}^{2} \Big(\beta_j \prod_{k=1}^{3} \mu_{jk}^{p_{ik}} \Big)^{b_{ij}} \end{aligned}$$

În consecinţă, log-verosimilitatea datelor complete este:

$$\ln P(Pos, B \mid \theta) = \sum_{i=1}^{n}\sum_{j=1}^{2} b_{ij}\Big(\ln \beta_j + \sum_{k=1}^{3} p_{ik} \ln \mu_{jk}\Big)$$

b. Probabilitatea condiţionată cerută este:

$$\begin{aligned} P(B_i = j \mid Pos_i = k, \theta^{(t)}) &\overset{T.\ Bayes}{=} \frac{P(Pos_i = k \mid B_i = j, \theta^{(t)}) P(B_i = j \mid \theta^{(t)})}{\sum_{j'=1}^{2} P(Pos_i = k \mid B_i = j', \theta^{(t)}) P(B_i = j' \mid \theta^{(t)})} \\ &= \frac{\beta_j^{(t)} \prod_{k=1}^{3} (\mu_{jk}^{(t)})^{p_{ik}}}{\sum_{j'=1}^{2} \beta_{j'}^{(t)} \prod_{k=1}^{3} (\mu_{j'k}^{(t)})^{p_{ik}}} \end{aligned}$$

Prin urmare, mediile variabilelor neobservabile b_{ij} la iteraţia t vor avea şi ele aceeaşi expresie:

$$E[b_{ij}^{(t)}] \overset{not.}{=} E[b_{ij} \mid Pos, \theta^{(t)}] = \frac{\beta_j^{(t)} \prod_{k=1}^{3} (\mu_{jk}^{(t)})^{p_{ik}}}{\sum_{j'=1}^{2} \beta_{j'}^{(t)} \prod_{k=1}^{3} (\mu_{j'k}^{(t)})^{p_{ik}}}$$

c. Calculăm media log-verosimilităţii datelor complete combinând rezultatele de la punctele a şi b şi ţinând cont de proprietatea de liniaritate a mediilor:

$$Q(\theta \mid \theta^{(t)}) = \sum_{i=1}^{n}\sum_{j=1}^{2} E[b_{ij}^{(t)}]\Big(\ln \beta_j + \sum_{k=1}^{3} p_{ik} \ln \mu_{jk}\Big)$$

d. Pentru a calcula $\theta^{(t+1)}$, trebuie maximizată funcţia $Q(\theta \mid \theta^{(t)})$ în raport cu θ. Vom calcula separat valorile optime ale parametrilor, şi anume mai întâi pentru β_j, cu $j \in \{1, 2\}$ şi apoi pentru μ_{jk}, cu $j \in \{1, 2\}$ şi $k \in \{1, 2, 3\}$.

Ştim că variabilele β_j se supun restricţiei $\beta_1 + \beta_2 = 1$. Reţinând din expresia funcţiei $Q(\theta \mid \theta^{(t)})$ doar termenii care conţin variabilele β_j, rezultă că vom avea de optimizat funcţia

$$\sum_{i=1}^{n}(E[b_{i1}^{(t)}] \ln \beta_1 + E[b_{i2}^{(t)}] \ln \beta_2) = \sum_{i=1}^{n}(E[b_{i1}^{(t)}] \ln \beta_1 + E[b_{i2}^{(t)}] \ln(1 - \beta_1)$$

Derivând ultima expresie în raport cu β_1 şi egalând apoi cu 0, obţinem:[1008]

$$\sum_{i=1}^{n}\left(\frac{E[b_{i1}^{(t)}]}{\beta_1} - \frac{E[b_{i2}^{(t)}]}{1 - \beta_1}\right) = 0 \Leftrightarrow (1 - \beta_1)\sum_{i=1}^{n} E[b_{i1}^{(t)}] = \beta_1 \sum_{i=1}^{n} E[b_{i2}^{(t)}] \Leftrightarrow$$

$$\sum_{i=1}^{n} E[b_{i1}^{(t)}] = \beta_1 \sum_{i=1}^{n}(E[b_{i1}^{(t)}] + E[b_{i2}^{(t)}]) \Leftrightarrow \sum_{i=1}^{n} E[b_{i1}^{(t)}] = \beta_1 \sum_{i=1}^{n} E[\underbrace{b_{i1}^{(t)} + b_{i2}^{(t)}}_{1}] \Leftrightarrow$$

$$\sum_{i=1}^{n} E[b_{i1}^{(t)}] = n\beta_1 \Leftrightarrow \beta_1 = \frac{1}{n}\sum_{i=1}^{n} E[b_{i1}^{(t)}]$$

[1008] Se demonstrează uşor că luând baza logaritmului supraunitară (aşa cum am procedat de-a lungul întregului capitol), rezultă că soluţiile derivatelor parţiale de ordinul întâi — care sunt calculate aici şi mai jos — reprezintă într-adevăr punctul de maxim al funcţiei $Q(\theta \mid \theta^{(t)})$.

Este imediat că

$$\beta_2 = \frac{1}{n}\sum_{i=1}^{n} E[b_{i2}^{(t)}],$$

iar atât β_1 cât şi β_2 se situează în intervalul $[0,1]$, întrucât mediile $E[b_{ij}^{(t)}]$ sunt de fapt tot nişte probabilităţi.

Pentru a calcula valorile optime ale parametrilor μ_{jk}, cu indicii $j \in \{1,2\}$ şi $k \in \{1,2,3\}$ vom apela la metoda multiplicatorilor Lagrange întrucât ştim că aceşti parametri se supun constrângerii $\mu_{j1} + \mu_{j2} + \mu_{j3} = 1$. Aşadar, vom avea problema de optimizare cu obiectivul $\max Q(\theta \mid \theta^{(t)})$ şi restricţiile $\sum_{k=1}^{3} \mu_{jk} - 1 = 0$ şi $\mu_{ij} \geq 0$. Funcţia Lagrange asociată acestei probleme este:

$$\Lambda(\mu, \lambda) = \sum_{i=1}^{n}\sum_{j=1}^{2}\sum_{k=1}^{3} E[b_{ij}^{(t)}]p_{ik} \ln \mu_{jk} + \sum_{j=1}^{2} \lambda_j \left(\sum_{k=1}^{3} \mu_{jk} - 1 \right)$$

Egalând cu 0 derivata parţială a lui $\Lambda(\mu, \lambda)$ în raport cu μ_{jk}, obţinem:

$$\frac{\partial \Lambda(\mu,\lambda)}{\partial \mu_{jk}} = 0 \Leftrightarrow \frac{1}{\mu_{jk}} \cdot \sum_{i=1}^{n} E[b_{ij}^{(t)}]p_{ik} + \lambda_j = 0 \Leftrightarrow \mu_{jk} = -\frac{1}{\lambda_j} \cdot \sum_{i=1}^{n} E[b_{ij}^{(t)}]p_{ik}, \text{ cu } \begin{matrix} j \in \{1,2\} \\ k \in \{1,2,3\} \end{matrix}$$

Ştim că

$$\frac{\partial \Lambda(\mu,\lambda)}{\partial \lambda_j} = 0 \Leftrightarrow \sum_{k=1}^{3} \mu_{jk} - 1 = 0, \text{ pentru } j \in \{1,2\}$$

Dacă în această relaţie vom înlocui μ_{jk} cu valorile obţinute anterior, vom determina valoarea lui λ_j pentru $j \in \{1,2\}$:

$$\sum_{k=1}^{3} \left(-\frac{1}{\lambda_j} \cdot \sum_{i=1}^{n} E[b_{ij}^{(t)}]p_{ik} \right) - 1 = 0 \Leftrightarrow \frac{1}{\lambda_j} \cdot \sum_{k=1}^{3}\sum_{i=1}^{n} E[b_{ij}^{(t)}]p_{ik} = -1$$

$$\Leftrightarrow \lambda_j = -\sum_{k=1}^{3}\sum_{i=1}^{n} E[b_{ij}^{(t)}]p_{ik}, \text{ cu } j \in \{1,2\}.$$

Prin urmare, valorile căutate pentru μ_{jk} sunt:

$$\begin{aligned} \mu_{jk}^{(t+1)} &= -\frac{1}{\lambda_j} \cdot \sum_{i=1}^{n} E[b_{ij}^{(t)}]p_{ik} = \frac{\sum_{i=1}^{n} E[b_{ij}^{(t)}]p_{ik}}{\sum_{l=1}^{3}\sum_{i=1}^{n} E[b_{ij}^{(t)}]p_{il}} \\ &= \frac{\sum_{i=1}^{n} E[b_{ij}^{(t)}]p_{ik}}{\sum_{i=1}^{n}\sum_{l=1}^{3} E[b_{ij}^{(t)}]p_{il}}, \text{ cu } j \in \{1,2\} \text{ şi } k \in \{1,2,3\} \end{aligned}$$

Este imediat că aceste valori se situează în intervalul $[0,1]$.

e. Aşa cum se observă şi din modelul grafic reprezentat la punctul a, dacă se cunoaşte tipul autobuzului B_i, nu se pot estima parametrii [reprezentând distribuţia] variabilei care reprezintă poziţia Pos_i ocupată de student în autobuz. Altfel spus, variabilele observabile B_i nu oferă nicio informaţie relevantă pentru a putea calcula variabilele neobservabile Pos_i.

11. **(Rezolvarea unei mixturi de doi vectori de distribuţii categoriale, folosind presupoziţia de independenţă condiţională de tip Bayes Naiv,[1009] aka algoritmul Bayes Naiv nesupervizat)**

prelucrare de Liviu Ciortuz, după
■ □ ● ○ ∗ ***CMU, 2014 spring, A. Singh, B. Poczos, HW3, pr. 2.1***

Fie setul de date etichetate $\mathcal{D} = \{(X_1, Y_1)), \ldots, (X_n, Y_n)\}$.

Atunci când se foloseşte presupoziţia de independenţă condiţională specifică clasificatorului Bayes Naiv, putem scrie verosimilitatea [comună a] datelor $\mathcal{D}$ astfel:

$$P(\mathcal{D}) \overset{i.i.d.}{=} \prod_{i=1}^{n} P(X_i, Y_i) = \prod_{i=1}^{n} (P(X_i|Y_i) \cdot P(Y_i)) = \prod_{i=1}^{n} \Big(P(Y_i) \cdot \prod_{j=1}^{d} P(X_i^j|Y_i) \Big)$$

Notaţia X_i^j, unde $j \in \{1, \ldots, d\}$, desemnează atributul de intrare j al instanţei etichetate (X_i, Y_i).

***Observaţie*: În cele ce urmează vom nota variabilele aleatoare cu majuscule, iar valorile lor cu minuscule.**[1010]

Presupunem că $y_i \in \{0, 1\}\ \forall i$ şi $x_i^j \in \{1, 2, \ldots, V\}\, \forall i, j$. Aşadar, considerăm (doar pentru simplitate) că V, numărul de valori, este acelaşi pentru toate atributele X^j, cu $j \in \{1, \ldots, d\}$ şi, în plus, mulţimile de valori ale acestor atribute coincid. Vom nota $\pi_0 = P(Y = 0)$, $\pi_1 = P(Y = 1)$ şi $\pi = (\pi_0, \pi_1)$. Similar, pentru fiecare $j \in \{1, \ldots, d\}$, vom nota parametrii distribuţiilor $P(X^j|Y)$ astfel: $\beta_0^j = (\beta_{10}^j, \beta_{20}^j, \ldots, \beta_{V0}^j)$ şi $\beta_1^j = (\beta_{11}^j, \beta_{21}^j, \ldots, \beta_{V1}^j)$, deci $\beta_{k0}^j \overset{not.}{=} P(X^j = k|Y = 0)$ şi $\beta_{k1}^j \overset{not.}{=} P(X^j = k|Y = 1)$, pentru fiecare $k \in \{1, 2, \ldots, V\}$. În sfârşit, $\beta \overset{not.}{=} (\beta_0^1, \beta_1^1, \ldots, \beta_0^d, \beta_1^d)$ şi $\theta \overset{not.}{=} (\pi, \beta)$.

Evident, în cazul în care dispunem de date etichetate — adică variabila Y_i este „observabilă“, pentru toate valorile lui i —, vom putea obţine uşor estimări (în sensul MLE) pentru parametrii π şi β, folosind procedeul clasic de enumerare (engl., counting) şi normalizare.

În acest exerciţiu vom presupune însă că nu dispunem de date etichetate. *Scopul* nostru este de a arăta că şi în situaţia aceasta putem estima parametrii θ (adică π şi β^j, cu $j = 1, \ldots, d$), şi anume folosind algoritmul de estimare-maximizare (EM).

a. Se ştie — vedeţi problema 1 — că funcţia de log-verosimilitate a datelor observabile ($\log P(x_1, \ldots, x_n|\theta)$) este minorată de o funcţie de două argumente $F(q, \theta)$, unde q este o distribuţie probabilistică oarecare definită peste variabilele-indicator z_i.[1011]

Vă cerem[1012]

[1009] În cazul $d = 1$ (vedeţi notaţiile din problemă), se obţine varianta de algoritm EM pentru o mixtură de două distribuţii categoriale, care a fost deja ilustrată la problema 10. Pentru mixturi de K distribuţii categoriale, vedeţi problema 28.

Pentru un exerciţiu (incluzând *implementare*) în care face clusterizare de caractere scrise de mână, aplicând un algoritm EM de tipul celui prezentat în problema de faţă, însă lucrând cu variabile Bernoulli în locul variabilelor categoriale, vedeţi exerciţiul 9.

[1010] De exemplu, Y_i este o variabilă aleatoare, iar y_i desemnează o instanţiere a lui Y_i la una din valorile ei posibile.

[1011] *Observaţie*: Veţi ţine cont că rolul acelor variabile z_i (de la problema 1) va fi jucat aici de către y_i.

[1012] *Sugestie*: Procedaţi în mod similar cu rezolvarea dată la problema 1 punctul *a*.

– să scrieţi expresia funcţiei de (log-)verosimilitate pe care urmăreşte să o maximizeze algoritmul EM în contextul prezentului exerciţiu;[1013]

– să introduceţi apoi probabilităţile $q(y_i)$ în mod forţat — însă într-o manieră convenabilă — în expresia scrisă anterior pentru funcţia de log-verosimilitate;

– şi, în sârşit, folosind inegalitatea lui Jensen, să obţineţi expresia marginii inferioare $F(q, \theta)$ pentru funcţia de log-verosimilitate care a fost menţionată mai sus.

***Observaţie importantă*: Spre deosebire de cum am procedat la toate exerciţiile de până acum legate de algoritmul EM, aici nu vom [mai] lucra cu funcţia „auxiliară“ Q,**[1014] **ci direct cu funcţia F.**

b. Pasul E: Calculaţi probabilitatea fiecărei asignări posibile [la o anumită clasă] pentru fiecare instanţă, date fiind valorile actuale ale parametrilor π şi β. Concret, veţi determina valorile optime pentru $q(y_i)$ (i.e., $q(Y_i = 0)$ şi $q(Y_i = 1)$), folosind derivatele parţiale ale funcţiei $F(q(y_i), \theta)$ care a fost dedusă la punctul precedent.

c. Pasul M: Deduceţi regulile pentru actualizarea parametrilor π şi β în funcţie de probabilităţile de asignare [la o anumită clasă] pentru fiecare instanţă, $q(y_i)$, care au fost calculate la pasul E.

d. Schiţaţi o strategie convenabilă pentru iniţializarea parametrilor (π şi β) şi indicaţi motivele care au stat la baza alegerii / conceperii respectivei strategii.

Răspuns:

a. Algoritmul EM urmăreşte să maximizeze funcţia de log-verosimilitate a datelor observabile, $\log P(x_1, \ldots, x_n|\theta)$. Fără a mai menţiona parametrul θ în calculul care urmează, putem scrie:

$$
\begin{aligned}
\log P(x_1, \ldots, x_n) &\overset{i.i.d.}{=} \log \prod_{i=1}^{n} P(x_i) = \log \prod_{i=1}^{n} \sum_{y_i \in \{0,1\}} P(x_i, y_i) \\
&= \sum_{i=1}^{n} \log \sum_{y_i \in \{0,1\}} P(x_i, y_i) = \sum_{i=1}^{n} \log \sum_{y_i \in \{0,1\}} P(x_i|y_i) P(y_i) \\
&= \sum_{i=1}^{n} \log \sum_{y_i \in \{0,1\}} P(y_i) \prod_{j=1}^{d} P(x_i^j|y_i) \\
&= \sum_{i=1}^{n} \log \sum_{y_i \in \{0,1\}} q(y_i) \frac{P(y_i) \prod_{j=1}^{d} P(x_i^j|y_i)}{q(y_i)} \\
&\overset{def.}{=} \sum_{i=1}^{n} \log E_q \left[\frac{P(y_i) \prod_{j=1}^{d} P(x_i^j|y_i)}{q(y_i)} \right] \\
&\geq \sum_{i=1}^{n} E_q \left[\log \frac{P(y_i) \prod_{j=1}^{d} P(x_i^j|y_i)}{q(y_i)} \right] \qquad (Jensen)
\end{aligned}
$$

[1013] *Indicaţie*: Verosimilitatea unei instanţe neetichetate, $P(x_i|\theta)$ se poate obţine însumând probabilităţile corespunzătoare tuturor asignărilor posibile y_i ale variabilei latente / neobservabile Y_i.
[1014] Vedeţi finalul rezolvării problemei 1 de la acest capitol.

$$\stackrel{def.}{=} \sum_{i=1}^{n} \sum_{y_i \in \{0,1\}} q(y_i) \log \frac{P(y_i) \prod_{j=1}^{d} P(x_i^j | y_i)}{q(y_i)}$$

$$= \sum_{i=1}^{n} \sum_{y_i \in \{0,1\}} q(y_i) \Big[\log P(y_i) + \Big(\sum_{j=1}^{d} \log P(x_i^j | y_i) \Big) - \log q(y_i) \Big].$$

Aşadar, reintroducând parametrul θ, marginea inferioară a funcţiei de log-verosimilitate a datelor observabile $x_1, \ldots, x_n$ este:

$$F(q, \theta) = \sum_{i=1}^{n} \sum_{y_i \in \{0,1\}} q(y_i) \Big[\log P(y_i | \theta) + \Big(\sum_{j=1}^{d} \log P(x_i^j | y_i, \theta) \Big) - \log q(y_i) \Big]. \qquad (429)$$

b. La pasul E, pentru $i = 1, \ldots, n$, vom actualiza $q(y_i)$ în funcţie de valorile curente pentru parametrii π şi β, mai exact în funcţie de $P_\pi(y_i) \stackrel{not.}{=} P(y_i \mid \pi)$ şi $P_\beta(x_i^j | y_i) \stackrel{not.}{=} P(x_i^j | y_i, \beta)$.[1015] Calculând derivata parţială a funcţiei F în raport cu $q(y_i)$, obţinem:

$$\log P_\pi(y_i) + \sum_{j=1}^{d} \log P_\beta(x_i^j | y_i) - \log q(y_i) - 1,$$

iar apoi egalându-o cu 0, deducem uşor următoarele reguli de actualizare:

$$y_i = 0 : \qquad q(Y_i = 0) \propto P_\pi(Y_i = 0) \prod_{j=1}^{d} P_\beta(x_i^j | Y_i = 0)$$

$$y_i = 1 : \qquad q(Y_i = 1) \propto P_\pi(Y_i = 1) \prod_{j=1}^{d} P_\beta(x_i^j | Y_i = 1),$$

unde simbolul $\propto$ înseamnă *proporţional cu*. Se poate observa uşor că factorul de proporţionalitate în precedentele două relaţii este acelaşi. Aşadar, ţinând cont de restricţia $q(y_i = 0) + q(y_i = 1) = 1$, putem scrie:

$$q(Y_i = 0) = \frac{P_\pi(Y_i = 0) \prod_{j=1}^{d} P_\beta(x_i^j | Y_i = 0)}{\sum_{\ell \in \{0,1\}} P_\pi(Y_i = \ell) \prod_{j=1}^{d} P_\beta(x_i^j | Y_i = \ell)}$$

$$q(Y_i = 1) = \frac{P_\pi(Y_i = 1) \prod_{j=1}^{d} P_\beta(x_i^j | Y_i = 1)}{\sum_{\ell \in \{0,1\}} P_\pi(Y_i = \ell) \prod_{j=1}^{d} P_\beta(x_i^j | Y_i = \ell)}$$

c. La pasul M, cunoaştem valorile $q(y_i)$ şi vom maximiza funcţia [obiectiv] $F(q, \theta)$ în raport cu parametrul θ (i.e., π şi β). După calcule relativ simple, cu

[1015] Pentru a determina valorile optime pentru $q(y_i = 0)$ şi $q(y_i = 1)$, am putea proceda ca şi la exerciţiile precedente:
i. fie ţinând cont de relaţia $q(y_i = 0) + q(y_i = 1) = 1$, $\forall i \in \{1, \ldots, n\}$ şi înlocuind, de exemplu, $q(y_i = 1)$ în funcţie de $q(y_i = 0)$ în relaţia (429), după care calculăm rădăcina derivatei $\frac{\partial F(q, \theta)}{\partial q(y_i = 0)}$ şi verificăm dacă această rădăcină aparţine intervalului $[0, 1]$;
ii. fie folosim metoda multiplicatorilor Lagrange pentru a rezolva (pentru fiecare $i \in \{1, \ldots, n\}$) problema de optimizare care are funcţia obiectiv (429) şi restricţia $q(y_i = 0) + q(y_i = 1) = 1$.
Însă rezolvarea pentru care am optat aici este de un alt tip, care într-o anumită măsură seamănă cu metoda *i*, dar inversează cei doi paşi.

care de acum suntem obişnuiţi, vom obţine următoarele reguli de actualizare pentru parametrii π:[1016]

$$\begin{cases} \pi_0 \overset{not.}{=} P(Y=0) \propto \sum_{i=1}^n q(Y_i=0) \\ \pi_1 \overset{not.}{=} P(Y=1) \propto \sum_{i=1}^n q(Y_i=1) \end{cases} \Rightarrow \begin{cases} \pi_0 = \dfrac{1}{n}\sum_{i=1}^n q(Y_i=0) \\ \pi_1 = \dfrac{1}{n}\sum_{i=1}^n q(Y_i=1) \end{cases}$$

şi apoi pentru parametrii β:[1017]

$$\begin{cases} \beta_{v0}^j \overset{not.}{=} P(X^j=v|Y=0) \propto \sum_{i=1}^n q(Y_i=0)\cdot 1_{\{x_i^j=v\}} \\ \beta_{v1}^j \overset{not.}{=} P(X^j=v|Y=1) \propto \sum_{i=1}^n q(Y_i=1)\cdot 1_{\{x_i^j=v\}}, \end{cases}$$

$$\Rightarrow \begin{cases} \beta_{v0}^j = \dfrac{\sum_{i=1}^n q(Y_i=0)\cdot 1_{\{x_i^j=v\}}}{\sum_{\ell\in\{0,1\}}\sum_{i=1}^n q(Y_i=\ell)\cdot 1_{\{x_i^j=v\}}} \\ \beta_{v1}^j = \dfrac{\sum_{i=1}^n q(Y_i=1)\cdot 1_{\{x_i^j=v\}}}{\sum_{\ell\in\{0,1\}}\sum_{i=1}^n q(Y_i=\ell)\cdot 1_{\{x_i^j=v\}}} \end{cases}$$

Notaţia $1_{\{\ldots\}}$ folosită mai sus desemnează o *funcţie-indicator*. Aşa cum ştim, ea ia valoarea 1 atunci când condiţia specificată [între acolade] este satisfăcută şi valoarea 0 în caz contrar.

Este interesant *de observat* că relaţiile obţinute la acest pas al algoritmului EM sunt foarte similare cu relaţiile corespunzătoare folosite [pentru estimarea parametrilor] de către clasificatorul Bayes Naiv (supervizat).

d. O strategie convenabilă pentru iniţializarea parametrilor mixturii ar trebui să ţină cont de cunoştinţele / informaţiile pe care le avem în legătură cu o soluţie plauzibilă. Dacă avem motive să credem că o anumită clasă este mai probabilă decât cealaltă clasă, vom putea „injecta" această informaţie în valoarea folosită la iniţializarea lui π. Pentru parametrii β, dacă intuim că o anumită valoare (k) a atributului X^j se corelează mai bine cu (adică este mai „indicativă" în raport cu) o anumită clasă ℓ, atunci vom da o valoare (probabilitate) mai mare parametrului $\beta_{k\ell}^j$, care corespunde respectivei valori a atributului X^j. Dacă nu avem niciun fel de informaţii / cunoştinţe a priori, putem să iniţializăm aceşti parametri în mod aleator.

[1016] Pentru actualizarea parametrilor π_0 şi π_1, putem folosi oricare dintre metodele i şi ii menţionate la nota de subsol precedentă.
În cazul i, reţinând din expresia (429) doar termenii care-l conţin pe $P(Y=y_i|\theta)$, vom avea:

$$\frac{\partial}{\partial P_\pi(Y=0)}\left(\sum_{i=1}^n \big(q(y_i=0)\log P_\pi(Y=0) + q(y_i=1)\log(1-P_\pi(Y-0))\big)\right) = 0 \Leftrightarrow \text{ ş.a.m.d.}$$

În cazul ii, lagrangeanul de optimizat va fi

$$\mathcal{L}(\lambda) = F(q,\theta) + \lambda(1 - P_\pi(Y=0) - P_\pi(Y=1)), \text{ deci}$$

$$\frac{\partial \mathcal{L}(\lambda)}{\partial P_\pi(Y=0)} = 0 \Leftrightarrow \frac{1}{P_\pi(Y=0)}\sum_{i=1}^n q(Y_i=0) = \lambda \Leftrightarrow P_\pi(Y=0) = \frac{1}{\lambda}\sum_{i=1}^n q(Y_i=0)$$

şi similar pentru $P_\pi(Y=1)$.

[1017] Folosim metoda multiplicatorilor lui Lagrange (ii), din cauza restricţiilor $\sum_{v=1}^V \beta_{v0}^j = 1$ şi $\sum_{v=1}^V \beta_{v1}^j = 1$. Detaliile de calcul sunt similare cu cele din ultima parte de la nota de subsol precedentă.

12. **(Algoritmul EM pentru mixturi de distribuţii categoriale; [aplicare la] identificarea domeniilor semantice asociate cuvintelor dintr-un document-text)**

■ □ • ○ ***CMU, 2012 fall, E. Xing, A. Singh, HW3, pr. 3***

În acest exerciţiu vi se va cere să deduceţi relaţiile corespunzătoare paşilor E şi M din corpul iterativ al algoritmului EM pentru „modelarea" variabilelor „neobservabile" care desemnează *domeniile semantice* (enlgl., topics) implicate în generarea unui document oarecare de tip text.

Vom considera că fiecare *cuvânt* din documentul dat este reprezentat de o variabilă aleatoare w care poate lua valorile $1, \ldots, V$ relativ la un *vocabular* dat. De fapt, în cele ce urmează vom desemna fiecare cuvânt w printr-un *vector-indicator* format din V componente astfel încât $w(i) = 1$ dacă w ia valoarea cuvântului de pe poziţia i din vocabular şi 0 în caz contrar. Aşadar, $\sum_{i=1}^{V} w(i) = 1$.

Dat fiind un *document* constituit din cuvintele w_j, $j = 1, \ldots, N$, unde N este lungimea documentului, vom presupune că aceste cuvinte sunt generate de către o mixtură de K distribuţii categoriale:

$$\begin{aligned} P(w_j) &= \sum_{s=1}^{K} \pi_s \, P(w_j|\beta_s) \\ P(w_j|\beta_s) &= \prod_{i=1}^{V} P(w_j(i) = 1|Z_j = s) = \prod_{i=1}^{V} \beta_s(i)^{w_j(i)}, \end{aligned}$$

unde

$\pi_s \overset{not.}{=} P(Z = s)$ este probabilitatea (a priori) ca variabila latentă Z, care desemnează domeniul semantic asociat unui cuvântul oarecare, să ia valoarea s;

$\beta_s \overset{not.}{=} (\beta_s(1), \ldots, \beta_s(i), \ldots, \beta_s(V))$, cu $\beta_s(i) \geq 0$ pentru $i = 1, \ldots, V$ şi $\sum_{i=1}^{V} \beta_s(i) = 1$, pentru fiecare $s = 1, \ldots, K$. Vectorul β_s determină distribuţia categorială asociată cu domeniul semantic s;

$\beta_s(i) \overset{not.}{=} P(w_j(i) = 1|Z_j = s)$ desemnează probabilitatea ca $w_j \overset{not.}{=} (w_j(1), \ldots, w_j(V))$ să fi fost generat de către domeniul semantic s.

***Observaţie*: Remarcaţi faptul că acest model nu ţine cont de ordinea / asocierea cuvintelor din documentul considerat. Deşi faptul aceasta constituie o încălcare evidentă a proprietăţilor definitorii ale unui text, modelul propus aici are totuşi o utilitate practică dovedită.[1018]**

a. Referitor la pasul E al algoritmului EM, pentru fiecare cuvânt w_j, calculaţi $q_j(s) \overset{not.}{=} P(Z_j = s|w_j; \theta')$, probabilitatea (a posteriori) ca, dat un cuvânt w_j, acesta să corespundă unui anumit domeniu semantic s din ansamblul celor K domenii semantice considerate. Prin θ desemnăm ansamblul parametrilor din modelul de mixtură considerat: π_s şi β_s cu $s = 1, \ldots, K$, iar θ' notează valoarea acestor parametri la pasul de iniţializare [al algoritmului EM], respectiv valoarea obţinută la iteraţia precedentă.

[1018] Vedeţi capitolul *Word Sense Disambiguation* din cartea *Foundations of Statistical Natural Language Processing*, Christopher Manning, Hinrich Schütze, MIT Press, 2002, pag 252-256.

b. Pentru pasul M, determinaţi valoarea parametrului θ care maximizează marginea inferioară (vedeţi problema 1) pentru log-verosimilitatea datelor observabile, i.e., a cuvintelor din documentul dat:

$$\ell(w|\theta) \stackrel{not.}{=} \ln \prod_{j=1}^{N} P(w_j|\theta).$$

***Indicaţie*: Procedând în mod clasic, adică sumând în raport cu variabila latentă s care reprezintă tematica / domeniul semantic, putem rescrie funcţia de log-verosimilitate $l(w|\theta)$ astfel:**

$$\ell(w|\theta) = \sum_{j=1}^{N} \ln \sum_{s} P(w_j, s|\theta) = \sum_{j=1}^{N} \ln \sum_{s} q_j(s) \frac{P(w_j, s|\theta)}{q_j(s)},$$

unde cantităţile $q_j(s)$ cu $j \in \{1, \ldots, V\}$ şi $s \in \{1, \ldots, K\}$ au fost calculate la pasul E al iteraţiei curente. Mai departe, folosind inegalitatea lui Jensen (vedeţi pr. 79 de la capitolul *Fundamente*), obţinem:

$$\begin{aligned}
\ell(w|\theta) \geq & \sum_{j=1}^{N} \sum_{s} q_j(s) \ln \frac{P(w_j, s|\theta)}{q_j(s)} = \\
& \sum_{j=1}^{N} \sum_{s} q_j(s) \ln P(w_j, s|\theta) - \sum_{j=1}^{N} \sum_{s} q_j(s) \ln q_j(s) = \\
& \sum_{j=1}^{N} \sum_{s} q_j(s) \ln P(w_j, s|\theta) + \sum_{j=1}^{N} H(q_j).
\end{aligned}$$

Am notat cu $H(q_j)$ entropia $\sum_s q_j(s) \ln q_j(s) = \sum_s P(s|w_j;\theta') \ln P(s|w_j;\theta')$.[1019] Evident, aceasta nu depinde de θ (deşi depinde de θ'). În concluzie, la pasul M veţi calcula acea valoare a lui θ pentru care se atinge maximul *funcţiei auxiliare*

$$Q(\theta|\theta') \stackrel{not.}{=} \sum_{j=1}^{N} \sum_{s} q_j(s) \ln P(w_j, s|\theta).$$

Răspuns:

a. La pasul E, calculăm probabilităţile a posteriori $q_j(s) \stackrel{not.}{=} P(Z_j = s|w_j;\theta')$. Pentru convenienţă, în rezolvarea / calculul care urmează vom renunţa la a folosi accentul ($'$) pentru componentele din setul de parametri $\theta' = (\pi', \beta'_1, \ldots, \beta'_K)$ indicat în enunţ — ar fi trebuit, de fapt, să scriem $\theta^{(t)} = (\pi^{(t)}, \beta_1^{(t)}, \ldots, \beta_K^{(t)})$ — deoarece notaţiile ar deveni prea complicate. (Cititorul atent va şti din context la ce anume se referă notaţia $\theta = (\pi, \beta_1, \ldots, \beta_K)$.)

$$\begin{aligned}
q_j(s) \stackrel{not.}{=} & P(Z_j = s|w_j;\theta) \stackrel{Bayes\ T.}{=} \frac{P(w_j|Z_j = s;\theta)\ P(Z_j = s|\theta)}{P(w_j|\theta)} \\
= & \frac{\pi_s\ P(w_j|\beta_s)}{\sum_{s'=1}^{K} \pi_{s'}\ P(w_j|\beta_{s'})} = \frac{\pi_s \prod_{l=1}^{V} \beta_s(l)^{w_j(l)}}{\sum_{s'=1}^{K} \pi_{s'} \prod_{l=1}^{V} \beta_{s'}(l)^{w_j(l)}}.
\end{aligned}$$

[1019] Pentru simplitate, am presupus că atât funcţia de log-verosimilitate cât şi entropia se calculează folosind pentru logaritm o aceeaşi bază, supraunitară.

b. Conform *Indicaţiei* din enunţ, calculăm mai întâi funcţia auxiliară asociată iteraţiei curente (t):

$$\begin{aligned}
Q(\theta|\theta') &\overset{not.}{=} \sum_{j=1}^{N}\sum_{s=1}^{K} q_j(s)\ln P(w_j, Z_j = s|\theta) \\
&= \operatorname*{argmax}_{\theta} \sum_{j=1}^{N}\sum_{s=1}^{K} q_j(s)\ln P(w_j|Z_j = s;\theta)\ P(Z_j = s|\theta) \\
&= \operatorname*{argmax}_{\theta} \sum_{j=1}^{N}\sum_{s=1}^{K} q_j(s)\ln\left(\pi_s \prod_{l=1}^{V}\beta_s(l)^{w_j(l)}\right) \\
&= \operatorname*{argmax}_{\theta} \sum_{j=1}^{N}\sum_{s=1}^{K} [q_j(s)\ln\pi_s + q_j(s)\sum_{l=1}^{V}\ln\beta_s(l)^{w_j(l)}] \\
&= \operatorname*{argmax}_{\theta} \sum_{j=1}^{N}\sum_{s=1}^{K} [q_j(s)\ln\pi_s + q_j(s)\sum_{l=1}^{V} w_j(l)\ln\beta_s(l)]. \qquad (430)
\end{aligned}$$

Se ştie că $\theta^{(t+1)} \overset{def.}{=} \operatorname{argmax}_\theta Q(\theta|\theta')$. Problema aceasta de optimizare se poate rezolva maximizând separat $Q(\theta|\theta')$ în raport cu parametrii $\beta_s(l)$ şi respectiv în raport cu parametrii π_s (unde $s \in \{1,\ldots,K\}$).[1020]

În ce priveşte optimizarea lui $Q(\theta|\theta')$ în raport cu $\beta_s(l)$:
După ce eliminăm din expresia (430) termenii care sunt constanţi în raport cu β_s, obţinem:

$$\sum_{j=1}^{N} q_j(s)\sum_{l=1}^{V} w_j(l)\ln\beta_s(l).$$

Apoi, deoarece $\beta_s \overset{not.}{=} (\beta_s(1),\ldots,\beta_s(i),\ldots,\beta_s(V))$, cu restricţiile $\beta_s(i) \geq 0$ pentru $i = 1,\ldots,V$ şi $\sum_{i=1}^{V}\beta_s(i) = 1$, vom folosi metoda multiplicatorilor lui Lagrange pentru a ne asigura că cea de-a doua restricţie este satisfăcută.[1021] **Aşadar, introducând variabila / multiplicatorul Lagrange $\lambda \in \mathbb{R}$, pornind de la expresia anterior obţinută vom scrie funcţia „lagrangeană“**

$$\mathcal{L}(\beta_s(l)) = \sum_{j=1}^{N} q_j(s)\sum_{l=1}^{V} w_j(l)\ln\beta_s(l) + \lambda\left(\sum_{l=1}^{V}\beta_s(l) - 1\right),$$

pe care o vom optimiza – prin derivare – în raport cu $\beta_s(l)$:

$$\begin{aligned}
&\frac{\partial}{\partial\beta_s(l)}\mathcal{L}(\beta_s(l)) = 0 \Leftrightarrow \sum_{j=1}^{N} q_j(s)\frac{w_j(l)}{\beta_s(l)} + \lambda = 0 \\
\Leftrightarrow \frac{1}{\beta_s(l)}\sum_{j=1}^{N} q_j(s)\ w_j(l) + \lambda = 0 &\Leftrightarrow \frac{1}{\beta_s(l)} = \frac{-\lambda}{\sum_{j=1}^{N} q_j(s)\ w_j(l)} \\
&\Leftrightarrow \beta_s(l) = \frac{\sum_{j=1}^{N} q_j(s)\ w_j(l)}{-\lambda}. \qquad (431)
\end{aligned}$$

[1020] Cititorul pedant va putea observa după calculele de mai jos că matricea hessiană a funcţiei auxiliare $Q(\theta|\theta')$ este diagonală şi negativ definită, deci $Q(\theta|\theta')$ este funcţie concavă, aşadar ea admite o singură valoare maximă, care este globală.
[1021] Veţi vedea că soluţia obţnută ulterior va satisface şi prima restricţie.

Întrucât $\sum_{l=1}^{V} \beta_s(l) = 1$, din relaţia (431) va rezulta

$$\sum_{l=1}^{V} \frac{\sum_{j=1}^{N} q_j(s)\ w_j(l)}{-\lambda} = 1 \quad \Leftrightarrow \quad -\lambda = \sum_{l=1}^{V} \sum_{j=1}^{N} q_j(s)\ w_j(l)$$

Substituind această valoare a lui $-\lambda$ în relaţia (431), obţinem ceea ce va constitui *noua valoare* a parametrului $\beta_s(l)$, pentru iteraţia $t+1$:

$$\begin{aligned} \beta_s(l) &= \frac{\sum_{j=1}^{N} q_j(s)\ w_j(l)}{\sum_{l=1}^{V} \sum_{j=1}^{N} q_j(s)\ w_j(l)} = \frac{\sum_{j=1}^{N} q_j(s)\ w_j(l)}{\sum_{j=1}^{N} \sum_{l=1}^{V} q_j(s)\ w_j(l)} \\ &= \frac{\sum_{j=1}^{N} q_j(s)\ w_j(l)}{\sum_{j=1}^{N} q_j(s) \underbrace{\sum_{l=1}^{V} w_j(l)}_{1}} = \frac{\sum_{j=1}^{N} q_j(s)\ w_j(l)}{\sum_{j=1}^{N} q_j(s)}. \end{aligned}$$

În mod intuitiv, expresia pe care tocmai am obţinut-o (şi care este, evident, nenegativă) poate fi interpretată ca fiind „ponderea" care îi revine cuvântului de pe poziţia l din vocabular — fiindcă $w(l) = 1$ — în ansamblul [clusterului corespunzător] domeniului semantic s.

În mod similar, pentru a optimiza funcţia auxiliară $Q(\theta|\theta')$ în raport cu probabilitatea de selecţie π_s, vom începe prin a elimina din expresia (430) termenii care sunt constanţi în raport cu π_s. Expresia care rezultă este:

$$\sum_{j=1}^{N} q_j(s) \ln \pi_s.$$

După aceea, folosind din nou metoda multiplicatorilor lui Lagrange pentru a impune satisfacerea restricţiei $\sum_{s=1}^{K} \pi_s = 1$, vom scrie funcţia „lagrangeană"

$$\mathcal{L}(\pi_s) = \sum_{j=1}^{N} q_j(s) \ln \pi_s + \lambda \left(\sum_{s=1}^{K} \pi_s - 1 \right).$$

Valoarea optimă pentru π_s va fi obţinută cu ajutorul derivatei parţiale:

$$\begin{aligned} \frac{\partial}{\partial \pi_s} \mathcal{L}(\pi_s) = 0 \Leftrightarrow \sum_{j=1}^{N} \frac{q_j(s)}{\pi_s} + \lambda = 0 \Leftrightarrow \frac{1}{\pi_s} \sum_{j=1}^{N} q_j(s) = -\lambda \\ \Leftrightarrow \pi_s = \frac{\sum_{j=1}^{N} q_j(s)}{-\lambda}. \end{aligned} \tag{432}$$

Întrucât $\sum_{s=1}^{K} \pi_s = 1$, vom avea:

$$\begin{aligned} \sum_{s=1}^{K} \frac{\sum_{j=1}^{N} q_j(s)}{-\lambda} = 1 \Leftrightarrow \frac{1}{-\lambda} \sum_{s=1}^{K} \sum_{j=1}^{N} q_j(s) = 1 \\ \Leftrightarrow -\lambda = \sum_{s=1}^{K} \sum_{j=1}^{N} q_j(s). \end{aligned}$$

Substituind această valoare a lui $-\lambda$ în expresia (432), obţinem *noua valoare* a parametrului π_s:

$$\pi_s = \frac{\sum_{j=1}^{N} q_j(s)}{\sum_{s=1}^{K}\sum_{j=1}^{N} q_j(s)} = \frac{\sum_{j=1}^{N} q_j(s)}{\sum_{j=1}^{N}\underbrace{\sum_{s=1}^{K} q_j(s)}_{1}} = \frac{\sum_{j=1}^{N} q_j(s)}{N}.$$

În mod intuitiv, această ultimă expresie (care este, evident, nenegativă) poate fi interpretată ca fiind „ponderea“ care revine clusterului s din ansamblul celor N cuvinte din vocabular.

Sumarizând, *regulile de actualizare* de la iteraţia t a algoritmului EM sunt

Pasul E:

$q_j(s) = \frac{\pi_s \prod_{l=1}^{V} \beta_s(l)^{w_j(l)}}{\sum_{s'=1}^{K} \pi_{s'} \prod_{l=1}^{V} \beta_{s'}(l)^{w_j(l)}}$ **pentru** $j = 1, \ldots, N$ **şi** $s = 1, \ldots, K$**,**

unde π_s, $\pi_{s'}$, β_s şi $\beta_{s'}$ provin de la pasul M al iteraţiei $t-1$ atunci când $t > 1$ (şi respectiv de la iniţializare, în cazul iteraţiei $t = 1$);

Pasul M:

$\beta_s(l) = \frac{\sum_{j=1}^{N} q_j(s)\ w_j(l)}{\sum_{j=1}^{N} q_j(s)}$ **pentru** $s = 1, \ldots, K$ **şi** $l = 1, \ldots, V$

$\pi_s = \frac{\sum_{j=1}^{N} q_j(s)}{N}$ **pentru** $s = 1, \ldots, K$**.**

8.1.3 Distribuţii binomiale / multinomiale

13. (Algoritmul EM: „învăţarea“ unei distribuţii probabiliste multinomiale (şi implicit a unei distribuţii categoriale), determinate de un singur parametru (de estimat), în condiţiile existenţei unei variabile neobservabile)

prelucrare de Liviu Ciortuz, după CMU, 2008 fall, Eric Xing, final exam, pr. 6

Considerăm un curs de învăţare automată la care probabilitatea ca un student să fie notat cu calificativul A este $P(A) = 1/2$, cu B este $P(B) = \theta$, cu C este $P(C) = 2\theta$, iar cu D este $P(D) = 1/2 - 3\theta$, unde $\theta \in (0, 1/6)$ este un parametru. Ni se mai spune că un număr de c studenţi au luat calificativul C, iar d studenţi au luat calificativul D. Nu ştim cu exactitate câţi studenţi au luat calificativul A sau câţi studenţi au luat calificativul B, dar ştim că în total h studenţi au luat fie calificativul A, fie calificativul B.

