ML course, 2026 fall

https://piazza.com/info.uaic.ro/fall2026/ml2026f

What you should know:

Week 1:

PART I (course): A brief introduction to Machine Learning

(slides 1-10 from https://edu.info.uaic.ro/invatare-automata/SLIDES/ml0.pdf)

PART II (seminary): Revision: Basic issues in Probabilities[1]

(slides 4-17 from https://edu.info.uaic.ro/invatare-automata/SLIDES/foundations.pdf)

Read: Chapter 2 (section 2.1) from the *Foundations of Statistical Natural Language Processing* book by Christopher Manning and Hinrich Schütze, MIT Press, 2002.[2]

PART II.1: Random events

Concepts/definitions:

- sample space, random event, event space
- probability function
- conditional probabilities
- independent random events (2 forms); conditionally independent random events (2 forms)

Theoretical results/formulas:

- elementary probability formula: $\frac{\#\text{ favorable cases}}{\#\text{ all possible cases}}$
- the "multiplication" rule; the "chain" rule
- "total probability" formula (2 forms)
- Bayes formula (2 forms)

Exercises illustrating the above concepts/definitions and theoretical results/formulas, in particular: proofs for certain properties derived from the *definition of the probability function* for instance: $P(\emptyset) = 0$, $P(\bar{A}) = 1 - P(A)$, $A \subseteq B \Rightarrow P(A) \leq P(B)$

Ciortuz et al.'s exercise book (2026f) ch. *Foundations*, ex. 1-5 [6-7] 8, 89-93 [94-95] 96

[1] Professors / teaching assistants who are in charge with the seminaries may decide to allocate for this revision more than one week or, alternatively, to address these probabilities issues "by need", i.e. when required by the machine learning algorithms that students will learn / apply in class.

[2] For a more concise / formal introductory text, see *Probability Theory Review for Machine Learning*, Samuel Ieong, November 6, 2006 (https://see.stanford.edu/materials/aimlcs229/cs229-prob.pdf) and/or *Review of Probability Theory*, Arian Maleki, Tom Do, Stanford University (https://cs229.stanford.edu/section/cs229-prob.pdf).

Part II.2: Random variables [and a few basic probabilistic distributions]

Concepts/definitions:

- random variables;
  random variables obtained through function composition
- discrete random variables;
  probability mass function (p.m.f.)
  examples: Bernoulli, categorical, binomial [multinomial, geometric, Poisson] distributions
- expectation (mean), variance, standard variation; covariance. (**See definitions!**)
- multi-valued random functions;
  joint, marginal, conditional distributions
- independence of random variables;
  conditional independence of random variables

Theoretical results/formulas:

- for any discrete variable X:
  $\sum_x p(x) = 1$, where p is the pmf of X

  for any continuous variable X:
  $\int p(x)\,dx = 1$, where p is the pdf of X
- $E[X + Y] = E[X] + E[Y]$
  $E[aX] = aE[X]$

  Corollary: the *liniarity* of expectation:
  $E[\sum_{i=1}^n a_i X_i] = \sum_{i=1}^n a_i E[X_i]$.

  $Var[aX] = a^2\, Var[X]$
  $Var[X] = E[X^2] - (E[X])^2$
  $Cov(X, Y) = E[XY] - E[X]E[Y]$
  $Var[X + Y] = Var[X] + Var[Y] + 2Cov(X, Y)$
- X, Y independent variables $\Rightarrow$
  $Var[X + Y] = Var[X] + Var[Y]$
- X, Y independent variables $\Rightarrow$
  $Cov(X, Y) = 0$, i.e. $E[XY] = E[X]E[Y]$

Exercises illustrating the above concepts/definitions and theoretical results/formulas, concentrating especially on:

- computing probabilities
- computing means / expected values of random variables
- verifying the [conditional] independence of two or more random variables
- identifying in a given problem's text the underlying probabilistic distribution: either a basic one (e.g., Bernoulli, binomial, categorial etc.), or one derived [by function composition or] by summation of identically distributed random variables

Ciortuz et al.'s exercise book: ch. *Foundations*, ex. 9.ab, [9.c-11] 12-16, 23-26 [27-28] 97-99 [100] 101-103