Ne propunem să utilizăm algoritmul Expectation-Maximization pentru a obţine o estimare de verosimilitate maximă a parametrului θ.

***Observaţii*:**

1. Remarcăm faptul că în această problemă nu „rezolvăm“ o mixtură de distribuţii, ci

învăţăm o distribuţie (categorială), a cărei definiţie depinde de un parametru (θ).

2. Am specificat distribuţia de învăţat în tabelul alăturat (mai precis, în primele două coloane din acest tabel). Evident, dacă am cunoaşte valoarea lui a, am putea calcula imediat estimarea de verosimilitate maximă (MLE) a lui θ.

	prob.	count-uri
A	$1/2$	a =?
B	θ	$b = h - a$
C	2θ	c
D	$1/2 - 3\theta$	d

De exemplu, folosind ultimele trei linii din tabel, am putea scrie:

$$\frac{3\theta}{\frac{1}{2}} = \frac{b+c}{b+c+d} \Leftrightarrow 6\theta = \frac{b+c}{b+c+d} \Leftrightarrow \theta = \frac{1}{6} \cdot \frac{b+c}{b+c+d} \Leftrightarrow \theta = \frac{1}{6} \cdot \frac{h-a+c}{h-a+c+d}$$

3. Conform formalizării generale a algoritmului EM, datele (observabile şi neobservabile) ar trebui să fie (în contextul acestei probleme) $x_1, \ldots, x_{h+c+d}$, fiecare x_i aparţinând mulţimii $\{A, B, C, D\}$. Datele observabile ar fi acele instanţe x_i care aparţin mulţimii $\{C, D\}$, iar datele neobservabile restul ($x_i \in \{A, B\}$). Cum ordinea instanţelor x_i nu contează (întrucât operăm cu distribuţia categorială, nu cea multinomială), se constată că este suficient să considerăm ca date observabile h, c şi d. Variabila neobservabilă va fi considerată a. (Corespunzător, vom avea $b = h - a$.) Se observă că toate cele patru linii din tabel vor fi necesare la scrierea funcţiei „auxiliare“ pentru algoritmul EM, care reprezintă media log-verosimilităţii datelor complete.

4. În mod normal, algoritmul EM face la pasul E calculul unei medii (care, în cazul mixturilor, este o probabilitate de forma $P(z_i = j | x_i, \theta^{(t)})$). În problema noastră, cunoscând $\theta^{(t)}$, valoarea parametrului θ la iteraţia t, vom putea calcula valoarea medie / „aşteptată“ a lui a, pe care o vom nota cu $\hat{a}$.

a. Pasul de calculare a mediilor (E): Care dintre formulele următoare reprezintă valorile „aşteptate“ (engl., expected values) pentru variabilele a şi b (văzute ca variabile aleatoare), exprimate în funcţie de parametrul θ?

(i) $\hat{a} = \frac{\frac{1}{2}}{\frac{1}{2}+h}\theta$ $\quad \hat{b} = \frac{\theta}{\frac{1}{2}+h}\theta$ $\qquad (ii)$ $\hat{a} = \frac{\frac{1}{2}}{\frac{1}{2}+\theta}h$ $\quad \hat{b} = \frac{\theta}{\frac{1}{2}+\theta}h$

(iii) $\hat{a} = \frac{\theta}{\frac{1}{2}+\theta}h$ $\quad \hat{b} = \frac{\frac{1}{2}}{\frac{1}{2}+\theta}h$ $\qquad (iv)$ $\hat{a} = \frac{\frac{1}{2}}{1+\theta^2}h$ $\quad \hat{b} = \frac{\theta}{1+\theta^2}h$

b. Pasul de maximizare (M): Având mediile pentru variabilele a şi b (notate cu $\hat{a}$ şi respectiv $\hat{b}$ ca mai sus), care dintre formulele următoare reprezintă estimarea de verosimilitate maximă a parametrului θ?

(i) $\hat{\theta} = \frac{h-\hat{a}+c}{6(h-\hat{a}+c+d)}$ $\qquad (ii)$ $\hat{\theta} = \frac{h-\hat{a}+d}{6(h-2\hat{a}-d)}$

(iii) $\hat{\theta} = \frac{h-\hat{a}}{6(h-2\hat{a}+c)}$ $\qquad (iv)$ $\hat{\theta} = \frac{2(h-\hat{a})}{3(h-\hat{a}+c+d)}$

Răspuns:

a. Valoarea medie pentru a se calculează astfel:

Notând $P(A \cup B)$ probabilitatea ca un student oarecare să ia fie calificativul A fie calificativul B, rezultă:[1022]

$$\hat{a} = P(A \mid A \cup B) \cdot h \overset{def.}{=} \frac{P(A)}{P(A) + P(B)} \cdot h = \frac{\frac{1}{2}}{\frac{1}{2} + \theta} \cdot h$$

La cea de-a doua egalitate am ţinut cont şi de faptul că A şi B, văzute ca mulţimi, sunt disjuncte.

Procedând în mod similar, vom obţine $\hat{b} = E[b \mid h, \theta] = \frac{\theta}{\frac{1}{2} + \theta} \cdot h$. Deci răspunsul corect este (ii).

b. La pasul M, algoritmul EM maximizează media log-verosimilităţii datelor complete în funcţie de θ. Ţinând cont de independenţa datelor, funcţia de verosimilitate a datelor complete se exprimă astfel:[1023]

$$P(a, b, c, d \mid \theta) = \left(\frac{1}{2}\right)^a (\theta)^b (2\theta)^c \left(\frac{1}{2} - 3\theta\right)^d.$$

Aşadar,

$$\ln P(a, b, c, d \mid \theta) = a \ln \frac{1}{2} + b \ln \theta + c \ln(2\theta) + d \ln \frac{1 - 6\theta}{2}$$

Înlocuind în această expresie variabilele necunoscute / neobservabile a şi b cu mediile lor, obţinem

$$E[\ln P(a, b, c, d \mid \theta)] = \hat{a} \ln \frac{1}{2} + \hat{b} \ln \theta + c \ln(2\theta) + d \ln \frac{1 - 6\theta}{2}$$

Maximizarea acestei medii (în raport cu parametrul θ) se realizează cu ajutorul derivatei de ordinul întâi:[1024]

$$\frac{\partial E[\ln p(a, b, c, d \mid \theta)]}{\partial \theta} = 0 \Leftrightarrow \frac{\hat{b}}{\theta} + \frac{2c}{2\theta} - \frac{3d}{\frac{1 - 6\theta}{2}} = 0 \Leftrightarrow \frac{\hat{b}}{\theta} + \frac{c}{\theta} - \frac{6d}{1 - 6\theta} = 0$$

$$\Leftrightarrow \frac{\hat{b} + c}{\theta} = \frac{6d}{1 - 6\theta} \Leftrightarrow 6d\theta = (\hat{b} + c)(1 - 6\theta) \Leftrightarrow 6\,\theta(d + \hat{b} + c) = \hat{b} + c$$

$$\Rightarrow \hat{\theta} = \frac{\hat{b} + c}{6(\hat{b} + c + d)} = \frac{h - \hat{a} + c}{6(h - \hat{a} + c + d)}$$

Se poate verifica imediat că $\hat{\theta} \in (0,\, 1/6)$. Prin urmare, răspunsul corect este (i).

***Observaţii*:**

[1022] Valoarea „aşteptată“ pentru a (numărul de studenţi care au obţinut calificativul A) este dată de media *distribuţiei binomiale* de parametri h şi $\frac{P(A)}{P(A) + P(B)}$. Se ştie că această medie este exact produsul celor doi parametri. Vedeţi problema 25.b de la capitolul de *Fundamente*.

[1023] În egalitatea următoare omitem factorul $\frac{(a + b + c + d)!}{a!\, b!\, c!\, d!}$ fiindcă nu depinde de θ.

[1024] Este imediat că derivata a doua este negativă pe tot domeniul ei de definiţie, deci funcţia de log-verosimilitate este strict concavă şi, în consecinţă, admite un singur punct de maxim.

5. Regula de actualizare care tocmai a fost obţinută la punctul b este un corespondent natural al formulei de estimare a parametrului θ corespunzătoare cazului când nu există date ascunse. (Vedeţi *Observaţia* 2 de mai sus.)
6. Folosind notaţia uzuală pentru algoritmul EM, putem spune că mediile $\hat{a} \overset{not.}{=} a^{(t+1)}$ şi $\hat{b} \overset{not.}{=} b^{(t+1)}$ au fost calculate în funcţie de „ipoteza“ curentă $\theta^{(t)}$, iar apoi expresia $E[\ln P(a, b, c, d \mid \theta)]$ a fost calculată în funcţie de θ, $\hat{a}$ şi $\hat{b}$ (deci în funcţie de θ şi $\theta^{(t)}$).

14. (Algoritmul EM pentru „învăţarea“ unei distribuţii multinomiale (şi implicit a unei distribuţii categoriale); o aplicaţie în domeniul bioinformaticii)

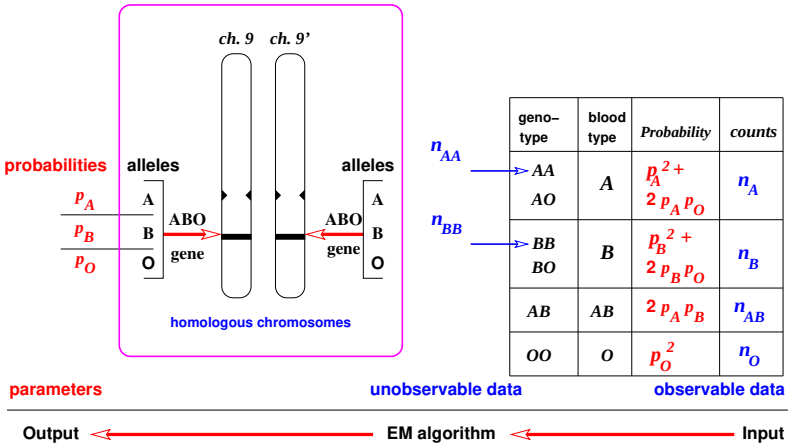
Liviu Ciortuz, 2017, pornind de la
■ □ ○ *Ex. 20.10 din "Probability for Statistics and Machine Learning" Anirban DasGupta, Springer, 2011*

Grupele sangvine ale oamenilor sunt determinate de variantele („alelele“) unei gene situate pe cromozomul 9 (mai exact, la poziţia 9q34.2), numită gena *ABO*. Se ştie că fiecare dintre noi dispunem de câte o pereche de astfel de cromozomi, deci de câte două copii ale genei *ABO*, şi anume una moştenită de la tată şi cealaltă moştenită de la mamă.

Se notează cu A, B şi O cele trei tipuri de alele ale genei *ABO*. Alelele A şi B sunt *dominante* în raport cu alela O. (Altfel spus, alela O este *recesivă* în raport cu fiecare dintre alelele A şi B.) Alelele A şi B sunt *codominante*. Prin urmare, grupele sangvine pot fi specificate conform tabelului alăturat.

grupe sangvine	alele moştenite
A	AA AO
B	BB BO
AB	AB
O	OO

Presupunem că, într-o populaţie oarecare, fiecare dintre alelele A, B şi O are la bărbaţi aceeaşi frecvenţă ca şi la femei. În cele ce urmează, aceste frecvenţe (notate respectiv cu p_A, p_B şi p_O) vor fi considerate a priori necunoscute, dar urmează să le determinăm.



a. Considerăm că într-un eşantion de populaţie format din n persoane sunt n_A persoane cu grupa sangvină A, n_B persoane cu grupa sangvină B, n_{AB} persoane cu grupa sangvină AB şi n_O persoane cu grupa sangvină O. (Evident, $n = n_a + n_B + n_{AB} + n_O$.) Pornind de la aceste date „observabile", să se deriveze algoritmul EM pentru determinarea probabilităţilor p_A, p_B şi p_O. (Evident, este suficient să se determine două dintre ele, fiindcă suma celor trei probabilităţi este 1.)

b. Implementaţi algoritmul EM pe care l-aţi conceput la punctul a şi rulaţi-l pentru inputul $n_A = 186$, $n_B = 38$, $n_{AB} = 13$, $n_O = 284$ ($n = 521$). Ca valori iniţiale pentru parametri, veţi lucra mai întâi cu $p_A = p_B = p_O = \frac{1}{3}$, iar apoi cu $p_A = p_O = 0.01$ şi $p_B = 0.98$. Pentru oprire, veţi cere ca $p_A^{(t)}$, $p_B^{(t)}$ şi $p_O^{(t)}$ să nu difere (fiecare în parte) cu mai mult de 10^{-4} faţă de valorile calculate la iteraţia precedentă. Comparaţi rezultatele obţinute pentru fiecare din cele două iniţializări.

***Indicaţii*:**

1. Presupunând că procesul de asociere a alelelor moştenite de către un individ de la părinţii lui respectă proprietăţile specifice evenimentelor aleatoare independente, rezultă că probabilităţile de realizare a combinaţiilor (în termeni genetici: „fenotipurile") AA, AO, BB, BO, AB şi OO într-o populaţie oarecare sunt p_A^2, $2p_Ap_O$, p_B^2, $2p_Bp_O$, $2p_Ap_B$, şi respectiv p_O^2.
2. Considerând $n_A = n_{AA} + n_{AO}$ şi $n_B = n_{BB} + n_{BO}$, unde semnificaţiile numerelor n_{AA}, n_{AO}, n_{BB} şi n_{BO} sunt similare cu semnificaţiile numerelor n_A, n_B, n_{AB}, şi n_O care au fost precizate mai sus, este natural ca în formularea algoritmului EM datele n_{AA} şi n_{BB} să fie considerate „neobservabile". Ca *parametri* ai modelului, se vor considera probabilităţile p_A, p_B şi p_O.
3. La pasul E al algoritmului EM veţi calcula $\hat{n}_{AA}$ şi $\hat{n}_{BB}$, care reprezintă numărul „aşteptat" de apariţii ale combinaţiei de alele AA şi respectiv numărul „aşteptat" de apariţii ale combinaţiei de alele BB în populaţia dată. Veţi scrie apoi funcţia de log-verosimilitate a datelor complete („observabile" şi „neobservabile"), exprimată cu ajutorul distribuţiei $Multinomial(n; p_A^2, 2p_Ap_O, p_B^2, 2p_Bp_O, 2p_Ap_B, p_O^2)$.
4. La pasul M al algoritmului EM, pornind de la media funcţiei de log-verosimilitate care a fost calculată la pasul E, veţi stabili regulile de actualizare pentru probabilităţile p_A, p_B şi p_O. Atenţie: suma acestor probabilităţi fiind 1, problema de optimizare pe care va trebui să o rezolvaţi la pasul M este una cu restricţii. În acest sens, *metoda multiplicatorilor lui Lagrange* vă poate fi de folos.

Răspuns:

Vom folosi următoarele notaţii:

- **setul de parametri: $p = \{p_A, p_B, p_O\}$;**
  de fapt, va fi suficient să estimăm doar p_A şi p_B, fiindcă $p_A + p_B + p_O = 1$
- **variabilele observabile: $n_{obs} = \{n_A, n_B, n_{AB}, n_O\}$**
- **variabilele neobservabile: $n_{unobs} = \{n_{AA}, n_{AO}, n_{BB}, n_{BO}\}$;**
  de fapt, vom reduce setul de variabile n_{unobs} la $\{n_{AA}, n_{BB}\}$, pentru că $n_{AO} = n_A - n_{AA}$ şi $n_{BO} = n_B - n_{BB}$;

– **datele complete:** $n_{compl} = n_{obs} \cup n_{unobs}$

– $n = n_A + n_B + n_{AB} + n_O$, $\quad n_A = n_{AA} + n_{AO}$, $\quad n_B = n_{BB} + n_{BO}$.

Pornind de la definiţia funcţiei masă de probabilitate (p.m.f.) pentru *distribuţia multinomială*, rezultă că expresia funcţiei de verosimilitate a datelor complete din problema noastră este următoarea:[1025]

$$L(p) \stackrel{not.}{=} P(n_{compl}|p) = \frac{n!}{n_{AA}!\, n_{AO}!\, n_{BB}!\, n_{BO}!\, n_{AB}!\, n_O!} \cdot (p_A^2)^{n_{AA}} \cdot (2\, p_A\, p_O)^{n_{AO}} \cdot (p_B^2)^{n_{BB}} \cdot (2\, p_B\, p_O)^{n_{BO}} \cdot (2\, p_A\, p_B)^{n_{AB}} \cdot (p_O^2)^{n_O}$$

Logaritmând această expresie, obţinem funcţia de log-verosimilitate a datelor complete:

$$\begin{aligned}
\ell(p) &\stackrel{def.}{=} \ln L(p) \\
&= c + n_{AA} \ln(p_A^2) + n_{AO} \ln(2\, p_A\, p_O) + n_{BB} \ln(p_B^2) + n_{BO} \ln(2\, p_B\, p_O) + \\
&\quad\; n_{AB} \ln(2\, p_A\, p_B) + n_O \ln(p_O^2) \\
&= c' + 2\, n_{AA} \ln p_A + n_{AO}(\ln p_A + \ln p_O) + \\
&\quad\; 2\, n_{BB} \ln p_B + n_{BO}(\ln p_B + \ln p_O) + n_{AB}(\ln p_A + \ln p_B) + 2\, n_O \ln p_O \\
&= c' + 2\, n_{AA} \ln p_A + (n_A - n_{AA})(\ln p_A + \ln p_O) + \\
&\quad\; 2\, n_{BB} \ln p_B + (n_B - n_{BB})(\ln p_B + \ln p_O) + n_{AB}(\ln p_A + \ln p_B) + 2\, n_O \ln p_O,
\end{aligned}$$

unde c şi c' sunt constante care nu depind de parametrul p.

Funcţia auxiliară se obţine din expresia acestei funcţii de log-verosimilitate, prin aplicarea operatorului E, care se referă la media variabilelor neobservabile:

$$\begin{aligned}
Q(p|p^{(t)}) &\stackrel{def.}{=} E[\ell(p)|n_{obs}\,; p^{(t)}] \\
&= c' + 2\, \hat{n}_{AA} \ln p_A + (n_A - \hat{n}_{AA})(\ln p_A + \ln p_O) + \\
&\quad\; 2\, \hat{n}_{BB} \ln p_B + (n_B - \hat{n}_{BB})(\ln p_B + \ln p_O) + n_{AB}(\ln p_A + \ln p_B) + 2\, n_O \ln p_O,
\end{aligned}$$

unde

$$\begin{aligned}
\hat{n}_{AA} &\stackrel{not.}{=} E[n_{AA}|n_{obs}\,; p^{(t)}] = E[n_{AA}|n_A, n_B, n_{AB}, n_O; p_A^{(t)}, p_B^{(t)}, p_O^{(t)}] \\
\hat{n}_{BB} &\stackrel{not.}{=} E[n_{BB}|n_{obs}\,; p^{(t)}] = E[n_{BB}|n_A, n_B, n_{AB}, n_O; p_A^{(t)}, p_B^{(t)}, p_O^{(t)}].
\end{aligned}$$

Pasul E:

După cum se observă în expresia funcţiei auxiliare $Q(p|p^{(t)})$, acum trebuie să calculăm mediile $\hat{n}_{AA}$ şi $\hat{n}_{BB}$, adică numărul „aşteptat“ (engl., expected number) de persoane care au *fenotipul* (adică, perechea de alele) AA, şi respectiv BB. Ţinând cont că n_{AA} (sau, echivalent, fenotipul AA), văzut ca variabilă aleatoare, urmează *distribuţia binomială* de parametri n_A şi $\frac{(p_A^{(t)})^2}{(p_A^{(t)})^2 + 2\, p_A^{(t)}\, p_O^{(t)}}$, rezultă că media sa este:[1026]

[1025] Observaţi că factorul $\frac{n!}{n_{AA}!\, n_{AO}!\, n_{BB}!\, n_{BO}!\, n_{AB}!\, n_O!}$ generalizează C_n^k (combinări de n luate câte k), ceea ce este natural, întrucât distribuţia multinomială este o generalizare a distribuţiei binomiale. (Vedeţi problema 25 de la capitolul de *Fundamente*.)

[1026] Vedeţi problema 25.b de la capitolul de *Fundamente*.

$$\begin{aligned}\hat{n}_{AA} &\overset{not.}{=} E[n_{AA}|n_A, n_B, n_{AB}, n_O; p_A^{(t)}, p_B^{(t)}, p_O^{(t)}] \\ &= \frac{(p_A^{(t)})^2}{(p_A^{(t)})^2 + 2\, p_A^{(t)}\, p_O^{(t)}} \cdot n_A. \end{aligned} \tag{433}$$

În mod similar,

$$\begin{aligned}\hat{n}_{BB} &\overset{not.}{=} E[n_{BB}|n_A, n_B, n_{AB}, n_O; p_A^{(t)}, p_B^{(t)}, p_O^{(t)}] \\ &= \frac{(p_B^{(t)})^2}{(p_B^{(t)})^2 + 2\, p_B^{(t)}\, p_O^{(t)}} \cdot n_B. \end{aligned} \tag{434}$$

Pasul M:

Întrucât probabilităţile p_A, p_B şi p_O trebuie să satisfacă restricţia $p_A + p_B + p_O = 1$, vom aplica metoda multiplicatorilor lui Lagrange şi vom introduce variabila / „multiplicatorul“ $\lambda \in \mathbb{R}$ cu scopul de rezolva *problema de optimizare* următoare:

$$\begin{aligned} p^{(t+1)} &\overset{not.}{=} (p_A^{(t+1)}, p_B^{(t+1)}, p_O^{(t+1)}) \\ &= \underset{p_A,\, p_B,\, p_O}{\operatorname{argmax}} [Q(p_A, p_B, p_O|p_A^{(t)}, p_B^{(t)}, p_O^{(t)}) + \lambda(1 - (p_A + p_B + p_O))] \\ &= \underset{p_A,\, p_B,\, p_O}{\operatorname{argmax}} [c' + 2\,\hat{n}_{AA} \ln p_A + (n_A - \hat{n}_{AA})(\ln p_A + \ln p_O) + 2\,\hat{n}_{BB} \ln p_B + \\ &\qquad (n_B - \hat{n}_{BB})(\ln p_B + \ln p_O) + n_{AB}(\ln p_A + \ln p_B) + 2\, n_O \ln p_O + \\ &\qquad \lambda(1 - (p_A + p_B + p_O))]. \end{aligned}$$

Calculând derivatele parţiale ale *funcţiei obiectiv* din această problemă de optimizare în raport cu p_A, p_B şi p_O şi egalându-le apoi cu 0, vom avea:

$$\frac{1}{p_A}(2\,\hat{n}_{AA} + n_A - \hat{n}_{AA} + n_{AB}) - \lambda = 0 \Rightarrow \hat{p}_A = \frac{1}{\lambda}(\hat{n}_{AA} + n_A + n_{AB})$$

$$\frac{1}{p_B}(2\,\hat{n}_{BB} + n_B - \hat{n}_{BB} + n_{AB}) - \lambda = 0 \Rightarrow \hat{p}_B = \frac{1}{\lambda}(\hat{n}_{BB} + n_B + n_{AB})$$

$$\frac{1}{p_O}(n_A - \hat{n}_{AA} + n_B - \hat{n}_{BB} + 2\, n_O) - \lambda = 0 \Rightarrow \hat{p}_O = \frac{1}{\lambda}(n_A - \hat{n}_{AA} + n_B - \hat{n}_{BB} + 2n_O).$$

În continuare, aplicând restricţia $\hat{p}_A + \hat{p}_B + \hat{p}_O = 1$, rezultă:

$$\frac{1}{\lambda}(\cancel{\hat{n}_{AA}} + n_A + n_{AB} + \cancel{\hat{n}_{BB}} + n_B + n_{AB} + n_A - \cancel{\hat{n}_{AA}} + n_B - \cancel{\hat{n}_{BB}} + 2n_O) = 1 \Leftrightarrow$$
$$\frac{2}{\lambda}(n_A + n_B + n_{AB} + n_O) = 1.$$

Întrucât $n = n_A + n_B + n_{AB} + n_O$, obţinem imediat $\frac{1}{\lambda} 2n = 1$ şi, în consecinţă, $\lambda = 2n$. Înlocuindu-l pe λ cu această valoare ($2n$) în expresiile pe care le-am obţinut anterior pentru $\hat{p}_A$, $\hat{p}_B$ şi $\hat{p}_O$, vom obţine următoarele *reguli de actualizare* pentru pasul M al algoritmului EM:

$$\hat{p}_A^{(t+1)} = \frac{1}{2n}(\hat{n}_{AA} + n_A + n_{AB}) \tag{435}$$

$$\hat{p}_B^{(t+1)} = \frac{1}{2n}(\hat{n}_{BB} + n_B + n_{AB}) \tag{436}$$

$$\hat{p}_O^{(t+1)} = \frac{1}{2n}(n_A - \hat{n}_{AA} + n_B - \hat{n}_{BB} + 2n_O) = \frac{1}{2n}(n_{AO} + n_{BO} + 2n_O). \tag{437}$$

Se observă uşor că toate aceste valori sunt în intervalul $[0, 1]$**.**

Sumarizând, relaţiile de actualizare pentru algoritmul EM sunt (433) **şi** (434) **pentru pasul E, şi respectiv** (435), (436) **şi** (437) **pentru pasul M.**

b. Rulând o implementare a algoritmului EM pe datele specificate în enunţ şi făcând iniţializările indicate acolo (şi menţionate în următoarele tabele), am obţinut rezultate:

Valori iniţiale:
$p_A = p_B = p_O = 1/3$**.**

Iteraţii:

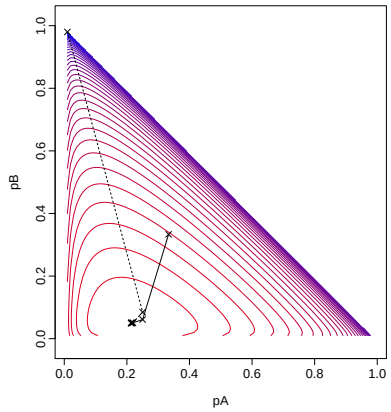
t	p_A	p_B	p_O
1	0.2505	0.0611	0.6884
2	0.2185	0.0505	0.7311
3	0.2142	0.0502	0.7357
4	0.2137	0.0501	0.7362
5	0.2136	0.0501	0.7363
6	0.2136	0.0501	0.7363

Valori iniţiale:
$p_A = p_O = 0.01$**,** $p_B = 0.98$**.**

Iteraţii:

t	p_A	p_B	p_O
1	0.2505	0.0847	0.6648
2	0.2193	0.0511	0.7296
3	0.2143	0.0502	0.7355
4	0.2137	0.0501	0.7362
5	0.2136	0.0501	0.7363
6	0.2136	0.0501	0.7363

Se constată că rezultatele finale care au fost obţinute de către algoritmul EM pornind de la cele două iniţializări diferite sunt identice. Acest fapt era de aşteptat, întrucât funcţia de log-verosimilitate a datelor observabile este strict concavă şi are un singur punct de maxim, aşa cum se observă şi în graficul alăturat, realizat de către Sebastian Ciobanu.



8.1.4 Sume de variabile aleatoare

15. **(Algoritmul EM: estimarea parametrilor pentru o sumă de două distribuţii exponenţiale)**

■ • ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW2, pr. 2.2***

Considerăm Z_1 **şi** Z_2 **două variabile aleatoare independente, distribuite exponenţial, de parametri** λ_1 **şi respectiv** λ_2**, cu** $\lambda_1 \neq \lambda_2$**.**[1027] **Vă reamintim că**

[1027] Notaţie: $Z_1 \sim exp\left(\frac{1}{\lambda_1}\right)$ şi $Z_2 \sim exp\left(\frac{1}{\lambda_2}\right)$.

funcţia densitate de probabilitate a unei variabile aleatoare exponenţiale Z de parametru $\lambda > 0$ este definită astfel:[1028]

$$f_\lambda(z) = \begin{cases} \lambda \cdot e^{-\lambda z} & \textbf{pentru } z \geq 0 \\ 0 & \textbf{pentru } z < 0. \end{cases}$$

Se consideră variabila aleatoare $X = Z_1 + Z_2$. Se dau instanţele $x_1, x_2, \ldots, x_n$ independente şi identic distribuite conform distribuţiei probabiliste a lui X.

a. Să se calculeze funcţia densitate de probabilitate a lui X în funcţie de parametrii λ_1 şi λ_2.

***Sugestie*: Calculaţi mai întâi funcţia de distribuţie cumulativă a lui X. Pentru aceasta, puteţi folosi formula $F(x) = \int_0^x \int_0^{x-z_1} f_{\lambda_1}(z_1) \cdot f_{\lambda_2}(z_2)\, dz_2\, dz_1$.**[1029]

b. Elaboraţi în detaliu cei doi paşi ai algoritmului EM pentru estimarea parametrilor λ_1 şi λ_2. Precizaţi la final care sunt ecuaţiile de actualizare pentru aceşti doi parametri, pornind de la setul de date $\{x_1, x_2, \ldots, x_n\}$.

Răspuns:

a. Funcţia de distribuţie cumulativă a lui X se calculează astfel:

$$\begin{aligned}
F(x) &= P(Z_1 + Z_2 < x) = \int_0^x \int_0^{x-z_1} f_{\lambda_1}(z_1) \cdot f_{\lambda_2}(z_2)\, dz_2\, dz_1 \\
&= \int_0^x \int_0^{x-z_1} \lambda_1 e^{-\lambda_1 z_1} \cdot \lambda_2 e^{-\lambda_2 z_2}\, dz_2\, dz_1 \\
&= \int_0^x (-\lambda_1) e^{-\lambda_1 z_1} \left(\int_0^{x-z_1} (-\lambda_2) e^{-\lambda_2 z_2}\, dz_2 \right) dz_1 \\
&= \int_0^x (-\lambda_1) e^{-\lambda_1 z_1} \left(e^{-\lambda_2 z_2} \Big|_0^{x-z_1} \right) dz_1 \\
&= \int_0^x (-\lambda_1) e^{-\lambda_1 z_1} \left(e^{-\lambda_2 (x-z_1)} - \underbrace{e^{-\lambda_2 \cdot 0}}_{1} \right) dz_1 \\
&= \int_0^x (-\lambda_1) e^{-\lambda_1 z_1} e^{-\lambda_2 (x-z_1)} dz_1 - \int_0^x (-\lambda_1) e^{-\lambda_1 z_1} dz_1 \\
&= \frac{-\lambda_1}{\lambda_2 - \lambda_1} e^{-\lambda_2 x} \int_0^x (\lambda_2 - \lambda_1) e^{(\lambda_2 - \lambda_1) z_1} dz_1 - \int_0^x (-\lambda_1) e^{-\lambda_1 z_1} dz_1 \\
&= -\frac{\lambda_1}{\lambda_2 - \lambda_1} e^{-\lambda_2 x} \left(e^{(\lambda_2 - \lambda_1) z_1} \Big|_0^x \right) - e^{-\lambda_1 z_1} \Big|_0^x \\
&= -\frac{\lambda_1}{\lambda_2 - \lambda_1} e^{-\lambda_2 x} \left(e^{(\lambda_2 - \lambda_1) x} - 1 \right) - \left(e^{-\lambda_1 x} - 1 \right) \\
&= -\frac{\lambda_1}{\lambda_2 - \lambda_1} e^{-\lambda_1 x} + \frac{\lambda_1}{\lambda_2 - \lambda_1} e^{-\lambda_2 x} - e^{-\lambda_1 x} + 1
\end{aligned}$$

[1028] Pentru graficul acestei funcţii, a se vedea problema 31.a de la capitolul de *Fundamente*.

[1029] Deducerea acestei formule este destul de simplă:

$$\begin{aligned}
F(x) &\overset{def.}{=} P(Z_1 + Z_2 < x) \\
&= \int_0^x \left(\int_0^{x-z_2} f_{\lambda_1}(z_1)\, dz_1 \right) \cdot f_{\lambda_2}(z_2)\, dz_2 \\
&= \int_0^x \left(\int_0^{x-z_1} f_{\lambda_2}(z_2)\, dz_2 \right) \cdot f_{\lambda_1}(z_1)\, dz_1 = \int_0^x \int_0^{x-z_1} f_{\lambda_1}(z_1) \cdot f_{\lambda_2}(z_2)\, dz_2\, dz_1.
\end{aligned}$$

$$
\begin{aligned}
&= 1 - \frac{1}{\lambda_2 - \lambda_1}\left(\lambda_1 e^{-\lambda_1 x} - \lambda_1 e^{-\lambda_2 x} + \lambda_2 e^{-\lambda_1 x} - \lambda_1 e^{-\lambda_1 x}\right) \\
&= 1 - \frac{\lambda_2 e^{-\lambda_1 x} - \lambda_1 e^{-\lambda_2 x}}{\lambda_2 - \lambda_1}
\end{aligned}
$$

Cunoscând funcţia de distribuţie cumulativă a lui X, vom calcula funcţia de densitate de probabilitate a lui X:

$$p(x) = \frac{\partial F(x)}{\partial x} = -\frac{1}{\lambda_2 - \lambda_1}\left(-\lambda_1\lambda_2 e^{-\lambda_1 x} + \lambda_1\lambda_2 e^{-\lambda_2 x}\right) = \frac{\lambda_1\lambda_2}{\lambda_2 - \lambda_1}\left(e^{-\lambda_1 x} - e^{-\lambda_2 x}\right)$$

b. Vom prezenta două variante / metode de elaborare a algoritmului EM pentru estimarea parametrilor λ_1 şi λ_2:

- **Prima metodă constă în aplicarea clasică a schemei algoritmice EM:**

***Pasul E (Expectation)*: Se calculează funcţia „auxiliară" (media log-verosimilităţii datelor complete) pentru iteraţia t:**

$$Q(\lambda \mid \lambda^{(t)}) = E_{P(Z|X,\lambda^{(t)})}[\log P(X, Z \mid \lambda)]$$

***Pasul M (Maximization)*: Se maximizează media log-verosimilităţii datelor complete, calculată la pasul E, în raport cu λ:**

$$\lambda^{(t+1)} = \underset{\lambda}{\operatorname{argmax}}\, Q(\lambda \mid \lambda^{(t)})$$

Notaţiile de mai sus au următoarele semnificaţii:

$$
\begin{aligned}
\lambda &= (\lambda_1, \lambda_2) \\
\lambda^{(t)} &= \text{ valoarea parametrului } \lambda \text{ la iteraţia } t \\
X &= \text{ variabila observabilă, cu instanţele } x_1, x_2, \ldots, x_n \\
Z &= (Z_1, Z_2) \text{ variabilele ascunse / neobservabile} \\
&\quad \text{având valorile } z_{1j}, z_{2j}, \ldots, z_{nj} \text{ cu } j \in \{1, 2\}, \\
&\quad \text{aşa încât } x_i = z_{i1} + z_{i2}.
\end{aligned}
$$

În continuare vom elabora calculele corespunzătoare celor doi paşi (E şi M).

(Pasul E) Expresia funcţiei „auxiliare" este:

$$
\begin{aligned}
Q(\lambda \mid \lambda^{(t)}) &= E_{P(Z|X,\lambda^{(t)})}\left[\log \prod_{i=1}^{n} p(x_i, z_{i1}, z_{i2} \mid \lambda)\right] \\
&= E_{P(Z|X,\lambda^{(t)})}\left[\log \prod_{i=1}^{n} f_{\lambda_1}(z_{i1}) \cdot f_{\lambda_2}(z_{i2})\right] \\
&= E_{P(Z|X,\lambda^{(t)})}\left[\sum_{i=1}^{n} \log\left(\lambda_1 e^{-\lambda_1 z_{i1}} \cdot \lambda_2 e^{-\lambda_2 z_{i2}}\right)\right] \\
&= E_{P(Z|X,\lambda^{(t)})}\left[\sum_{i=1}^{n} \left(\log\lambda_1 - \lambda_1 z_{i1} + \log\lambda_2 - \lambda_2 z_{i2}\right)\right] \\
&= E_{P(Z|X,\lambda^{(t)})}\left[n\log\lambda_1 + n\log\lambda_2 - \sum_{i=1}^{n}\left(\lambda_1 z_{i1} + \lambda_2 z_{i2}\right)\right] \\
&= n\log\lambda_1 + n\log\lambda_2 - \lambda_1 \sum_{i=1}^{n} E_{p(z_{i1}|x_i,\lambda^{(t)})}[z_{i1}] - \lambda_2 \sum_{i=1}^{n} E_{p(z_{i2}|x_i,\lambda^{(t)})}[z_{i2}]
\end{aligned}
$$

Ultima egalitate de mai sus are loc datorită proprietăţii de liniaritate a mediilor.

Vom trece acum la calcularea mediilor variabilelor neobservabile. Media variabilei z_{i1} în raport cu probabilitatea condiţionată $p(z_{i1} \mid x_i, \lambda^{(t)})$ este:

$$E_{p(z_{i1}|x_i,\lambda^{(t)})}[z_{i1}] \stackrel{def.}{=} \int_0^{x_i} z_{i1} \cdot p(z_{i1} \mid x_i, \lambda^{(t)})\, dz_{i1}$$

Aşadar, pentru a calcula această medie trebuie să obţinem mai întâi expresia probabilităţii condiţionate

$$p(z_{i1} \mid x_i, \lambda^{(t)}) \stackrel{def.}{=} \frac{p(z_{i1}, x_i \mid \lambda^{(t)})}{p(x_i \mid \lambda^{(t)})}$$

Probabilitatea de la numitor a fost calculată la punctul a, iar pentru cea de la numărător vom folosi relaţia $x_i = z_{i1} + z_{i2}$. Prin urmare,

$$p(z_{i1} \mid x_i, \lambda^{(t)}) = \frac{p(z_{i1}, z_{i2}|\lambda^{(t)})}{p(x_i \mid \lambda^{(t)})} = \frac{p(z_{i1}|\lambda_1^{(t)}) \cdot p(z_{i2}|\lambda_2^{(t)})}{p(x_i \mid \lambda^{(t)})},$$

ultima egalitate având loc datorită faptului că variabilele aleatoare Z_1 şi Z_2 sunt independente. În consecinţă,

$$\begin{aligned} p(z_{i1} \mid x_i, \lambda^{(t)}) &= \frac{f_{\lambda_1^{(t)}}(z_{i1}) \cdot f_{\lambda_2^{(t)}}(x_i - z_{i1})}{\frac{\lambda_1^{(t)}\lambda_2^{(t)}}{\lambda_2^{(t)} - \lambda_1^{(t)}} \left(e^{-\lambda_1^{(t)} x_i} - e^{-\lambda_2^{(t)} x_i}\right)} \\ &= (\lambda_2^{(t)} - \lambda_1^{(t)}) \cdot \frac{\lambda_1^{(t)} e^{-\lambda_1^{(t)} z_{i1}} \cdot \lambda_2^{(t)} e^{-\lambda_2^{(t)}(x_i - z_{i1})}}{\lambda_1^{(t)}\lambda_2^{(t)} \left(e^{-\lambda_1^{(t)} x_i} - e^{-\lambda_2^{(t)} x_i}\right)} \\ &= (\lambda_2^{(t)} - \lambda_1^{(t)}) \cdot \frac{e^{(\lambda_2^{(t)} - \lambda_1^{(t)}) z_{i1}}}{\left(e^{-\lambda_1^{(t)} x_i} - e^{-\lambda_2^{(t)} x_i}\right) \cdot e^{\lambda_2^{(t)} x_i}} = (\lambda_2^{(t)} - \lambda_1^{(t)}) \cdot \frac{e^{(\lambda_2^{(t)} - \lambda_1^{(t)}) z_{i1}}}{e^{(\lambda_2^{(t)} - \lambda_1^{(t)}) x_i} - 1} \end{aligned}$$

Aşadar, media variabilei z_{i1} în raport cu probabilitatea condiţionată $p(z_{i1} \mid x_i, \lambda^{(t)})$ este:

$$\begin{aligned} E_{p(z_{i1}|x_i,\lambda^{(t)})}[z_{i1}] &\stackrel{def.}{=} \int_0^{x_i} z_{i1} \cdot p(z_{i1} \mid x_i, \lambda^{(t)})\, dz_{i1} \\ &= \int_0^{x_i} z_{i1} \cdot (\lambda_2^{(t)} - \lambda_1^{(t)}) \cdot \frac{e^{(\lambda_2^{(t)} - \lambda_1^{(t)}) z_{i1}}}{e^{(\lambda_2^{(t)} - \lambda_1^{(t)}) x_i} - 1}\, dz_{i1} \\ &= \frac{1}{e^{(\lambda_2^{(t)} - \lambda_1^{(t)}) x_i} - 1} \int_0^{x_i} z_{i1} \cdot (\lambda_2^{(t)} - \lambda_1^{(t)}) \cdot e^{(\lambda_2^{(t)} - \lambda_1^{(t)}) z_{i1}}\, dz_{i1} \end{aligned}$$

Vom rezolva ultima integrală de mai sus utilizând formula de integrare prin părţi:

$$\int f \cdot g' = f \cdot g - \int f' \cdot g$$

Aşadar,

$$\begin{aligned}
&\int_0^{x_i} z_{i1}\cdot(\lambda_2^{(t)}-\lambda_1^{(t)})\cdot e^{(\lambda_2^{(t)}-\lambda_1^{(t)})z_{i1}}\,dz_{i1}\\
&= \int_0^{x_i} z_{i1}\cdot\frac{\partial}{\partial z_{i1}}\left(e^{(\lambda_2^{(t)}-\lambda_1^{(t)})z_{i1}}\right)dz_{i1}\\
&= \left(z_{i1}\cdot e^{(\lambda_2^{(t)}-\lambda_1^{(t)})z_{i1}}\right)\Big|_0^{x_i} - \int_0^{x_i} e^{(\lambda_2^{(t)}-\lambda_1^{(t)})z_{i1}}\,dz_{i1}\\
&= \left(x_i\cdot e^{(\lambda_2^{(t)}-\lambda_1^{(t)})x_i}-0\right) - \frac{1}{\lambda_2^{(t)}-\lambda_1^{(t)}}\cdot e^{(\lambda_2^{(t)}-\lambda_1^{(t)})z_{i1}}\Big|_0^{x_i}\\
&= x_i\cdot e^{(\lambda_2^{(t)}-\lambda_1^{(t)})x_i} - \frac{e^{(\lambda_2^{(t)}-\lambda_1^{(t)})x_i}-1}{\lambda_2^{(t)}-\lambda_1^{(t)}}
\end{aligned}$$

Prin urmare,

$$\begin{aligned}
E_{p(z_{i1}|x_i,\lambda^{(t)})}[z_{i1}] &= \frac{1}{e^{(\lambda_2^{(t)}-\lambda_1^{(t)})x_i}-1}\cdot\left(x_i\cdot e^{(\lambda_2^{(t)}-\lambda_1^{(t)})x_i} - \frac{e^{(\lambda_2^{(t)}-\lambda_1^{(t)})x_i}-1}{\lambda_2^{(t)}-\lambda_1^{(t)}}\right)\\
&= \frac{x_i\cdot e^{(\lambda_2^{(t)}-\lambda_1^{(t)})x_i}}{e^{(\lambda_2^{(t)}-\lambda_1^{(t)})x_i}-1} - \frac{1}{\lambda_2^{(t)}-\lambda_1^{(t)}}
\end{aligned}$$

Pentru a calcula media lui z_{i2} în raport cu probabilitatea condiţionată $p(z_{i2} \mid x_i, \lambda^{(t)})$, vom utiliza relaţia $x_i = z_{i1} + z_{i2}$ şi media lui z_{i1} calculată anterior:

$$\begin{aligned}
E_{p(z_{i2}|x_i,\lambda^{(t)})}[z_{i2}] &= E_{p(z_{i2}|x_i,\lambda^{(t)})}[x_i - z_{i1}] = x_i - E_{p(z_{i2}|x_i,\lambda^{(t)})}[z_{i1}]\\
&= x_i - E_{p(z|x_i,\lambda^{(t)})}[z_{i1}] = x_i - E_{p(z_{i1}|x_i,\lambda^{(t)})}[z_{i1}]\\
&= x_i - \frac{x_i\cdot e^{(\lambda_2^{(t)}-\lambda_1^{(t)})x_i}}{e^{(\lambda_2^{(t)}-\lambda_1^{(t)})x_i}-1} + \frac{1}{\lambda_2^{(t)}-\lambda_1^{(t)}}\\
&= \frac{1}{\lambda_2^{(t)}-\lambda_1^{(t)}} - \frac{x_i}{e^{(\lambda_2^{(t)}-\lambda_1^{(t)})x_i}-1}
\end{aligned}$$

(Pasul M) Etapa de maximizare: Folosind expresia obţinută pentru funcţia auxiliară Q, vom calcula

$$\lambda^{(t+1)} = \underset{\lambda_1>0,\lambda_2>0}{\operatorname{argmax}}\Big(n\log\lambda_1 + n\log\lambda_2 - \lambda_1\sum_{i=1}^{n}E_{p(z_{i1}|x_i,\lambda^{(t)})}[z_{i1}] - \\ -\lambda_2\sum_{i=1}^{n}E_{p(z_{i2}|x_i,\lambda^{(t)})}[z_{i2}]\Big)$$

Valorile parametrilor λ_1 şi λ_2 corespunzătoare maximului acestei funcţii se obţin folosind metoda derivatelor parţiale:

$$\frac{\partial}{\partial\lambda_1}Q(\lambda \mid \lambda^{(t)}) = 0 \Leftrightarrow \frac{n}{\lambda_1} = \sum_{i=1}^{n}E_{p(z_{i1}|x_i,\lambda^{(t)})}[z_{i1}] \Rightarrow \lambda_1^{(t+1)} = \frac{n}{\sum_{i=1}^{n}E_{p(z_{i1}|x_i,\lambda^{(t)})}[z_{i1}]} > 0$$

$$\frac{\partial}{\partial\lambda_2}Q(\lambda \mid \lambda^{(t)}) = 0 \Leftrightarrow \frac{n}{\lambda_2} = \sum_{i=1}^{n}E_{p(z_{i2}|x_i,\lambda^{(t)})}[z_{i2}] \Rightarrow \lambda_2^{(t+1)} = \frac{n}{\sum_{i=1}^{n}E_{p(z_{i2}|x_i,\lambda^{(t)})}[z_{i2}]} > 0$$

Se verifică imediat că aceste soluţii într-adevăr maximizează $Q(\lambda \mid \lambda^{(t)})$.

• **O a doua metodă de estimare a parametrilor λ_1 şi λ_2 constă în aplicarea schemei algoritmice EM pentru funcţia / variabila $X = Z_1 + Z_2$, urmând ideile de bază ale acestei scheme, dar făcând la pasul M [în locul calculelor] analogia uzuală cu soluţia metodei de estimare directă (MLE) pentru parametrul unei distribuţii exponenţiale. Ca şi mai înainte, variabilele Z_1 şi Z_2 pot fi văzute ca variabile ascunse.**

***Pasul E (Expectation)*:**

Se calculează mediile variabilelor ascunse z_{i1} şi z_{i2}:

$$E_{p(z_{i1}|x_i,\lambda^{(t)})}[z_{i1}] \text{ şi } E_{p(z_{i1}|x_i,\lambda^{(t)})}[z_{i2}]$$

Calculele efective sunt cele realizate mai sus.

***Pasul M (Maximization)*:**

Se calculează / actualizează parametrul $\lambda \overset{not.}{=} (\lambda_1, \lambda_2)$ în funcţie de valorile medii obţinute la pasul precedent:[1030]

Deoarece $Z_1 \sim exp\left(\frac{1}{\lambda_1}\right)$, iar estimarea în sensul verosimilităţii maxime (MLE) a parametrului distribuţiei exponenţiale este inversul mediei setului de exemple (vedeţi problema 49 de la capitolul de *Fundamente*), aplicând operatorul E (media) şi ţinând cont de proprietatea de liniaritate a mediilor, rezultă:

$$\frac{1}{\lambda_1} = \frac{1}{n}\sum_{i=1}^{n} E_{p(z_{i1}|x_i,\lambda^{(t)})}[z_{i1}] \Rightarrow \lambda_1^{(t+1)} = \frac{n}{\sum_{i=1}^{n} E_{p(z_{i1}|x_i,\lambda^{(t)})}[z_{i1}]}$$

Similar, pentru $Z_2 \sim exp\left(\frac{1}{\lambda_2}\right)$ obţinem:

$$\frac{1}{\lambda_2} = \frac{1}{n}\sum_{i=1}^{n} E_{p(z_{i2}|x_i,\lambda^{(t)})}[z_{i2}] \Rightarrow \lambda_2^{(t+1)} = \frac{n}{\sum_{i=1}^{n} E_{p(z_{i2}|x_i,\lambda^{(t)})}[z_{i2}]}$$

16. **(Algoritmul EM pentru învăţarea parametrilor a două distribuţii gaussiene, pornind de la instanţe generate de *suma* a două variabile care urmează aceste distribuţii)**

■ □ • ○ ***Stanford, 2016 fall, A. Ng, J. Duchi, HW4, pr. 2***

La o conferinţă de învăţare automată au fost trimise spre recenzare şi, eventual, publicare P lucrări (engl., papers). Comitetul de recenzare a lucrărilor este format din R recenzori. Fiecare dintre aceşti R recenzori va citi toate cele P lucrări şi va da fiecărei lucrări un scor, indicând astfel cât de bună crede el că este lucrarea respectivă. Vom nota cu x_{pr} scorul pe care recenzorul r îl atribuie lucrării p. Dacă scorul este mare, înseamnă că recenzorului i-a plăcut lucrarea respectivă; acest scor reprezintă o recomandare din partea recenzorului ca lucrarea respectivă să fie acceptată pentru prezentare la conferinţă. Dacă scorul este mic, înseamnă că recenzorului nu i-a plăcut lucrarea respectivă.

[1030] Este imediat că estimarea parametrului λ_1 poate fi făcută în mod independent de estimarea parametrului λ_2 (până la distribuţia probabilistă în raport cu care s-au calculat mediile de la pasul E).

Vom presupune că fiecare lucrare p are o anumită valoare „intrinsecă“, pe care o vom nota cu μ_p; atunci când valoarea aceasta este mare înseamnă că lucrarea respectivă este bună. Fiecare recenzor încearcă să estimeze, în urma citirii / analizării lucrării p, cât este μ_p. În mod concret, scorul x_{pr}, care este raportat de către recenzorul r, reprezintă încercarea acestuia de a ghici cât este μ_p.

Există tot soiul de factori aleatori care influenţează procesul de recenzare a lucrărilor. Din acest motiv, în această problemă vom folosi / propune un model care incorporează mai multe surse de „zgomot“ / perturbaţii (engl., noise).

Unii recenzori sunt înclinaţi să creadă că toate lucrările sunt bune; ei tind să acorde tuturor lucrărilor scoruri mari. Alţi recenzori sunt, din contră, foarte severi şi tind să acorde scoruri mici tuturor lucrărilor. În mod similar, este foarte posibil ca scorurile pe care doi recenzori diferiţi le acordă lucrărilor pe care le recenzează să manifeste varianţe / dispersii foarte diferite, ceea ce înseamnă că unii recenzori sunt mai credibili decât alţii.

Vom nota cu ν_r *bias*-ul recenzorului r; asta înseamnă că scorurile puse de acest recenzor tind în general să fie cu cantitatea ν_r mai mari decât ar trebui să fie.

Aşadar, din punct de vedere formal vom presupune că scorurile puse de recenzori sunt generate de către un proces aleatoriu, care este definit astfel:

$$\begin{aligned} y_{pr} &\sim \mathcal{N}(\mu_p, \sigma_p^2), \\ z_{pr} &\sim \mathcal{N}(\nu_r, \tau_r^2), \\ x_{pr}|y_{pr}, z_{pr} &\sim \mathcal{N}(y_{pr} + z_{pr}, \sigma^2). \end{aligned}$$

Variabilele y_{pr} şi z_{pr} sunt independente; variabilele comune (x, y, z) care corespund unor perechi diferite de tip lucrare-recenzor sunt de asemenea independente. Pe lângă aceasta, trebuie precizat că noi observăm doar valorile variabilelor x_{pr}; aşadar, toate variabilele y_{pr} şi z_{pr} sunt latente / neobservabile.

Ceea ce dorim este să estimăm valorile parametrilor μ_p, σ_p^2, ν_r şi τ_r^2 pentru $p = 1, \ldots, P$ şi $r = 1, \ldots, R$. Din raţiuni de simplitate, vom trata σ^2 (varianţa condiţională a variabilei x_{pr} în raport cu y_{pr} şi z_{pr}) ca şi cum ar fi o constantă cunoscută, fixată. Dacă vom obţine estimări bune pentru valorile „intrinseci“ (μ_p) ale lucrărilor, atunci aceste estimări vor putea fi folosite pentru fundamentarea deciziilor care trebuie luate în privinţa acceptării / respingerii lucrărilor pentru [prezentare la] conferinţă.

Vom estima valorile parametrilor μ_p, σ_p^2, ν_r şi τ_r^2 maximizând verosimilitatea datelor observabile $\{x_{pr} \ ; \ p = 1, \ldots, P, \ r = 1, \ldots, R\}$. În această problemă, aşa cum am precizat deja, variabilele latente sunt y_{pr} şi z_{pr}, iar maximizarea verosimilităţii nu se poate face în mod direct (engl., in closed form). Prin urmare, vom folosi algoritmul EM.

Sarcina dumneavoastră va fi să derivaţi regulile de actualizare specifice celor doi paşi (E şi M) ai algoritmului EM. Aceste reguli vor putea să conţină doar operatori de adunare, scădere, înmulţire, împărţire, log, exp **şi extragere de rădăcină pătrată ($\sqrt{\ }$) din constante reale, precum şi adunare, scădere, înmulţire, inversare de matrice de numere reale şi calculul de determinanţi.**

a. La acest punct veţi obţine relaţiile de actualizare pentru pasul E:

***i*. Distribuţia comună $p(y_{pr}, z_{pr}, x_{pr})$ este de tip gaussian multidimensional. Găsiţi vectorul de medii asociat acestei distribuţii, precum şi matricea de**

covarianţă corespunzătoare, în funcţie de parametrii μ_p, σ_p^2, ν_r, τ_r^2 şi σ^2.

***Sugestie*: x_{pr} poate fi scris ca $x_{pr} = y_{pr} + z_{pr} + \varepsilon_{pr}$, unde $\varepsilon_{pr} \sim \mathcal{N}(0, \sigma^2)$ este un „zgomot", reprezentat de o variabilă de tip gaussian, independentă în raport cu y_{pr} şi z_{pr}.**

***ii*. Deduceţi expresia distribuţiei condiţionale $q_{pr}(y_{pr}, z_{pr}) \stackrel{def.}{=} p(y_{pr}, z_{pr}|x_{pr})$ de la pasul E, folosind regulile pentru condiţionare în raport cu submulţimi de variabile aleatoare gaussiene comune.**[1031]

b. Deduceţi regulile de actualizare de la pasul M pentru parametrii μ_p, ν_r, σ_p^2 şi τ_r^2.

***Sugestie*: S-ar putea să vă fie de folos să exprimaţi marginea inferioară a log-verosimilităţii datelor observabile ca o medie (engl., expectation) pentru valorile (y_{pr}, z_{pr}) generate de o distribuţie aleatoare având funcţia de densitate $q_{pr}(y_{pr}, z_{pr})$.**

***Comentariu*:** În articolul *Learning from the Wisdom of Crowds by Minimax Entropy* (NIPS, 2012), autorii Dengyong Zhou, John C. Platt,[1032] Sumit Basu şi Yi Mao au descris implementarea unei metode care este destul de asemănătoare cu cea pe care am prezentat-o în acest exerciţiu, cu scopul de a estima valorile „intrinseci" ale lucrărilor, μ_p. În acel articol, problema este ceva mai complicată, fiindcă nu toţi recenzorii evaluează fiecare lucrare, însă ideile de bază sunt în esenţă aceleaşi cu cele din acest exerciţiu. Întrucât modelul acesta încearcă să estimeze şi să corecteze bias-urile recenzorilor (ν_r), estimările obţinute pentru μ_p sunt mult mai folositoare pentru fundamentarea deciziilor de acceptare / respingere a lucrărilor decât scorurile „brute" acordate de recezori.

Răspuns:

Vom desemna cu θ întregul set de parametri pe care îi estimăm.
Paşii algoritmului EM pentru problema noastră se scriu — la un nivel înalt — astfel:

Pasul E: Pentru fiecare pereche p, r vom calcula

$$q_{pr}^{(t)}(y_{pr}, z_{pr}) \stackrel{not.}{=} p(y_{pr}, z_{pr}|x_{pr}; \theta^{(t)});$$

Pasul M: $\theta^{(t+1)} = \text{argmax}_\theta \sum_{p=1}^{P} \sum_{r=1}^{R} E_{q_{pr}^{(t)}(y_{pr}, z_{pr})} \ln p(x_{pr}, y_{pr}, z_{pr}; \theta)$.

Chestiunea care se pune acum este cum se calculează în mod concret aceste reguli de actualizare.

a. Pentru pasul E, facem următoarea *observaţie importantă*:

Dacă la calcularea probabilităţilor $p(y_{pr}, z_{pr}|x_{pr}; \theta^{(t)})$ am folosi regula lui Bayes combinată cu formula probabilităţii totale, atunci la numitor ar apărea integralele unor funcţii de densitate gaussiene, care n-ar fi deloc uşor de calculat. În schimb, putem observa că funcţia densitate de probabilitate

$$\begin{aligned} p(y_{pr}, z_{pr}, x_{pr}) &= p(y_{pr}, z_{pr}) \cdot p(x_{pr}|y_{pr}, z_{pr}) \\ &= p(y_{pr}) \cdot p(z_{pr}) \cdot p(x_{pr}|y_{pr}, z_{pr}) \end{aligned}$$

[1031] Vedeţi relaţia (39) din *Observaţia* finală de la rezolvarea problemei 38 de la capitolul de *Fundamente*.
[1032] John C. Platt este autorul algoritmului SMO (Sequential Minimal Optimization), care este „inima" implementărilor pentru maşinile cu vectori-suport (SVM). Vedeţi problemele 23 şi 52 de la capitolul *Maşini cu vectori-suport*.

este un produs de alte trei funcţii de densitate gaussiene, deci este şi ea o densitate *gaussiană multidimensională*.[1033]

Ca o *consecinţă*, distribuţia [comună şi] condiţionată $y_{pr}, z_{pr}|x_{pr}$ va fi, la rândul ei, tot o distribuţie gaussiană, pe care o vom putea calcula folosind *regulile pentru condiţionarea gaussienelor*.

Pentru a obţine expresia funcţiei de densitate pentru distribuţia comună (y_{pr}, z_{pr}, x_{pr}), vom ţine cont de faptul că o distribuţie gaussiană multidimensională este parametrizată [complet] de către vectorul medie şi matricea de covarianţă.

Pentru a calcula *vectorul medie* al distribuţiei comune (y_{pr}, z_{pr}, x_{pr}), vom rescrie x_{pr} în forma următoare:

$$x_{pr} = y_{pr} + z_{pr} + \varepsilon_{pr},$$

unde variabila $\varepsilon_{pr} \sim \mathcal{N}(0, \sigma^2)$ reprezintă un „zgomot“ (engl., noise) care urmează o distribuţie gaussiană şi este independentă de variabilele y_{pr} şi z_{pr}.[1034] **Aşadar,**

$$\begin{aligned} E[y_{pr}] &= \mu_p \\ E[z_{pr}] &= \nu_p \\ E[x_{pr}] &= E[y_{pr} + z_{pr} + \varepsilon_{pr}] = E[y_{pr}] + E[z_{pr}] + E[\varepsilon_{pr}] \\ &= \mu_p + \nu_r + 0 = \mu_p + \nu_r. \end{aligned}$$

Pentru a calcula *matricea de covarianţă* a distribuţiei comune (y_{pr}, z_{pr}, x_{pr}), observăm că $Var(y_{pr}) = \sigma_p^2$, $Var(z_{pr}) = \tau_r^2$, iar $Cov(y_{pr}, z_{pr}) = Cov(z_{pr}, y_{pr}) = 0$ întrucât y_{pr} şi z_{pr} sunt independente.[1035] **De asemenea, fiindcă y_{pr}, z_{pr} şi ε_{pr} sunt independente, vom avea:**[1036]

$$\begin{aligned} Var(x_{pr}) &= Var(y_{pr} + z_{pr} + \varepsilon_{pr}) = Var(y_{pr}) + Var(z_{pr}) + Var(\varepsilon_{pr}) \\ &= \sigma_p^2 + \tau_r^2 + \sigma^2. \end{aligned}$$

În fine,

$$\begin{aligned} Cov(y_{pr}, x_{pr}) &= Cov(x_{pr}, y_{pr}) = Cov(y_{pr} + z_{pr} + \varepsilon_{pr}, y_{pr}) \\ &= Cov(y_{pr}, y_{pr}) + Cov(z_{pr}, y_{pr}) + Cov(\varepsilon_{pr}, y_{pr}) \\ &= \sigma_p^2 + 0 + 0 = \sigma_p^2, \end{aligned}$$

unde antipenultima egalitate se verifică uşor,[1037] **iar penultima egalitate rezultă din independenţa variabilelor y_{pr}, z_{pr} şi ε_{pr}. În mod similar obţinem**

[1033] Vedeţi problema 34 de la capitolul de *Fundamente*, coroborată cu explicaţia de la nota de subsol 1034.

Se poate demonstra că în general (deci nu doar pentru distribuţii gaussiene independente), produsul a două distribuţii gaussiene multidimensionale este tot o distribuţie gaussiană multidimensională. Vedeţi Appendix A.2 în cartea *Gaussian processes in machine learning*, Karl Rasmussen, Christopher Williams, MIT Press, 2006.

[1034] Pentru a vedea de ce aceasta rezultă din definiţia dată în enunţul problemei, observaţi că probabilitatea ca $\varepsilon_{pr} = x_{pr} - y_{pr} - z_{pr}$ să ia o anumită valoare ε este

$$\begin{aligned} p(\varepsilon_{pr} = \varepsilon | y_{pr}, z_{pr}) &= p(x_{pr} - y_{pr} - z_{pr} = \varepsilon | y_{pr}, z_{pr}) \\ &= p(x_{pr} = \varepsilon + y_{pr} + z_{pr} | y_{pr}, z_{pr}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}\varepsilon^2\right), \end{aligned}$$

care nu depinde nici de y_{pr} şi nici de z_{pr}. Astfel, ε_{pr} poate fi privit ca un „zgomot“ independent, gaussian, cu media 0 şi varianţa σ^2.

[1035] Vedeţi problema 10 de la capitolul de *Fundamente*.

[1036] Vedeţi problema 23 de la capitolul de *Fundamente*, corelată cu problema 10 de la acelaşi capitol.

[1037] Se poate demonstra proprietatea $Cov(X, Y + Z) = Cov(Y + Z, X) = Cov(Y, X) + Cov(Z, X)$ pentru orice variabile aleatoare X, Y şi Z. În locul lui $Y + Z$ se poate apoi considera orice sumă finită de variabile aleatoare.

$Cov(z_{pr}, x_{pr}) = Cov(x_{pr}, x_{pr}) = \tau_r^2$. **Aşadar, putem scrie:**

$$y_{pr}, z_{pr}, x_{pr} \sim \mathcal{N}\left(\left[\begin{array}{c} \mu_p \\ \nu_r \\ \mu_p + \nu_r \end{array} \right], \left[\begin{array}{ccc} \sigma_p^2 & 0 & \sigma_p^2 \\ 0 & \tau_r^2 & \tau_r^2 \\ \sigma_p^2 & \tau_r^2 & \sigma_p^2 + \tau_r^2 + \sigma^2 \end{array} \right] \right)$$

Acum vom folosi un rezultat clasic de la *condiţionarea subseturilor de variabile* din distribuţiile gaussiene multidimensionale — şi anume, relaţia (39) de la problema 39 de la capitolul de *Foundamente* — pentru a obţine expresia căutată pentru $p(y_{pr}, z_{pr}|x_{pr}; \theta^{(t)})$:

$$q_{pr}^{(t)}(y_{pr}, z_{pr}) \overset{not.}{=} p(y_{pr}, z_{pr}|x_{pr}; \theta) = \mathcal{N}\left(\left[\begin{array}{c} \mu_{pr,Y} \\ \mu_{pr,Z} \end{array} \right], \left[\begin{array}{cc} \Sigma_{pr,YY} & \Sigma_{pr,ZY} \\ \Sigma_{pr,YZ} & \Sigma_{pr,ZZ} \end{array} \right] \right), \tag{438}$$

unde[1038]

$$\mu_{pr} \overset{not.}{=} \left[\begin{array}{c} \mu_{pr,Y} \\ \mu_{pr,Z} \end{array} \right] = \left[\begin{array}{c} \mu_p + \dfrac{\sigma_p^2}{\sigma^2 + \sigma_p^2 + \tau_r^2}(x_{pr} - \mu_p - \nu_r) \\ \\ \nu_r + \dfrac{\tau_p^2}{\sigma^2 + \sigma_p^2 + \tau_r^2}(x_{pr} - \mu_p - \nu_r) \end{array} \right] \tag{439}$$

$$\Sigma_{pr} \overset{not.}{=} \left[\begin{array}{cc} \Sigma_{pr,YY} & \Sigma_{pr,ZY} \\ \Sigma_{pr,YZ} & \Sigma_{pr,ZZ} \end{array} \right] = \frac{1}{\sigma^2 + \sigma_p^2 + \tau_r^2} \left[\begin{array}{cc} \sigma_p^2(\tau_r^2 + \sigma^2) & -\sigma_p^2\tau_r^2 \\ -\sigma_p^2\tau_r^2 & \tau_r^2(\sigma_p^2 + \sigma^2) \end{array} \right]. \tag{440}$$

Am omis detaliile de calcul, dar ele nu sunt dificil de elaborat.

b. Pentru pasul M, este bine să ne readucem aminte că distribuţia a posteriori $q^{(t)}$ este calculată în funcţie de $\theta^{(t)}$, în vreme ce parametrii pe care vrem să-i calculăm pentru pasul următor sunt $\theta^{(t+1)}$. Aceasta înseamnă că parametrii din distribuţia $q^{(t)}$ sunt constanţi în raport cu acei parametri în funcţie de care se face maximizarea de la acest pas. Desemnând — într-o formă uşor simplificată — cu $E_q[\cdot]$ media calculată în raport cu distribuţia de probabilitate $q_{pr}^{(t)}(y_{pr}, z_{pr})$ pentru fiecare p şi r, vom face maximizarea media log-verosimilităţii datelor observabile:

$$\theta^{(t+1)} = \underset{\theta}{\operatorname{argmax}}\, Q(\theta|\theta^{(t)}),$$

unde

$$\begin{aligned} Q(\theta|\theta^{(t)}) &= \sum_{p=1}^{P}\sum_{r=1}^{R} E_q \ln p(x_{pr}, y_{pr}, z_{pr}|\theta) \\ &\overset{i.i.d.}{=} \textstyle\sum_{p=1}^{P}\sum_{r=1}^{R} E_q \ln \left[\dfrac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x_{pr} - y_{pr} - z_{pr})^2} \cdot \dfrac{1}{\sqrt{2\pi}\sigma_p} e^{-\frac{1}{2\sigma_p^2}(y_{pr} - \mu_p)^2} \cdot \right. \\ &\qquad \left. \dfrac{1}{\sqrt{2\pi}\tau_r} e^{-\frac{1}{2\tau_r^2}(z_{pr} - \nu_r)^2} \right] \\ &= \textstyle\sum_{p=1}^{P}\sum_{r=1}^{R} E_q \Big[\quad \ln \dfrac{1}{(\sqrt{2\pi})^{3/2}\sigma\sigma_p\tau_r} - \dfrac{1}{2\sigma^2}(x_{pr} - y_{pr} - z_{pr})^2 \\ &\qquad -\dfrac{1}{2\sigma_p^2}(y_{pr} - \mu_p)^2 - \dfrac{1}{2\tau_r^2}(z_{pr} - \nu_r)^2 \Big] \end{aligned}$$

[1038] Pentru a creşte lizibilitatea, în expresiile următoare am renunţat la a mai scrie la exponent numărul iteraţiei (t) pentru parametrii μ, σ^2, τ^2 şi Σ.

$$\stackrel{\theta=(\sigma_p,\tau_r)}{=} \sum_{p=1}^{P}\sum_{r=1}^{R} E_q\left[\ln\frac{1}{\sigma_p\tau_r} - \frac{1}{2\sigma_p^2}(y_{pr}-\mu_p)^2 - \frac{1}{2\tau_r^2}(z_{pr}-\nu_r)^2\right]$$

$$= \sum_{p=1}^{P}\sum_{r=1}^{R} E_q\Bigg[\ln\frac{1}{\sigma_p\tau_r} - \frac{1}{2\sigma_p^2}((y_{pr})^2 - 2y_{pr}\mu_p + \mu_p^2) - \frac{1}{2\tau_r^2}((z_{pr})^2 - 2z_{pr}\nu_r + \nu_r^2)\Bigg]$$

$$\stackrel{lin.\ med.}{=} \sum_{p=1}^{P}\sum_{r=1}^{R}\Bigg[\ln\frac{1}{\sigma_p\tau_r} - \frac{1}{2\sigma_p^2}(E_q[(y_{pr})^2] - 2E_q[y_{pr}]\mu_p + \mu_p^2) - \frac{1}{2\tau_r^2}(E_q[(z_{pr})^2] - 2E_q[z_{pr}]\nu_r + \nu_r^2)\Bigg]$$

$$\stackrel{lin.\ med.}{=} \sum_{p=1}^{P}\sum_{r=1}^{R}\Bigg[\ln\frac{1}{\sigma_p\tau_r} - \frac{1}{2\sigma_p^2}(\Sigma_{pr,YY} + \mu_{pr,Y}^2 - 2\mu_{pr,Y}\cdot\mu_p + \mu_p^2) - \frac{1}{2\tau_r^2}(\Sigma_{pr,ZZ}) + \mu_{pr,Z}^2 - 2\mu_{pr,Z}\cdot\nu_r + \nu_r^2)\Bigg].$$

Ultima egalitate este o consecinţă a faptului că $E_q[y_{pr}] = \mu_{pr,Y}$ şi $E_q[(y_{pr})^2] = (E_q[(y_{pr})^2] - E_q[y_{pr}]^2) + E_q[y_{pr}]^2 = \Sigma_{pr,YY} + \mu_{pr,Y}^2$. Relaţii similare se scriu şi pentru $E_q[z_{pr}]$ şi $E_q[(z_{pr})^2]$.

Calculând derivatele parţiale ale funcţiei „auxiliare" $Q(\theta|\theta^{(t)})$ în raport cu parametrii μ_p, ν_r, σ_p şi respectiv τ_r şi egalându-le apoi cu 0, vom obţine:

$$-\frac{1}{2\sigma_p^2}\sum_{r=1}^{R}(2\mu_p - 2\mu_{pr,Y}) = 0 \Rightarrow \mu_p^{(t+1)} = \frac{1}{R}\sum_{r=1}^{R}\mu_{pr,Y} \tag{441}$$

$$-\frac{1}{2\tau_r^2}\sum_{p=1}^{P}(2\nu_r - 2\mu_{pr,Z}) = 0 \Rightarrow \nu_p^{(t+1)} = \frac{1}{P}\sum_{p=1}^{P}\mu_{pr,Z} \tag{442}$$

$$\sum_{r=1}^{R}\left[-\frac{1}{\sigma_p} + \frac{1}{\sigma_p^3}(\Sigma_{pr,YY} + \mu_{pr,Y}^2 - 2\mu_{pr,Y}\,\mu_p + \mu_p^2)\right] = 0 \Rightarrow$$

$$(\sigma_p^2)^{(t+1)} = \frac{1}{R}\sum_{r=1}^{R}(\Sigma_{pr,YY} + \mu_{pr,Y}^2 - 2\mu_{pr,Y}\,\mu_p + \mu_p^2) \tag{443}$$

$$\sum_{p=1}^{P}\left[-\frac{1}{\tau_r} + \frac{1}{\tau_r^3}(\Sigma_{pr,ZZ} + \mu_{pr,Z}^2 - 2\mu_{pr,Z}\,\nu_r + \nu_r^2)\right] = 0 \Rightarrow$$

$$(\tau_r^2)^{(t+1)} = \frac{1}{P}\sum_{p=1}^{P}(\Sigma_{pr,ZZ} + \mu_{pr,Z}^2 - 2\mu_{pr,Z}\,\nu_r + \nu_r^2) \tag{444}$$

Folosind acum rezultatele pe care le-am obţinut, putem reformula paşii E şi M ai algoritmului EM astfel:

Pasul E: Pentru fiecare p şi r, calculează $q_{pr}^{(t)}(y_{pr}, z_{pr})$, conform relaţiei (438), folosind μ_{pr} şi Σ_{pr} din relaţiile (439) şi respectiv (440);

Pasul M: Pentru fiecare p şi r, calculează $\mu_p^{(t+1)}$, $\nu_r^{(t+1)}$, $(\sigma_p^2)^{(t+1)}$ şi $(\tau_r^2)^{(t+1)}$ folosind relaţiile (441), (442), (443) şi respectiv (444).

8.1.5 Alte instanţe ale schemei algoritmice EM

17. **(Algoritmul EM pentru învăţarea parametrului distribuţiei Poisson, când se consideră că o parte din date lipsesc)**

formulare de Liviu Ciortuz, după
■ □ ***Ex. 20.8 din "Probability for Statistics and Machine Learning", Anirban DasGupta, Springer, 2011***

Presupunem că vrem să modelăm statistic numărul de accidente uşoare care s-au produs în n locaţii într-un anumit interval de timp, să zicem o săptămână. În acest scop, vom folosi o distribuţie Poisson de parametru λ.[1039] La sfârşitul perioadei de timp respective, ni se transmite de la m din cele n locaţii câte o „înregistrare" (notată cu x_i), reprezentând numărul de accidente uşoare produse în locaţia i.

În mod implicit, ar trebui să considerăm că în cele $n-m$ locaţii de la care n-am primit înregistrări nu s-a produs niciun accident. Însă, în urma „inspectării" datelor suntem determinaţi să luăm în considerare *presupunerea* că în unele din aceste $n-m$ locaţii s-a produs câte un [singur] accident uşor, care a fost „trecut sub tăcere" la raportare.

Aşadar, formalizând, vom considera datele „neobservabile" $z_1 = 0, \ldots, z_{n_0} = 0, z_{n_0+1} = 1, \ldots, z_{n_0+n_1} = 1$, cu $n_0 + n_1 = n - m$, alături de datele observabile $x_1, \ldots, x_m \in \mathbb{N}$. Toate aceste date sunt produse de variabile aleatoare urmând distribuţia *Poisson* de [acelaşi] parametru λ.

Elaboraţi algoritmul EM (în speţă pasul E şi pasul M) pentru estimarea parametrului λ. *Sugestie*: În loc să se lucreze cu z_j, cu $j = 1, \ldots, m$, va fi suficient să consideraţi ca date neobservabile n_0 şi n_1. Mai mult, considerând numerele n şi m cunoscute, va fi suficient să lucraţi doar cu n_1 ca dată neobservabilă (bineînţeles, pe lângă datele observabile x_i).

Răspuns:

Ţinând cont de faptul că datele $z_1, \ldots, z_{n_0}, z_{n_0+1}, \ldots, z_{n_0+n_1}, x_1, \ldots, x_m$ urmează distribuţia Poisson de parametru λ, a cărei funcţie de densitate este $P(x|\lambda) = \frac{1}{e^\lambda} \cdot \frac{\lambda^x}{x!}$, putem scrie expresia care ne dă verosimilitatea datelor „complete":

$$
\begin{aligned}
L(\lambda) &\overset{def.}{=} P(z_1, \ldots, z_{n_0}, z_{n_0+1}, \ldots, z_{n_0+n_1}, x_1, \ldots, x_m|\lambda) \\
&\overset{i.i.d.}{=} \prod_{j=1}^{n_0+n_1} P(z_i|\lambda) \cdot \prod_{i=1}^{m} P(x_i|\lambda) \\
&= \prod_{j=1}^{n_0} \frac{1}{e^\lambda}\frac{\lambda^0}{0!} \cdot \prod_{j=n_0+1}^{n_1} \frac{1}{e^\lambda}\frac{\lambda}{1!} \cdot \prod_{i=1}^{m} \frac{1}{e^\lambda}\frac{\lambda^{x_i}}{x_i!} \\
&= \frac{1}{(e^\lambda)^{n_0+n_1+m}} \cdot \lambda^{n_1} \cdot \prod_{i=1}^{m} \frac{\lambda^{x_i}}{x_i!} = \frac{1}{(e^\lambda)^n} \cdot \lambda^{n_1} \cdot \prod_{i=1}^{m} \frac{\lambda^{x_i}}{x_i!} \\
&= e^{-n\lambda} \cdot \lambda^{n_1} \cdot \lambda^{\sum_{i=1}^m x_i} \cdot \frac{1}{\prod_{i=1}^m x_i!} = e^{-n\lambda} \cdot \lambda^{n_1+\sum_{i=1}^m x_i} \cdot \frac{1}{\prod_{i=1}^m x_i!}
\end{aligned}
$$

[1039] Am precizat deja la pr. 46 de la capitolul de *Fundamente* faptul că distribuţia Poisson este utilă la modelarea fenomenelor rare.

Log-verosimilitatea datelor complete este:

$$\ell(\lambda) \stackrel{def.}{=} \ln L(\lambda) = -n\lambda + \Big(n_1 + \sum_{i=1}^{m} x_i\Big) \ln \lambda - \sum_{i=1}^{m} \ln x_i!$$

Funcţia „auxiliară“ va fi scrisă cu ajutorul distribuţiei a posteriori a datelor „neobservabile“ (n_1) în raport cu datele observabile (x_i, cu $i = 1, \ldots, m$) şi cu cu $\lambda^{(t)}$, care desemnează valoarea parametrului λ la iteraţia t. Aşadar,

$$Q(\lambda|\lambda^{(t)}) \stackrel{def.}{=} E_{n_1|x_i,\lambda^{(t)}}[\ell(\lambda)] = -n\lambda + \Big(E[n_1|x_i, \lambda^{(t)}] + \sum_{i=1}^{m} x_i\Big) \ln \lambda - \sum_{i=1}^{m} \ln x_i!.$$

***Pasul E*:**

$En_1|x_i, \lambda^{(t)}]$ **este numărul „aşteptat“ de instanţe neobservabile z_j care au valoarea 1, din totalul de $n - m$ instanţe neobservabile. În cazul distribuţiei Poisson, definiţia funcţiei de densitate implică $P(x = 1|\lambda) = \dfrac{1}{e^\lambda} \cdot \dfrac{\lambda^1}{1!} = \dfrac{1}{e^\lambda} \cdot \lambda$ şi $P(x = 0|\lambda) = \dfrac{1}{e^\lambda} \cdot \dfrac{\lambda^0}{0!} = \dfrac{1}{e^\lambda}$. Rezultă că $E[n_1|x_i, \lambda^{(t)}]$ este chiar media distribuţiei binomiale $Bin\Big(n - m; \dfrac{\lambda^{(t)}}{1 + \lambda^{(t)}}\Big)$,[1040] deci**

$$E[n_1|x_i, \lambda^{(t)}] = (n - m)\frac{\lambda^{(t)}}{1 + \lambda^{(t)}}.$$

***Pasul M*:**

Ţinând cont de expresia care tocmai a fost obţinută la pasul E, vom rescrie funcţia auxiliară sub forma următoare:

$$Q(\lambda|\lambda^{(t)}) = -n\lambda + \Big[(n - m)\frac{\lambda^{(t)}}{1 + \lambda^{(t)}} + \sum_{i=1}^{m} x_i\Big] \ln \lambda - \sum_{i=1}^{m} \ln x_i!.$$

Derivatele întâi şi a doua ale acestei funcţii în raport cu λ sunt:

$$\begin{aligned} \frac{\partial}{\partial\lambda} Q(\lambda|\lambda^{(t)}) &= -n + \Big[(n - m)\frac{\lambda^{(t)}}{1 + \lambda^{(t)}} + \sum_{i=1}^{m} x_i\Big]\frac{1}{\lambda} \\ \frac{\partial^2}{\partial\lambda^2} Q(\lambda|\lambda^{(t)}) &= -\Big[(n - m)\frac{\lambda^{(t)}}{1 + \lambda^{(t)}} + \sum_{i=1}^{m} x_i\Big]\frac{1}{\lambda^2} \end{aligned}$$

Conform enunţului, $n > m$ şi $\sum_{i=1}^{m} x_i \geq 0$. La pasul de iniţializare al algoritmului EM, parametrului λ i se asignează o valoare ($\lambda^{(0)}$) pozitivă, ceea ce implică (prin inducţie completă) că $\lambda^{(t)} > 0$ la orice iteraţie $t > 0$, întrucât valoarea pentru care se anulează derivata întâi este

$$\lambda^{(t+1)} = \frac{1}{n}\Big[(n - m)\frac{\lambda^{(t)}}{1 + \lambda^{(t)}} + \sum_{i=1}^{m} x_i\Big].$$

Rezultă că derivata a doua a funcţiei auxiliare $Q(\lambda|\lambda^{(t)})$ este totdeauna negativă, ceea ce înseamnă că această funcţie admite un maxim, şi anume exact pentru $\lambda^{(t+1)}$.

[1040] Vedeţi problema 25.b de la capitolul de *Fundamente*.

18. **(Algoritmul EM: estimarea probabilităţii de selecţie a unei componente din cadrul unei mixturi – combinaţie liniară de două distribuţii probabiliste oarecare)**

■ • ***CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, HW2, pr. 2.1***
CMU, 2006 spring, Carlos Guestrin, final exam, pr. 8

Se consideră variabila aleatoare X având funcţia densitate de probabilitate sub forma unei mixturi de două componente:

$$p_\alpha(x) = \alpha \cdot p_1(x) + (1 - \alpha) \cdot p_2(x)$$

Se cunosc componentele $p_1(x)$ şi $p_2(x)$ care reprezintă modele / funcţii de densitate de probabilitate (considerate nespecificate aici), dar nu se cunoaşte valoarea parametrului $\alpha \in [0, 1]$.

Cunoscând exemplele $\{x_1, x_2, \ldots, x_n\}$ care au fost generate în mod independent şi sunt identic distribuite conform distribuţiei variabilei X, formulaţi algoritmul EM pentru estimarea parametrului α. Descrieţi în mod explicit cei doi paşi — pasul E (pentru calculul mediilor) şi pasul M (de maximizare) — ai algoritmului.

***Observaţie*: Se poate face în mod natural extensia la mixturi cu un număr arbitrar (dar fixat) de componente probabiliste.**

Răspuns:

Pentru a aplica algoritmul EM considerăm variabilele ascunse $\{z_1, z_2, \ldots, z_n\}$, cu $z_i \in \{1, 2\}$, pentru $i = \overline{1, n}$. Pentru fiecare valoare (fixată) a lui i, dacă $z_i = 1$, atunci exemplul x_i a fost generat de componenta $p_1(x)$ a mixturii, iar dacă $z_i = 2$, atunci exemplul x_i a fost generat de componenta $p_2(x)$.