Advanced issues (I):

- Taking *A self evaluation test for the ML course*, CMU, 2014 fall, W. Cohen:
  http://www.cs.cmu.edu/∼wcohen/10-601/self-assessment/Intro_ML_Self_Evaluation.pdf

◦ Similar tests:
http://www.cs.cmu.edu/∼ninamf/courses/601sp15/hw/homework1.pdf (CMU, 2015 spring, N. Balcan)
http://curtis.ml.cmu.edu/w/courses/images/8/88/Homework1.pdf (CMU, 2016 spring, W. Cohen, N. Balcan)
http://www.cs.cmu.edu/∼mgormley/courses/10601b-f16/files/hw1_questions.pdf (CMU, 2016 fall, N. Balcan, M. Gormley)

Advanced issues (II):

- probability density function (p.d.f.), cumulative function distribution (c.d.f.)
- continuous random variables;
  examples: uniform, Gaussian, [exponential, Gamma, Beta, Laplace] distributions
- correlation coefficient of two random variables

Ciortuz et al.'s exercise book: ch. *Foundations*, ex. 17-19 [30-31] 32-35 [36-38] 104-108 [109] 110-112, 115-116, 122

Advanced issues (III):

Concepts/definitions:

- vector of random variables; covariance matrix for a vector of random variables
  pozitive [semi-]definite matrices
  negative [semi-]definite matrices

Theoretical result:

- For any vector of random variables, the covariance matrix is symmetric and positive semi-definite.

  Ciortuz et al.'s exercise book: ch. *Foundations*, ex. 20

Advanced issues (IV):

- the *likelihood function* (see *Estimating Probabilities*, additional chapter to the *Machine Learning* book by Tom Mitchell, 2016)
- the *Bernoulli distribution*: MLE and MAP estimation of the parameter

Ciortuz et al.'s exercise book: ch. *Foundations*, ex. 42-43

Advanced issues: The Gaussian distribution

Read: "Pattern Classification", R. Duda, P. Hart, D. Stork, 2nd ed. (Wiley-Interscience, 2000), Appendices A.4.12, A.5.1 and A.5.2.

The uni-variate Gaussian distribution: p.d.f., mean and variance, and c.d.f.;
the variable transformation enabling to go from the non-standard case to the standard case;

The multi-variate Gaussian distribution: pdf, the case of diagonal covariance matrix

Ciortuz et al.'s exercise book, ch. *Foundations*:
ex. 30-32 (unidimensional), 35, 38, 117 (bidimensional), 36-37 (multidimensional);
parameter estimation: ex. 50-51, 135-136;
the central limit theorem (the i.i.d. case) and the law of large numbers: ex. 40.

Week 2.$\frac{1}{2}$: Introduction to Information Theory

Read: Chapter 2 (section 2.2) from the *Foundations of Statistical Natural Language Processing* book by Christopher Manning and Hinrich Schütze, MIT Press, 2002.
(slides 32-35 [36-39] from https://edu.info.uaic.ro/invatare-automata/SLIDES/foundations.pdf)

Concepts/definitions:

- entropy;
  specific conditional entropy;
  average conditional entropy;
  information gain (mutual information)
  joint entropy;

Theoretical results/formulas:

- $0 \leq H(X) \leq H(\underbrace{1/n, 1/n, \ldots, 1/n}_{n \text{ times}}) = \log_2 n$
- $IG(X;Y) \stackrel{def.}{=} H(X) - H(X|Y) = H(Y) - H(Y|X)$
- $IG(X;Y) \geq 0$
- $IG(X;Y) = 0$ iff X and Y are independent
  - $IG(X;X) = H(X)$
- $H(X,Y) = H(X) + H(Y|X) = H(Y) + H(X|Y)$
  (generalisation: the chain rule, $H(X_1, \ldots, X_n) = H(X_1) + H(X_2 \mid X_1) + \ldots + H(X_n \mid X_1, \ldots, X_{n-1})$)
- $H(X,Y) = H(X) + H(Y)$ iff X and Y are indep.