Pasul E al algoritmului EM constă în calcularea mediei log-verosimilităţii datelor complete:

$$Q(\alpha \mid \alpha^{(t)}) \overset{def.}{=} E_{P(Z|X,\alpha^{(t)})}[\log P(X, Z \mid \alpha)]$$

Deoarece fiecare variabilă z_i depinde doar de x_i, iar datele x_i sunt independente între ele, putem scrie:

$$\begin{aligned}
Q(\alpha \mid \alpha^{(t)}) &\overset{def., \ i.i.d.}{=} E_{P(Z|X,\alpha^{(t)})}[\sum_{i=1}^{n} \log p(x_i, z_i \mid \alpha)] \\
&\overset{lin. \ med.}{=} \sum_{i=1}^{n} E_{p(z_i|x_i,\alpha^{(t)})}[\log p(x_i, z_i \mid \alpha)] \\
&\overset{def. \ E}{=} \sum_{i=1}^{n} \Big[\sum_{z_i \in \{1,2\}} p(z_i \mid x_i, \alpha^{(t)}) \cdot \log p(x_i, z_i \mid \alpha) \Big] \\
&= \sum_{i=1}^{n} \Big[p(z_i = 1 \mid x_i, \alpha^{(t)}) \cdot \log p(x_i, z_i = 1 \mid \alpha) + \\
&\qquad + p(z_i = 2 \mid x_i, \alpha^{(t)}) \cdot \log p(x_i, z_i = 2 \mid \alpha) \Big] \\
&= \sum_{i=1}^{n} \Big[p(z_i = 1 \mid x_i, \alpha^{(t)}) \cdot \log(\alpha p_1(x_i)) + \\
&\qquad + p(z_i = 2 \mid x_i, \alpha^{(t)}) \cdot \log((1 - \alpha) p_2(x_i)) \Big]
\end{aligned}$$

$$= \sum_{i=1}^{n} p(z_i = 1 \mid x_i, \alpha^{(t)}) \cdot \log(\alpha) + \sum_{i=1}^{n} p(z_i = 1 \mid x_i, \alpha^{(t)}) \cdot \log p_1(x_i) +$$
$$\sum_{i=1}^{n} p(z_i = 2 \mid x_i, \alpha^{(t)}) \cdot \log(1-\alpha) + \sum_{i=1}^{n} p(z_i = 2 \mid x_i, \alpha^{(t)}) \cdot \log p_2(x_i)$$

Probabilităţile condiţionate implicate în această formulă pot fi calculate cu ajutorul teoremei lui Bayes:

$$p(z_i = 1 \mid x_i, \alpha^{(t)}) \overset{T.B.}{=} \frac{p(x_i \mid z_i = 1, \alpha^{(t)}) \cdot p(z_i = 1 \mid \alpha^{(t)})}{p(x_i, \alpha^{(t)})}$$
$$= \frac{p_1(x_i) \cdot \alpha^{(t)}}{p_1(x_i) \cdot \alpha^{(t)} + p_2(x_i) \cdot (1 - \alpha^{(t)})}$$

$$p(z_i = 2 \mid x_i, \alpha^{(t)}) \overset{T.B.}{=} \frac{p(x_i \mid z_i = 2, \alpha^{(t)}) \cdot p(z_i = 2 \mid \alpha^{(t)})}{p(x_i, \alpha^{(t)})}$$
$$= \frac{p_2(x_i) \cdot (1 - \alpha^{(t)})}{p_1(x_i) \cdot \alpha^{(t)} + p_2(x_i) \cdot (1 - \alpha^{(t)})}$$

Pasul M al algoritmului EM constă în determinarea valorii parametrului α pentru care se obţine maximul expresiei $Q(\alpha \mid \alpha^{(t)})$, care a fost deja calculată mai sus. Pentru a determina $\alpha^{(t+1)} \overset{def.}{=} \operatorname{argmax}_\alpha Q(\alpha \mid \alpha^{(t)})$ vom folosi derivata de ordinul întâi în raport cu α:

$$\frac{\partial Q(\alpha \mid \alpha^{(t)})}{\partial \alpha} = 0 \Leftrightarrow \frac{1}{\alpha} \sum_{i=1}^{n} p(z_i = 1 \mid x_i, \alpha^{(t)}) - \frac{1}{1-\alpha} \sum_{i=1}^{n} p(z_i = 2 \mid x_i, \alpha^{(t)}) = 0$$
$$\Leftrightarrow (1-\alpha) \sum_{i=1}^{n} p(z_i = 1 \mid x_i, \alpha^{(t)}) - \alpha \sum_{i=1}^{n} p(z_i = 2 \mid x_i, \alpha^{(t)}) = 0$$
$$\Leftrightarrow \sum_{i=1}^{n} p(z_i = 1 \mid x_i, \alpha^{(t)}) - \alpha \left(\sum_{i=1}^{n} p(z_i = 1 \mid x_i, \alpha^{(t)}) + \sum_{i=1}^{n} p(z_i = 2 \mid x_i, \alpha^{(t)}) \right) = 0$$
$$\Leftrightarrow \sum_{i=1}^{n} p(z_i = 1 \mid x_i, \alpha^{(t)}) - \alpha \sum_{i=1}^{n} \Big(\underbrace{p(z_i = 1 \mid x_i, \alpha^{(t)}) + p(z_i = 2 \mid x_i, \alpha^{(t)})}_{=1} \Big) = 0$$
$$\Leftrightarrow \sum_{i=1}^{n} p(z_i = 1 \mid x_i, \alpha^{(t)}) - n \cdot \alpha = 0 \Rightarrow \alpha = \frac{1}{n} \sum_{i=1}^{n} p(z_i = 1 \mid x_i, \alpha^{(t)}).$$

Se observă uşor că această soluţie corespunde unui punct de maxim (de fapt, unicul punct de maxim) al funcţiei Q. Prin urmare, ţinând cont de expresia probabilităţii $p(z_i = 1 \mid x_i, \alpha^{(t)})$ care a fost calculată mai sus (vedeţi pasul E), formula de actualizare pentru parametrul α este:

$$\alpha^{(t+1)} = \frac{1}{n} \sum_{i=1}^{n} \frac{\alpha^{(t)} \cdot p_1(x_i)}{\alpha^{(t)} \cdot p_1(x_i) + (1 - \alpha^{(t)}) \cdot p_2(x_i)}$$

***Comentariu*: Pentru o *aplicare* relativ simplă a acestui algoritm EM (extins pentru mai multe componente în mixtură) vedeţi paginile 9-11 din documentul *Handout 12*, de Brani Vidakovic, de la Georgia Institute of Technology, *Bayesian Statistics course* (ISyE 8843A), 2004.**

19. **(Adevărat ori Fals?)**

a. ***CMU, 2002 fall, Andrew Moore, final exam, pr. 1.e***

Spre deosebire de metoda gradientului, care se poate bloca în poziţia unui optim local, algoritmul EM (Expectation-Maximization) identifică întotdeauna optimul global.

b. ***CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, midterm, pr. 1.8.b***

Algoritmul EM nu micşorează valoarea funcţiei obiectiv de la o iteraţie la alta.

Răspuns:

a. Fals. Ambele metode se pot bloca într-un optim local.

b. Adevărat. La problema 1 s-a arătat că algoritmul EM maximizează o margine inferioară ($F(q(z), \theta)$) pentru funcţia de log-verosimilitate a datelor observabile ($\log P(x|\theta) = \log \sum_z P(x, z|\theta)$). Metoda folosită pentru maximizare este una iterativă ("coordinate ascent"). La problema 2 s-a demonstrat că valoarea funcţiei de log-verosimilitate a datelor observabile nu se micşorează de la o iteraţie la alta a algoritmului EM.

8.2 Schema algoritmică EM — Probleme propuse

8.2.1 Fundamente teoretice

20. **(ELBO pentru funcţia obiectiv a algoritmului EM)**

□ • ***CMU, 2020 fall, E. Xing, Z. Bar-Joseph, HW4, pr. 4.3***

Considerăm datele observabile $X = \{x_i\}_{i=1}^n$, împreună cu variabilele neobservabile / ascunse $Z = \{z_i\}_{i=1}^n$, unde $z_i \in \{1, \ldots, K\}$. Parametrii modelului [de tip mixtură de distribuţii probabiliste pe care vrem să-l asociem acestor date] sunt desemnaţi prin simbolul θ, iar cu $\theta^{(t)}$ vom nota valoarea estimată pentru aceşti parametri la pasul t al algoritmului EM. Vrem să maximizăm log-verosimilitatea datelor „incomplete“, $\ell(X;\theta) \overset{not.}{=} \ln p(X|\theta)$.

Care dintre următoarele margini inferioare (engl., lower-bounds) pentru log-verosimilitatea datelor „incomplete“ sunt valide, adică $F(q,\theta) \leq \ell(X;\theta)$?

□ $\sum_{i=1}^{n} \ln p(x_i|\theta)$

□ $\sum_{i=1}^{n} \ln \left(\sum_{k=1}^{K} p(x_i, z_i = k|\theta) \right)$

□ $\sum_{i=1}^{n} \sum_{k=1}^{K} p(z_i = k|x_i, \theta^{(t)}) \ln \frac{p(x_i, z_i = k|\theta)}{p(z_i = k|x_i, \theta^{(t)})}$

□ $\frac{1}{K} \sum_{i=1}^{n} \sum_{k=1}^{K} \ln p(x_i, z_i = k|\theta) + n \ln K$

□ **niciuna dintre cele de mai sus.**

21. **(MAP EM: o întrebare)**

□ • ○ ***CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, midterm, pr. 1.j***

Să zicem că vi se dă un algoritm EM care maximizează verosimilitatea [datelor observabile] folosind un model [probabilist] cu variabile latente. Vi se cere să modificaţi algoritmul EM astfel încât să maximize [nu verosimilitatea, ci] probabilitatea a posteriori (MAP) a datelor observabile. Care pas (sau paşi) ai algoritmului EM trebuie modificaţi:

A. pasul E (expectation)
B. pasul M (maximization)
C. ambii paşi
D. nu este necesară nicio modificare.

22. **(Algoritmul EM semi-supervizat)**

□ • ***Stanford, 2020 summer, Andrew Ng, HW3, pr. 4.a***

EM (Expectation Maximization) este un exemplu clasic de algoritm folosit pentru *învăţare nesupervizată* (adică, învăţare în care sunt folosite variabile „ascunse“ sau latente). În această problemă vom explora una dintre modalităţile în care algoritmul EM poate fi adaptat pentru a realiza *învăţare semi-supervizată*, caz în care avem atât exemple de antrenament etichetate cât şi exemple neetichetate.

În cadrul standard al *învăţării nesupervizate*, avem $n \in \mathbb{N}$ exemple neetichetate $\{x^{(1)}, \ldots, x^{(n)}\}$ şi dorim să învăţăm parametrii distribuţiei $p(x, z; \theta)$ pornind de la date, însă variabilele $z^{(i)}$ nu sunt observabile. *Algoritmul EM clasic*, care a fost conceput exact pentru acest scop, maximizează în mod indirect $p(x; \theta)$, verosimilitatea datelor observabile — a realiza această sarcină în mod direct este în general dificil, eventual chiar nerealizabil —, executând în mod iterativ pasul E şi pasul M şi maximizând de fiecare dată o *margine inferioară* (engl., lower bound) pentru $p(x; \theta)$ (ceea ce în general este o sarcină fezabilă). Funcţia obiectiv poate fi scrisă în mod concret astfel:

$$\ell_{\textbf{unsup}}(\theta) = \sum_{i=1}^{n} \ln p(x^{(i)}; \theta) = \sum_{i=1}^{n} \ln \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}; \theta)$$

Aici vom urmări să construim o extensie / variantă a algoritmului EM care să funcţioneze pentru *cadrul semi-supervizat*. Vom presupune că avem $\tilde{n} \in \mathbb{N}$ exemple *adiţionale* etichetate, $\{(\tilde{x}^{(1)}, \tilde{z}^{(1)}), \ldots, (\tilde{x}^{(\tilde{n})}, \tilde{z}^{(\tilde{n})})\}$, unde atât $\tilde{x}^{(i)}$ cât şi $\tilde{z}^{(i)}$ sunt observabile. Dorim să maximizăm simultan *verosimilitatea marginală* (engl., marginal likelihood) a parametrilor în raport cu *exemplele neetichetate*, precum şi *verosimilitatea totală* (engl., full likelihood) a parametrilor în raport cu *exemplele etichetate*, şi anume optimizând suma lor ponderată (cu ajutorul unui anumit hiperparametru α). Mai precis, funcţia obiectiv semi-supervizată $\ell_{\textbf{semi-sup}}(\theta)$ poate fi scrisă astfel:

$$\ell_{\textbf{semi-sup}}(\theta) = \ell_{\textbf{unsup}}(\theta) + \alpha \ell_{\textbf{sup}}(\theta),$$

unde

$$\ell_{\textbf{sup}}(\theta) = \sum_{i=1}^{\tilde{n}} \ln p(\tilde{x}^{(i)}, \tilde{z}^{(i)}; \theta).$$

a. Formulele de actualizare de la paşii E şi M ai algoritmului EM pentru cazul semi-supervizat se pot determina folosind aceeaşi abordare ca şi în cazul nesupervizat. Arătaţi că obţinem:

Pasul E (semi-supervizat): pentru $i \in \{1, \ldots, n\}$, calculează

$$q^{(t)}(z^{(i)}) = p(z^{(i)} | x^{(i)}; \theta^{(t)}).$$

Pasul M (semi-supervizat):

$$\theta^{(t+1)} = \arg\max_{\theta} \left[\sum_{i=1}^{n} \left(\sum_{z^{(i)}} q^{(t)}(z^{(i)}) \ln \frac{p(x^{(i)}, z^{(i)}; \theta)}{q^{(t)}(z^{(i)})} \right) + \alpha \left(\sum_{i=1}^{\tilde{n}} \ln p(\tilde{x}^{(i)}, \tilde{z}^{(i)}; \theta) \right) \right].$$

b. [Convergenţa.]
Demonstraţi că acest algoritm converge. Pentru aceasta, este suficient să arătaţi că $\ell_{\text{semi-sup}}(\theta)$, adică funcţia obiectiv pentru cazul semi-supervizat, creşte monoton la fiecare iteraţie a noului algoritm EM. Mai specific, arătaţi că $\ell_{\text{semi-sup}}(\theta^{(t+1)}) \geq \ell_{\text{semi-sup}}(\theta^{(t)})$, unde cu $\theta^{(t)}$ am notat valorile parametrilor θ obţinute la sfârşitul iteraţiei t a acestui algoritm EM.

***Observaţie*: pentru două „instanţe" ale acestei scheme algoritmice de tip EM semi-supervizat, puteţi vedea problema 71 de la capitolul de *Clusterizare* (EM/GMM semi-supervizat) şi problema 38 de la prezentul capitol (EM/BernoulliMM semi-supervizat).**

23. **(Algoritmul "hard" EM; exemplificare pentru rezolvarea unei mixturi de vectori de distribuţii categoriale, folosind presupoziţia de independenţă condiţională de tip Bayes Naiv, cu asignare "hard" a instanţelor la clustere)**

prelucrare de Liviu Ciortuz, după
□ • ○ ***CMU, 2014 spring, A. Singh, B. Poczos, HW3, pr. 2.2***

Fie un set de instanţe neetichetate $\mathcal{D} = \{x_1, \ldots, x_n\}$. La problema 1.a (fundamentarea teoretică a algoritmului EM), am folosit la scrierea funcţiei de log-verosimilitate ($\log P(\mathcal{D}|\theta) = \sum_i \log P(x_i|\theta)$) faptul că $P(x_i|\theta)$ poate fi rescris / exprimat însumând probabilităţile comune (engl., joint probabilities) corespunzătoare lui x_i (pe de o parte) şi fiecărei asignări posibile a variabilelor latente / neobservabile (pe de altă parte).

În acest exerciţiu ne interesează să vedem ce se întâmplă atunci când în loc să facem sumarea despre care am vorbit mai sus, vom seta variabile neobservabile la valorile lor cele mai probabile în raport cu estimările / valorile actuale ale parametrilor. Această strategie este numită adeseori *algoritmul "hard" EM*. În anumite cazuri, el funcţionează bine.[1041]

a. Arătaţi că şi în situaţia în care în loc să procedăm la sumarea probabilităţilor pentru toate asignările posibile ale variabilelor latente — $P(X_i = x_i|\pi, \beta) = \sum_z P(X_i = x_i, Z = z|\pi, \beta)$ — facem maximizare, are loc o optimizare a unei margini inferioare pentru funcţia de log-verosimilitate a datelor observabile.

b. Elaboraţi pasul E pentru varianta "hard" a algoritmului EM.

c. Elaboraţi pasul M pentru varianta "hard" a algoritmului EM într-un cadru particular, şi anume atunci când *obiectivul* nostru este să rezolvăm problema de *învăţare nesupervizată de tip Bayes Naiv*, însă nu folosind asignare "soft" a instanţelor la clustere cum am procedat la exerciţiul 11, ci folosind asignare "hard".

[1041] Spre exemplu, putem obţine o variantă a algoritmului K-means aplicând algoritmul "hard" EM unei mixturi de distribuţii gaussiene.

24. **(Algoritmul EM generalizat)**

□ • ○ ***Stanford, 2007 fall, Andrew Ng, HW3, pr. 5***

Uneori, atunci când încercăm să rulăm algoritmul EM, se poate să avem dificultăţi în a executa pasul M în manieră exactă. (Vă reamintim că adeseori trebuie să implementăm proceduri numerice de optimizare ca să efectuăm maximizarea, ceea ce poate fi costisitor.) Aşadar, în loc să găsim maximul global pentru marginea inferioară a funcţiei de log-verosimilitate, ar fi suficient să obţinem o anumită creştere pe această margine inferioară (văzută ca funcţie), executând (de exemplu) o iteraţie a algoritmului gradientului ascendent. Această variantă a algoritmului EM este cunoscută îndeobşte sub numele de *algoritmul EM generalizat* (engl., Generalized EM, GEM).

Ca să ne exprimăm într-o manieră ceva mai formală, vă readucem aminte că la pasul M din algoritmul EM standard se execută maximizarea

$$\theta \leftarrow \operatorname*{argmax}_{\theta} \sum_i \sum_{z^{(i)}} q_i(z^{(i)}) \ln \frac{p(x^{(i)}, z^{(i)}; \theta)}{q_i(z^{(i)})}.$$

În schimb, algoritmul GEM foloseşte la pasul M următoarea regulă de actualizare:

$$\theta \leftarrow \theta + \alpha \nabla_\theta \left(\sum_i \sum_{z^{(i)}} q_i(z^{(i)}) \ln \frac{p(x^{(i)}, z^{(i)}; \theta)}{q_i(z^{(i)})} \right),$$

unde α este rata de învăţare, despre care presupunem că este aleasă suficient de mică, în aşa fel încât funcţia obiectiv să nu descrească atunci când se execută acest pas [al gradientului ascendent].

a. Demonstraţi că algoritmul GEM descris mai sus converge. Aceasta revine la a arăta că funcţia de verosimilitate este monoton crescătoare, ca şi în cazul algoritmului EM. Aşadar, demonstraţi că $\ell(\theta^{(t+1)}) \geq \ell(\theta^{(t)})$.

b. Să presupunem că în loc să folosim algoritmul EM am vrea să aplicăm algoritmul gradientului ascendent pentru a maximiza funcţia de log-verosimilitate în mod direct. Cu alte cuvinte, încercăm să maximizăm funcţia (non-convexă)

$$\ell(\theta) = \sum_i \ln \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}; \theta),$$

folosind regula de actualizare

$$\theta \leftarrow \theta + \alpha \nabla_\theta \Big(\sum_i \ln \sum_{z^{(i)}} p(x^{(i)}, z^{(i)}; \theta) \Big).$$

Demonstraţi că această regulă de actualizare este de fapt aceeaşi cu regula de actualizare de la algoritmul GEM descris mai sus.

8.2.2 Mixturi de distribuţii Bernoulli / categoriale

25. **(Două extensii ale algoritmului EM pentru estimarea parametrilor unei mixturi de două distribuţii Bernoulli (cazul general))**

□ · ***Liviu Ciortuz, 2023***

a. Modificaţi algoritmul EM de la problema 6 astfel încât să putem rezolva mixturi de două distribuţii Bernoulli folosind (pentru eficienţă) două variabile aleatoare de tip binomial, $\tilde{z}_1$ şi $\tilde{z}_0$.[1042] Aceste două variabile aleatoare vor avea următoarea semnificaţie: $\tilde{z}_1$ va desemna numărul (necunoscut) de aruncări — din totalul celor n_s aruncări (număr cunoscut) la care s-a obţinut faţa *stemă* — la care s-a folosit prima monedă; similar, $\tilde{z}_0$ va desemna numărul (necunoscut) de aruncări — din totalul celor n_B aruncări (număr cunoscut) la care s-a obţinut faţa *ban* — la care s-a folosit prima monedă.

Veţi folosi de asemenea notaţiile $r_1^{(t)} \stackrel{not.}{=} P(z_i = 1|x_i = 1; \theta^{(t)})$ şi $r_0^{(t)} \stackrel{not.}{=} P(z_i = 1|x_i = 0; \theta^{(t)})$, unde $\theta^{(t)} \stackrel{not.}{=} (p^{(t)}, q^{(t)}, \pi^{(t)})$ ca şi la problema 6.[1043]

Implementaţi algoritmul EM pentru rezolvarea acestui model de mixtură şi apoi rulaţi-l pe inputul $x = \{1, 1, 0, 1, 0, 0, 1, 0, 0, 0, 1, 1\}$, cu valorile iniţiale $1/3$, $2/3$ şi $1/2$ pentru parametrii p, q şi respectiv π. Ce se observă în privinţa convergenţei? Dar dacă se iniţializează parametrii p, q şi π cu alte valori?

b. Modificaţi algoritmul EM de la problema 6 astfel încât să putem rezolva mixturi de K (deci nu doar două) distribuţii Bernoulli. Parametrii acestor distribuţii vor fi notaţi cu $p_1, \ldots, p_K$, iar probabilităţile de selecţie [ale respectivelor distribuţii] cu $\pi_1, \ldots, \pi_K$, unde $p_i \in [0, 1]$ şi $\pi_i \in [0, 1]$ pentru $i = 1, \ldots, K$, iar $\pi_1 + \ldots + \pi_K = 1$.

***Sugestie*: La acest punct ar trebui ca variabilele neobservabile Z_i să ia valori în mulţimea $\{1, \ldots, K\}$. Însă pentru a putea aplica uşor artificiul exponenţierii (engl., the exponentiation trick), este mai convenabil ca în locul fiecărei astfel de variabile Z_i să folosiţi variabilele-indicator $Z_{i1}, \ldots, Z_{iK}$, cu proprietatea $Z_{ij} = 1$ dacă instanţa de antrenment x_i a fost generată de către distribuţia j şi $Z_{ij} = 0$ în cazul contrar. De asemenea, întrucât avem restricţia $\pi_1 + \ldots + \pi_K = 1$, va trebui să folosiţi metoda multiplicatorilor lui Lagrange. Vă puteţi inspira din rezolvarea problemei 9 (sau 28).**

***Observaţie*: Este posibil să combinaţi cele două extensii — care au fost descrise la punctele a şi b — într-una singură, însă este ceva mai dificil.[1044]**

[1042] Pentru o variantă similară de algoritm EM, vedeţi problema 5 şi în special *Observaţia importantă* de la finalul rezolvării acelei probleme.

[1043] Faţa *stemă* este desemnată cu cifra 1, iar faţa *ban* este desemnată cu cifra 0.

[1044] Cu titlu de *Sugestie*, vă sfătuim să folosiţi $2K$ variabile aleatoare de tip binomial, şi anume $\tilde{z}_{1j}$ şi $\tilde{z}_{0j}$, cu $j = 1, \ldots, K$. Variabila $\tilde{z}_{1j}$ va desemna numărul (necunoscut) de aruncări — din totalul celor n_s aruncări la care s-a obţinut faţa *stemă* — la care s-a folosit moneda j; similar, variabila $\tilde{z}_{0j}$ va desemna numărul (necunoscut) de aruncări — din totalul celor n_B aruncări la care s-a obţinut faţa *ban* — la care s-a folosit moneda j.

26. **(Algoritmul EM pentru mixturi de distribuţii Bernoulli folosind două variabile aleatoare de tip binomial: executarea manuală a unei iteraţii)**

prelucrare de Liviu Ciortuz, după
□ • ***CMU, 2020 fall, E. Xing, Z. Bar-Joseph, HW4, pr. 5***

Presupunem că dispunem de două monede, $\mathcal{C}_1$ şi $\mathcal{C}_2$, pentru care probabilităţile [respective] de apariţie a feţei *stemă* sunt necunoscute. Vom nota aceste două probabilităţi cu p şi respectiv q. Moneda $\mathcal{C}_1$ este aleasă cu probabilitatea π, iar moneda $\mathcal{C}_2$ este aleasă cu probabilitatea $1-\pi$. Aruncăm moneda aleasă o dată, iar apoi notăm rezultatul. Acest experiment probabilist este repetat de cinci ori, iar rezultatul obţinut în final este $X = \{H, H, T, H, T\}$.

Obiectivul **acestei probleme este să estimăm parametrii $\theta \overset{not.}{=} (p, q, \pi)$ folosind algoritmul EM. Facem *convenţia* următoare: dacă *variabila neobservabilă* Z_i ia valoarea 1, atunci înseamnă că la aruncarea cu numărul i a fost folosită moneda $\mathcal{C}_1$, iar dacă $Z_i = 0$ atunci a fost folosită moneda $\mathcal{C}_2$. Veţi iniţializa parametrii cu următoarele valori: $p^{(0)} = \frac{1}{4}$, $q^{(0)} = \frac{2}{3}$ şi $\pi^{(0)} = \frac{1}{2}$.**

Observaţie**: La rezolvarea acestei probleme veţi utiliza algoritmul EM/BMM care face obiectul problemei 25.a.**

Notă**: Pentru întrebările $b - h$ de mai jos, vă cerem să scrieţi răspunsul sub forma unei *fracţii ireductibile* de forma $\frac{a}{b}$, unde a şi b sunt numere naturale prime între ele. De *exemplu*, dacă răspunsul este 0.4125, va trebui să-l scrieţi sub forma $\frac{33}{80}$.**

a. Faceţi o *reprezentare grafică* pentru mixtura descrisă în enunţul problemei.
b. Pasul E: Calculaţi $r_1^{(0)} \overset{not.}{=} P(z_i = 1 | x_i = H; \theta^{(0)})$, unde $\theta^{(0)} \overset{not.}{=} (p^{(0)}, q^{(0)}, \pi^{(0)})$.

c. Notăm cu $\tilde{z}_1$ numărul de feţe *stemă* (engl., head, H) produse de moneda $\mathcal{C}_1$. Observaţi faptul că $\tilde{z}_1$ este o variabilă aleatoare (latentă / ascunsă / neobservabilă). Calculaţi media (sau, valoarea „aşteptată"; engl., expected value) pentru variabila $\tilde{z}_1$ la prima iteraţie a algoritmului EM. Justificaţi răspunsul în mod detaliat.

d. Pasul E: Calculaţi $r_0^{(0)} \overset{not.}{=} P(z_i = 1 | x_i = T, \theta^{(0)})$.

e. Notăm cu $\tilde{z}_0$ numărul de feţe *ban* **(engl., tail, T) produse de moneda $\mathcal{C}_1$. Observaţi faptul că $\tilde{z}_0$ este o variabilă aleatoare (de asemenea latentă). Calculaţi media (sau, valoarea „aşteptată"; engl., expected value) a lui $\tilde{z}_0$ la prima iteraţie a algoritmului EM. Justificaţi răspunsul în mod detaliat.**

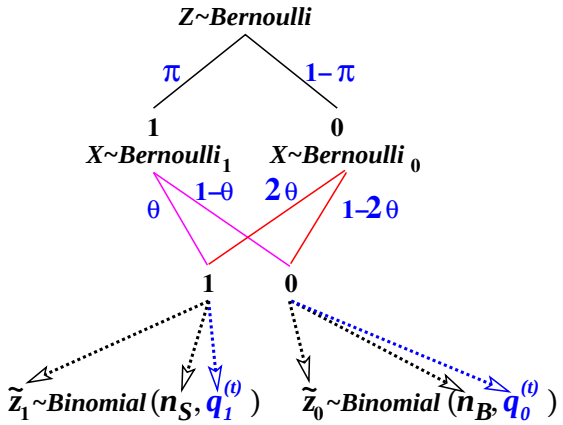
f. Pasul M: Calculaţi $\pi^{(1)}$, $p^{(1)}$, $q^{(1)}$.

g. Scrieţi expresia funcţiei $L(\theta^{(t)}) \overset{not.}{=} P(X | \theta^{(t)})$, verosimilitatea datelor observabile la iteraţia t. Apoi calculaţi $L(\theta^{(1)})$ şi $L(\theta^{(0)})$. La final, verificaţi că $L(\theta^{(1)}) \geq L(\theta^{(0)})$.

27. **(Estimarea parametrului de selecţie dintr-o mixtură de distribuţii Bernoulli: exemplificare folosind un set simplu de date; deducerea regulilor de actualizare pentru varianta algoritmului EM care foloseşte două distribuţii binomiale)**

□ ∗ ***L. Ciortuz, 2025, folosind datele de la CMU, 2005 fall, T. Mitchell, A. Moore, midterm, pr. 1.3***

În acest exerciţiu vom arăta că putem să reformulăm algoritmul EM pentru estimarea parametrilor unei mixturi de distribuţii Bernoulli care a fost prezentat la problema 7.C.a sub o *altă formă* (care este mult mai convenabilă din punctul de vedere al eficienţei), folosind două distribuţii binomiale, similare cu cele de la problema 5. (Vedeţi *Observaţia importantă* de la finalul rezolvării problemei 5.)



a. La rezolvarea dată la punctul a de la problema 7.C, pentru *pasul* E (aşadar, când probabilitatea de selecţie, π, era considerată fixată), am obţinut

$$\mu_i^{(t)} = P(z_i = 1|x_i, \theta^{(t)}) \overset{F.\ Bayes}{=} \frac{\pi\,(\theta^{(t)})^{x_i}(1-\theta^{(t)})^{1-x_i}}{\pi\,(\theta^{(t)})^{x_i}(1-\theta^{(t)})^{1-x_i} + (1-\pi)(2\theta^{(t)})^{x_i}(1-2\theta^{(t)})^{1-x_i}}.$$

Particularizaţi această formulă pentru *cazul* $x_i = 1$ (adică, pentru $x_i \equiv$ *stemă*)

$$q_1^{(t)} \overset{not.}{=} P(z_i = 1|x_i = 1, \theta^{(t)}) = \ldots$$

şi respectiv pentru *cazul* $x_i = 0$ (adică, pentru $x_i \equiv$ *ban*)

$$q_0^{(t)} \overset{not.}{=} P(z_i = 1|x_i = 0, \theta^{(t)}) = \ldots$$

Exprimaţi în cuvinte proprii care sunt semnificaţiile acestor două probabilităţi.

b. Fie $\tilde{z}_1$ numărul (neobservabil) de apariţii ale feţei *stemă* pentru moneda 1, din totalul de n_S apariţii ale stemei în datele noastre.[1045] Similar, fie $\tilde{z}_0$ numărul (neobservabil) de apariţii ale feţei *ban* pentru moneda 1, din totalul de $n_B 3$ apariţii ale banului în datele noastre. Vom considera datele (din tabelul prezentat la secţiunea B de la problema 7) „sintetizate“ sub forma următoare: două apariţii ale stemei ($n_S \overset{not.}{=} 2$) şi trei apariţii ale banului ($n_B \overset{not.}{=} 3$).

Scrieţi expresiile funcţiilor de verosimilitate şi log-verosimilitate a datelor complete.

Deduceţi apoi expresia funcţiei auxiliare de la iteraţia t a algoritmului EM.

c. Pentru *Pasul E* al algoritmului EM, calculaţi mediile

[1045] Vă rugăm să *remarcaţi* faptul că variabila $\tilde{z}_1$ este diferită de variabila z_1 care a fost folosită în rezolvările de la punctele a şi b de la problema 7. Sperăm ca această „suprascriere“ (parţială) la nivel de notaţie să nu incomodeze cititorul. Am preferat / optat să păstrăm astfel convenţia din problema 1, şi anume: cu x se notează variabilele observabile şi cu z variabilele neobservabile.

• $E[\tilde{z}_1]$ — **numărul aşteptat (engl., expected number) de apariţii ale feţei** *stemă* **pentru moneda 1 din totalul de** $n_S \overset{not.}{=} 2$ **apariţii ale stemei în datele noastre, şi**
• $E[\tilde{z}_0]$ — **numărul aşteptat (engl., expected number) de apariţii ale feţei** *ban* **pentru moneda 1 din totalul de** $n_B \overset{not.}{=} 3$ **apariţii ale banului în datele noastre.**

d. Pentru *Pasul M* al algoritmului EM, deduceţi regula de actualizare a parametrului θ **la iteraţia** t**.**

e. În final, scrieţi pseudo-codul algoritmului EM pentru rezolvarea acestei mixturi de distribuţii Bernoulli.

***Observaţie*: Similar putem transforma varianta de algoritm EM care a fost prezentată la problema 7.C.b, unde parametrii liberi sunt** π **şi** θ**.**

28. **(Algoritmul EM: estimarea parametrilor unei mixturi de distribuţii categoriale: cazul general[1046])**

prelucrare de Liviu Ciortuz, după
■ □ • ○ ***CMU, 2015 spring, T. Mitchell, N. Balcan, HW6, pr. 1***

La acest exerciţiu vom lucra cu modele de mixturi de distribuţii *categoriale*.[1047]

Fie un set de date $x = \{x_1, \ldots, x_n\}$**, unde fiecare** $x_i \in \{v_1, \ldots, v_M\}$ **este generat [în mod independent de ceilalţi** x_j**, cu** $j \neq i$**] de către una din** K **distribuţii categoriale posibile, notată cu** X_i**. Vom considera că distribuţia care l-a generat pe** x_i **a fost desemnată de către o altă variabilă aleatoare categorială, luând valori în mulţimea** $\{1, \ldots, K\}$ **şi având vectorul de parametri** $\pi \in [0,1]^K$**, cu** $\sum_k \pi_k = 1$**. Vom nota cu** $\theta_k \in \mathbb{R}^M$ **parametrul distribuţiei categoriale asociate cu componenta** k **din mixtură, deci** $\theta_k \in (0,1)^M$ **şi** $\sum_{j=1}^M \theta_{kj} = 1$ **pentru** $k = 1, \ldots, K$**.**

***Procesul generativ* pentru un *model de mixtură categorială* poate fi sumarizat astfel:** $Z_i \sim \mathit{Categorial}(\pi), X_i \sim \mathit{Categorial}(\theta_{Z_i})$**. Pentru acest model, în care observăm variabilele** X **dar nu şi variabilele** Z**, obiectivul este să învăţăm parametrii** $\Theta = \{\pi, \theta_1, \ldots, \theta_K\}$**. Veţi folosi algoritmul EM pentru a realiza acest obiectiv.**

***Observaţie*: Atunci când lucrăm cu distribuţii categoriale, este utilă folosirea funcţiilor indicator [în locul variabilelor-indicator]. Prin definiţie, funcţia-indicator** $1_{\{x=j\}}$ **are valoarea** 1 **dacă** $x = j$ **şi** 0 **în caz contrar.**[1048]

[1046] Problema aceasta reprezintă o generalizare în raport cu problema 6, unde am prezentat algoritmul EM pentru o mixtură de două distribuţii Bernoulli. Generalizarea se referă la faptul că în locul distribuţiilor Bernoulli aici se consideră distribuţii categoriale, iar pe de altă parte, numărul de distribuţii cu care se lucrează poate fi oarecare ($K \geq 1$).

Problema 10 este un caz particular (sau, o ilustrare) pentru problema de faţă. Pe de altă parte, atunci când mixtura are $K = 2$ componente, problema 11 reprezintă o generalizare a problemei de faţă.

[1047] La problema 44 de la capitolul de *Fundamente* am arătat cum se face estimarea în sens MLE pentru parametrii unei distribuţii categoriale. Aici vom estima parametrii unei mixturi de K distribuţii categoriale.

[1048] De exemplu, dacă vom considera o variabilă aleatoare Y care ia valori în mulţimea $\{1, \ldots, N\}$ şi urmează o distribuţie categorială (notaţie: $Y \sim \mathit{Categorial}(\phi)$, unde $\phi \in (0,1)^N$), cu $\phi \overset{not.}{=} (\phi_1, \ldots, \phi_N)$, vom putea să exprimăm probabilitatea ca Y să ia o anunită valoare în felul următor:

$$P(Y) = \prod_{i=1}^{N} \phi_i^{1_{\{Y=i\}}}.$$

a. Calculaţi *distribuţia comună* a datelor observabile (X) şi neobservabile (Z): $P(X, Z; \Theta)$.

b. Calculaţi *probabilităţile a posteriori* corespunzătoare *variabilelor latente*, $P(Z_i = k|X_i; \Theta)$.

c. Calculaţi *media log-verosimilităţi datelor complete*,

$$Q(\Theta|\Theta') \stackrel{def.}{=} E_{Z|X;\Theta'}[\log P(X, Z; \Theta)].$$

d. Deduceţi regulile de actualizare pentru parametrii Θ. Altfel spus, care este valoarea lui Θ care maximizează funcţia $Q(\Theta|\Theta')$ calculată la punctul c?

***Indicaţie*: Aveţi grijă că soluţia pe care o veţi obţine trebuie să satisfacă restricţia ca parametrii distribuţiilor categoriale trebuie să se sumeze la valoarea 1. Metoda multiplicatorilor Lagrange este o cale foarte convenabilă pentru rezolvarea unor astfel de probleme de optimizare cu restricţii.**

29. (EM pentru combinarea unei clasificări de tip Bayes Naiv cu o mixtură de vectori de distribuţii Bernoulli, folosind şi un termen de penalizare / regularizare. Aplicaţie: recunoaşterea cifrelor scrise de mână)

● ○ ***U. Toronto, Radford Neal, "Statistical Methods for Machine Learning and Data Mining" course, 2014 spring, HW2***

***Obiectivul* dumneavoastră în această problemă va fi să clasificaţi cifre scrise de mână, cu ajutorul unor modele de mixturi [de distribuţii probabiliste] care vor fi obţinute prin maximimzarea verosimilităţii datelor însoţită de un termen de penalizare, folosind algoritmul EM.**

Datele pe care le veţi folosi constau din 800 de imagini de antrenament şi 1000 de imagini de test pentru cifre scrise de mână (din codurile poştale de pe plicuri din SUA). Aceste imagini au fost extrase din bine-cunoscutul set de date MNIST, prin selectare aleatorie dintr-un total de 60000 de imagini de antrenament furnizate, reducând rezoluţia acestor imagini de la 28×28 de pixeli la 14×14 de pixeli, făcând media valorilor pixelilor pentru fiecare bloc de pixeli de dimensiune 2×2, iar apoi folosind nişte praguri pentru aceste medii în aşa fel încât să obţinem valori binare. Pe pagina web a acestei culegeri[1049] este furnizat un fişier de date de antrenament având 800 de linii, fiecare linie conţinând 196 de valori pentru pixeli şi anume, fie 0, fie 1. Tot acolo este furnizat şi un alt fişier care conţine etichetele corespunzătoare acestor 800 de cifre (de la 0 to 9). În mod similar, vă punem la dispoıţie şi un fişier cu 1000 de imagini de test, precum şi un fişier conţinând etichetele acestor 1000 de imagini de test. (Nu veţi avea voie să vă uitaţi la etichetele acestor imagini de test decât la sfârşit de tot, şi anume atunci când va trebui să analizaţi cât de bine se comportă algoritmii de învăţare automată.)

Veţi clasifica aceste imagini pentru cifre scrise de mână folosind un *model generativ*, din care, dată fiind imaginea unei astfel de cifre, veţi deriva probabilităţile pentru cele 10 clase posibile. *Clasa asociată unei cifre de test* este cea care are probabilitatea maximă.

[1049] Vedeţi https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/UToronto.2014s.RNeal.HW2.EM-for-BernoulliMM-using-the-NBayes-assumption.handwritten-char-reco.data+R-code+sol/

A. Modelul generativ pe care îl vom folosi estimează *probabilităţile claselor* ca [fiind] *frecvenţele* lor din setul de date de antrenament (ceea ce va determina o distribuţie aproximiativ uniformă — însă nu exact uniformă — peste cele 10 cifre). De asemenea, acest model estimează *distribuţiile de probabilitate ale imaginilor din cadrul fiecărei clase* cu ajutorul unui *model de mixtură cu* K *componente*, fiecare componentă modelând valorile celor 196 de pixeli ca [fiind] variabile aleatoare *independente*. Va fi convenabil să combinăm toate aceste 10 modele de mixturi [de distribuţii probabiliste] într-un singur *model de mixtură cu* $10K$ *componente*, care modelează atât valorile pixelilor cât şi etichetele claselor. Totuşi, *probabilităţile* [*condiţionate ale*] *componentelor* din cadrul modelului fiecărei clase vor fi *fixate*, astfel încât K componente vor atribui probabilitatea condiţionată 1 cifrei 0, alte K componente vor atribui probabilitatea condiţionată 1 cifrei 1, alte K vor atribui probabilitatea condiţionată 1 cifrei 2 ş.a.m.d.

Aşadar, modelul probabilist pentru *distribuţia comună* asupra etichetelor y_i, precum şi asupra valorilor pixelilor $x_{i,1}, \ldots, x_{i,196}$ pentru $i = 0, \ldots, 9$ poate fi exprimat sub forma următoare:

$$P(y_i, x_i|\pi, \theta) = \sum_{k=1}^{10K} \left(\pi_k q_{k,y_i} \prod_{j=1}^{196} \theta_{k,j}^{x_{i,j}} (1 - \theta_{k,j})^{1-x_{i,j}} \right).$$

Instanţele etichetate (x_i, y_i) se presupun a fi independente.

***Parametrii* acestui model sunt *probabilităţile de mixare / selecţie*, $\pi_1, \ldots, \pi_{10K}$, precum şi probabilităţile $\theta_{k,j}$ asociate pixelilor (mai precis, pentru ca aceştia să fie setaţi la valoarea 1), pentru fiecare componentă în parte, adică pentru $k = 1, \ldots, 10K$ şi $j = 1, \ldots, 196$. Probabilităţile condiţionate $q_{k,y}$ ale componentelor fiecărei clase sunt fixate astfel:**

$$q_{k,y} = \begin{cases} 1 & \text{atunci când } k \in \{Ky + 1, \ldots, Ky + K\} \\ 0 & \text{în caz contrar,} \end{cases}$$

pentru $k = 1, \ldots, 10K$ şi $y = 0, \ldots, 9$.

La sfârşitul acestui exerciţiu veţi implementa în Python / R / Matlab o funcţie care identifică valorile parametrilor care maximizează log-verosimilitatea datelor de antrenament plus un *termen de penalizare* (engl., penalty term).[1050]

Algoritmul EM poate fi uşor adaptat pentru a găsi [în locul estimării verosimilităţii maxime] estimarea verosimilitatăţii maxime penalizate. În raport cu versiunea generală a algoritmului EM [care a fost prezentată la curs], pasul E se păstrează neschimbat, însă pasul M va maximiza acum o funcţie care reprezintă media log-verosimilităţii penalizate a datelor complete, $E_Q[\ln P(x, z|\theta) + G(\theta)]$, unde $G(\theta)$ este termenul de penalizare.

***Scopul penalizării* este acela de a evita ca estimările probabilităţilor asociate pixelilor să fie 0 sau aproape de 0, situaţii care pot cauza probleme la clasificarea instanţelor de test. (Este imediat că dacă o astfel de probabilitate este 0, rezultă că probabilitatea oricărei instanţe de test va fi 0 pentru clasa respectivă.) Termenul de penalizare care se sumează la log-verosimilitate trebuie să fie**

$$G(\theta) = \alpha \sum_{k=1}^{10K} \sum_{j=1}^{196} [\log(\theta_{k,j}) + \log(1 - \theta_{k,j})],$$

[1050] Cu cât valorile acestui termen de penalizare vor fi mai mari, cu atât va fi mai bine.

unde constanta α controlează „magnitudinea“ penalizării.[1051]

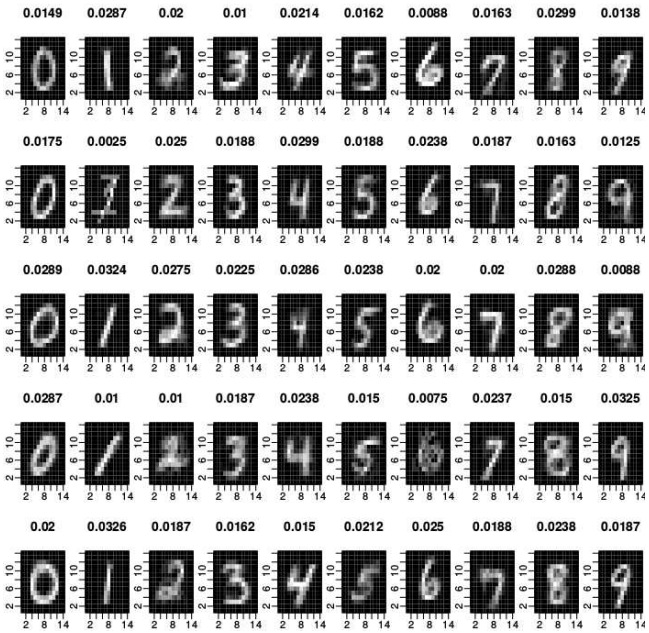
a. Demonstraţi că într-adevăr [regula de actualizare de la] pasul E rămâne la fel ca în cazul versiunii generale a algoritmului EM.

b. La pasul M, reestimarea probabilităţilor de selecţie π va rămâne de asemenea neschimbată [în raport cu cazul general],[1052] **însă pentru reestimarea parametrilor θ va trebui să luaţi în considerare şi termenul de penalizare. Demonstraţi că formula pentru actualizarea parametrilor $\theta_{k,j}$ la pasul M este**

$$\hat{\theta}_{k,j} = \frac{\alpha + \sum_{i=1}^{n} r_{i,k} x_{i,j}}{2\alpha + \sum_{i=1}^{n} r_{i,k}},$$

unde $r_{i,k}$ este probabilitatea ca instanţa i să fi fost generată de către componenta k a mixturii, probabilitate care a fost estimată la pasul E.[1053]

***Comentariu*: Pentru $K = 5$, putem ilustra sub forma unor imagini valorile parametrilor θ care au fost învăţate pentru cele 50 de componente ale mixturii:**



Este clar că în general cele cinci componente corespunzătoare fiecărei cifre au identificat diferite variante rezonabile de caligrafiere a cifrelor, exceptând probabil câteva cazuri, cărora le corespund probabilităţi de selecţie destul de mici (indicate sub formă numerică, deasupra imaginilor), aşa cum este caracterul “1” de pe cel de-al doilea rând de imagini.

B. Funcţia / procedura în cadrul căreia veţi implementa algoritmul EM va trebui să ia ca argumente [numele fişierelor care conţin] imaginile din setul de

[1051] Pentru acest exerciţiu, α va putea fi fixat la valoarea 0.05, deşi în aplicaţii reale el va trebui probabil să fie determinat folosind, de exemplu, metoda cross-validării.

[1052] Va trebui să vă convingeţi singuri de acest fapt.

[1053] Ca să obţineţi această formulă, aţi putea să porniţi de la demonstraţia care a fost prezentată la problema 9, modificând-o în aşa fel încât să includă şi termenul de penalizare.

date de antrenament şi etichetele asociate acestor imagini, numărul de componente ale mixturii (acest număr, K, este acelaşi pentru toate clasele / cifrele), „magnitudinea“ penalizării (α), precum şi numărul de iteraţii pe care trebuie să le execute algoritmul EM. Procedura aceasta va trebui să returneze o listă formată din estimările parametrilor (π şi θ), precum şi „responsabilităţile“ (r). Va trebui să startaţi algoritmul cu anumite valori iniţiale pentru „responsabilităţi“, iar apoi să intraţi în pasul M. „Responsabilitatea“ pentru componenta k a instanţei / imaginii i trebuie să fie 0 în cazul în care pentru componenta k avem $q_{k,y_i} = 0$. Petru cazul $q_{k,y_i} = 1$, va trebui să setaţi în mod aleatoriu $r_{i,k}$ conform distribuţiei continue uniforme definite pe intervalul $[1, 2]$, iar apoi să rescalaţi aceste valori astfel încât pentru fiecare valoare $i \in \{1, \ldots, n\}$ să avem $\sum_{k=1}^{K} r_{i,k} = 1$.

c. După fiecare iteraţie, procedura dumneavoastră [care implementează algoritmul] EM va trebui să imprime valoarea log-verosimilităţii, precum şi valoarea sumei dintre log-verosimilitate şi termenul de penalizare. Suma aceasta nu trebuie să descrească de la o iteraţie la alta — în cazul în care ea descreşte, înseamnă că aveţi o eroare în procedura EM. Va trebui să executaţi suficiente iteraţii pentru ca aceste valori să se devină [aproape] stabilizate atunci când se ajunge la ultima iteraţie.

d. Veţi implementa de asemenea o procedură care să ia ca input valorile parametrilor care au fost returnate de către procedura de antrenare EM şi le foloseşte pentru a prezice clasa asociată unei imagini oarecare de test. Această procedură va trebui să folosească regula lui Bayes pentru a calcula probabilitaţile ca imaginea de test respectivă să fi fost generată de fiecare dintre cele $10K$ componente ale mixturii, iar apoi să adune probabilităţile corespunzătoare celor K componente asociate fieacărei clase / cifre, obţinând astfel probabilităţile ca imaginea să reprezinte respectiv fiecare cifră de la 0 la 9. Procedura va trebui să returneze aceste probabilităţi, iar ele vor putea fi folosite ulterior pentru a decide cărei cifre îi corespunde imaginea respectivă de test, identificând cifra cu cea mai mare probabilitate.

e. Veţi rula procedura EM pe datele de antrenament şi apoi procedura de predicţie pe toate datele de test pentru $K = 1$; aceasta din urmă ar trebui să producă exact aceleaşi rezultate ca şi algoritmul Bayes Naiv.[1054]

Apoi veţi executa 10 rulări pentru $K = 5$, folosind diferite numere [ca "seed"-uri] pentru iniţializări aleatorii, iar după aceea veţi afişa [ca rezultat] acurateţea la predicţie / testare pentru fiecare dintre aceste 10 rulări.

Pentru fiecare imagine de test şi pentru fiecare clasă, veţi calcula media probabilităţilor de apartenţă a imaginii la clasa respectivă (aceste probabilităţi au fost obţinute la cele 10 rulări), iar apoi veţi folosi aceste probabilităţi medii pentru a clasifica imaginile de test.

Veţi încheia comparând mai întâi acurateţea produsă de către Bayes Naiv ($K = 1$) cu cea produsă folosind mixtura (cu $K = 5$), iar apoi acurateţea produsă de către „ansamblul“ de predicţii cu acurateţile produse folosind câte o singură rulare a algoritmului EM şi, în sfârşit, cu acurateţea produsă de către cea mai bună rulare a algoritmului EM în raport cu log-verosimilitatea datelor de antrenament (cu şi, respectiv, fără termenul de penalizare).

[1054] Veţi oberva că algoritmul EM converge imediat în cazul $K = 1$.

30. **(Modele de mixturi pentru analiza semantică a documentelor de tip text: *modelul K-bag-of-words* şi *modelul domeniilor semantice*)**

□ • ○ ∗ ***CMU, 2011 fall, Eric Xing, HW5, pr. 2***

***Introducere*:**

În acest exerciţiu veţi compara două modele de analiză semantică pentru documente de tip text: modelul K-bag-of-words (în secţiunea A de mai jos) şi modelul mixturii domeniilor semantice (în secţiunea B).[1055] Ambele modele sunt bazate pe *variable latente / neobservabile*: valorile anumitor variabile sunt necunoscute, iar noi suntem interesaţi să găsim aceste valori. În modelul K-bag-of-words, vrem să aflăm pentru fiecare document (i) dintr-o colecţie de documente date care este domeniul semantic „ascuns“ (t_i) asociat documentului respectiv (în manieră "soft"), dintr-o mulţime de K domenii. În modelul domeniilor semantice, suntem interesaţi mai ales de distribuţiile „ascunse“ ($\theta_i \overset{not.}{=} (\theta_i^1, \ldots, \theta_i^K)$, cu $\theta_i^k \in [0,1]$ şi $\sum_{k=1}^K \theta_i^k = 1$) asociate documentului i şi de domeniul semantic (sau: sensul) asociat (şi aici în manieră "soft") fiecărui cuvânt (z_{ij}).

A. Modelul K-bag-of-words — *un sens pentru fiecare document*

În cadrul primei părţi a acestui exerciţiu veţi face cunoştinţă cu o mixtură (oarecum simplificată) de distribuţii categoriale care este folosită pentru a modela [*generarea* de] cuvinte în documente de tip text. Acest model este cunoscut sub numele de *modelul celor K urne de cuvinte* (engl., **K-bag-of-words).**

Punctul de plecare îl constituie K *distribuţii categoriale* care sunt asociate **„urnelor“ de cuvinte. Fiecare dintre aceste K distribuţii categoriale este definită peste cuvintele dintr-un (acelaşi!) vocabular. Numărul de cuvinte din vocabular este V. *Parameterii*** acestor distribuţii sunt desemnaţi respectiv **prin $\beta_1, \ldots, \beta_K$. Fiecare β_k (cu $k = 1, \ldots, K$) este un vector V-dimensional, ale cărui componente sunt numere nenegative care, însumate dau valoarea 1.**

Vom considera N *documente*, numerotate cu $1, \ldots, N$, fiecare document i având M_i cuvinte. Remarcaţi faptul că documentele pot conţine un număr diferit de cuvinte. Pentru fiecare document i, cuvântul cu numărul de ordine j va fi desemnat prin $w_{ij} \in \{1, \ldots, V\}$. Aşadar, vom folosi numere întregi (reprezentând *indicele* poziţiei corespunzătoare din vocabular) pentru a desemna cuvintele din documente.

Cuvintele **(de fapt, *apariţiile* cuvintelor) din cele N documente sunt modelate după cum urmează:**

Se consideră [încă] o distribuţie de probabilitate de tip discret, [la care ne vom referi cu termenul *a priori* şi] care este reprezentată prin vectorul π. Pentru fiecare document i se alege în mod aleatoriu, conform distribuţiei menţionate, un *indicator* [de *domeniu semantic* / tematică] $t_i \in \{1, \ldots, K\}$. Indicatorul [tematic] t_i ne spune care dintre cele K distribuţii categoriale generează cuvintele din documentul i. Apoi, extragem (adică, generăm) fiecare cuvânt w_{ij} al documentului i în conformitate cu parametrii β_{t_i} ai distribuţiei categoriale t_i, extragerile făcându-se în mod independent unele de altele.[1056]

[1055] O versiune simplificată a modelului mixturii domeniilor semantice a fost prezentată la problema 12.
[1056] Aşadar, nu se ţine cont de *ordinea* de apariţie a cuvintelor în document.

Acest design corespunde următorului *proces generativ*:

$$\begin{aligned} t_i &\sim \textit{Categorial}(\pi) \text{ pentru } i \in \{1,\ldots,N\} \\ w_{ij} &\sim \textit{Categorial}(\beta_{t_i}) \text{ pentru } i \in \{1,\ldots,N\} \text{ şi } j \in \{1,\ldots,M_i\}. \end{aligned}$$

Remarcaţi faptul că în acest model, în raport cu modelul clasic de mixtură de K distribuţii categoriale, diferenţa este faptul că se consideră seturi de instanţe / date „observate" care au în comun acelaşi indicator de mixtură t_i (şi anume, cuvintele w_{ij}, cu $j = 1,\ldots,M_i$).

***Indicaţie*: Vă cerem ca, în rezolvarea acestui exerciţiu, să folosiţi indici superiori (engl., superscripts) pentru a desemna componentele vectorilor. De exemplu, prin notaţia π^k veţi desemna elementul de pe poziţia k din vectorul π. De asemenea, variabila-indicator t_i care a fost introdusă mai sus va fi rescrisă sub forma vectorului-indicator $t_i \overset{not.}{=} (t_i^1, t_i^2, \ldots, t_i^K)$. Deci prin t_i^k veţi indica elementul de pe poziţia k din vectorul-indicator t_i. Precizăm că, prin convenţie, în acest vector doar una dintre componente este 1,[1057] toate celelalte fiind 0.**

a. Vectorul de probabilităţi de selecţie π se notează cu $\pi = (\pi^1, \ldots, \pi^K)$. Am precizat deja că probabilitatea selectării unui domeniu semantic $t_i \in \{1,\ldots,K\}$ este π^{t_i}. Scriem formal acest fapt astfel: $P(t_i|\pi) = \pi^{t_i}$.

Vă cerem să exprimaţi într-o manieră unitară probabilităţile $P(t_i|\pi)$ pentru $i = 1,\ldots,N$, folosind un artificiu de genul celui pe care l-am introdus la rezolvarea mixturilor de distribuţii Bernoulli (vedeţi relaţia (417) de la problema 6), extins acum la distribuţia categorială.

b. Exprimaţi probabilitatea $P(w_{ij}|t,\beta,\pi)$ într-o formă cât mai simplă, ţinând cont de relaţiile de (in)dependenţă. Altfel spus, va trebui să scrieţi o egalitate de forma $P(w_{ij}|t,\beta,\pi) = P(w_{ij}|\ldots)$, unde simbolul '...' desemnează o anumită submulţime din $\{t,\beta,\pi\}$. Simbolii t şi β scrişi fără indici inferiori (engl., subscripts) reprezintă ansamblul tuturor vectorilor $t_1,\ldots,t_N$, respectiv $\beta_1,\ldots,\beta_K$.

***Sugestie*: Încercaţi să reprezentaţi grafic modelul de mixtură de tip *K-bag-of-words*. Probabilitatea pe care o veţi indica în răspunsul final va trebui să fie condiţionată în raport cu toţi vectorii $\beta_1,\ldots,\beta_K$.**

c. Explicitaţi probabilitatea „simplificată" pe care aţi indicat-o în răspunsul de la punctul b, folosind definiţia distribuţiei categoriale.

B. Modelul mixturii domeniilor semantice — *un sens pentru fiecare cuvânt*

În cea de-a doua parte a exerciţiului nostru vom explora „mixtura de mixturi" care formează baza aşa-numitului *model al [mixturii] domeniilor semantice* (engl., topic model) pentru clusterizare de documente de tip text.[1058] Putem gândi acest model ca fiind obţinut prin operarea a două modificări asupra modelului (mai simplu) *K-bag-of-words* care a fost prezentat în secţiunea A:

În primul rând, în loc ca documentului i să-i fie asignat un singur *domeniu semantic*[1059] t_i, acum îi vom permite acestuia să fie caracterizat / modelat de o *mixtură de domenii semantice*. Vom asocia acestei mixturi un vector $\theta_i \overset{not.}{=} (\theta_i^1, \ldots, \theta_i^K)$, cu $\theta_i^k \in [0,1]$ şi $\sum_{k=1}^K \theta_i^k = 1$. Mai precis, θ_i trebuie văzut ca o distribuţie de probabilitate definită peste K domenii semantice reprezentate

[1057] Şi anume, poziţia corespunzătoare *variabilei*-indicator t_i, care a fost introdusă anterior.
[1058] Pentru mai multe informaţii, vă recomandăm articolul *Latent Dirichlet Allocation*, Blei et al, 2003.
[1059] Sau: tematică.

respectiv de tuplurile de parametri $\beta_1, \ldots, \beta_K$. Ca şi în secţiunea A, pentru fiecare $k \in \{1, \ldots, K\}$, β_k este un vector, $(\beta_k^1, \ldots, \beta_k^V)$, cu $\beta_k^v \in [0,1]$ şi $\sum_{v=1}^V \beta_k^v = 1$, unde V este mărimea vocabularului. Acesta din urmă este comun pentru toate documentele ($i = 1, \ldots, N$). Distribuţia probabilistă desemnată prin $\theta_i = (\theta_i^1, \ldots, \theta_i^K)$ este una de tip *Dirichlet* şi va fi prezentată mai jos.

În al doilea rând, introducem pentru fiecare cuvânt w_{ij} câte un *indicator* $z_{ij} \in \{1, \ldots, K\}$ care determină domeniul semantic asociat cuvântului w_{ij}. Remarcaţi cum diferă acest design de modelul K-bag-of-words: acum permitem fiecărui cuvânt să aibă propriul său domeniu semantic, în loc să-i impunem să „adopte" domeniul semantic al documentului (t_i).[1060] Desigur, vom genera valorile variabilei z_{ij} conform distribuţiei $\theta_i = (\theta_i^1, \ldots, \theta_i^K)$ asociate documentului i.

Aceste două schimbări dau naştere următorului *proces generativ*:

$$\theta_i = (\theta_i^1, \ldots, \theta_i^K) \sim \textit{Dirichlet}(\alpha) \textbf{ pentru } i \in \{1, \ldots, N\}$$
$$z_{ij} \in \{1, \ldots, K\} \sim \textit{Categorial}(\theta_i) \textbf{ pentru } i \in \{1, \ldots, N\} \textbf{ şi } j \in \{1, \ldots, M_i\},$$
$$w_{ij} \in \{1, \ldots, V\} \sim \textit{Categorial}(\beta_{z_{ij}}) \textbf{ pentru } i \in \{1, \ldots, N\} \textbf{ şi } j \in \{1, \ldots, M_i\},$$

unde M_i este lungimea documentului i, iar $\alpha > 0$ este un parametru scalar pentru distribuţia (simetrică) Dirichlet. Distribuţia Dirichlet este definită sub forma următoare:

$$P(\theta_i|\alpha) = \frac{\Gamma(K\alpha)}{[\Gamma(\alpha)]^K} \prod_{k=1}^K (\theta_i^K)^{\alpha-1},$$

simbolul Γ desemnând funcţia Gamma.[1061]

Observaţi că acest model este într-adevăr o „mixtură de mixturi" de distribuţii categoriale: fiecărui document i i se asociază o mixtură θ_i peste domeniile $\beta_1, \ldots, \beta_K$, şi există N astfel de mixturi, $\theta_1, \ldots, \theta_N$, care constituie împreună mixtura de mixturi de distribuţii categoriale.

d. Exprimaţi probabilitatea $P(z_{ij}|\theta, \alpha, \beta)$ într-o formă cât mai simplă, ţinând cont de relaţiile de (in)dependenţă. Simbolii θ şi β scrişi fără indici (engl., subscripts) reprezintă ansamblurile de parametri $\theta_1, \ldots, \theta_N$ şi respectiv $\beta_1, \ldots, \beta_K$.

e. Explicitaţi probabilitatea „simplificată" pe care aţi indicat-o ca răspuns la punctul precedent, folosind definiţia distribuţiei categoriale.

f. Exprimaţi probabilitatea $P(w_{ij}|z, \theta, \alpha, \beta)$ într-o formă cât mai simplă, ţinând cont de relaţiile de (in)dependenţă.

g. Explicitaţi probabilitatea „simplificată" pe care aţi indicat-o ca răspuns la punctul precedent, folosind definiţia distribuţiei categoriale.

C. Comparaţie

Punctele / întrebările următoare vă vor ajuta să faceţi o *comparaţie* între modelul K-bag-of-words şi modelul mixturii domeniilor semantice pe care le-am prezentat la secţiunile A şi respectiv B.

[1060] Cele două abordări / principii sunt cunoscute în domeniul lingvisticii computaţionale sub forma *un sens la fiecare apariţie* (engl., "one sense per occurrence"), respectiv *un [singur] sens pe întregul document* (engl., "one sense per document").

[1061] Vedeţi problema 46 de la capitolul de *Fundamente*, precum şi http://en.wikipedia.org/wiki/Gamma_function.

h. Atât indicatorul t_i din modelul K-bag-of-words, cât şi distribuţia θ_i din modelul mixturii domeniilor semantice spun ceva anume despre conţinutul tematic (sau: domeniul semantic) al documentului i. Formulaţi într-o singură frază diferenţa principală dintre t_i şi θ_i.

i. Discutaţi implicaţiile răspunsului de la punctul h. În ce fel este mai utilă modelarea de tipul mixturii domeniilor semantice decât *modelarea* de tip K-bag-of-words?

j. Atât în modelul K-bag-of-words cât şi în modelul mixturii domeniilor semantice, parametrii β_k reprezintă vocabulare pentru fiecare domeniu semantic k. Nu discutăm aici despre învăţarea valorilor parametrilor β, dar se poate arăta că anumite strategii de învăţare clasice (de exemplu, *Algoritmul EM*[1062] şi *Gibbs sampling*) vor „produce“ uneori domenii semantice care folosesc în comun (engl., share) anumite cuvinte — altfel spus, putem avea $\beta_k^v > 0$ şi $\beta_l^v > 0$ pentru un cuvânt v şi două domenii semantice distincte k şi l. De ce este oare util aşa ceva (ne referim la *word sharing*)?

8.2.3 Distribuţii binomiale / multinomiale

31. **(Estimarea în sens MLE a parametrilor unor [mixturi de distribuţii Bernoulli, cu ajutorul unor] distribuţii binomiale: rezolvare folosind 1. metode clasice de optimizare şi 2. algoritmul EM)**

prelucrare de Liviu Ciortuz, după
□ • ○ ***U. Toronto, 2015 fall, Statistical Computation course, Radford Neal, HW1 and HW2, pr. 1***

A. Să presupunem că un student de la Universitatea din Toronto este interesat să afle ce procentaj din populaţia adultă din Toronto joacă jocul Minecraft în mod regulat. În acest scop, el a trimis chestionare la 130 de adulţi pe care i-a selectat în mod uniform aleatoriu dintre toţi adulţii din Toronto. În mod surprinzător, el a primit respunsuri de la toţi cei cărora le-a trimis chestionare.[1063] Dintre cele $n = 130$ persoane care au fost incluse în acest sondaj, $x = 75$ au spus că joacă Minecraft.

Estimarea de verosimilitate maximă (MLE) pentru procentajul de adulţi din Toronto care joacă Minecraft se poate obţine uşor: $x/n = 75/130$. Din păcate, studentul îşi dă seama doar acum că nu doar lucrul acesta îl interesează, ci el ar dori să cunoască şi procentajele de bărbaţi şi respectiv de femei [din Toronto] care joacă Minecraft. Însă el nu cunoaşte genul persoanelor care au răspuns la chestionar.

Pentru a rezolva această chestiune, studentul decide să trimită încă un chestionar la 25 de bărbaţi selectaţi în mod aleatoriu dintre bărbaţii din Toronto şi la 25 de femei selectate în mod aleatoriu dintre femeile din Toronto. (El

[1062] Vedeţi problema 12 de la acest capitol.
[1063] Desigur, în cazul sondajelor reale, nereturnarea răspunsurilor la chestionare este o mare problemă, pe care însă noi o vom ignora în cazul de faţă.

nu poate să-şi permită să lucreze cu un eşantion mai mare, fiindcă nu mai are multe fonduri disponibile.) Din fericire, norocul este în continuare de partea lui şi toate persoanele cărora le-a adresat acest al doilea chesionar îi răspund. Dintre cei $m_1 = 25$ de bărbaţi care la care a fost trimis acest al doilea chestionar, $x_1 = 20$ spun că joacă Minecraft, iar dintre cele $m_2 = 25$ de femei la care a fost trimis acest chestionar, $x_2 = 6$ spun că joacă Minecraft.

În cele două secţiuni care urmează, vă vom cere să aflaţi (în două maniere diferite) estimările de verosimilitate maximă pentru procentajul de bărbaţi din Toronto care joacă Minecraft (p_1) şi respectiv procentajul de femei din Toronto care joacă Minecraft (p_2), folosind datele de mai sus. Veţi folosi presupoziţia că populaţia oraşului Toronto este alcătuită în mod egal din bărbaţi şi femei şi veţi presupune, de asemenea, că faptul că o persoană joacă Minecraft este independent de faptul că o altă persoană joacă sau nu Minecraft.

B. Funcţia de verosimilitate a datelor din secţiunea A poate fi scrisă ţinând cont de următoarele fapte:

x urmează distribuţia binomială de parametri n şi $(p_1 + p_2)/2$,

x_1 urmează distribuţia binomială de parametri m_1 şi p_1, iar

x_2 urmează distribuţia binomială de parametri m_2 şi p_2.

Aşadar, putem să exprimăm funcţia de verosimilitate astfel:

$$\begin{aligned} L(p_1, p_2) &= C_n^x((p_1+p_2)/2)^x(1-(p_1+p_2)/2)^{n-x} \\ &\quad \times C_{m_1}^{x_1} p_1^{x_1}(1-p_1)^{m_1-x_1} \times C_{m_2}^{x_2} p_2^{x_2}(1-p_2)^{m_2-x_2} \end{aligned}$$

***Obiectivul* în această secţiune a problemei va fi să implementaţi proceduri care să identifice estimările de verosimilitate maximă pentru procentajele p_1 şi p_2 pentru datele din secţiunea A folosind mai multe metode clasice de optimizare, iar apoi să comparaţi cum anume se comportă ele. În toate aceste metode veţi maximiza log-verosimilitatea, nu verosimilitatea datelor.**

Metodele de optimizare pe care va trebui să le testaţi sunt:

- **Metoda gradientului ascendent.**
- **Metoda Newton-Raphson.**[1064]
- **Metoda maximizării alternante [pe coordonate], folosind metoda bisecţiei pentru a găsi în mod alternativ maximul în raport cu p_1 şi respectiv în raport cu p_2.**[1065]

Veţi testa aceste metode pe datele care au fost descrise mai sus şi veţi discuta cât de bine lucrează ele. În particular, veţi analiza cel puţin următoarele aspecte:

i. cât de sensibile sunt aceste metode la iniţializările aleatorii;

ii. dacă aceste metode produc acelaşi răspuns (cu condiţia unei / unor iniţializări aleatorii corespunzătoare), iar în cazul contrar, care dintre răspunsuri este cel mai bun (adică, se apropie cel mai mult de maximul log-verosimilităţii);

iii. cât de rapid converg aceste metode şi, în mod specific dacă ele converg în timp liniar [LC: în raport cu numărul de instanţe de antrenament], pătratic sau de o altă natură;

[1064] Pentru o uşoară introducere în această metodă, vedeţi problema 80 de la capitolul de *Fundamente*.
[1065] Vedeţi paginile 26-27 din http://cs229.stanford.edu/notes2020fall/notes2020fall/cs229-notes3.pdf.

iv. cât de uşor sunt de implementat aceste metode.

Va trebui să calculaţi funcţia de log-verosimilitate, vectorul ei gradient şi matricea hessiană corespunzătoare. De asemenea, veţi indica estimările pe care le-aţi obţinut, precum şi orice alte reprezentări grafice sau textuale care credeţi că sunt relevante pentu analiza pe care aţi făcut-o.

C. În această secţiune, veţi concepe, veţi implementa şi veţi testa un algoritm EM pentru a găsi estimarea de verosimilitate maximă tot pentru datele care au fost prezentate în secţiunea A, însă acum veţi considera că datele respective sunt modelate folosind o mixtură de distribuţii probabiliste, iar ce componentă anume a mixturii a generat o anumită instanţă poate uneori să fie observabil, iar alteori nu. Datele „neobservabile" manipulate cu ajutorul algoritmului EM vor fi variabilele-indicator care arată ce componentă a mixturii a generat o instanţă oarecare, atunci când această componentă nu este cunoscută.

Aşadar, aici veţi găsi estimările de verosimilitate maximă pentru procentajele p_1 şi p_2 folosind algoritmul EM. Veţi furniza rezultatele pentru aceleaşi date ca mai sus, şi anume $n = 130$, $m_1 = 25$, $m_2 = 25$, $x = 75$, $x_1 = 20$ şi $x_2 = 6$.

Datele „neobservabile" din această problemă sunt genul celor x persoane care au răspuns *da* la primul chestionar, precum şi genul celor $n - x$ persoane care au răspuns *nu* la acel chestionar.

a. Pentru pasul E al algoritmului, veţi calcula valorile medii / „aşteptate" (engl., expected values) pentru aceste două numere, ţinând cont pe de o parte de datele observate şi pe de altă parte de estimările curente pentru p_1 şi p_2.

b. La pasul M, veţi maximiza valoarea medie (engl., expected value) pentru cât ar trebui să fie log-verosimilitatea datelor dacă am cunoaşte aceste numere, această medie fiind calculată în raport cu distribuţia probabilistă care a fost calculată la pasul E (şi care este fixată pe durata execuţiei pasului M).

***Observaţie*: La pasul E, puteţi să calculaţi valorile „aşteptate" ale datelor neobservabile doar în cazurile în care ele sunt neapărat necesare pentru pasul M (întrcât este posibil ca doar mediile anumitor cantităţi să fie necesare pentru calculele de la pasul M).**

c. Calculaţi expresia log-verosimilităţii datelor „observate" şi verificaţi la fiecare iteraţie a algoritmului EM că ea nu descreşte niciodată (probabil cu foarte puţine excepţii, datorate unor erori de rotunjire).

d. Comentaţi succint cât este (sau nu) de uşor să se folosească algoritmul EM pentru această problemă, şi cât de repede se ajunge la convergenţă.

32. (Algoritmului EM pentru „învăţarea“ unei distribuţii multinomiale care este definită cu ajutorul unui [singur] parametru; o aplicaţie în domeniul bioinformaticii)

Liviu Ciortuz, 2017, după
■ □ ○ ***Georgia Institute of Technology,***
Bayesian Statistics course (ISyE 8843A),
Brani Vidakovic, 2004, Handout 12, sec. 1.2.1

***Introducere*:** La problema 14 am arătat cum se pot estima parametrii unei distribuţii categoriale (şi, în mod implicit, ai unei distribuţii multinomiale) în cazul unei probleme de bioinformatică, în care este implicată o genă (numită ABO) care are trei alele. Distribuţia categorială din acea problemă avea trei parametri (de fapt doi, fiindcă suma celor trei parametri trebuie să fie 1). În problema de faţă, vom folosi un singur parametru (ψ) pentru a estima probabilităţile unei alte distribuţii multinomiale, care este specifică unei alte probleme de bioinformatică, unde de această dată sunt considerate două perechi de gene, fiecare genă având câte două alele, iar probabilităţile de „recombinare“ pentru aceste alele sunt diferite în cazul bărbaţilor, faţă de cazul femeilor.[1066]

A. Fie o variabilă aleatoare X care ia valori în mulţimea $\{v_1, v_2, v_3, v_4\}$ şi urmează distribuţia $Multinomial\left(n; \frac{2+\psi}{4}, \frac{1-\psi}{4}, \frac{1-\psi}{4}, \frac{\psi}{4}\right)$, unde ψ este un para-

[1066] Pentru mai multe explicaţii de tip biologic / genetic pentru această problemă, redăm aici secţiunea 1.2.1 din documentul indicat mai sus: *Handout 12*, de la cursul de Statistică bayesiană (ISyE 8843A) care a fost ţinut de către profesorul Brani Vidakovic la Georgia Institute of Technology, în anul 2004.
Vom considera două locusuri bi-alelice legate, A şi B, având alelele A şi a, respectiv B şi b, unde alela A este dominantă în raport cu alela a, iar alela B este dominantă în raport cu alela b.
Un heterozigot dublu $AaBb$ va produce gameţi de patru tipuri: AB, Ab, aB şi ab. Întrucât locusurile sunt legate, tipurile AB şi ab vor apărea cu o frecvenţă diferită de frecvenţa tipurilor Ab şi aB, să zicem $1-r$ şi respectiv r la bărbaţi, şi $1-r'$ şi respectiv r' la femei.
Aici vom presupune că originea parentală a acestor heterozigoţi este cauzată de împerecherea $AABB \times aabb$, astfel încât r şi r' sunt probabilităţile / ratele de recombinare ale celor două locusuri la bărbaţi şi respectiv la femei.
Problema este să estimăm probabilităţile r şi r', dacă este posibil, din descendenţii imediaţi ai heterozigoţilor dubli.
Întrucât gameţii AB, Ab, aB şi ab sunt produşi în proporţii de $(1-r)/2$, $r/2$, $r/2$ şi $(1-r)/2$ de către părinţi masculini, respectiv $(1-r')/2$, $r'/2$, $r'/2$ şi $(1-r')/2$ de către părinţi feminini, zigoţii care au genotipurile $AABB$, $AaBB$, ... etc., sunt produşi cu frecvenţele $(1-r)(1-r')/4$, $(1-r)r'/4$, etc.
Problema este următoarea: deşi există 16 genotipuri distincte de descendenţi, luând în considerare originea parentală, relaţiile de dominare implică faptul că noi observăm doar 4 fenotipuri distincte, pe care le desemnăm prin $A*B*$, $A*b*$, $a*B*$ şi $a*b*$. Aici $A*$ (respectiv $B*$) desemnează ceea ce este dominant, în vreme ce $a*$ (respectiv $b*$) desemnează fenotipurile recesive determinate de alele lui A (respectiv B).
Astfel, indivizii care au genotipurile $AABB$, $AaBB$, $AABb$ sau $AaBb$, care reprezintă 9/16 din combinaţiile de gameţi (verificaţi!), manifestă fenotipul $A*B*$, adică alternativa dominantă în raport cu ambele caractere, în vreme ce indivizii care au genotipurile $AAbb$ sau $Aabb$ (3/16) manifestă fenotipul $A*b*$, indivizii cu genotipurile $aaBB$ şi $aaBb$ (3/16) manifestă fenotipul $a*B*$ şi, în fine, indivizii cu genotipul dublu recesiv $aabb$ (1/16) manifestă fenotipul $a*b*$.
Este cumva surprinzător faptul că probabilităţile celor patru clase fenotipice pot fi definite folosind parametrul $\psi \overset{not.}{=} (1-r)(1-r')$, după cum urmează: $a*b*$ are probabilitatea $\psi/4$ (este uşor de verificat), $a*B*$ şi $A*b*$ au ambele probabilităţile $(1-\psi)/4$, în vreme ce $A*B*$ are probabilitatea 1 minus suma probabilităţillor precedente, adică $(2+\psi)/4$.
Presupunem acum că avem un eşantion aleatoriu format din n descendenţi imediaţi ai heterozigoţilor noştri dubli. Astfel, cele 4 clase fenotipice vor fi reprezentate grosso-modo proporţional cu probabilităţile lor teoretice, distribuţia lor comună fiind de tip multinomial:

$$Multinomial\left(n; \frac{2+\psi}{4}, \frac{1-\psi}{4}, \frac{1-\psi}{4}, \frac{\psi}{4}\right).$$

Observaţi faptul că aici niciuna dintre probabilităţile r şi r' nu poate fi estimată în mod separat din aceste date, ci doar produsul $(1-r)(1-r')$.

metru cu valori în intervalul $(0,1)$. Cele patru probabilităţi listate în definiţia acestei distribuţii multinomiale sunt în corespondenţă directă cu valorile v_1, v_2, v_3 şi respectiv v_4.

a. Presupunem că n_1, n_2, n_3 şi n_4 sunt numărul de „realizări" ale valorilor v_1, v_2, v_3 şi v_4 în totalul celor n „observaţii". (Pentru fixarea ideilor, vom considera $n_1 = 125$, $n_2 = 18$, $n_3 = 20$ şi $n_4 = 34$.) Calculaţi în *manieră analitică* estimarea de verosimilitate maximă (MLE) a parametrului ψ.

b. Fie acum variabila aleatoare $X' \sim \mathit{Multinomial}\left(n; \frac{1}{2}, \frac{\psi}{4}, \frac{1-\psi}{4}, \frac{1-\psi}{4}, \frac{\psi}{4}\right)$. Valorile luate de variabila X', corespunzător acestor cinci probabilităţi, sunt (în ordine) v_1, v_1, v_2, v_3 şi v_4.[1067] Vom considera n_{11}, n_{12}, n_2, n_3 şi respectiv n_4 numărul de „realizări" ale acestor valori, însă de data aceasta n_{11} şi n_{12} vor fi neobservabile. În schimb, vom furniza suma $n_1 = n_{11} + n_{12}$ ca dată „observabilă", alături de celelalte date observabile, n_2, n_3 şi n_4.

Să se estimeze parametrul ψ folosind algoritmul EM. Cum este această estimare faţă de valoarea obţinută la punctul a?

***Indicaţie*: Veţi elabora formulele corespunzătoare pasului E şi pasului M. Apoi veţi face o implementare şi veţi rula algoritmul EM (pornind, de exemplu, cu valoarea iniţială 0.5 pentru ψ) până când valorile acestui parametru până la cea de-a şasea zecimală nu se mai modifică.**

c. Calculaţi — şi apoi reprezentaţi grafic pe intervalul $(0,1)$ — funcţia de log-verosimilitate a datelor observabile

$$\ell_c(\psi) \stackrel{def.}{=} \ln \sum_{n_{11}+n_{12}=n_1} g_c(\underbrace{n_{11}, n_{12}}_{neobs.}, \underbrace{n_2, n_3, n_4}_{obs.}, \psi)$$

unde

$$\begin{aligned} & g_c(n_{11}, n_{12}, n_2, n_3, n_4, \psi) \\ &= \frac{n!}{n_{11}!\, n_{12}!\, n_2!\, n_3!\, n_4!} \left(\frac{1}{2}\right)^{n_{11}} \left(\frac{\psi}{4}\right)^{n_{12}} \left(\frac{1-\psi}{4}\right)^{n_2} \left(\frac{1-\psi}{4}\right)^{n_3} \left(\frac{\psi}{4}\right)^{n_4} \\ &= \frac{n!}{n_{11}!\, n_{12}!\, n_2!\, n_3!\, n_4!} \left(\frac{1}{2}\right)^{n_{11}} \left(\frac{\psi}{4}\right)^{n_{12}+n_4} \left(\frac{1-\psi}{4}\right)^{n_2+n_3} \end{aligned}$$

cu $n = n_{11} + n_{12} + n_2 + n_3 + n_4$.

d. Pentru a pune în evidenţă convergenţa algoritmului EM pe datele de mai sus, la iteraţiile $t = 0, 1, 2$ ale algoritmului EM, la pasul M, calculaţi $\psi^{(t+1)}$ şi trasaţi graficul funcţiei auxiliare
$Q(\psi|\psi^{(t)}) = \ln \frac{n!}{\hat{n}_{11}!\, \hat{n}_{12}!\, n_2!\, n_3!\, n_4!} + \hat{n}_{11} \ln \frac{1}{2} + (\hat{n}_{12} + n_4) \ln \frac{\psi}{4} + (n_2 + n_3) \ln \frac{1-\psi}{4}$**.**

B. Vom presupune acum că parametrul necunoscut ψ urmează o distribuţie a priori, şi anume distribuţia Beta[1068]

$$p(\psi) = \frac{1}{B(\nu_1, \nu_2)} \psi^{\nu_1 - 1} (1-\psi)^{\nu_2 - 1},$$

[1067] Observaţi că, în raport cu distribuţia multinomială de la punctul a, am „descompus" probabilitatea $\frac{2+\psi}{4}$ în două probabilităţi, $\frac{1}{2}$ şi $\frac{\psi}{4}$, ca şi cum ele ar corespunde unor evenimente disjuncte.

[1068] Distribuţia Beta este un caz particular al distribuţiei Dirichlet, care poate fi folosită ca distribuţie a priori pentru parametrii distribuţiei categoriale. Vedeţi ex. 129 şi ex. 130.d de la capitolul de *Fundamente*.

unde $B(\nu_1,\nu_2) \overset{def.}{=} \int_0^1 \psi^{\nu_1-1}(1-\psi)^{\nu_2-1}d\psi \overset{calcul}{=} \dfrac{\Gamma(\nu_1)\Gamma(\nu_2)}{\Gamma(\nu_1+\nu_2)}$**, iar** Γ **este funcţia Gamma a lui Euler.**[1069] **Distribuţia Beta este o conjugată naturală pentru distribuţii ale datelor lipsă (engl., missing data), ca în cazul variabilei** n_{12}**, care urmează distribuţia** $Binomial\left(n_1, \dfrac{\psi/4}{1/2+\psi/4}\right)$**.**[1070]

e. Demonstraţi că expresia log-verosimilităţii condiţionale — ignorând anumiţi termeni, care se însumează şi nu depind de ψ **— este următoarea:**

$$\ln L(\psi) + \ln p(\psi) = (n_{12} + n_4 + \nu_1 - 1)\ln\psi + (n_2 + n_3 + \nu_2 - 1)\ln(1-\psi).$$

f. Scrieţi o explicaţie succintă pentru faptul că pasul E al algoritmului EM pentru această problemă de estimare în sens MAP coincide cu pasul E din algoritmul EM standard. Aşadar, el revine la a înlocui variabila n_{12} **cu media sa condiţionată,** $n_1 \dfrac{\psi^{(t)}/4}{1/2+\psi^{(t)}/4}$**.**

Apoi demonstraţi că regula de actualizare de la pasul M este

$$\psi^{(t+1)} = \frac{n_{12}^{(t)} + n_4 + \nu_1 - 1}{n_{12}^{(t)} + n_2 + n_3 + n_4 + \nu_1 + \nu_2 - 2}.$$

***Observaţie*: Atunci când distribuţia Beta este uniformă (adică, pentru** $\nu_1 = \nu_2 = 1$**), soluţia MAP coincide cu soluţia de verosimilitate maximă (engl., maximum likelihood) standard.**

g. Adaptaţi implementarea pe care aţi realizat-o la punctul b **pentru a lua în considerare şi distribuţia a priori Beta asupra parametrului** ψ**.**

33. **(Algoritmul EM pentru „învăţarea“ parametrilor unei distribuţii multinomiale; aplicare pe datele unui „studiu“ epidemiologic)**

□ • · ***University of Chicago, 2004 spring, Statistics, course, Michael Eichler, HW3, pr. 2***

A. Datele din această problemă au fost preluate dintr-un studiu epidemiologic care a fost realizat în urma înregistrării mai multor cazuri de intoxicaţii alimentare care au apărut după un prânz organizat pentru personalul unei companii de asigurări. În total, 419 persoane au fost contactate pentru acest studiu epidemiologic. În următorul tabel apar două variabile, *Illness*, **care indică dacă persoana respectivă a suferit de intoxicaţie alimentară, precum şi** *Fish*, **care indică dacă persoana respectivă a mâncat peşte la felul principal de la acel prânz:**

	Fish		
Illness	Yes(1)	No(0)	Unknown
Yes(1)	142	4	62
No(0)	104	54	53

[1069] Vedeţi ex. 31.b de la capitolul de *Fundamente*.

[1070] La ex. 43 şi ex. 130 de la capitolul de *Fundamente* aveţi două exemple de folosire a distribuţiei Beta ca distribuţie a priori pentru parametrul distribuţiei Bernoulli şi respectiv al distribuţiei geometrice.

O parte din aceste date sunt necunsocute (engl., unknown / missing) fiindcă doar $m = 304$ persoane au răspuns la solicitările realizatorilor acest studiu. Totuşi, folosind datele companiei a fost posibil să aflăm cine anume a avut intoxicaţie alimentară după acel prânz.

Pentru $k = 1, \ldots, n = 419$, vom considera

$$X_k = \begin{cases} 1 & \text{dacă persoana } k \text{ a suferit de intoxicaţie alimentară,} \\ 0 & \text{în cazul contrar} \end{cases}$$

$$Y_k = \begin{cases} 1 & \text{dacă persoana } k \text{ a mâncat peşte,} \\ 0 & \text{în cazul contrar} \end{cases}$$

şi vom presupune că valorile variabilelor $Y_{m+1}, \ldots, Y_n$ sunt necunoscute. Mai departe, vom nota

$$N_{ij}^{(o)} = \sum_{k=1}^{m} 1_{\{X_k = i \text{ and } Y_k = j\}} \quad \text{şi} \quad N_{ij}^{(m)} = \sum_{k=m+1}^{n} 1_{\{X_k = i \text{ and } Y_k = j\}},$$

pentru $i, j \in \{0, 1\}$.[1071]
Datele observate sunt

$$N_{obs} = (N_{00}^{(o)}, N_{10}^{(o)}, N_{01}^{(o)}, N_{11}^{(o)}, N_{0+}^{(m)}, N_{1+}^{(m)}),$$

cu $N_{i+}^{(m)} = N_{i0}^{(m)} + N_{i1}^{(m)}$ pentru $i \in \{0, 1\}$.
Fie

$$N_{ij} = N_{ij}^{(o)} + N_{ij}^{(m)} \text{ pentru } i, j \in \{0, 1\}$$

datele complete. Presupunem că $N = (N_{ij})$ este distribuit multinomial, cu parametrul $\theta = (\theta_{ij})$,

$$N \sim \mathit{Multinomial}(n, \theta_{00}, \theta_{10}, \theta_{01}, \theta_{11}).$$

a. Formulaţi un algoritm EM pentru acest model. În acest scop, va trebui să specificaţi care anume sunt datele neobservabile şi apoi să deduceţi regulile de actualizare de la pasul E şi respectiv pasul M.

b. Implementaţi algoritmul pe care l-aţi conceput la punctul precedent (în Python, R, Matlab sau limbajul de programare pe care îl preferaţi) şi aplicaţi-l pe datele de mai sus. Veţi iniţializa parametrii $\theta_{00}, \ldots, \theta_{11}$ cu valorile estimărilor de verosimilitate maximă (MLE) care se obţin din primele două coloane ale tabelului din enunţ, adică $\theta_{00}^{(0)} = \frac{54}{304}, \ldots, \theta_{11}^{(0)} = \frac{142}{304}$.