Exercises illustrating the above concepts/definitions and theoretical results/formulas, concentrating especially on:

- computing different types of entropies:
  Ciortuz et al.'s exercise book: ch. *Foundations*, ex. 55-59 [60-61] 140-142;
- proof of some basic properties:
  Ciortuz et al.'s exercise book: ch. *Foundations*, ex. [62] 143-146, 151.

Advanced issues:

- relative entropy (Kullback-Leibler divergence)
  Ciortuz et al.'s exercise book: ch. *Foundations*, ex. 63, 148-149;
- cross-entropy
  Ciortuz et al.'s exercise book: ch. *Foundations*, ex. 64-65, 147.

Weeks 2.$\frac{2}{2}$, 3 and 4: Decision Trees — illustrating basic issues in ML

Read: Chapter 3 from Tom Mitchell's *Machine Learning* book.

Important Note:
See (i.e., do *not* skip!) the Overview (rom.: "Sumar") section for the *Decision Trees* chapter in Ciortuz et al.'s exercise book. It is in fact a "road map" for what we will be doing here. (This *note* applies also to all chapters.)

Week 2.$\frac{2}{2}$:

Decision trees and the **ID3 algorithm:**
applications;
analysis of the ID3 algorithm (as an algorithm *per se*);
properties of ID3 trees:
Ciortuz et al.'s exercise book, ch. *Decision trees*, ex. 1-9, 32-46 [58]

- **decision trees:**
seen as data structures: ex. 1, 7.b, 32
and as logic programs: ex. 2.e, 40.bc
- **ID3 algorithm:**
simple applications: ex. 2-3, 5, 38-39, 41-42
- **analysis of ID3 as an algorithm *per se*:**
recursive, divide-et-impera
search algorithm (1-step look-ahead): ex. 3, 39
greedy: ex. 4, 21.a, 40
- **properties of ID3 trees:** ex. 2-4, 7-9, 21.a, 33, 39-40 [45] 56-57
- **implementation exercises:** ex. 35 [58]
- **revision:** ex. 21, 56-57

Important Note:
Some of the exercises listed above would be done in class (i.e., at seminaries) in an easier / nicer way if students would priorily do at home the exercise 35, which asks for the **implementation** of the **information gain** (and also entropy, specific conditional entropy and average conditional entropy), starting form the counts (more precisely, from the data partitions) associated to the leaf nodes of a **decision stump**.[3] Alternatively, the exercise 36 advises the student on how to conveniently use a **pocket calculator** in order to calculate the above mentioned entropies and the information gain.

Implementation exercises:

CMU, 2012 spring, Roni Rosenfeld, HW3[4]
- Complete a given C (incomplete) implementation for ID3.
- Work firstly on a simple example (Play Tennis from Tom Mitchell's *Machine Learning* book) and secondly on a real dataset (Agaricus-Lepiota Mushrooms).
  - Perform *reduced-error (top-down vs. bottom-up) pruning* to cope with *overfitting*.[5]

[3] This implementation could be later extanded to an implementation of ID3 algorithm (the basic form); see ex. 54.

[4] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2012s.RR.HW3.DT+pruning.C-code.PlayTennis+mushrooms/

[5] CMU, 2011 spring, T. Mitchell, A. Singh, HW1, pr. 3 — https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2011s.TM+AS.HW1.pr3.DT+epsilon-pruning.DATA-mushrooms.C-code/ — is similar to the above problem, except that *pruning* a node is conditioned on getting at least an ε increase in accuracy. (Dataset: mushrooms.)

Note: CMU, 2011 spring, Roni Rosenfeld, HW3[6] – a similar problem to the above one — uses a *chess* dataset generated by Alen Shapiro (see *Structured Induction in Expert Systems*, 1983, 1987).