B. Pentru a identifica posibilele cauze ale intoxicării alimentare, vrem să ştim dacă îmbolnăvirile sunt datorate alegerii felului principal de mâncare. În primul rând, va trebui să antrenăm (engl., fit) modelul folosind „ipoteza nulă“ că X_k şi Y_k sunt independente. Cu această presupoziţie, parametrii vor satisface proprietatea

$$\theta_{ij} = (\theta_{i0} + \theta_{i1})(\theta_{0j} + \theta 1 j).$$

Aşadar, modelul va putea fi reparametrizat cu noii parametri

$$\alpha = \theta_{10} + \theta_{11} \quad \text{şi} \quad \beta = \theta_{01} + \theta_{11}.$$

[1071] Vă readucem aminte că notaţia $1_{\{\ \}}$ desemnează o *funcţie-indicator*, a cărei valoare este 1 atunci când este indeplinită condiţia scrisă între acolade, şi 0 în cazul contrar.

c. Folosiţi o nouă variantă a algoritmului EM pentru a antrena (engl., fit) modelul folosind „ipoteza nulă“. Explicaţi de ce pasul E este identic cu cel [de la algoritmul EM] de mai sus şi apoi formulaţi pasul M.

d. Implementaţi noul algoritm EM (în Python, R, Matlab sau limbajul de programare pe care îl preferaţi) şi apoi aplicaţi-l de datele de mai sus. Ca şi la punctul b, veţi iniţializa parametrii α şi β cu valorile estimărilor de verosimilitate maximă (MLE) care se obţin din primele două coloane ale tabelului din enunţ, adică $\alpha^{(0)} = \frac{142+4}{304}$ şi $\beta_{11}^{(0)} = \frac{142+104}{304}$.

e. Verificaţi dacă „ipoteza nulă“ este adevărată. Puteţi realiza aceasta folosind testul raportului de verosimilitate (engl., likelihood ratio test). Testaţi dacă cele două variabile sunt independente. (*Sugestie*: Folosiţi statistica χ^2.)

8.2.4 Alte instanţe ale schemei algoritmice EM

34. **(Algoritmul EM: estimarea parametrilor unei mixturi de distribuţii Poisson)**

□ • · ***University of Chicago, 2004 spring, Statistics course, Michael Eichler, HW3, pr. 3***

Următorul tabel conţine [într-o manieră compactă] numărul de decese ale femeilor în vârstă de 80 de ani sau mai mult care au fost raportate în fiecare zi în cursul anilor 1910-1912.

Numărul de decese (k)	0	1	2	3	4	5	6	7	8	9
Frecvenţa observată (n_k)	162	267	271	185	111	61	27	8	3	1

a. Dacă decesele sunt independente unele de altele şi frecvenţa lor nu se schimbă în timp, atunci pentru a modela aceste date este natural să folosim distribuţia Poisson. Antrenaţi (engl., fit) o distribuţie Poisson pe aceste date. Analizând rezultatul pe care l-aţi obţinut, consideraţi că datele sunt într-adevăr distribuite conform distribuţiei Poisson?

b. Este posibil ca vara şi respectiv iarna să apară pattern-uri diferite cu privire la numărul de decese înregistrate, din care cauză o mixtură de două distribuţii Poisson ar putea să constituie un model mai bun [pentru datele de mai sus] decât o singură distribuţie Poisson.

Considerând că se foloseşte o mixtură de două distribuţii Poisson, scrieţi expresia verosimilităţii datelor observabile.

c. Formulaţi un algoritm EM pentru acest model de mixtură. Pentru aceasta, va trebui să specificaţi care anume sunt datele complete şi datele lipsă şi apoi să deduceţi regulile de actualizare de la pasul E şi respectiv pasul M.

d. Implementaţi în Python / R / Matlab algoritmul EM pe care l-aţi conceput la punctul precedent şi aplicaţi-l pe datele de mai sus. Cât de bine se potriveşte acest model cu datele, comparativ cu distribuţia Poisson de la punctul a? Care dintre cele două modele descrie mai bine datele?

35. **(Algoritmul EM: estimarea parametrilor unei mixturi de vectori de distribuţii Poisson i.[i.]d.)**

■ ● ○ ***Univ. of Utah, 2008 fall, Hal Daumé III, HW9, pr. 1***

Am făcut cunoştinţă cu distribuţia Poisson la problemele 27 şi 46 de la capitolul de *Fundamente*. Vă reamintim că această distribuţie este definită peste numerele naturale pozitive, şi anume: dat fiind parametrul λ, funcţia masă de probabilitate (p.m.f.) a distribuţiei Poisson este dată de expresia

$$p(x|\lambda) = \frac{1}{e^{\lambda}} \cdot \frac{\lambda^x}{x!}, \text{ pentru orice } x \in \mathbb{N}.$$

În problema 46 menţionată mai sus, am văzut că, dată fiind o secvenţă de numere naturale pozitive $x_1, \ldots, x_n$, estimarea de verosimilitate maximă (MLE) pentru parametrul λ este $\frac{1}{n}\sum_{i=1}^{n} x_i$, adică exact media [aritmetică a] numerelor date.

În acest exerciţiu vom considera o generalizare a acestei distribuţii: modelul mixturii de [vectori de] distribuţii Poisson. Nu ştim dacă aţi aflat [sau nu] până acum, însă acest model este folosit la monitorizarea serverelor de internet: numărul de cereri de acces care sunt adresate unui server de internet într-o unitate de timp (de exemplu, un minut) urmează de obicei o distribuţie Poisson.

Considerăm că avem n servere de internet şi că le monitorizăm pe fiecare pe o durată de M minute. Aşadar, vom obţine $n \times M$ numere pozitive (counturi); vom nota cu $x_{i,m}$ numărul de cereri adresate serverului i în minutul m. *Scopul* nostru este să *clusterizăm* serverele de internet în funcţie de frecvenţa cererilor adresate lor [în timp].[1072]

Presupunem că dorim să formăm K clustere. Definiţi un *model* de tip *mixtură de* [*vectori de*] *distribuţii Poisson* pentru această problemă. Veţi nota cu z_i variabila latentă care desemnează cărui cluster de servere de internet îi aparţine serverul i (mai precis, indicând un număr de la 1 la K). De asemenea, veţi nota cu λ_k parametrul distribuţiei Poisson care este urmată de fiecare componentă a vectorilor de variabile Poisson din clusterul k, iar cu π_k probabilitatea de selecţie a respectivului cluster.

Elaboraţi algoritmul EM pentru rezolvarea acestui de tip *mixtură de* [*vectori de*] *distribuţii Poisson*. Stabiliţi pentru pasul E relaţiile de calcul pentru mediile variabilelor latente z_i, iar pentru pasul M relaţiile de actualizare pentru parametrii λ şi probabilităţile de selecţie π pentru distribuţiile din acest model.

***Indicaţie*: Verosimilitatea datelor complete va arăta astfel:**

$$L(\lambda, \pi) \overset{def.}{=} P(\bar{x}, z|\lambda, \pi) \overset{i.i.d.}{=} \prod_{i=1}^{n} P(x_i, z_i|\lambda, \pi) = \prod_{i=1}^{n} P(x_i|z_i, \lambda, \pi) \cdot \underbrace{P(\underbrace{z_i}_{k}|\lambda, \pi)}_{\pi_k}$$

$$= \prod_{i=1}^{n}\prod_{k=1}^{K}\left[\pi_k \prod_{m=1}^{M} p(x_{i,m}|\lambda_k)\right]^{1_{\{z_i=k\}}},$$

unde

[1072] În cazul (limită!) în care $M = 1$, modelul nostru va deveni modelul unei mixturi de K distribuţii de tip Poisson (nu *vectori* de distribuţii de tip Poisson). Pentru cazul (şi mai restrictiv!) $K = 2$, vedeţi pr. 34.

$\bar{x} \overset{not.}{=} (x_1, \ldots, x_n)$**, cu** $x_i \overset{not.}{=} (x_{i,1}, \ldots, x_{i,M})$ **pentru** $i = 1, \ldots, n$**,**
$z \overset{not.}{=} (z_1, \ldots, z_K)$**,** $\pi \overset{not.}{=} (\pi_1, \ldots, \pi_K)$**,** $\lambda \overset{not.}{=} (\lambda_1, \ldots, \lambda_K)$**, iar**
$1_{\{z_i=k\}}$ **este funcţia-indicator; ea ia valoarea** 1 **dacă** $z_i = k$ **şi** 0 **în caz contrar.**

36. **(Algoritmul EM: pasul E pentru estimarea parametrilor unei mixturi de distribuţii Gamma)**

prelucrare de Liviu Ciortuz, după
■ □ • ○ ***CMU, 2012 fall, T. Mitchell, Z. Bar-Joseph, final exam, pr. 2***

Consderăm instanţele $X_i \in \mathbb{R}^+$, $(i = 1, \ldots, n)$ **produse de către următoarea mixtură de distribuţii:**

$$Z_i \sim Categorial(\pi_1, \pi_2, \ldots, \pi_K)$$
$$X_i \sim Gamma(2, \beta_{Z_i})$$

Funcţia de densitate a distribuţiei de probabilitate $Gamma(2, \beta)$ **este definită astfel:** $P(X = x) = \beta^2 x e^{-\beta x}$**.**

a. Vom considera numărul de distribuţii din mixtură $K = 3$ **şi parametrii (secunzi) ai acestor distribuţii,** $\beta_1 = 1, \beta_2 = 2, \beta_3 = 4$**. Calculaţi** $P(Z_i = 1 | X_i = 1)$**.**

b. Elaboraţi pasul E al algoritmului EM pentru estimarea parametrilor acestei mixturi, scriind câte o formulă / relaţie matematică pentru fiecare expresie / cantitate care trebuie calculată la acest pas.

c. Credeţi că acest model de mixtură de distribuţii Gamma poate opera cu clustere non-disjuncte (engl., overlapping), similar cu modelul mixturii de distribuţii gaussiene?

37. **(EM pentru învăţare supervizată: cazul mixturilor de regresori liniari)**

□ • · ***Stanford, 2007 fall, Andrew Ng, HW4, pr. 1***

Algoritmul EM, aşa cum a fost prezentat la curs, se foloseşte pentru învăţare nesupervizată. În particular, el serveşte pentru a reprezenta distribuţia instanţelor de antrenament $p(x)$ **prin marginalizare în raport cu o variabilă aleatoare latentă** z**:**

$$p(x) = \sum_z p(x, z) = \sum_z p(x|z)\, p(z).$$

Însă algoritmul EM poate fi aplicat de asemenea şi pentru învăţare supervizată, iar în această problemă vom discuta *modelul unei mixturi de regresori liniari* (engl., mixture of linear regressors).[1073]

Urmărind să reprezentăm distribuţia condiţionată $p(y|x)$**, unde** $x \in \mathbb{R}^n$ **şi** $y \in \mathbb{R}$**, vom introduce din nou o variabilă aleatoare latentă** z **discretă:**

$$p(y|x) = \sum_z p(y, z|x) = \sum_z p(y|x, z)\, p(z|x).$$

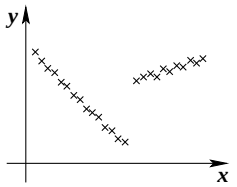
[1073] Mixtura de regresori liniari este un caz particular pentru ceea ce se numeşte *modelul mixturii ierarhice de experţi* (engl., the Hierarchical Mixture of Experts model).

Pentru simplitate vom presupune că z are valori binare, $p(y|x,z)$ este o funcţie densitate de probabilitate (p.d.f) gaussiană, iar distribuţia $p(z|x)$ este produsă de către un model de regresie logistică. În mod formal, vom scrie:

$$p(z|x;\phi) = g(\phi^\top x)^Z (1 - g(\phi^\top x))^{1-z}$$
$$p(y|x, z = i; \theta_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\Big(\frac{-(y - \theta_i^\top x)^2}{2\sigma^2}\Big),\ i = 1, 2,$$

unde g este funcţia logistică / sigmoidală, σ este un parametru cunoscut, iar ϕ, θ_0 şi $\theta_1 \in \mathbb{R}^n$ sunt parametrii modelului.[1074]

În mod ituitiv, *procesul stochastic* care corespunde acestui model poate fi imaginat după cum urmează. Dată fiind o instanţă x, folosind un model de regresie logistică, vom determina mai întâi dacă instanţa respectivă aparţine uneia sau celeilalte dintre cele două clase „ascunse", $z = 0$ sau $z = 1$. Apoi, determinăm y ca o funcţie liniară de x (avem funcţii liniare distincte pentru diferitele valori ale lui z) şi adăugăm zgomotul gaussian, exact ca în modelul standard de regresie liniară. De *exemplu*, următorul set de date poate fi reprezentat bine cu ajutorul unui astfel de model, însă nu poate fi reprezentat bine cu regresia liniară standard.



a. Presupunem că x, y şi z sunt observabile, aşa că obţinem un set de date de antrenament $\{(x^{(1)}, y^{(1)}, z^{(1)}), \ldots, (x^{(m)}, y^{(m)}, z^{(m)})\}$. Scrieţi log-verosimilitatea acestor date în funcţie de parametrii ϕ, θ_0 şi θ_1, iar apoi calculaţi estimările de verosimilitate maximă pentru aceşti parametri. Remarcaţi că datorită faptului că $p(z|x)$ este un model de regresie logistică, nu va exista o formulă analitică (engl., closed form formula) pentru estimarea lui ϕ. Aşadar, calculaţi vectorul-gradient şi matricea hessiană pentru funcţia de verosimilitate a datelor în raport cu ϕ; în practică, acestea pot fi folosite pentru a calcula numeric estimarea de verosimilitate maximă (MLE) a parametrilor.

b. Acum vom presupune că z este o variabilă aleatoare latentă (neobservabilă). Scrieţi log-verosimilitatea datelor în funcţie de parametri, iar apoi deduceţi un algoritm EM care identifică maximul acestei log-verosimilităţi. Formulaţi în mod clar pasul E şi pasul M. (Precizăm din nou că pasul M necesită o rezolvare numerică, aşadar va trebui să calculaţi vectorii-gradient şi matricele hessiene corespunzătoare.)

[1074] Aici folosim indicele $i \in \{1, 2\}$ pentru θ ca să desemnăm doi vectori de parametri distincţi, nu pentru a indexa în aceşti vectori o anumită componentă.

38. **(Algoritmul EM semi-supervizat: cazul mixturilor de distribuţii Bernoulli; aplicare)**

formulare de Liviu Ciortuz, după
□ • ***Stanford, 2020 summer, Andrew Ng, HW3, pr. 4.bc***
CMU, 2012 spring, Ziv Bar-Joseph, HW4, pr. 2

La problema 22 am prezentat o formă generală a algoritmului EM semi-supervizat, precum şi proprietăţile sale de bază: forma regulilor de actualizare pentru pasul E şi pasul M, precum şi proprietatea de convergenţă. Aici vom reveni asupra modelului de mixturi de distribuţii Bernoulli (engl., Bernoulli Mixture Model, BMM), şi-i vom aplica algoritmul EM semi-supervizat.

Vom considera că datele sunt generate de către $K \in \mathbb{N}^*$ distribuţii Bernoulli, având probabilităţile necunoscute $p_j \in [0, 1]$, cu $j \in \{1, \ldots, K\}$. Avem n instanţe $x^{(i)} \in \{0, 1\}$, cu $i \in \{1, \ldots, n\}$, iar fiecărei instanţe îi este asociată o variabilă latentă (ascunsă / necunoscută) $z^{(i)} \in \{1, \ldots, K\}$, indicând ce distribuţie a generat instanţa $x^{(i)}$. În mod specific, $z^{(i)} \sim Categorial(\pi)$, cu $\sum_{j=1}^{K} \pi_j = 1$ şi $\pi_j \geq 0$ pentru orice j, iar $(x^{(i)}|z^{(i)}) \sim Bernoulli(p_{z^{(i)}})$ sunt i.i.d. Aşadar, p şi π, unde $p \overset{not.}{=} (p_1, \ldots, p_K)$ şi $\pi \overset{not.}{=} (\pi_1, \ldots, \pi_K)$, sunt parametrii modelului.

Vom mai considera $\tilde{n}$ instanţe suplimentare, $\tilde{x}^{(i)} \in \mathbb{R}^d$ cu $i \in \{1, \ldots, \tilde{n}\}$, precum şi variabilele *observate* asociate $\tilde{z}^{(i)} \in \{1, \ldots, K\}$, fiecare $\tilde{z}^{(i)}$ indicând distribuţia care a generat instanţa $\tilde{x}^{(i)}$. Remarcaţi faptul că $\tilde{z}^{(i)}$ sunt constante cunoscute (în contrast cu $z^{(i)}$, care sunt variabile *aleatoare* necunoscute). Ca şi mai înainte, vom presupune că $(\tilde{x}^{(i)}|\tilde{z}^{(i)}) \sim Bernoulli(p_{z^{(i)}})$ sunt i.i.d.

Rezumând, avem $n + \tilde{n}$ exemple, dintre care n sunt instanţe neetichetate $x^{(i)}$, asociate cu variabilele neobservabile $z^{(i)}$, iar $\tilde{n}$ sunt instanţe etichetate $\tilde{x}^{(i)}$, având asociate în mod corespunzător etichetele observabile $\tilde{z}^{(i)}$. Algoritmul EM tradiţional este conceput să ia ca input doar n exemple neetichetate şi învaţă un model al cărui parametri sunt p şi π.

Acum va trebui să aplicaţi algoritmul EM semi-supervizat la modelul de mixturi Bernoulli (BMM) pentru a folosi şi cele $\tilde{n}$ instanţe etichetate suplimentare, şi să obţineţi regulile de actualizare specifice pentru paşii E şi M în varianta semi-supervizată.

Pentru *fixarea ideilor*, puteţi începe să lucraţi folosind *setul de date* din tabelul alăturat, care reprezintă rezultatul unui experiment probabilist în care sunt utilizate două monede, care modelează distribuţii Bernoulli.[1075] (Remarcaţi faptul că în acest caz $K = 2$, ceea ce face problema semnificativ mai uşoară.)

Veţi presupune că în coloana *moneda* sunt indicate valorile $z^{(i)}$ (respectiv $\tilde{z}^{(i)}$), şi că $H = 1$ şi $T = 0$.

moneda	$x^{(i)}/\tilde{x}^{(i)}$
?	H
?	H
?	T
?	H
?	T
1	H
2	T
2	T
2	T
2	H

a. [Pasul E, semi-supervizat]
Specificaţi în mod clar care sunt variabilele latente care trebuie să fie rees-

[1075] Datele supervizate sunt luate de la pr. 42 de la capitolul de *Fundamente*. Datele nesupervizate sunt luate de la CMU, 2020 fall, E. Xing, Z. Bar-Joseph, HW4, pr. 5.

timate la pasul E. Deduceţi formulele de actualizare de la pasul E pentru reestimarea tuturor variabilelor latente pe care le-aţi menţionat. Expresia finală de la pasul E trebuie să conţină doar x, z, p, π şi constante universale.

b. [Pasul M, semi-supervizat]
Specificaţi în mod clar care sunt parametrii care trebuie să fie recalculaţi la pasul M. Deduceţi formulele de actualizare pentru toţi parametrii pe care i-aţi menţionat. Mai precis, deduceţi formule analitice (engl., closed form expressions) pentru actualizarea parametrilor $p_j^{(t+1)}$ şi $\pi_j^{(t+1)}$, pornind de la funcţia-obiectiv semi-supervizată.

c. [Aplicare]
Executaţi o iteraţie a algoritmului EM semi-supervizat pe datele din tabelul de mai sus, luând $\alpha = 1$ şi folosind ca valori iniţiale pentru parametrii p_1, p_2, π_1 şi π_2 estimările de verosimilitate maximă (MLE) care pot fi calculate din instanţele complet observabile (adică, etichetate).

39. **(Algoritmul EM: chestiuni metodologice)**

□ • ○ ***CMU, 2014 spring, A. Singh, B. Poczos, HW3, pr. 2.3***

Algoritmul EM converge în general la un optim local. Legat de această chestiune, formulaţi în mod succint câteva strategii care ar putea fi folosite pentru a se obţine totuşi estimări acceptabil de bune ale parametrilor, atunci când se utilizează algoritmul EM în practică.

40. **(Adevărat ori Fals?)**

∗ ***CMU, 2010 spring, E. Xing, T. Mitchell, A. Singh, midterm exam., pr. 1.8.ab***

a. Algoritmul EM optimizează o margine inferioară (engl., lower bound) a funcţiei sale obiectiv, $\ln \prod_i P(x_i \mid \theta)$, unde $x_1, \ldots, x_n$ sunt datele observate, iar θ sunt parametrii modelului asociat acestor date.

b. Funcţia obiectiv optimizată de algoritmul EM poate fi optimizată şi cu metoda gradientului, care va găsi optimul global, în vreme ce EM găseşte soluţia sa mai rapid dar poate returna doar un optim local. Adevărat sau fals? Justificaţi.



© *M. Romanică*

9 Modele Markov ascunse

Sumar

- **Noţiuni preliminare: programare dinamică, schema algoritmică EM.**
- **Verificarea înţelegerii unor noţiuni de bază în referitoare la modelele Markov ascunse (engl., Hidden Markov Models, HMM): ex. 1-4, ex. 15-17.**
- **Model Markov vizibil: exemplificare, legătura cu HMM: ex. 5.**
- **HMM ca model probabilist total: ex. 14.**
- **Algoritmul Forward:**[1076]
  exemple de aplicare: ex. 6, ex. 7, ex. 9.a;
  demonstrarea formulei de la pasul inductiv: ex. 18.a;
  calcularea probabilităţii de emitere a unei secvenţe, folosind probabilităţile Forward: ex. 18.b.
- **Algoritmul Backward:**[1077]
  demonstrarea formulei de la pasul inductiv: ex. 8;
  exemplu de aplicare: ex. 9.a.
- **Algoritmul Viterbi:**[1078]
  exemplu de aplicare: ex. 9.b;
  determinarea căii celei mai probabile de generare a unei secvenţe: ex. 9.c;
  determinarea probabilităţii ca un anumit simbol [din secvenţa de semnale] să fi fost generat într-o anumită stare: ex. 9.d;
  o variaţie pe tema algoritmului Viterbi: ex. 20.
- **Algoritmul Forward-Backward / Baum-Welch / EM pentru HMM:**[1079]
  exemplu de aplicare: ex. 11.b;
  demonstrarea formulei necesare pentru calcularea mediilor variabilelor neobservabile care corespund tranziţiilor: ex 10;
  demonstrarea faptului că algoritmul Forward-Backward lasă neschimbate probabilităţile de tranziţie sau de emisie care sunt nule [la iniţializare]: ex 12.
  - **Demonstrarea unei formule alternative — în raport cu formulele bazate pe probabilităţile Forward şi respectiv probabilităţile Backward — pentru calcularea probabilităţii de emitere a unei secvenţe: ex. 19.**
- **HMM cu emisii gaussiene: ex. 13, ex. 22.**

[1076] Pentru calculul probabilităţilor Forward, notate cu $\alpha_i(t) = P(O_1, \ldots, O_t, X_{t+1} = S_i)$.
[1077] Pentru calculul probabilităţilor Backward, notate cu $\beta_i(t) = P(O_t O_{t+1} \ldots O_T \mid X_t = S_i)$.
[1078] Pentru calculul cantităţilor $\delta_i(t) = \max_{X_1 \ldots X_{t-1}} P(X_1 \ldots X_{t-1}, O_1 \ldots O_{t-1}, X_t = s_i)$.
[1079] Pentru „învăţarea“ parametrilor (i.e., a probabilităţilor de tranziţie şi respectiv de emisie) ai / ale unui HMM.

9.1 Modele Markov ascunse — Probleme rezolvate

1. **(Verificarea înţelegerii unor noţiuni de bază)**

CMU, 2001 fall, Andrew Moore, final, pr. 7

Fie un model Markov ascuns (engl., Hidden Markov Model, HMM) cu trei stări s_1, s_2, s_3 şi trei simboli de emisie observabili X, Y, Z. Probabilităţile de start, de tranziţie şi respectiv de emisie sunt definite de următoarele matrice:

$$\begin{bmatrix} \pi_1 = 1 \\ \pi_2 = 0 \\ \pi_3 = 0 \end{bmatrix}$$

$$\begin{bmatrix} a_{11} = 1/2 & a_{12} = 1/2 & a_{13} = 0 \\ a_{21} = 0 & a_{22} = 1/2 & a_{23} = 1/2 \\ a_{31} = 0 & a_{32} = 0 & a_{33} = 1 \end{bmatrix}$$

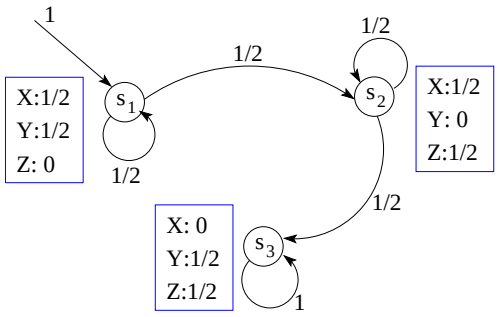
$$\begin{bmatrix} b_1(X) = 1/2 & b_1(Y) = 1/2 & b_1(Z) = 0 \\ b_2(X) = 1/2 & b_2(Y) = 0 & b_2(Z) = 1/2 \\ b_3(X) = 0 & b_3(Y) = 1/2 & b_3(Z) = 1/2 \end{bmatrix}$$

a. Desenaţi automatul nedeterminist corespunzător acestui model Markov.

b. Presupunînd că acest model a emis secvenţa de semnale $XZXYYZYZZ$, identificaţi cea mai probabilă secvenţă de stări corespunzătoare, precum şi probabilitatea acestei emisii. Se consideră că emisiile au loc la intrarea în stările respective.[1080] **(Nu este necesară aplicarea algoritmului Viterbi.)**

Răspuns:

a. Automatul este cel din figura alăturată:



b. Deoarece în acest caz particular, în fiecare stare a automatului este o probabilitate de emisie nulă, putem identifica relativ uşor din structura sa cea mai probabilă secvenţă de stări pentru a fi emisă secvenţa de semnale $XZXYYZYZZ$, şi anume: $s_1 s_2 s_2 s_3 s_3 s_3 s_3 s_3 s_3 s_3$.

Probabilitatea acestei emisii se poate calcula astfel:

$$P(X) = \pi_1 \cdot b_1(X) = \frac{1}{2}$$

$$P(XZ) = P(X) \cdot a_{12} b_2(Z) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2^3}$$

$$P(XZX) = P(XZ) \cdot a_{22} b_2(X) = \frac{1}{2^3} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2^5}$$

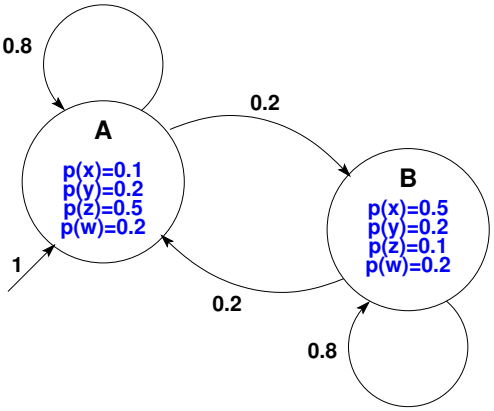
$$P(XZXY) = P(XZX) \cdot a_{23} b_3(Y) = \frac{1}{2^7}$$

[1080] *Atenţie*! La fel vom proceda şi în toate problemele care urmează, dacă nu se specifică altfel, în mod explicit.

$$P(XZXYY) = P(XZXY) \cdot a_{33}b_3(Y) = \frac{1}{2^8}$$
$$P(XZXYYZ) = P(XZXYY) \cdot a_{33}b_3(Z) = \frac{1}{2^9}$$
$$P(XZXYYZY) = P(XZXYYZ) \cdot a_{33}b_3(Y) = \frac{1}{2^{10}}$$
$$P(XZXYYZYZ) = P(XZXYYZY) \cdot a_{33}b_3(Z) = \frac{1}{2^{11}}$$
$$P(XZXYYZYZZ) = P(XZXYYZYZ) \cdot a_{33}b_3(Z) = \frac{1}{2^{12}}.$$

2. **(Verificarea înţelegerii unor noţiuni de bază)**
CMU, 2002 fall, Andrew Moore, final, pr. 3

Considerăm modelul Markov ascuns ilustrat în figura alăturată. Notaţie: q_t şi O_t sunt starea şi respectiv semnalul observat la momentul t.



Calculaţi:

a. $P(q_2 = A)$
b. $P(O_2 = x)$
c. $P(q_2 = A \mid O_2 = x)$
d. $P(O_{100} = w)$
e. Care este cea mai probabilă secvenţă de stări dacă $O_1 = O_2 = O_3 = O_4 = O_5 = x$. (Nu este necesară aplicarea algoritmului Viterbi.)

Răspuns:

a. Probabilitatea ca la momentul 2 automatul să fie în starea A este:

$$P(q_2 = A) = 1 \cdot 0.8 = 0.8$$

b. Probabilitatea ca la momentul 2 automatul să emită simbolul x este:

$$\begin{aligned} P(O_2 = x) &= P(O_2 = x, q_2 = A) + P(O_2 = x, q_2 = B) \\ &= 0.8 \cdot 0.1 + 0.2 \cdot 0.5 = 0.08 + 0.1 = 0.18 \end{aligned}$$

c. Probabilitatea ca automatul să fie în starea A la momentul 2, ştiind că al doilea simbol emis este x:

$$P(q_2 = A \mid O_2 = x) = \frac{P(q_2 = A, O_2 = x)}{P(O_2 = x)} = \frac{0.1 \cdot 0.8}{0.18} = \frac{0.08}{0.18} = \frac{4}{9}$$

d. Probabilitatea cerută este:

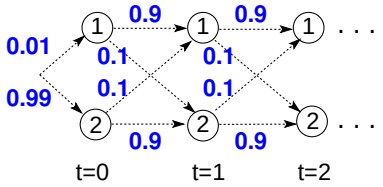
$$\begin{aligned} P(O_{100} = w) &= P(q_{100} = A, O_{100} = w) + P(q_{100} = B, O_{100} = w) \\ &= 0.2 \cdot P(q_{100} = A) + 0.2 \cdot P(q_{100} = B) \\ &= 0.2 \cdot [P(q_{100} = A) + P(q_{100} = B)] = 0.2 \cdot 1 = 0.2 \end{aligned}$$

e. Cea mai probabilă secvenţă de stări este $ABBBB$, întrucât probabilitatea de a emite x este mai mare în starea B decât în starea A. Se intră iniţial în starea A deoarece $\pi_A = 1$, dar apoi orice rămânere în starea A conduce la penalizare.

3. **(Verificarea înţelegerii unor noţiuni de bază)**

CMU, 2003 fall, T. Mitchell, A. Moore, final, pr. 10

Considerăm un model Markov ascuns care modelează aruncarea unei monede imperfecte. Figura alăturată ilustrează tranziţiile împreună cu probabilităţile asociate lor în acest model.



Probabilitatea ca în urma aruncării monedei să se obţină *cap* respectiv *pajură* este:

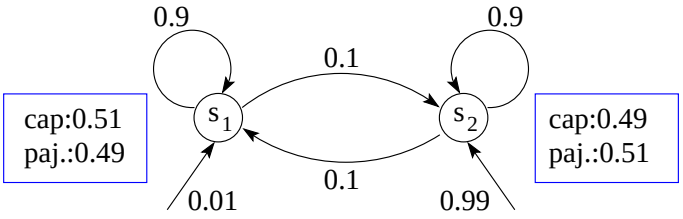
$$P(x = cap \mid s = 1) = 0.51,\ P(x = pajură \mid s = 1) = 0.49$$
$$P(x = cap \mid s = 2) = 0.49,\ P(x = pajură \mid s = 2) = 0.51$$

a. Presupunînd că la trei aruncări succesive ale monedei s-a obţinut de fiecare dată *cap*, care este cea mai probabilă secvenţă de stări? (Nu este necesar să se folosească algoritmul Viterbi.)

b. Care este cea mai probabilă secvenţă de stări dacă la un număr foarte mare (de exemplu: 10^6) de aruncări consecutive ale monedei obţinem un acelaşi rezultat: *cap*.

Răspuns:

Automatul este:



a. Cea mai probabilă secvenţă de stări este $2, 2, 2$. Probabilităţile de a obţine simbolul *cap* sunt aproape identice pentru cele două stări, este foarte probabil ca sistemul să înceapă cu starea 2 şi să rămână aici. Se observă că se pierde mult din probabilitate dacă se trece în starea 1. (Emiterea acestei secvenţe de simboluri în starea 2 se face cu probabilitatea $0.99 \cdot 0.49 \cdot 0.9 \cdot 0.49 \cdot 0.9 \cdot 0.49$.)

b. Când creşte foarte mult numărul de aruncări şi de obţinere a rezultatului *cap*, creşte şi presiunea sistemului de a merge în starea 1, aceasta având un mic avantaj. La un moment dat se va face trecerea în starea 1, iar de aici nu există niciun avantaj de a merge înapoi în starea 2. Costul tranziţiei de la starea 2 la starea 1 este acelaşi indiferent de moment, însă realizarea tranziţiei mai devreme e mai favorabilă, întrucât creşte probabilitatea obţinerii unei serii lungi de *cap*. Aşadar, cel mai bine este să se înceapă cu s_2 şi să se continue imediat cu s_1. Se obţine astfel secvenţa de stări: $2, 1, 1, \ldots 1$ şi probabilitatea $0.99 \cdot 0.49 \cdot 0.1 \cdot 0.51 \cdot 0.9 \cdot 0.51 \cdot 0.9 \ldots 0.51 \cdot 0.9$.

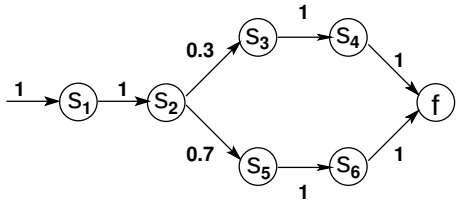
4. **(Verificarea înţelegerii unor noţiuni de bază)**

CMU, 2004 fall, T. Mitchell, Z. Bar-Joseph, final, pr. 5

Fie modelul Markov ascuns definit de matricea de tranziţii şi emisii din tabelul următor:

	0	s_1	s_2	s_3	s_4	s_5	s_6	f	A	C	G	T
0	0	1	0	0	0	0	0	0				
s_1	0	0	1	0	0	0	0	0	0.5	0.3	0	0.2
s_2	0	0	0	0.3	0	0.7	0	0	0.1	0.1	0.2	0.6
s_3	0	0	0	0	1	0	0	0	0.2	0	0.1	0.7
s_4	0	0	0	0	0	0	0	1	0.1	0.3	0.4	0.2
s_5	0	0	0	0	0	0	1	0	0.1	0.3	0.3	0.3
s_6	0	0	0	0	0	0	0	1	0.2	0.3	0	0.5

Automatul este cel din figura următoare:



La fiecare din punctele de mai jos, puneţi unul din semnele $<$, $>$, $=$ între cei doi termeni specificaţi.

a. $P(O_1 = A,\, O_2 = C,\, O_3 = T,\, O_4 = A,\, X_1 = s_1,\, X_2 = s_2)$
$P(O_1 = A,\, O_2 = C,\, O_3 = T,\, O_4 = A \mid X_1 = s_1,\, X_2 = s_2)$.

b. $P(O_1 = A,\, O_2 = C,\, O_3 = T,\, O_4 = A,\, X_3 = s_3,\, X_4 = s_4)$
$P(O_1 = A,\, O_2 = C,\, O_3 = T,\, O_4 = A \mid X_3 = s_3,\, X_4 = s_4)$.

c. $P(O_1 = A,\, O_2 = C,\, O_3 = T,\, O_4 = A,\, X_3 = s_3,\, X_4 = s_4)$
$P(O_1 = A,\, O_2 = C,\, O_3 = T,\, O_4 = A,\, X_3 = s_5,\, X_4 = s_6)$.

d. $P(O_1 = A,\, O_2 = C,\, O_3 = T,\, O_4 = A)$
$P(O_1 = A,\, O_2 = C,\, O_3 = T,\, O_4 = A,\, X_3 = s_3,\, X_4 = s_4)$.

e. $P(O_1 = A,\, O_2 = C,\, O_3 = T,\, O_4 = A)$
$P(O_1 = A,\, O_2 = C,\, O_3 = T,\, O_4 = A \mid X_3 = s_3,\, X_4 = s_4)$.

f. $P(O_1 = A,\, O_2 = C,\, O_3 = T,\, O_4 = A)$
$P(O_1 = A,\, O_2 = T,\, O_3 = T,\, O_4 = G)$.

Răspuns:

a. $p_1 = P(O_1 = A, O_2 = C, O_3 = T, O_4 = A, X_1 = s_1, X_2 = s_2)$,
$p_2 = P(O_1 = A, O_2 = C, O_3 = T, O_4 = A \mid X_1 = s_1, X_2 = s_2)$.

Se observă că $p_1 = p_2 \cdot P(X_1 = s_1, X_2 = s_2) = p_2 \cdot 1 \cdot 1$ **(**$\pi_1 = 1$ **şi** $a_{s_1 s_2} = 1$**). Deci rezultă** $p_1 = p_2$.

b. $p_1 = P(O_1 = A, O_2 = C, O_3 = T, O_4 = A, X_3 = s_3, X_4 = s_4)$,
$p_2 = P(O_1 = A, O_2 = C, O_3 = T, O_4 = A \mid X_3 = s_3, X_4 = s_4)$.

Putem scrie că $p_1 = p_2 \cdot P(X_3 = s_3, X_4 = s_4) = p_2 \cdot 0.3 \cdot 1$**. Aşadar,** $p_1 < p_2$**.**

c. $p_1 = P(O_1 = A, O_2 = C, O_3 = T, O_4 = A, X_3 = s_3, X_4 = s_4)$**,**
$p_2 = P(O_1 = A, O_2 = C, O_3 = T, O_4 = A, X_3 = s_5, X_4 = s_6)$**.**

În acest caz se vor calcula cele două probabilităţi:

$$\begin{aligned} p_1 &= P(O_1 = A, O_2 = C, O_3 = T, O_4 = A \mid X_3 = s_3, X_4 = s_4) \cdot P(X_3 = s_3, X_4 = s_4) \\ &= P(O_1 = A, O_2 = C) \cdot P(O_3 = T, O_4 = A, X_3 = s_3, X_4 = s_4) \\ &= 0.5 \cdot 0.1 \cdot 0.7 \cdot 0.1 \cdot 0.3 = 0.05 \cdot 0.021 \\ p_2 &= P(O_1 = A, O_2 = C, O_3 = T, O_4 = A \mid X_3 = s_5, X_4 = s_6) \cdot P(X_3 = s_5, X_4 = s_6) \\ &= P(O_1 = A, O_2 = C) \cdot P(O_3 = T, O_4 = A, X_3 = s_5, X_4 = s_6) \\ &= 0.5 \cdot 0.1 \cdot 0.3 \cdot 0.2 \cdot 0.7 = 0.05 \cdot 0.042 \end{aligned}$$

Prin urmare, $p_1 < p_2$**.**

d. $p_1 = P(O_1 = A, O_2 = C, O_3 = T, O_4 = A)$**,**
$p_2 = P(O_1 = A, O_2 = C, O_3 = T, O_4 = A, X_3 = s_3, X_4 = s_4)$**.**

Se observă că $p_1 = p_2 + P(O_1 = A, O_2 = C, O_3 = T, O_4 = A, X_3 = s_5, X_4 = s_6)$**, iar cel de-al doilea termen al doilea al acestei sume este strict pozitiv (conform calculelor de mai sus). În consecinţă,** $p_1 > p_2$**.**

e. $p_1 = P(O_1 = A, O_2 = C, O_3 = T, O_4 = A)$**,**
$p_2 = P(O_1 = A, O_2 = C, O_3 = T, O_4 = A \mid X_3 = s_3, X_4 = s_4)$**.**

$$\begin{aligned} p_1 &= P(O_1 = A, O_2 = C, O_3 = T, O_4 = A, X_3 = s_3, X_4 = s_4) + \\ &\quad + P(O_1 = A, O_2 = C, O_3 = T, O_4 = A, X_3 = s_5, X_4 = s_6) \\ &= p_2 \cdot P(X_3 = s_3, X_4 = s_4) + \\ &\quad + P(O_1 = A, O_2 = C, O_3 = T, O_4 = A \mid X_3 = s_5, X_4 = s_6) \cdot P(X_3 = s_5, X_4 = s_6) \end{aligned}$$

Rezultă că $p_1 < p_2$**.**

f. $p_1 = P(O_1 = A, O_2 = C, O_3 = T, O_4 = A)$**,**
$p_2 = P(O_1 = A, O_2 = T, O_3 = T, O_4 = G)$**.**

Se calculează imediat
$p_1 = 0.5 \cdot 0.1 \cdot (0.3 \cdot 0.7 \cdot 0.1 + 0.7 \cdot 0.3 \cdot 0.2) = 0.5 \cdot 0.1 \cdot 0.7 \cdot (0.03 + 0.06) = 0.5 \cdot 0.1 \cdot 0.7 \cdot 0.09$
şi $p_2 = 0.5 \cdot 0.6 \cdot (0.3 \cdot 0.7 \cdot 0.4 + 0.7 \cdot 0.3 \cdot 0) = 0.5 \cdot 0.6 \cdot 0.7 \cdot 0.3 \cdot 0.4$**.**
$p_1 < p_2 \Leftrightarrow 0.1 \cdot 0.09 < 0.6 \cdot 0.12$ **(A). În concluzie,** $p_1 < p_2$**.**

5. **(Model Markov vizibil: exemplificare; legătura cu un model Markov ascuns)**

Helélène Touzet, Université de Lille 1, France

Se consideră secvenţa de nucleotide a unui organism imaginar *Bizarrus Examinus*, care are următoarele caracteristici:

- **Nucleotida actuală este cu probabilitate de** 25% **din bazele A, C, G sau T, dacă precedentele două nucleotide sunt identice.**

- **Altfel (adică: în cazul în care precedentele două nucleotide sunt diferite), probabilitatea ca nucleotida actuală să fie C sau G este de două ori mai mare decat A sau T. Mai mult, între C şi G (respectiv A şi T), *purinele* (A şi G) vor apărea mai frecvent decât *pirimidinele* (C sau T), şi anume în 60% din cazuri.**

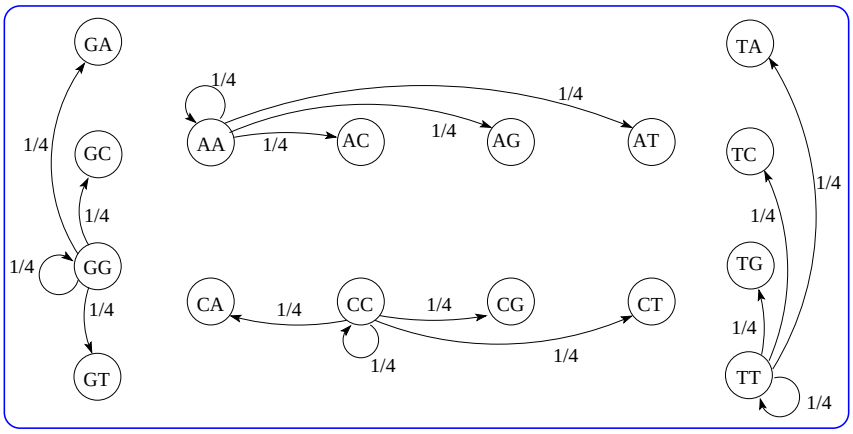
Arătaţi că secvenţa *Bizarrus Examinus* poate fi modelată folosind un model Markov vizibil; veţi indica stările şi probabilităţile de tranziţie.

Răspuns:

Stările modelului Markov cerut sunt toate combinaţiile posibile de câte 2 nucleotide, care înseamnă ultimele două nucleotide produse, fiind deci posibilă tranziţia dintr-o stare în alta în cazul în care ultima nucleodită a primei stări este egală cu prima nucleotidă din starea a doua.

***Observaţie*: Fiecare stare a modelului Markov are 4 posibile tranziţii. Cum există $4 \cdot 4 = 16$ stări, vor fi 72 de tranziţii.**

Să începem cu stările cu nucleotide identice. Din enunţ obţinem că probabilităţile de tranziţie din aceste stări sunt $\frac{1}{4}$. Putem reprezenta aceste tranziţii astfel:



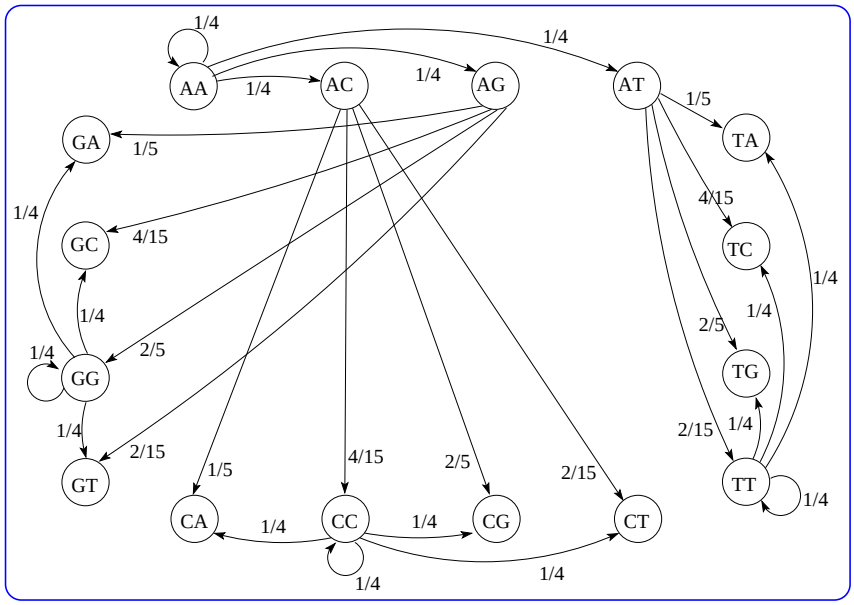
În continuare ne vom ocupa de tranziţiile din stările cu nucleotide diferite. Să notăm cu pq prima stare, unde $p \neq q$, şi probabilităţile vor fi:

$$\begin{aligned} P(X_{i-1} = pq, X_i = qA) &= \frac{1}{3} \cdot \frac{6}{10} = \frac{1}{5} \\ P(X_{i-1} = pq, X_i = qC) &= \frac{2}{3} \cdot \frac{4}{10} = \frac{4}{15} \\ P(X_{i-1} = pq, X_i = qG) &= \frac{2}{3} \cdot \frac{6}{10} = \frac{2}{5} \\ P(X_{i-1} = pq, X_i = qT) &= \frac{1}{3} \cdot \frac{4}{10} = \frac{2}{15} \end{aligned}$$

Altfel scris, probabilităţile de tranziţie dintr-o stare cu nucleotide diferite sunt în funcţie de nucleotida actuală:

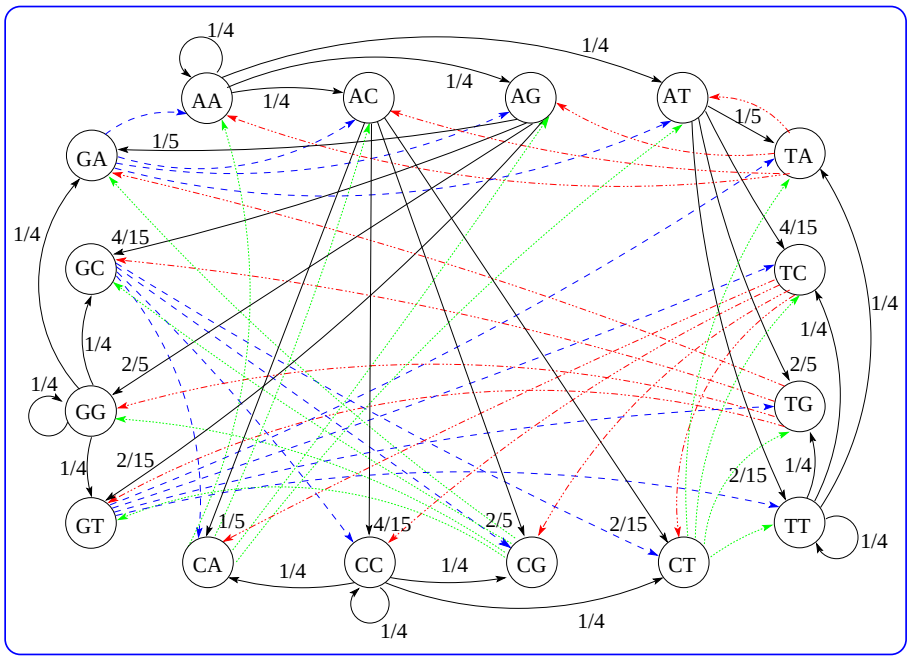
$$A : \frac{1}{5}, \quad C : \frac{4}{15}, \quad G : \frac{2}{5}, \quad T : \frac{2}{15}$$

Dacă trasăm tranziţiile din stările AC, AG şi AT obţinem:



În mod similar vor fi trasate şi tranziţiile din celelalte stări (GA, GC, GT, CA, CG, CT, TA, TC şi TG).

Rezultatul este un model Markov de forma:



Simplificând, putem asocia acestui model Markov vizibil un model Markov ascuns cu

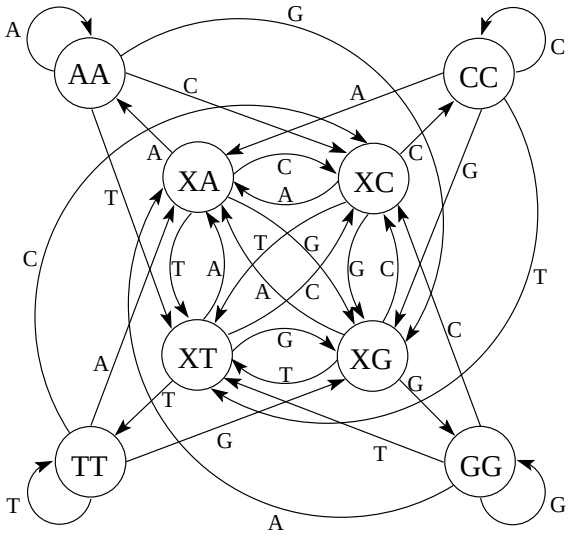
– **8 stări: AA, XA (unde X semnifică literă diferită de A), CC, XC (cu X diferit de C), GG, XG (cu X diferit de G), TT, XT (cu X diferit de T);**

– **emisii de câte un caracter, conform desenului de mai jos, având probabilitatea 1;**

– **tranziţii având probabilităţile (calculate mai sus):**

AA, AA	$1/4$	GG, XA	$1/4$
AA, XC	$1/4$	GG, XC	$1/4$
AA, XG	$1/4$	GG, GG	$1/4$
AA, XT	$1/4$	GG, XT	$1/4$
CC, XA	$1/4$	TT, XA	$1/4$
CC, CC	$1/4$	TT, XC	$1/4$
CC, XG	$1/4$	TT, XG	$1/4$
CC, XT	$1/4$	TT, TT	$1/4$

XA, AA	$1/5$	XG, XA	$1/5$
XA, XC	$4/15$	XG, XC	$4/15$
XA, XG	$2/5$	XG, GG	$2/5$
XA, XT	$2/15$	XG, XT	$2/15$
XC, XA	$1/5$	XT, XA	$1/5$
XC, CC	$4/15$	XT, XC	$4/15$
XC, XG	$2/5$	XT, XG	$2/5$
XC, XT	$2/15$	XT, TT	$2/15$



6. **(Algoritmul Forward: aplicare)**

CMU, 2006 fall, E. Xing, T. Mitchell, final exam, pr. 6

Fie un model Markov ascuns cu stările $X_t \in \{S_1, S_2, S_3\}$, observaţiile $O_t \in \{A, B, C\}$ şi parametrii

$\pi_1 = 1$	$a_{11} = 1/2$	$a_{12} = 1/4$	$a_{13} = 1/4$	$b_1(A) = 1/2$	$b_1(B) = 1/2$	$b_1(C) = 0$
$\pi_2 = 0$	$a_{21} = 0$	$a_{22} = 1/2$	$a_{23} = 1/2$	$b_2(A) = 1/2$	$b_2(B) = 0$	$b_2(C) = 1/2$
$\pi_3 = 0$	$a_{31} = 0$	$a_{32} = 0$	$a_{33} = 1$	$b_3(A) = 0$	$b_3(B) = 1/2$	$b_3(C) = 1/2$

a. Calculaţi $P(X_5 = S_3)$**.**

Pentru punctele b**,** c **şi** d**, vom presupune că observăm secvenţa** $AABCABC$**. Se consideră că emisia simbolului** O_t **are loc la ieşirea din starea** X_t**.**

b. Calculaţi $P(X_5 = S_3 \mid O = AABCABC)$**.**

c. Completaţi următorul tabel, presupunând observaţia $O = AABCABC$**. Notaţie:** $\alpha_i(t) = P(O_1, \ldots, O_t, X_{t+1} = S_i)$**.**

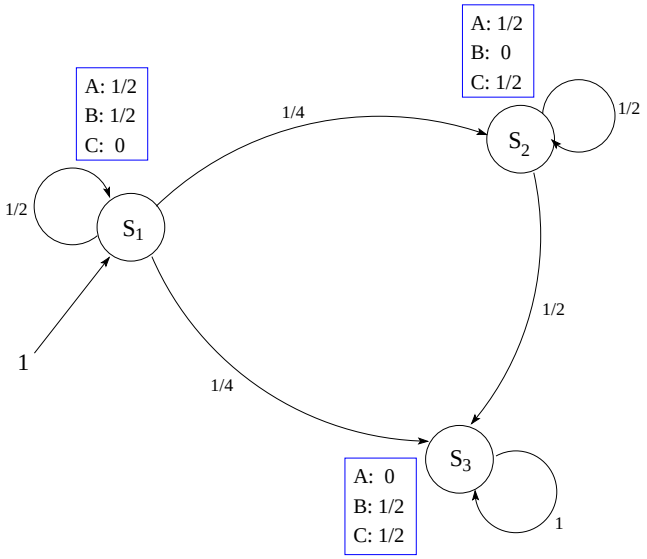
t	1	2	3	4	5	6	7	8
$\alpha_1(t)$								
$\alpha_2(t)$								
$\alpha_3(t)$								

d. Ştiind că a fost emisă secvenţa $O = AABCABC$, scrieţi secvenţa de stări X care are probabilitate maximă a posteriori, $P(X|O)$.

Cât este această probabilitate a posteriori?

Răspuns:

Modelul Markov este cel din figura alăturată:



a. A calcula $P(X_5 = S_3)$ enumerând toate posibilităţile pentru X_1, X_2 X_3 şi X_4 conduce la prea multe variante. Probabilitatea evenimentului contrar, $P(X_5 \neq S_3) = P(X_5 = S_1$ sau $X_5 = S_2)$ este însă mult mai uşor de calculat. Aşadar, vom folosi formula $P(X_5 = S_3) = 1 - P(X_5 = S_1) - P(X_5 = S_2)$.

Pentru ca la momentul 5 automatul să fie în starea 1, se observă că există o singură posibilitate, şi anume ca automatul să intre în starea 1 şi să rămână acolo, deci:

$$P(X_5 = S_1) = 1 \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{16}$$

Pentru $P(X_5 = S_2)$ avem patru posibilităţi, corespunzătoare celor patru momente în care se poate face trecerea din starea 1 în starea 2, deci:

$$P(X_5 = S_2) = 4 \cdot 1 \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{4} = \frac{1}{8}$$

$$P(X_5 = S_3) = 1 - \frac{1}{16} - \frac{1}{8} = \frac{13}{16}$$

b. Întrucât la momentul 5 s-a emis simbolul A, iar probabilitatea de emitere a lui A în starea 3 este 0, rezultă $P(X_5 = S_3 \mid O = AABCABC) = 0$.

c. Tabelul completat arată astfel:

t	1	2	3	4	5	6	7	8
O_t	A	A	B	C	A	B	C	
$\alpha_1(t)$	1	$\frac{1}{2^2}$	$\frac{1}{2^4}$	$\frac{1}{2^6}$	0	0	0	0
$\alpha_2(t)$	0	$\frac{1}{2^3}$	$\frac{1}{2^4}$	$\frac{1}{2^7}$	$\frac{1}{2^9}$	$\frac{1}{2^{11}}$	0	0
$\alpha_3(t)$	0	$\frac{1}{2^3}$	$\frac{1}{2^4}$	$\frac{5}{2^7}$	$\frac{11}{2^9}$	$\frac{1}{2^{11}}$	$\frac{1}{2^{12}}$	$\frac{1}{2^{13}}$

d. Secvenţa de stări este $X = S_1S_1S_1S_2S_2S_3S_3$ şi are probabilitatea a posteriori $P(X \mid O) = 1$ datorită probabilităţilor specifice de emisie ale acestui model Markov.

7. **(Algoritmul Forward: aplicare)**

CMU, 2008 spring, Eric Xing, HW4, pr. 3.1

Fie modelul Markov ascuns pentru cazinoul necinstit.[1081] Probabilitatea de a începe cu un zar corect / cinstit este 0.99. $P(1) = P(2) = P(3) = P(4) = P(5) = P(6) = 1/6$ pentru un zar cinstit, iar $P(1) = P(2) = P(3) = P(4) = P(5) = 1/10$ şi $P(6) = 1/2$ pentru un zar măsluit. Probabilitatea de rămânere în starea „cinstit“ este 0.95, iar cea de rămînere în starea „măsluit“ este 0.90. Se consideră că emisia unui simbol (oarecare) are loc la ieşirea dintr-o stare.

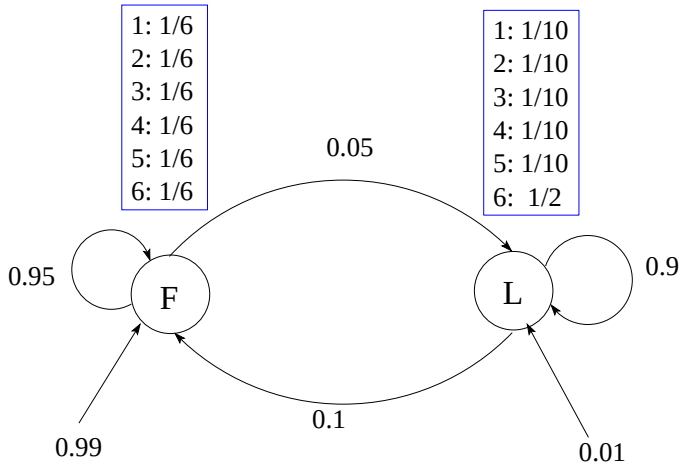
a. Calculaţi probabilitatea ca în patru aruncări de zar, primele trei să provină de la un zar cinstit, cu rezultatele 3, 1 şi 5, iar cea de-a patra aruncare de zar să provină de la un zar măsluit, cu rezultatul 6, aşadar $P(O_1 = 3, O_2 = 1, O_3 = 5, O_4 = 6, X_1 = F, X_2 = F, X_3 = F, X_4 = L)$.

b. Calculaţi $P(O_1 = 3, O_2 = 1, O_3 = 5)$ folosind algoritmul Forward.

Răspuns:

Modelul Markov este următorul (se folosesc notaţiile: F – Fair dice = zar cinstit, L – Loaded dice = zar măsluit):

[1081] A se vedea *Biological sequence analysis*, R. Durbin et al, 1998, pag. 54-55, 56-57.



a. Probabilitatea cerută este:

$$P(O_1 = 3, O_2 = 1, O_3 = 5, O_4 = 6, X_1 = F, X_2 = F, X_3 = F, X_4 = L) =$$
$$= P(X_1 = F) \cdot P(O_1 = 3 \mid X_1 = F, X_2 = F) \cdot P(X_2 = F \mid X_1 = F) \cdot$$
$$P(O_2 = 1 \mid X_2 = F, X_3 = F) \cdot P(X_3 = F \mid X_2 = F) \cdot P(O_3 = 5 \mid X_3 = F, X_4 = L) \cdot$$
$$P(X_4 = L \mid X_3 = F) \cdot P(O_4 = 6 \mid X_4 = L)$$
$$= \pi_F \cdot b_{FF3} \cdot a_{FF} \cdot b_{FF5} \cdot a_{FF} \cdot b_{FL1} \cdot a_{FL} \cdot b_{LL6}$$
$$= 0.99 \cdot \frac{1}{6} \cdot 0.95 \cdot \frac{1}{6} \cdot 0.95 \cdot \frac{1}{6} \cdot 0.05 \cdot \frac{1}{2} = 0.000103411$$

b. Probabilităţile Forward se calculează folosind un tabel corespunzător:

Output		3		1		5	
t	1		2		3		4
$\alpha_F(t)$	0.99		0.15685		0.024926083		0.003967936
$\alpha_L(t)$	0.01		0.00915		0.002130583		0.00039947
$P(O_1 \dots O_{t-1})$	1		0.166		0.027056666		0.004367406

Deci $P(O_1 = 3, O_2 = 1, O_3 = 5) = 0.004367406$**.**

8. (Algoritmul Backward; demonstraţie – pasul inductiv)

Liviu Ciortuz

a. Redăm mai jos demonstraţia formulei de recurenţă de la pasul inductiv al algoritmului Backward pentru HMM. Pentru fiecare egalitate din cadrul demonstraţiei, precizaţi motivul/motivele pentru care are loc egalitatea respectivă. Acolo unde este cazul, detaliaţi.

$$\begin{aligned}
\beta_i(t) &= P(O_t O_{t+1} \dots O_T \mid X_t = i, \mu) \\
&= \sum_{j=1}^{N} P(O_t O_{t+1} \dots O_T \mid X_t = i, X_{t+1} = j, \mu) P(X_{t+1} = j \mid X_t = i, \mu) \\
&= \sum_{j=1}^{N} P(O_{t+1} \dots O_T \mid O_t, X_t = i, X_{t+1} = j, \mu) P(O_t \mid X_t = i, X_{t+1} = j, \mu) a_{ij}
\end{aligned}$$

$$= \sum_{j=1}^{N} P(O_{t+1} \dots O_T \mid X_{t+1} = j, \mu) b_{ijO_t} a_{ij}$$
$$= \sum_{j=1}^{N} \beta_j(t+1) b_{ijO_t} a_{ij} \quad (445)$$

b. Vă reamintim că la pasul de iniţializare al algoritmului Backward se fixează $\beta_i(T+1) = 1$ pentru $i = 1, \dots, N$, unde N este numărul de stări din modelul Markov ascuns μ. Cum se poate calcula $P(O_1 O_2 \dots O_T \mid \mu)$, probabilitatea de emisie a unei secvenţe de semnale $O_1 O_2 \dots O_T$ de către modelul μ, folosind probabilităţile Backward?

Răspuns:

a. Demonstrăm relaţia (1): Din definiţie, $\beta_i(t)$ este probabilitatea de a emite secvenţa de simboluri de la O_t la O_T cu condiţia ca la momentul de timp t să fie în starea s_i.

Demonstrăm că (2)=(1): Pentru probabilităţile din această egalitate se aplică definiţia probabilităţii condiţionate, obţinându-se:

$$(2) = \frac{\sum_{j=1}^{N} P(O_t \dots O_T, X_t = i, X_{t+1} = j, \mu)}{P(X_t = i, X_{t+1} = j, \mu)} \cdot \frac{P(X_{t+1} = j, X_t = i, \mu)}{P(X_t = i, \mu)}$$

Simplificând această relaţie şi apoi renunţând la sumă (folosind formula probabilităţii totale), se obţine probabilitatea din relaţia (1):

$$(2) = \frac{P(O_t O_{t+1} \dots O_T, X_t = i \mid \mu)}{P(X_t = i \mid \mu)} = P(O_t O_{t+1} \dots O_T \mid X_t = i, \mu) = (1)$$

Demonstrăm egalitatea (3)=(2): Se aplică definiţia probabilităţii de tranziţie din starea i în starea j, $a_{ij} = P(X_{t+1} = j \mid X_t = i, \mu)$, apoi definiţia probabilităţilor condiţionate şi se simplifică:

$$(3) = \sum_{j=1}^{N} P(O_{t+1} \dots O_T \mid O_t, X_t = i, X_{t+1} = j, \mu) P(O_t \mid X_t = i, X_{t+1} = j, \mu) a_{ij}$$
$$= \sum_{j=1}^{N} \frac{P(O_t \dots O_T, X_t = i, X_{t+1} = j, \mu)}{P(O_t, X_t = i, X_{t+1} = j, \mu)} \cdot \frac{P(O_t, X_t = i, X_{t+1} = j, \mu)}{P(X_t = i, X_{t+1} = j, \mu)} \cdot$$
$$P(X_{t+1} = j \mid X_t = i, \mu)$$
$$= \sum_{j=1}^{N} \frac{P(O_t \dots O_T, X_t = i, X_{t+1} = j, \mu)}{P(X_t = i, X_{t+1} = j, \mu)} \cdot P(X_{t+1} = j \mid X_t = i, \mu)$$
$$= \sum_{j=1}^{N} P(O_t \dots O_T \mid X_t = i, X_{t+1} = j, \mu) P(X_{t+1} = j \mid X_t = i, \mu) = (2)$$

Demonstrăm egalitatea (4)=(3): b_{ijO_t} reprezintă probabilitatea de emitere a simbolului O_t la trecerea de la starea i la starea j, şi deci are formula $b_{ijO_t} = P(O_t \mid X_t = i, X_{t+1} = j, \mu)$. Probabilitatea $P(O_{t+1} \dots O_T \mid O_t, X_t = i, X_{t+1} = j, \mu)$

aflată într-o sumă după j, nu depinde nici de O_t, nici de $X_t = i$, deci este egală cu $P(O_{t+1} \dots O_T \mid X_{t+1} = j, \mu)$. Egalitatea este astfel evidentă.

Egalitatea (4)=(5) este o chestiune de notaţie, $\beta_j(t+1)$ fiind probabilitatea de a emite secvenţa $O_{t+1} \dots O_T$ cu condiţia ca la momentul $t+1$ să se afle în starea j.

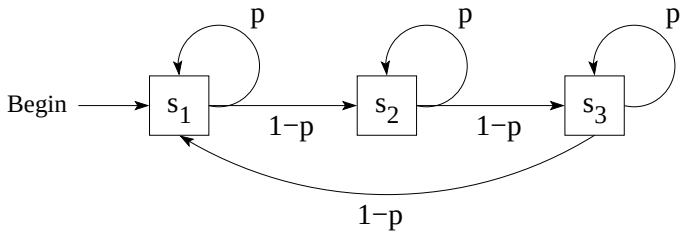
b. Probabilitatea de emisie a unei secvenţe de semnale $O = O_1 O_2 \dots O_T$ folosind probabilităţile Backward este:

$$P(O \mid \mu) = \sum_{i=1}^{N} \pi_i \beta_i(1).$$

9. (Algoritmii Forward, Backward, Viterbi: aplicare)

Marita Olsson, Univ. of Stockholm

Fie modelul Markov ascuns (HMM) de mai jos.



Fiecare stare/nod reprezintă o urnă care conţine un număr de bile, dintre care unele sunt roşii (R) iar celelalte sunt albastre (B). O secvenţă observabilă x este generată de acest HMM în modul următor:

– se alege la întâmplare o bilă din prima urnă, după care se notează culoarea ei (R sau B), apoi bila este repusă în urnă;

– procedura de mai sus se repetă de mai multe ori, cu singura diferenţă că bila se alege cu probabilitatea p din aceeaşi urnă ca la pasul precedent şi cu probabilitatea $1 - p$ din urna următoare.

În prima urnă sunt 3 bile roşii şi 2 bile albastre, a doua urnă conţine o bilă roşie şi 4 bile albastre, iar în a treia urnă sunt 4 bile roşii şi o bilă albastră. Se consideră $p = 1/3$.

a. Calculaţi $P(x = RBB)$.

În continuare, se consideră secvenţa $y = RRBBB$. Pentru a determina π^*, calea cea mai probabilă pentru generarea acestei secvenţe, s-a rulat algoritmul Viterbi şi s-au obţinut rezultatele de mai jos:

O		R		R		B		B		B	
t	1		2		3		4		5		6
$\delta_1(t)$	1		1/5		1/25		2/375		32/5625		128/84375
$\delta_2(t)$	0		2/5		2/25		8/375		32/5625		128/84375
$\delta_3(t)$	0		0		4/75		16/375		64/5625		256/84375
$\psi_1(t)$	1		1		1		1		3		3
$\psi_2(t)$	–		1		1		2		2		1 **sau** 2
$\psi_3(t)$	–		–		2		2		2		2

b. Arătaţi cum a fost calculat $\delta_2(3)$.

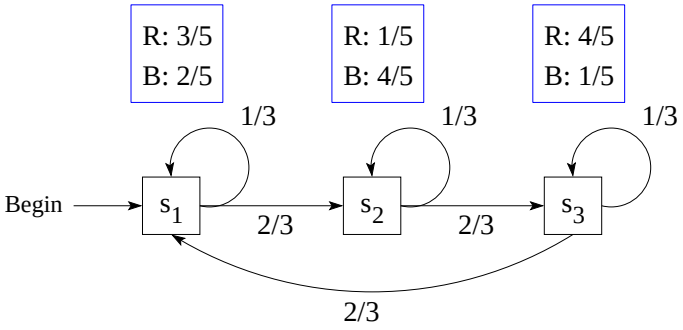
***Observaţie*: Am notat cu $\delta_t(i)$ probabilitatea calculată de către algoritmul Viterbi: cea mai mare dintre probabilităţile de generare a secvenţei de semnale $O_1 \dots O_t$, cu ajungere (la finalul generării secvenţei) în starea i.**

c. Determinaţi π^*.

d. Presupunând că se dă secvenţa $O = O_1 O_2 \dots O_{100}$, precizaţi cum anume se poate calcula probabilitatea ca O_{77} să fie generat în starea s_3.

Răspuns:

Modelul Markov ascuns (HMM) corespunzător datelor problemei este:



a. Probabilitatea cerută $P(x = RBB)$ se poate calcula în mai multe moduri:

***Metoda 1 (Forţă brută)*: Pentru producerea secvenţei de semnale RBB sunt posibile 4 drumuri: s_1, s_1, s_1; s_1, s_1, s_2; s_1, s_2, s_2; sau s_1, s_2, s_3. Se poate calcula probabilitatea pentru fiecare dintre acestea:**

$$p_1 = P(x = RBB, X = 111) = \frac{3}{5} \cdot \frac{1}{3} \cdot \frac{2}{5} \cdot \frac{1}{3} \cdot \frac{2}{5} = \frac{2^2}{5^3 \cdot 3}$$
$$p_2 = P(x = RBB, X = 112) = \frac{3}{5} \cdot \frac{1}{3} \cdot \frac{2}{5} \cdot \frac{2}{3} \cdot \frac{4}{5} = \frac{2^4}{5^3 \cdot 3}$$
$$p_3 = P(x = RBB, X = 122) = \frac{3}{5} \cdot \frac{2}{3} \cdot \frac{4}{5} \cdot \frac{1}{3} \cdot \frac{4}{5} = \frac{2^5}{5^3 \cdot 3}$$
$$p_4 = P(x = RBB, X = 123) = \frac{3}{5} \cdot \frac{2}{3} \cdot \frac{4}{5} \cdot \frac{2}{3} \cdot \frac{1}{5} = \frac{2^4}{5^3 \cdot 3}$$

Probabilitatea secvenţei de semnale RBB este suma acestor probabilităţi:

$$P(x = RBB) = \sum_{i=1}^{4} p_i = \frac{4 + 16 + 32 + 16}{375} = \frac{68}{375}$$

***Metoda 2 (Algoritmul Forward)*: Se calculează probabilităţile Forward $\alpha_i(t)$:**[1082]

[1082] $\alpha_i(t) \overset{def.}{=} P(O_1 \dots O_t, X_{t+1} = s_i)$, cu $\alpha_i(1) = \pi_i$, pentru $i = 1, 2, 3$, iar pentru $t \leq T$, $\alpha_i(t+1) = \sum_{j=1}^{N} \alpha_j(t) \cdot a_{ji} \cdot b_{ji O_t}$, unde N este numărul de stări, a_{ji} este probabilitatea de tranziţie din starea s_i în starea s_j, iar $b_{ji O_t}$ este probabilitatea de emisie a simbolului O_t la trecerea din starea s_i în starea s_j.

x		R		B		B	
t	1		2		3		4
$\alpha_1(t)$	1		1/5		2/75		36/1125
$\alpha_2(t)$	0		2/5		4/25		56/1125
$\alpha_3(t)$	0		0		16/75		112/1125
$P(x)$	1		3/5		2/5		68/375

$$\alpha_1(2) = 1 \cdot 1/3 \cdot 3/5 = 1/5$$
$$\alpha_2(2) = 1 \cdot 2/3 \cdot 3/5 = 2/5$$
$$\alpha_3(2) = 0$$
$$\alpha_1(3) = 1/5 \cdot 1/3 \cdot 2/5 = 2/75$$
$$\alpha_2(3) = 1/5 \cdot 2/3 \cdot 2/5 + 2/5 \cdot 1/3 \cdot 4/5 = 4/75 + 8/75 = 12/75 = 4/25$$
$$\alpha_3(3) = 2/5 \cdot 2/3 \cdot 4/5 = 16/75$$
$$\alpha_1(4) = 2/75 \cdot 1/3 \cdot 2/5 + 16/75 \cdot 2/3 \cdot 1/5 = \frac{36}{3^2 \cdot 5^3}$$
$$\alpha_2(4) = 2/75 \cdot 2/3 \cdot 2/5 + 4/25 \cdot 1/3 \cdot 4/5 = \frac{56}{3^2 \cdot 5^3}$$
$$\alpha_3(4) = 4/25 \cdot 2/3 \cdot 4/5 + 16/75 \cdot 1/3 \cdot 1/5 = \frac{112}{3^2 \cdot 5^3}$$

***Metoda 3 (Algoritmul Backward)*: Se calculează probabilităţile Backward $\beta_i(t)$:**[1083]

x		R		B		B	
t	1		2		3		4
$\beta_1(t)$	68/375		4/15		2/5		1
$\beta_2(t)$	34/1125		8/25		4/5		1
$\beta_3(t)$	4/25		1/15		1/5		1
$P(x)$	68/375						

$$\beta_1(3) = 1 \cdot 1/3 \cdot 2/5 \cdot 1 + 1 \cdot 2/3 \cdot 2/5 = 2/5$$
$$\beta_2(3) = 1 \cdot 1/3 \cdot 4/5 + 1 \cdot 2/3 \cdot 4/5 = 4/5$$
$$\beta_3(3) = 1 \cdot 1/3 \cdot 1/5 + 1 \cdot 2/3 \cdot 1/5 = 1/5$$
$$\beta_1(2) = 2/5 \cdot 1/3 \cdot 2/5 + 4/5 \cdot 2/3 \cdot 2/5 = 4/15$$
$$\beta_2(2) = 4/5 \cdot 1/3 \cdot 4/5 + 1/5 \cdot 2/3 \cdot 4/5 = 8/25$$
$$\beta_3(2) = 2/5 \cdot 2/3 \cdot 1/5 + 1/5 \cdot 1/3 \cdot 1/5 = 1/15$$
$$\beta_1(1) = 4/15 \cdot 1/3 \cdot 3/5 + 8/25 \cdot 2/3 \cdot 3/5 = \frac{4}{3 \cdot 5^2} + \frac{16}{5^3} = \frac{68}{3 \cdot 5^3}$$
$$\beta_2(1) = 8/25 \cdot 1/3 \cdot 1/5 + 1/15 \cdot 2/3 \cdot 1/5 = \frac{8}{3 \cdot 5^3} + \frac{2}{3^2 \cdot 5^2} = \frac{34}{3^2 \cdot 5^3}$$
$$\beta_3(1) = 4/15 \cdot 2/3 \cdot 4/5 + 1/15 \cdot 1/3 \cdot 4/5 = \frac{32}{3^2 \cdot 5^2} + \frac{4}{3^2 \cdot 5^2} = \frac{4}{5^2}$$

[1083] $\beta_i(t) \stackrel{def.}{=} P(O_t O_{t+1} \dots O_T | X_t = s_i)$, cu $\beta_i(T+1) = 1$, pentru $i = 1, 2, 3$, iar pentru $t \leq T$, conform problemei 8, $\beta_i(t) = \sum_{j=1}^{N} \beta_j(t+1) \cdot a_{ji} \cdot b_{ji O_t}$.

b. $\delta_2(3)$ **se calculează după formula:**

$$\delta_2(3) = \max\left(\delta_1(2)\cdot a_{12}\cdot b_1(R); \delta_2(2)\cdot a_{22}\cdot b_2(R); \delta_3(2)\cdot a_{32}\cdot b_3(R)\right)$$

$$\Rightarrow \delta_2(3) = \max\left(\frac{1}{5}\cdot\frac{2}{3}\cdot\frac{3}{5}; \frac{2}{5}\cdot\frac{1}{3}\cdot\frac{1}{5}; 0\right) \Rightarrow \delta_2(3) = \max\left(\frac{2}{25}; \frac{2}{75}\right) \Rightarrow \delta_2(3) = \frac{2}{25}$$

c. Pentru a determina π^* **folosind algoritmul Viterbi se procedează astfel: Se alege maximul de pe ultima coloană (din primele 3 rânduri - adică dintre** $\delta_i(T)$**). Acesta este obţinut pe linia 3, deci** $\delta_3(6)$**. Aşadar, ultima stare a drumului va fi** s_3**. În continuare se trece în tabel la** $\psi_3(6)$**, unde este valoarea 2. Această valoare indică penultima stare ca fiind** s_2 **şi de asemenea linia pe care se trece în continuare pentru a reconstitui calea cea mai probabilă de generare a secvenţei respective. Se trece pe linia 2; de aici se continuă tot pe linia 2 de două ori, apoi se trece la linia 1. Se obţine în final drumul:** $s_1s_1s_2s_2s_2s_3$**.**

O		R		R		B		B		B	
t	1		2		3		4		5		6
δ_1	1		1/5		1/25		2/375		32/5625		128/84375
δ_2	0		2/5		2/25		8/375		32/5625		128/84375
δ_3	0		0		4/75		16/375		64/5625		256/84375
Ψ_1	1		1		1		1		3		3
Ψ_2	–		1		1		2		2		1 sau 2
Ψ_3	–		–		2		2		2		2

d. Ştim că $P(O, X_t = s_i) = \alpha_i(t)\cdot\beta_i(t)$**, unde** $\alpha_i(t)$ **şi** $\beta_i(t)$ **sunt probabilităţile Forward şi respectiv Backward. Aşadar:**

$$P(X_{77} = s_3 \mid O) = \frac{P(X_{77} = s_3, O_1 \ldots O_{100})}{P(O \mid \mu)} = \frac{\alpha_3(77)\cdot\beta_3(77)}{\sum_{i=1}^{3}\alpha_i(101)}$$

10. **(Demonstraţia formulei necesare pentru calculul mediilor variabilelor neobservabile corespunzătoare tranziţiilor în algoritmul Forward-Backward)**

Liviu Ciortuz

Fie un model Markov ascuns având probabilităţile de tranziţie $a_{ij} = P(X_{t+1} = j \mid X_t = i, \mu)$ **şi probabilităţile de emisie** $b_{ijO_t} = P(O_t \mid X_{t+1} = j, X_t = i, \mu)$**.**

Demonstraţi formula:

$$P(X_t = i, X_{t+1} = j \mid O, \mu) = \frac{\alpha_i(t)a_{ij}b_{ijO_t}\beta_j(t+1)}{P(O \mid \mu)}$$

unde $\alpha_i(t) \overset{not.}{=} P(O_1O_2\cdots O_{t-1}, X_t = i \mid \mu)$ **şi** $\beta_i(t) \overset{not.}{=} P(O_tO_{t+1}\cdots O_T \mid X_t = i, \mu)$**.**

Răspuns:

Se porneşte de la membrul stâng al relaţiei şi se aplică formula probabilităţii condiţionate de mai multe ori (fie în sens direct fie în sens invers):

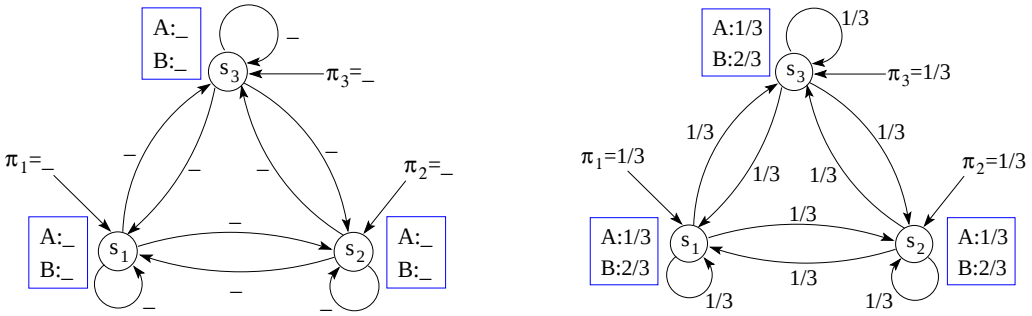
$$
\begin{aligned}
& P(X_t = i, X_{t+1} = j \mid O, \mu) = \frac{P(X_t = i, X_{t+1} = j, O \mid \mu)}{P(O \mid \mu)} \\
= \;& \frac{1}{P(O,\mu)} \cdot P(O_1 O_2 \ldots O_{t-1} O_t O_{t+1} \ldots O_T, X_t = i, X_{t+1} = j) \\
= \;& \frac{1}{P(O,\mu)} P(O_{t+1} \ldots O_T \mid X_t = i, X_{t+1} = j, O_1 \ldots O_t) P(X_t = i, X_{t+1} = j, O_1 \ldots O_t) \\
= \;& \frac{1}{P(O,\mu)} P(O_{t+1} \ldots O_T \mid X_{t+1} = j) P(X_t = i, X_{t+1} = j, O_1 \ldots O_t) \\
= \;& \frac{1}{P(O,\mu)} \cdot \beta_j(t+1) \cdot P(X_t = i, X_{t+1} = j, O_1 \ldots O_t) \\
= \;& \frac{1}{P(O,\mu)} \cdot \beta_j(t+1) \cdot P(O_1 \ldots O_{t-1} O_t, X_t = i, X_{t+1} = j) \\
= \;& \frac{1}{P(O,\mu)} \cdot \beta_j(t+1) P(O_1 \ldots O_{t-1} \mid O_t, X_t = i, X_{t+1} = j) P(O_t, X_t = i, X_{t+1} = j) \\
= \;& \frac{1}{P(O,\mu)} \cdot \beta_j(t+1) \cdot P(O_1 \ldots O_{t-1} \mid X_t = i) \cdot P(O_t, X_t = i, X_{t+1} = j) \\
= \;& \frac{1}{P(O,\mu)} \cdot \beta_j(t+1) \cdot \frac{P(O_1 \ldots O_{t-1}, X_t = i)}{P(X_t = i)} \cdot P(O_t, X_t = i, X_{t+1} = j) \\
= \;& \frac{1}{P(O,\mu)} \cdot \beta_j(t+1) \cdot \alpha_i(t) \cdot \frac{P(O_t, X_t = i, X_{t+1} = j)}{P(X_t = i)} \\
= \;& \frac{1}{P(O,\mu)} \cdot \beta_j(t+1) \cdot \alpha_i(t) \cdot \frac{P(O_t \mid X_t = i, X_{t+1} = j) \cdot P(X_t = i, X_{t+1} = j)}{P(X_t = i)} \\
= \;& \frac{1}{P(O,\mu)} \cdot \beta_j(t+1) \cdot \alpha_i(t) \cdot b_{ijO_t} \cdot \frac{P(X_t = i, X_{t+1} = j)}{P(X_t = i)} \\
= \;& \frac{1}{P(O,\mu)} \cdot \beta_j(t+1) \cdot \alpha_i(t) \cdot b_{ijO_t} \cdot P(X_{t+1} = j \mid X_t = i) \\
= \;& \frac{1}{P(O,\mu)} \cdot \beta_j(t+1) \cdot \alpha_i(t) \cdot b_{ijO_t} \cdot a_{ij} = \frac{\alpha_i(t) a_{ij} b_{ijO_t} \beta_j(t+1)}{P(O \mid \mu)}
\end{aligned}
$$

11. **(HMM: noţiuni de bază; algoritmul Forward-Backward)**

CMU, 2003 fall, T. Mitchell, A. Moore, final, pr. 10.cd

a. Se consideră modelul Markov ascuns de mai jos, în partea stângă, cu trei stări s_1, s_2 şi s_3. Probabilităţile de intrare în fiecare dintre stări sunt π_1, π_2, respectiv π_3.

Stabiliţi un set de probabilităţi astfel încât modelul Markov rezultat să maximizeze probabilitatea obţinerii secvenţei ABA.



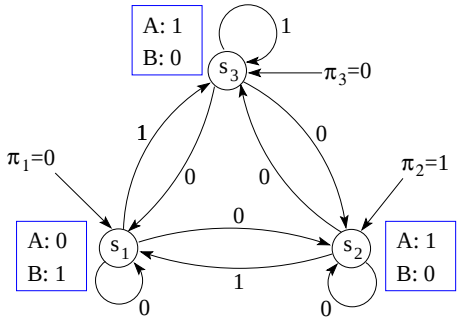
b. Se va folosi algoritmul EM (cunoscut pentru HMM şi sub numele Forward-Backward sau Baum-Welch[1084]) pentru estimarea parametrilor modelului Markov ascuns. Înaintea primei iteraţii s-au realizat iniţializările prezentate în figura de mai sus din partea dreaptă. Pentru aceste valori iniţiale, va converge cu succes algoritmul EM la un model care maximizează probabilitatea secvenţei ABA?