[6] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2011s.RoniRosenfeld.HW3.DT.C-code.chess/

Weeks 3-4:
extensions of the ID3 algorithm;
analysis of ID3 as a Machine Learning algorithm;
Ciortuz et al.'s ex. book, ch. *Decision trees*, ex. 10-16 [17-18] 19-21 [47] 48-50 [52-53] 54-57

• **extensions of the ID3 algorithm**
– handling of continuous attributes: ex. 10-12, 47-50
– decision surfaces, decision boundaries: ex. 10, 48, and ch. *Instance-based learning*, ex. 11.b

◦ **other extensions to the ID3 algorithm**
– handling of attributes with many values: ex. 13, 52
– handling of attributes with costs: ex. 14
– using other impurity measures as local optimality criterion in ID3: ex. 15
– reducing the greedy behaviour of the ID3 algorithm: ex. 17-18

• **analysis: ID3 as a Machine Learning algorithm**
– *inductive bias* for ID3:
[LC: a hierarchical structure of the model, compatibility/consistency with the data, and]
compactness of the resulting decision tree;
– error analysis/computation: training error, validation error, n-fold cross-validation, CVLOO:
ex. 6-8, 10, 21.d, 44-46, 48-49, 50.d
– ID3 as "eager" learner: ex. 16
– ID3 and [non-]robustness to noises, and *overfitting*: ex. 10, 21.bc, 49
– *pruning* strategies for decision trees: ex. 19 [20] [53] 54 [55]

Implementation exercises:

CMU, 2011 fall, T. Mitchell, A. Singh, HW1, pr. 2[7]
- Working with continuous attributes on a real *dataset*: Breast Cancer.
- Complete a given a Matlab/Octave implementation for ID3.
- Perform *reduced-error pruning*.
- Implement another splitting criterion: the *weighted misclassification rate*.

[7] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2011f.TM+AS.HW1.pr2.DT-with-continuous-attributes.BreastCancer.Matlab-code/

Weeks 5-6: Bayesian Classifiers

Read:
Chapter 6 from T. Mitchell's *Machine Learning* book (except subsections 6.11 and 6.12.2);
(slides #4-6, 12-14 in https://edu.info.uaic.ro/invatare-automata/SLIDES/ml6.pdf)

Week 5: The Naive Bayes and Joint Bayes classifiers

- Bayes' theorem:
  Ciortuz et al.'s exercise book, ch. *Foundations*, ex. 6-7, 94-95;
- conditionally independent random [events,] variables:
  Ciortuz et al.'s exercise book, ch. *Foundations*, ex. 15-16, 99-102 [103];
- classes of machine learning hypotheses: MAP hypotheses vs. ML hypotheses:
  Ciortuz et al.'s exercise book, ch. *Bayesian classification*,[8] ex. 1-4, 24-26, 42;
- pseudo-code: ML book, page 177, and slides #12-14 in
  https://edu.info.uaic.ro/invatare-automata/SLIDES/ml6.pdf
- **applications of Naive Bayes and Joint Bayes algorithms:** ex. 5-9, 27-31.
- the *complexity of the model* created by these algorithms, i.e. the number of parameters to be estimated from training data: linear for Naive Bayes $(2d + 1)$ and exponential for Joint Bayes $(2^{d+1} - 1)$:[9] ex. 7.e, ex. 29.ab, ex. 35.ac;

Week 6:

- **computation of the [training] error rate of Naive Bayes:** ex. 10-11 [12] 32-36;
- *sample complexity* of Naive Bayes and Joint Bayes: ex. 13;
- **the nature of the *decision boundary* determined by Naive Bayes** (and the relationship to logistic regression): ex. 14, 38;
- comparisons with other classifiers: ex. 40-41;
- revision: ex. 43.

Advanced issues (I):

- an information-theoretic view on Naive Bayes algorithm [see the link with the equivalence relation max likelihood $\Leftrightarrow$ min cross-entropy]: ex. 149 (and ex. 148) from the *Foundations* chapter.

Advanced issues (II): Gaussian Bayesian classification[10]

- applications: the *uni-variate* case: ex. 15-16, 45-46, 54;
  the *bi-variate* case: GNB: ex. 44.c; GJB: ex. 20-21, 51.ef, 52-53;
- the nature of the *decision boundary*): GNB: ex. 17, 49; GJB: ex. 18-19, 51.ab;
- the relationship between Gaussian Bayesian classification and logistic regression: ex. 22, 23;
- the number of parameters to be estimated: GNB: ex. 47.b, GJB: ex. 47.d;
- GNB: the link between the estimation of parameters and the maximization of the log-verosimility of the data: ex. 48.