Răspuns:

a. Pentru ca modelul Markov ascuns obţinut să fie valid, trebuie ca parametrii acestuia, adică probabilităţile, să respecte următoarele ecuaţii:

$$\begin{aligned} \pi_1 + \pi_2 + \pi_3 &= 1 \\ a_{i1} + a_{i2} + a_{i3} &= 1,\ \forall i \in \{1,2,3\} \\ b_i(A) + b_i(B) &= 1,\ \forall i \in \{1,2,3\} \end{aligned}$$

Sunt mai multe modele Markov ascunse pentru care probabilitatea obţinerii secvenţei ABA să fie 1, adică maximă. Unul dintre acestea este următorul:



Într-adevăr, în acest model, secvenţa ABA poate fi obţinută pe următoarea cale: $q_1 = s_2$, $q_2 = s_1$, $q_3 = s_3$.

[1084] Fizicianul Ruslan Stratonovich este cel dintâi care are prezentat algoritmul Forward-Backward, în anul 1960, în articolul *Conditional Markov Processes* din revista *Theory of Probability and Its Applications*, 5(2):156–178, în limba rusă. Procesele Markov poartă numele matematicianului rus Andrei Markov (1856-1922). Andrei Markov şi fratele său Vladimir Markov (1871-1896) sunt cei care au demonstrat inegalitatea probabilistă cunoscută sub numele de inegalitatea [fraţilor] Markov; vedeţi pr. 21.a de la capitolul de *Fundamente*. Vladimir Markov a murit de tuberculoză la vârsta de doar 25 de ani.

Leonard Baum (1931 – 2017) şi Lloyd Welch (1927-) au descris algoritmul Forward-Backward (şi formalismul teoretic al modelelor Markov ascunse) în limba engleză, în cadrul unor cercetări întreprinse la sfâşitul anilor '60 şi începutul anilor '70 din secolul trecut, la *Institute for Defence Analyses* din S.U.A. (Cf. Lawrence Rabiner, *Fist-Hand: The Hidden Markov Model*, https://ethw.org/First-Hand:The_Hidden_Markov_Model.)

b. Algoritmul EM (Forward-Backward) constă în repetarea iterativă a următorilor doi paşi:

***Estimare*: Folosind probabilităţi Forward şi Backward, $\alpha_i(t)$ şi respectiv $\beta_i(t)$, se calculează $p_t(i,j) = \dfrac{\alpha_i(t)\, a_{ij}\, b_{ij}(O_t)\, \beta_j(t+1)}{\sum_{m=1}^{N} \alpha_m(t)\, \beta_m(t)}$ pentru $t \in \{1,2,3\}$ şi $i,j \in \{1,2,3\}$.**[1085]

***Maximizare*: Se recalculează probabilităţile iniţiale ale modelului**

$$\hat{\pi}_i = \sum_{j=1}^{N} p_1(i,j) \qquad \hat{a}_{ij} = \frac{\sum_{t=1}^{T} p_t(i,j)}{\sum_{l=1}^{N}\sum_{t=1}^{T} p_t(i,l)} \qquad \hat{b}_{ijk} = \frac{\sum_{t:O_t=k,1\le t\le T} p_t(i,j)}{\sum_{t=1}^{T} p_t(i,j)}$$

Vom aplica prima iteraţie a algoritmului EM. Pentru aceasta trebuie calculate probabilităţile Forward şi Backward. Datorită simetriei probabilităţilor celor trei stări s_1, s_2 şi s_3 în modelul considerat, vom avea $\alpha_1(t) = \alpha_2(t) = \alpha_3(t)$ şi $\beta_1(t) = \beta_2(t) = \beta_3(t)$, $\forall t$ fixat. Aceste probabilităţi sunt:

Output		*A*		*B*		*A*	
t	1		2		3		4
$\alpha_i(t),\ i \in \{1,2,3\}$	$1/3$		$1/3^2$		$2/3^3$		$2/3^4$
$\beta_i(t),\ i \in \{1,2,3\}$	$2/3^3$		$2/3^2$		$1/3$		1

Se observă că probabilitatea de a obţine secvenţa ABA este foarte mică:

$$P(ABA) = \sum_i \alpha_i(4) = 3 \cdot \frac{2}{3^4} = \frac{2}{3^3} \approx 0.074$$

Pasul de estimare:

$$p_1(i,j) = \frac{\alpha_i(1)a_{ij}b_{ij}(A)\beta_j(2)}{\sum_{m=1}^{3}\alpha_m(1)\beta_m(1)} = \frac{\frac{1}{3}\cdot\frac{1}{3}\cdot\frac{1}{3}\cdot\frac{2}{3^2}}{3\left(\frac{1}{3}\cdot\frac{2}{3^3}\right)} = \frac{1}{3^2}$$

$$p_2(i,j) = \frac{\alpha_i(2)a_{ij}b_{ij}(B)\beta_j(3)}{\sum_{m=1}^{3}\alpha_m(2)\beta_m(2)} = \frac{\frac{1}{3^2}\cdot\frac{1}{3}\cdot\frac{2}{3}\cdot\frac{1}{3}}{3\left(\frac{1}{3^2}\cdot\frac{2}{3^2}\right)} = \frac{1}{3^2}$$

$$p_3(i,j) = \frac{\alpha_i(3)a_{ij}b_{ij}(A)\beta_j(4)}{\sum_{m=1}^{3}\alpha_m(3)\beta_m(3)} = \frac{\frac{2}{3^3}\cdot\frac{1}{3}\cdot\frac{1}{3}\cdot 1}{3\left(\frac{2}{3^3}\cdot\frac{1}{3}\right)} = \frac{1}{3^2}$$

Pasul de maximizare:

$$\hat{\pi}_i = \sum_{j=1}^{3} p_1(i,j) = 3\cdot\frac{1}{3^2} = \frac{1}{3} \text{ pentru } i \in \{1,2,3\}$$

[1085] Numitorul $\sum_{m=1}^{N}\alpha_m(t)\,\beta_m(t)$ este probabilitatea generării secvenţei de semnale $O_1 \ldots O_T$. (Formula aceasta este una dintre posibilităţile de calculare a acestei probabilităţi.) Numărătorul $\alpha_i(t)\, a_{ij}\, b_{ij}(O_t)\beta_j(t+1)$ este probabilitatea ca la momentul t să se treacă din starea i în starea j, ştiind că a fost emisă secvenţa $O_1 \ldots O_T$. Raportul acestor două probabilităţi, adică $p_t(i,j)$, poate fi interpretat ca fiind media unei variabile (aleatoare) indicator neobservabile, $X_t(i,j)$, care ia valoarea 1 dacă la momentul t se face trecerea din starea i în starea j (şi se emite semnalul O_t), şi 0 în caz contrar.

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{3} p_t(i,j)}{\sum_{l=1}^{3}\sum_{t=1}^{3} p_t(i,l)} = \frac{3 \cdot \frac{1}{3^2}}{3 \cdot 3 \cdot \frac{1}{3^2}} = \frac{1}{3} \textbf{ pentru } i,j \in \{1,2,3\}$$

$$\hat{b}_{ij}(A) = \frac{\sum_{t\in\{1,3\}} p_t(i,j)}{\sum_{t=1}^{3} p_t(i,j)} = \frac{2 \cdot \frac{1}{3^2}}{3 \cdot \frac{1}{3^2}} = \frac{2}{3} \textbf{ pentru } i,j \in \{1,2,3\}$$

$$\hat{b}_{ij}(B) = \frac{\sum_{t\in\{2\}} p_t(i,j)}{\sum_{t=1}^{3} p_t(i,j)} = \frac{1 \cdot \frac{1}{3^2}}{3 \cdot \frac{1}{3^2}} = \frac{1}{3} \textbf{ pentru } i,j \in \{1,2,3\}$$

Se observă că nu se modifică probabilităţile de intrare π_i si nici probabilităţile de tranziţie a_{ij}. De asemenea, deşi se modifică probabilităţile de emisie, acestea îşi păstrează simetria: $b(A)$, şi respectiv, $b(B)$ sunt egale pentru toate cele 3 stări. (De fapt, s-au inversat valorile precedente pentru $b(A)$ şi $b(B)$.)

Probabilităţile Forward şi Backward recalculate sunt:

Output		A		B		A	
t	1		2		3		4
$\alpha_i(t),\ i \in \{1,2,3\}$	$1/3$		$2/3^2$		$2/3^3$		$4/3^4$
$\beta_i(t),\ i \in \{1,2,3\}$	$4/3^3$		$2/3^2$		$2/3$		1

iar probabilitatea de a obţine secvenţa ABA s-a dublat:

$$P(ABA) = \sum_i \alpha_i(4) = 3 \cdot \frac{4}{3^4} = \frac{4}{3^3} \approx 0.148$$

Se poate constata uşor (prin calcule) că orice iteraţie ulterioară a algoritmului EM nu va aduce nicio modificare, toate probabilităţile rămânând neschimbate.

În concluzie, pentru valorile iniţiale date în enunţ, algoritmul EM nu converge la un model care maximizează probabilitatea secvenţei ABA. (Vă reamintim că la punctul a am demonstrat că această probabilitate maximă este 1.)

12. **(O proprietate a algoritmului Forward-Backward)**

Christopher Manning, Hinrich Schütze,
Foundations of statistical natural language processing, 1999

Demonstraţi că dacă într-un HMM o emisie/tranziţie are probabilitatea 0, atunci algoritmul Forward-Backward o lasă neschimbată.

Răspuns:

Algoritmul Forward-Backward are ca date de intrare probabilităţile modelului Markov: π_i, a_{ij} şi b_{ijk}. Folosind probabilităţi Forward şi Backward, $\alpha_i(t)$ şi respectiv $\beta_i(t)$, algoritmul recalculează probabilităţile iniţiale, obţinând $\hat{\pi}_i$, $\hat{a}_{ij}$ şi $\hat{b}_{ijk}$.

Arătăm că dacă $a_{ij} = 0$ atunci $\hat{a}_{ij} = 0$, iar dacă $b_{ijk} = 0$ atunci $\hat{b}_{ijk} = 0$.

Algoritmul Forward-Backward constă din mai multe iteraţii, fiecare iteraţie constând în doi paşi:

- ***Estimare*: Se calculează** $p_t(i,j) = \dfrac{\alpha_i(t)a_{ij}b_{ijk}\beta_j(t+1)}{\sum_{m=1}^{N}\alpha_m(t)\beta_m(t)}$.

  Dacă oricare dintre a_{ij} **sau** b_{ijk} **(unde** $O_t = k$**) este** 0**, rezultă că** $p_t(i,j) = 0$**, indiferent de valoarea lui** t**.**

- ***Maximizare*:**

  $\hat{a}_{ij} = \dfrac{\sum_{t=1}^{T} p_t(i,j)}{\sum_{l=1}^{N}\sum_{t=1}^{T} p_t(i,l)}$

  Dacă $a_{ij} = 0$**, atunci** $p_t(i,j) = 0$**, deci** $\hat{a}_{ij} = 0$**.**

  $\hat{b}_{ijk} = \dfrac{\sum_{t:O_t=k,1\le t\le T} p_t(i,j)}{\sum_{t=1}^{T} p_t(i,j)}$

  Dacă $b_{ijk} = 0$**, atunci** $p_t(i,j) = 0$ **pentru orice** t **pentru care** $O_t = k$**, deci** $\hat{b}_{ijk} = 0$**.**

Am demonstrat astfel că algoritmul Forward-Backward lasă aceste probabilităţi neschimbate, dacă ele sunt iniţial nule.

13. **(Un model Markov ascuns cu emisii gaussiene)**

□ • ○ ***CMU, 2008 fall, Eric Xing, final, pr. 8***

Fie un model Markov ascuns cu emisii continue, având K **stări. Notăm cu** x_i **emisia de la momentul** i**, iar cu** z_i **starea ascunsă corespunzătoare. Emisiile stării** $k \in \{1,\ldots,K\}$ **au o distribuţie gaussiană de medie** μ_k **şi deviaţie standard** σ_k**.**[1086] **Prin urmare, probabilităţile de emisie sunt** $P(x_i \mid z_i = k, \theta) = \mathcal{N}(x_i \mid \mu_k, \sigma_k)$**.** θ **reprezintă mulţimea parametrilor modelului, adică probabilităţile iniţiale** π**, matricea probabilităţilor de tranziţie** a**, mediile** $\mu_1,\ldots,\mu_K$ **şi deviaţiile standard** $\sigma_1,\ldots,\sigma_K$**.**

a. Calculaţi log-verosimilitatea secvenţei de emisii $(x_1,\ldots,x_n)$ **dacă succesiunea de stări în care se produc aceste emisii este** $(z_1,\ldots,z_n)$**.**

b. Calculaţi formulele de update pentru probabilităţile Forward şi Backward pentru acest HMM şi explicaţi pe scurt diferenţa faţă de cele studiate în cazul discret.

c. Presupunem că ni se dau secvenţa de observaţii $X = (x_1,\ldots,x_n)$ **şi secvenţa de stări corespunzătoare,** $Z = (z_1,\ldots,z_n)$**. Vrem să găsim valorile parametrilor** θ **pentru acest model Markov ascuns, folosind un algoritm de tipul EM / Forward-Bacxkward.**

Sunt oare relaţiile de actualizare pentru probabilităţile a_{ij} **şi** π_i **diferite de cele ale modelului Markov ascuns cu emisii discrete? Explicaţi de ce da, sau de ce nu.**

Care vor fi relaţiile de actualizare pentru parametrii μ_k **şi** σ_k **ai distribuţiilor gaussiene care modelează emisiile?** ***Sugestie*: Nu este necesar să derivaţi aceste relaţii. Date fiind stările, emisiile sunt independente unele de altele, iar fiecare emisie urmează distribuţia gaussiană asociată stării în care se produce emisia respectivă.**

[1086] Se va considera că aceste emisii depind doar de starea k, nu şi de stările în care se poate trece pornind din starea k.

Pentru punctele d şi e de mai jos, vom presupune că ni se dă doar secvenţa de observaţii $X = (x_1, \ldots, x_n)$. Vrem să găsim valorile parametrilor θ pentru acest model Markov ascuns.

d. Algoritmul de învăţare nesupervizată EM / Forward-Backward optimizează la fiecare iteraţie media log-verosimilităţii datelor complete. Oare de ce este aceasta o funcţie obiectiv rezonabilă?

Scrieţi expresia acestei funcţii (notată cu $E[\ell_c(x, y; \theta)]$) pentru modelul Markov ascuns cu emisii gaussiene. Nu este necesar să demonstraţi cum anume se obţine această expresie.

e. Presupunem că vrem să găsim estimările de verosimilitate maximă ($\hat{\mu}_k$ şi $\hat{\sigma}_k$) pentru parametrii μ_k şi σ_k. Vor avea oare formulele [pentru obţinerea] acestor estimări exact aceeaşi formă ca şi în cazul mixturilor de distribuţii gaussiene? Justificaţi pe scurt. *Sugestie*: Identificaţi acei termeni din expresia funcţiei $E[\ell_c(x, y; \theta)]$ care sunt relevanţi pentru optimizare (adică, termenii care conţin parametrii μ_k şi / sau σ_k).

Răspuns:

a. Log-verosimilitatea condiţională cerută este:

$$\ln P(x_1, \ldots, x_n \mid z_1, \ldots, z_n) = \ln \prod_{i=1}^{n} P(x_i \mid z_i) = \ln \prod_{i=1}^{n} \mathcal{N}(x_i \mid \mu_{z_i}, \sigma_{z_i})$$
$$= \sum_{i=1}^{n} \ln \mathcal{N}(x_i \mid \mu_{z_i}, \sigma_{z_i}) = \sum_{i=1}^{n} \Big[-\ln(\sqrt{2\pi}\sigma_{z_i}) - \frac{(x_i - \mu_{z_i})^2}{2\sigma_{z_i}^2} \Big]$$

b. Probabilităţile Forward se calculează astfel:

$$\begin{aligned}
\alpha_k(t+1) &\overset{def.}{=} P(x_1, \ldots, x_t, z_{t+1} = k) \\
&= \sum_i P(x_1, \ldots, x_t, z_{t+1} = k, z_t = i) \\
&= \sum_i P(x_1, \ldots, x_{t-1}, z_t = i) \cdot P(x_t, z_{t+1} = k \mid x_1, x_2, \ldots, x_{t-1}, z_t = i) \\
&= \sum_i \alpha_i(t) \cdot P(x_t, z_{t+1} = k \mid z_t = i) \\
&= \sum_i \alpha_i(t) \cdot P(x_t \mid z_{t+1} = k, z_t = i) \cdot P(z_{t+1} = k \mid z_t = i) \\
&= \sum_i \alpha_i(t) \cdot \mathcal{N}(x_t \mid \mu_k, \sigma_k) \cdot a_{ik} = \mathcal{N}(x_t \mid \mu_k, \sigma_k) \sum_i \alpha_i(t) \cdot a_{ik}
\end{aligned}$$

Similar, probabilităţile Backward se calculează astfel:

$$\begin{aligned}
\beta_k(t) &\overset{def.}{=} P(x_t, \ldots, x_T \mid z_t = k) \\
&= \sum_i P(x_t, \ldots, x_T, z_{t+1} = i \mid z_t = k) \\
&= \sum_i P(x_t, \ldots, x_T \mid z_{t+1} = i, z_t = k) \cdot \underbrace{P(z_{t+1} = i \mid z_t = k)}_{= a_{ki}} \\
&= \sum_i P(x_{t+1}, \ldots, x_T \mid x_t, z_{t+1} = i, z_t = k) \cdot P(x_t \mid z_{t+1} = i, z_t = k) \cdot a_{ki}
\end{aligned}$$

$$= \sum_i P(x_{t+1}, \ldots, x_T \mid z_{t+1} = i) \cdot \mathcal{N}(x_t \mid \mu_i, \sigma_i) \cdot a_{ki}$$
$$= \sum_i \beta_i(t+1) \cdot \mathcal{N}(x_t \mid \mu_i, \sigma_i) \cdot a_{ki}$$

Aşadar, formulele de recurenţă pentru un model Markov ascuns cu emisii continue au o formă similară cu cele din cazul discret,[1087] **singura diferenţă constând în probabilităţile de emisie, care în cazul continuu sunt modelate de distribuţii probabiliste gaussiene.**

c. Relaţiile de actualizare pentru probabilităţile a_{ij} şi π_i sunt aceleaşi cu cele din cazul emisiilor discrete, fiindcă ele implică doar numărarea tranziţiilor dintre stări şi sunt independente de forma aleasă pentru probabilităţile de emisie.

Relaţiile de actualizare pentru parametrii μ_k şi σ_k ai distribuţiilor gaussiene care modelează emisiile sunt următoarele:

$$\mu_k = \frac{\sum_i 1_{\{z_i=k\}} x_i}{\sum_i 1_{\{z_i=k\}}}$$
$$\sigma_k = \frac{\sum_i 1_{\{z_i=k\}} (x_i - \mu_k)^2}{\sum_i 1_{\{z_i=k\}}}$$

pentru $k = 1, \ldots, K$.[1088] **(Vă reamintim că notaţia $1_{\{\cdot\}}$ desemnează funcţia indicator.)**

d. Media log-verosimilităţii datelor complete (X şi Z) este o margine inferioară pentru funcţia de log-verosimilitate a datelor observabile.[1089] **Algoritmul EM / Forward-Backward converge la un optim local al funcţiei de log-verosimilitate a datelor observabile, aşadar este rezonabil să folosim ca funcţie obiectiv [la fiecare iteraţie a acestui algoritm] media log-verosimilităţii datelor complete.**

$$E[\ell_c(\theta; x, y)] = \sum_n E[y^i_{n,1}]\pi_i + \sum_n \sum_{t=2}^{T} E[y^i_{n,t-1} y^j_{n,t}] \ln a_{ij} + \sum_n \sum_{t=1}^{T} E[y^i_{n,t}] \ln \mathcal{N}(x_n | \mu_i, \sigma_i),$$

unde n indexează stările modelului Markov ascuns,[1090] **iar $y^i_{n,t}$ este o variabilă aleatoare care ia valoarea 1 dacă la momentul de timp t modelul se află în starea cu indicele i şi respectiv valoarea 0 în cazul contrar.**[1091]

e. Da, formulele pentru estimările de verosimilitate maximă pentru parametrii μ_k şi σ_k au aceeaşi formă în ambele cazuri. În expresia funcţiei $E[\ell_c(\theta; x, y)]$ termenul relevant este (doar) ultimul termen, care seamănă foarte mult cu cel din expresia mediei log-verosimilităţii datelor complete de la mixtura de distribuţii gaussiene.[1092]

[1087] Pentru formulele de calcul ale probabilităţilor Forward şi respectiv Backward în cazul emisiilor discrete, vedeţi problemele 18.a şi 8.a, relaţiile (446) şi (445).
[1088] Vedeţi problemele 50.a şi respectiv 51.a de la capitolul de *Fundamente*.
[1089] Vedeţi problema 1 de la capitolul *Schema algoritmică EM*.
[1090] Nu are nicio legătură cu indicele n din notaţia secvenţei $X = (x_1, \ldots, x_n)$.
[1091] Pentru fiecare valoare fixată pentru $t \in \{1, \ldots, T\}$, numai una dintre variabilele $y^i_{n,t}$ ia valoare diferită de 0.
[1092] Vedeţi problema 18 de la capitolul *Clusterizare*.

9.2 Modele Markov ascunse — Probleme propuse

14. **(Modelul Markov – model probabilistic total)**

prelucrare de Liviu Ciortuz, după
Problems and solutions in biological sequence analysis,
M. Borodvsky, S. Ekisheva, 2006, pr. 3.1-3.3

Fie μ un model Markov de ordinul întâi.

a. Suma probabilităţilor tuturor secvenţelor de stări de lungime L se poate scrie ca

$$\sum_x P(x) = \sum_{x_1} \sum_{x_2} \ldots \sum_{x_L} P(x_1) \Pi_{i=2}^{L} a_{x_{i-1}\, x_i}$$

Arătaţi că această sumă este 1.

b. Presupunem că acest model are o stare finală 'silent' notată *End*, şi că tranziţiile din orice stare în starea *End* au aceeaşi probabilitate, τ. Arătaţi că suma probabilităţilor tuturor secvenţelor de lungime L (şi care se termină prin tranziţie în starea *End*) este $\tau(1-\tau)^{L-1}$.

c. Arătaţi că suma probabilităţilor tuturor secvenţelor posibile (de orice lungime) este 1. Aceasta dovedeşte că un model/lanţ Markov este într-adevăr o distribuţie de probabilitate peste spaţiul tuturor secvenţelor definite peste alfabetul respectiv.

15. **(Verificarea înţelegerii unor noţiuni de bază)**

∗ ***CMU, 2003 fall, T. Mitchell, A. Moore, HW7, pr. 2.a***

Fie un model Markov ascuns cu două simboluri observate: X şi Y. Se consideră următoarea probabilitate (cu notaţiile tradiţionale pentru HMM):

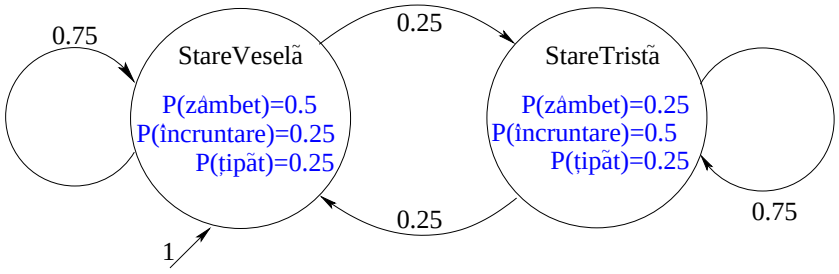
$$P(q_{t+2} = s_i \wedge q_{t+3} = s_j \wedge O_{t+3} = X \mid q_t = s_k \wedge q_{t+1} = s_m \wedge O_t = X)$$

Calculaţi valoarea acestei probabilităţi, în funcţie de elementele corespunzătoare din matricea de tranziţie A şi din matricea de emisie B.

16. **(Verificarea înţelegerii unor noţiuni de bază)**

∗ ***CMU, 2003 fall, T. Mitchell, A. Moore, HW7, pr. 2.b***

Vulpea Foxy duce o viaţă simplă. În unele zile e veselă iar în alte zile e tristă. Dar îşi ascunde starea emoţională, şi poţi să observi doar dacă zâmbeşte, se încruntă sau ţipă. Începem în prima zi cu starea veselă şi continuăm cu o tranziţie în fiecare zi. Vom reprezenta evoluţia stărilor emoţionale a lui Foxy cu ajutorul unui model Markov ascuns, şi anume:



Dacă notăm prin q_t starea din ziua t, iar prin O_t observaţia din ziua t, realizaţi următoarele cerinţe:

a. Calculaţi $P(q_2 = \textit{Tristă})$.

b. Calculaţi $P(O_2 = \textit{zâmbet})$.

c. Calculaţi $P(q_2 = \textit{Tristă} \mid O_2 = \textit{zâmbet})$.

d. Cât este $P(O_2 = \textit{ţipăt})$.

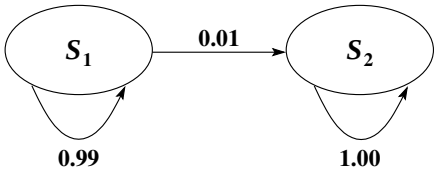
e. Notăm $\phi_t = P(q_t = \textit{Veselă})$. Este evident că $\phi_1 = 1$. Se poate da o definiţie inductivă a lui ϕ_{t+1} în funcţie de ϕ_t printr-o expresie de forma: $\phi_{t+1} = X + Y \cdot \phi_t$. Calculaţi valorile numerice ale lui X şi Y.

f. Se consideră $O_1 = \textit{încruntare}$, $O_2 = \textit{încruntare}$, $O_3 = \textit{încruntare}$, $O_4 = \textit{încruntare}$ şi $O_5 = \textit{încruntare}$. Care este cea mai probabilă secvenţă de stări? (Nu este necesar să aplicaţi algoritmul Viterbi.)

17. **(Verificarea înţelegerii unor noţiuni de bază)**

CMU, 2010 fall, Aarti Singh, HW4, pr. 5

Figura de mai jos reprezintă un model Markov ascuns cu două stări. Probabilităţile de tranziţie sunt înscrise pe diagramă. Probabilităţile de emisie corespunzătoare fiecărei stări sunt definite pe alfabetul de simboli $\{1, 2, 3, 4\}$ şi apar în the tabelul de lângă diagramă. Se consideră că probabilităţile de intrare sunt aceleaşi pentru ambele stări.



	S_1	S_2
$P(X = 1)$	0	0.1
$P(X = 2)$	0.199	0
$P(X = 3)$	0.8	0.7
$P(X = 4)$	0.001	0.2

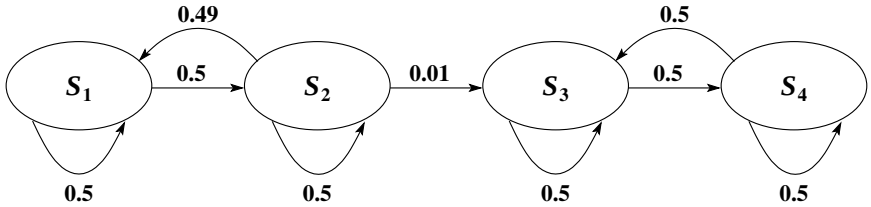
a. Daţi un exemplu de *secvenţă de ieşire* de lungime 2 care nu poate fi generată de modelul Markov dat. Justificaţi răspunsul.

b. Am generat o secvenţă de 10701^{2010} de simboli cu acest HMM şi am observat că ultimul simbol din secvenţă a fost 3. Care este starea cea mai probabilă în care a fost generat acest ultim simbol?

c. Considerăm secvenţa de semnale $\{3, 3\}$. Care este cea mai probabilă secvenţă de stări corespunzătoare acestei secvenţe de semnale? Faceţi calculele.

d. Acum vom considera secvenţa de semnale $\{3, 3, 4\}$. Care sunt primele două stări ale celei mai probabile secvenţe de stări corespunzătoare secvenţei de semnale considerate? Faceţi calculele.

e. Putem încerca să extindem întrucâtva capacitatea de modelare a acestui HMM prin „duplicarea“ fiecăreia din cele două stări. Urmând această idee, am creat diagrama de mai jos. Putem stabili probabilităţile de intrare şi cele de emisie în aşa fel încât acest model cu 4 stări, având probabilităţile de tranziţie indicate în diagramă, să fie echivalent cu modelul original cu 2 stări (adică, pentru orice secvenţă de semnale O, $P(O \mid HMM_{2states}) = P(O \mid HMM_{4states})$)? Dacă da, cum anume? Dacă nu, justificaţi.



18. **(Algoritmul Forward; demonstraţie – pasul inductiv)**

Liviu Ciortuz, 2020

Fie un model Markov ascuns μ.

a. La pasul de iniţializare al algoritmului Forward se fixează $\alpha_i(1) = \pi_i$ pentru $i = 1, \ldots, N$, unde N este numărul de stări din modelul Markov ascuns μ, iar π_i este probabilitatea de intrare în starea i. Formula de recurenţă de la pasul inductiv al algoritmului Forward pentru HMM este următoarea:

$$\alpha_j(t+1) = \sum_{i=1}^{N} \alpha_i(t) a_{ij} b_{ijO_t}, \text{ pentru } j = 1, \ldots, N \text{ şi } t = 1, \ldots, T. \tag{446}$$

Faceţi demonstraţia acestei formule de recurenţă. Pentru fiecare egalitate din cadrul demonstraţiei, precizaţi motivul/motivele pentru care are loc egalitatea respectivă.

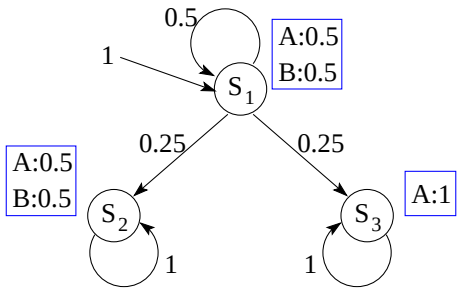
b. Cum se poate calcula $P(O_1 O_2 \ldots O_T \mid \mu)$, probabilitatea de emisie a unei secvenţe de semnale $O_1 O_2 \ldots O_T$ de către modelul μ, folosind probabilităţile Forward (α)?

19. **(HMM, calculul probabilităţii secvenţelor emise: o formulă alternativă; demonstraţie şi aplicare)**

* ***CMU, 2009 spring, Ziv Bar-Joseph, final, pr. 5***

Pentru calcularea probabilităţii unei secvenţe de observaţii, $P(O_1, \ldots, O_T)$, suntem obişnuiţi cu folosirea probabilităţilor Forward sau a probabilităţilor Backward (vedeţi problemele 18.b şi respectiv 8.b). În această problemă veţi folosi alte probabilităţi pentru a determina o formulă de calcul pentru $P(O_1, \ldots, O_T)$.

Apoi veţi aplica noua formulă pentru secvenţe de observaţii corespunzătoare următorului model Markov ascuns:



a. Fie următoarele notaţii:

$$\nu_i^t = P(O_1, \ldots, O_T \mid q_t = s_i) \text{ şi } p_t(i) = P(q_t = s_i).$$

Determinaţi o formulă pentru $P(O_1, \ldots, O_T)$ care să folosească *doar* ν_i^t şi $p_t(i)$.

b. Folosind formula găsită la punctul precedent, calculaţi $P(O_1 = B, \ldots, O_{200} = B)$ pentru modelul dat, adică probabilitatea observării a 200 de emisii B consecutive. Pentru aceasta trebuie să găsiţi un moment t convenabil, să calculaţi ν_i^t şi $p_t(i)$ corespunzători şi apoi să folosiţi aceste valori pentru a calcula probabilitatea cerută. (*Indicaţie*: pentru calcularea $p_t(i)$ se observă că tranziţiile din şi spre stările S_2 şi S_3 sunt simetrice pentru orice t, deci $p_t(S_2) = p_t(S_3)$.)

c. Folosind formula găsită la punctul a, calculaţi $P(O_1 = A, \ldots, O_{200} = A)$ pentru modelul dat, adică probabilitatea observării a 200 de emisii A consecutive. Trebuie să găsiţi din nou un moment t convenabil, dar de data aceasta este permisă calcularea în funcţie de ν_1^t, pentru care nu trebuie obţinută valoarea numerică. Toate celelalte probabilităţi implicate în formulă trebuie însă calculate. (*Indicaţie*: momentul t ales la acest punct poate fi diferit de cel ales la puctul b.)

20. **(HMM: o variaţie pe tema algoritmului Viterbi)**

York University, UK, 2001 spring,
Univ. of York (UK), J. Cussens, D. Kudenko, open exam., pr. 1.1

Trebuie să călătoreşti între două puncte, A şi B. Aceste puncte sunt despărţite de n râuri. Peste fiecare din cele n râuri sunt construite m poduri. Poţi trece peste un râu doar dacă foloseşti unul din cele m poduri de pe râul respectiv. Considerăm că se cunosc distanţele dintre orice două poduri care sunt situate pe râuri vecine.

a. Defineşte un algoritm eficient care găseşte cel mai scurt drum dintre A şi B şi returnează (şi) lungimea acestui drum. *Observaţie*: Nu trebuie să implementezi acest algoritm, ci doar să descrii în pseudo-cod cum lucrează.

b. Compară algoritmul rezultat la punctul a cu algoritmul Viterbi.

21. **(Algoritmul Forward-Backward: implementare)**

∗ ***Liviu Ciortuz***

Pentru un model Markov ascuns se consideră date (ca variabile globale):

- **N (numărul de stări) şi o secvenţă O de lungime T peste un alfabet de K simboli;**
- **vectorul de probabilităţi de intrare pi[N], matricea de probabilităţi de tranziţie a[N,N] şi matricea probabilităţi de emisie b[N,N,K];**
- **matricele alpha[N,T+1] şi beta[N,T+1] care se presupun a fi calculate de algoritmii Forward şi respectiv Backward;**
- **matricea p[T+1][N][N] pentru claculul mediilor variabilelor neobservabile (reprezentând, fiecare, tranziţia de la starea i la starea j la momentul t).**

Definiţi în limbajul Python / C două funcţii, expectation() **şi** maximization()**, care servesc la implementarea algoritmului EM/Forward-Backward pentru HMM. Funcţia** expectation **va calcula matricea p[T+1,N,N], iar funcţia** maximization **va înlocui pi, a şi b cu noile valori calculate de algoritmul EM.**

Vă reamintim algoritmul EM/Forward-Backward:

• Estimare:
Dată o secvenţă (observaţie) O, se defineşte probabilitatea traversării unui anumit arc la momentul de timp t:

$$\begin{aligned} p_t(i,j) &\overset{def.}{=} P(X_t = i, X_{t+1} = j | O, \mu) \\ &= \frac{P(X_t = i, X_{t+1} = j, O|\mu)}{P(O|\mu)} = \frac{\alpha_i(t) a_{ij} b_{ijO_t} \beta_j(t+1)}{\sum_{m=1}^{N} \alpha_m(t)\beta_m(t)} \\ &= \frac{\alpha_i(t) a_{ij} b_{ijO_t} \beta_j(t+1)}{\sum_{m=1}^{N}\sum_{n=1}^{N} \alpha_m(t) a_{mn} b_{mnO_t} \beta_n(t+1)} \end{aligned}$$

Sumând după t, obţinem:
$\sum_{t=1}^{T} p_t(i,j)$ **= media (valoarea aşteptată) a numărului de tranziţii de la s_i la s_j în timpul emisiei lui O;**
$\sum_{j=1}^{N}\sum_{t=1}^{T} p_t(i,j)$ **= media (valoarea aşteptată) a numărului de tranziţii din s_i în timpul emisiei lui O.**

• Maximizare:
Folosind $\mu = (A, B, \Pi)$, se calculează $\hat{\mu} = (\hat{A}, \hat{B}, \hat{\Pi})$:

$$\begin{aligned} \hat{\pi}_i &= \frac{\sum_{j=1}^{N} p_1(i,j)}{\sum_{l=1}^{N}\sum_{j=1}^{N} p_1(l,j)} = \sum_{j=1}^{N} p_1(i,j) = \gamma_i(1) \\ \hat{a}_{ij} &= \frac{\sum_{t=1}^{T} p_t(i,j)}{\sum_{l=1}^{N}\sum_{t=1}^{T} p_t(i,l)} \\ \hat{b}_{ijk} &= \frac{\sum_{t:O_t=k,\ 1\le t\le T} p_t(i,j)}{\sum_{t=1}^{T} p_t(i,j)}. \end{aligned}$$

22. **(Un HMM cu emisii gaussiene: varianta normalizată pentru probabilităţile Forward, Backward şi pentru algoritmul EM/Forward-Backward; implementare şi aplicare)**

• ○ ***CMU, 2009 spring, Ziv Bar-Joseph, HW4, pr. 3***

Robert merge să înoate zilnic. El foloseşte fie culoarul 1, fie culoarul 2 din piscină. Dacă piscina este aglomerată, atunci el alege să înoate pe culoarul 2, în caz contrar alege culoarul 1. Îşi notează întotdeauna timpul dus-întors de înot. Descoperă astfel că performanţa sa pe culoarul 2 corespunde unei distribuţii normale în jurul a 60 de secunde cu deviaţia standard 1, iar pe culoarul 1 unei distribuţii normale în jurul a 63 de secunde cu deviaţia standard 1. Robert observă de asemenea un tipar al utilizării piscinei: dacă într-o zi este aglomerat, atunci probabilitatea ca în ziua următoare să fie de asemenea aglomerat este de 0.3, iar dacă nu este aglomerat, atunci probabilitatea ca în ziua următoare să fie aglomerat este de 0.6.

Deşi îşi notează zilnic timpul efectuat, Robert nu îşi notează pe ce culoar înoată. Astfel în însemnările sale se găsesc următoarele date:

Ziua (n)	**Timpul** (x) **în secunde**
1	60.0
2	62.0
3	59.0
4	63.0
5	62.0
6	64.0
7	61.0

a. Desenaţi un *model grafic* care să descrie cât mai bine măsurătorile făcute de Robert (notate sub forma x_n) precum şi variabilele ascunse (notate cu z_n).

b. Calculaţi distribuţia condiţionată a variabilelor z_n în raport cu $x_1, \ldots, x_n$. Vă cerem să elaboraţi calculul/demonstraţia în întregime.

***Indicaţie*: Aţi putea formula $\alpha(z_n) = P(z_n \mid x_1, \ldots, x_n)$ în funcţie de $\alpha(z_{n-1})$, într-o formă parametrizată.**

c. Derivaţi ecuaţia de recursie înapoi (engl., backward equation) pentru probabilitatea a posteriori $P(z_n \mid x_1, \ldots, x_N)$, notată cu $\beta(z_n)$, unde n ia valori în $\{1, \ldots, N\}$. Faceţi calculul/demonstraţia în întregime.

***Indicaţie*: Aţi putea formula $\beta(z_n)$ în funcţie de $\beta(z_{n+1})$ şi $\alpha(z_n)$, într-o formă parametrizată.**

d. Cât este media condiţionată pentru tranziţia z_n, z_{n+1} dată fiind secvenţa de emisii $x_1, \ldots, x_N$, adică $E(z_n, z_{n+1}^T \mid x_1, \ldots, x_N)$? Faceţi calculul / demonstraţia în întregime.

***Indicaţie*: Ţineţi cont că aveţi variabile ascunse discrete (de tip categorial) pentru stări şi variabile continue (de tip gaussian) pentru observaţii. Aşadar, $E(z_n z_{n+1}^T \mid x_1, \ldots, x_N)$ va fi o matrice 2×2 în acest caz.**

Pentru punctele următoare $(e - h)$, este necesar ca în prealabil să faceţi implmenetarea formulelor pe care le-aţi determinat la punctele $b - d$.

e. Folosind datele din tabelul de mai sus, stabiliţi dacă piscina este aglomerată sau nu în ziua a patra.

f. Robert nu este sigur în legătură cu estimările sale pentru culoarele 1 şi 2. Aţi putea să faceţi o re-estimare? În mod concret, puteţi presupune că performanţele lui pe culoarele 1 şi 2 corespund distribuţiilor normale $\mathcal{N}(\mu_1, \sigma_1^2)$ şi respectiv $\mathcal{N}(\mu_2, \sigma_2^2)$. Puteţi estima parametrii $\mu_1, \sigma_1, \mu_2, \sigma_2$?

***Sugestie*: Veţi putea apela la un algoritm de tip EM care foloseşte probabilităţile $\alpha(z_n)$ şi $\beta(z_n)$ determinate anterior. Veţi considera că probabilităţile de tranziţie, precum şi valorile iniţiale ale parametrilor $\mu_1, \sigma_1, \mu_2, \sigma_2$ nu se re-estimează aici.**

g. Pe de altă parte, Robert nu este sigur în legătură cu pattern-ul zilelor aglomerate de la piscină, adică nu ştie dacă probabilităţile de tranziţie au fost corect estimate ca fiind 0.3 şi 0.6. Aţi putea să faceţi o re-estimare a acestor probabilităţi folosind observaţiile de mai sus?

***Sugestie*: Veţi putea apela la un [alt] algoritm de tip EM care foloseşte probabilităţile $\alpha(z_n)$ şi $\beta(z_n)$ determinate anterior. Veţi considera că valorile iniţiale ale probabilităţilor de tranziţie nu se re-estimează aici. Veţi folosi ca valori pentru parametrii celor două distribuţii gaussiene mediile $\mu_1 = 60, \mu_2 = 63$ şi varianţele $\sigma_1^2 = \sigma_2^2 = 1$.**

h. Să presupunem că Robert nu este sigur nici în legătură cu parametrii $\mu_1, \sigma_1, \mu_2, \sigma_2$ şi nici cu parametrii zilelor aglomerate de la piscină. Aţi putea să faceţi o re-estimare a tuturor acestor parametri folosind observaţiile de mai sus?

Vă solicităm să implementaţi algoritmul dumneavoastră de învăţare automată în limbajul de programare preferat. Indicaţi rezultatele obţinute în urma estimării parametrilor folosind datele din tabel.

„Mai bun este sfârşitul unui lucru decât începutul lui...“

Cartea Eclesiastului 7:8.a

Alcătuirea acestei culegeri a fost pentru noi, autorii ei, asemenea unui drum, pe care l-am parcurs sub îndrumarea unor profesori şi asistenţi de departe. La capătul acestui drum, ne dăm seama că lungimea drumului şi greutăţile prin care am trecut n-au fost determinante, ci cel mai mult au contat „peisajele“ care se desfăşoară de o parte şi de alta a drumului.

Dragă „cititorule“, sperăm că după ce vei fi parcurs, la rândul tău, alături de noi o parte din acest drum (care pe noi ne-a încântat!), ţi-am devenit şi noi călăuze, dar despărţindu-ne aici, îţi vei continua propriul tău drum, fie aplicând cele arătate aici, fie parcurgând alte capitole. Sau, pur şi simplu, folosind abilităţile pe care le-ai dobândit *antrenându-te* cu astfel de exerciţii, vei putea să le *generalizezi* în propriile tale domenii de interes. Îţi dorim sincer *Mult succes!*

Autorii



© *M. Romanică*