[8] This chapter is equally the source for all exercises listed below.
[9] The number of parameters indicated in parantheses refer to the case when both the input attributes and the output attribute are Bernoulli.
[10] I.e., Bayesian classifiers with continuous [numerical] input attributes — the case of Gaussian attributes.

Implementation exercises:

0. CMU, 2010 fall, Ziv Bar-Joseph, HW1, pr. 4[11]
Implement the Naive Bayes classification algorithm,
and perform CVLOO on a toy ("weather prediction") dataset;
do *feature selection* based on CVLOO.

1. Stanford, 2012 spring, Andrew Ng, pr. 6[12]
Implement the Naive Bayes (train and test) algorithm;
use it as a spam filter on a subset of the Ling-Spam dataset.

2. CMU, 2011 spring, Tom Mitchell, HW2, pr. 3[13]
Implement the Naive Bayes classification algorithm and perform n-ary classification on the *20 Newsgroups* datset;
for the $P(X_i|Y)$ parameters, do MAP estimation (instead of MLE) using as prior the Dirichlet distribution;[14]
identify the *key words* (for classification) using *conditional entropy*; analyse its effectiveness relative to the *information gain*.

[11] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2010f.ZBJ.HW1.pr4.NBayes.wheatherPrediction.data/

[12] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/Stanford.2012.OpenCourse.ANg.Ex6.NaiveBayes.spam-filtering.data.Matlab-code.sol/

[13] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2011s.TM.HW2.pr3.NB+featSelection.20newsgroups.DATA+CODE+sol/

[14] An earlier exercise, CMU, 2009 spring, T. Mitchell, HW3, pr. 2 — https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2009s.TM.HW3.pr2.textClassif.NB+LR.hokey-vs-baseball.data.Matlab-code.featureSelection.solution/ —, centered on Naive Bayes and the MAP estimation with Dirichlet prior, but instead of asking the student to perform n-ary classification (on the 20 newsgroups dataset), it limited itself to binary classification on a much simpler dataset: hokey vs. baseball newsgroups. A similar exercise — CMU, 2014 fall, W. Cohen and Z. Bar-Joseph, HW2, pr. 6, https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2014f.WCohen+ZBJ.HW2.pr6.Byes+text-classification.DATA+code+sol/ — uses a simple measure for feature (i.e. keyword) selection, and classifies texts from The Economist and The Onion.

Week 7: Estimating the parameters of some discrete probabilistic distributions

Maximum Likelihood Estimation (MLE) — An Introduction;
Maximum A posteriori Probability (MAP) Estimation — An Introduction.

- The **likelihood function, maximum likelihood estimation (MLE):**
  Ciortuz et al.'s exercise book, ch. *Foundations*, ex. 42.
- The **Bernoulli distribution, MLE:**
  Ciortuz et al.'s exercise book, ch. *Foundations*, ex. 43.A, 124, 125.ac, 126.a, 127.a.
- The **Bernoulli distribution, MAP estimation:**
  Ciortuz et al.'s exercise book, ch. *Foundations*, ex. 43.B, 126.bd.
- The **categorical distribution, MLE [and MAP estimation]:**
  Ciortuz et al.'s exercise book, ch. *Foundations*, ex. 44 [0.129].

Read: *Estimating Probabilities*, additional chapter to the *Machine Learning* book by Tom Mitchell, 2016. http://www.cs.cmu.edu/%7Etom/mlbook/Joint_MLE_MAP.pdf

Advanced issues:

- **MLE and MAP estimation for the Poisson distributions:**
  Ciortuz et al.'s exercise book: ch. *Foundations*, ex. 0.46
- **MLE and MAP estimation for the geometrical distributions:**
  Ciortuz et al.'s exercise book: ch. *Foundations*, ex. 0.130

Week 8: midterm EXAM

Week 9: Instance-Based Learning

Read: Chapter 8 from Tom Mitchell's *Machine Learning* book.

application of the k-NN algorithm:
Ciortuz et al.'s exercise book, ch. *Instance-based learning*, ex. 1-7, 12.ab, 16-25, 26.a, 28;
comparisons with other classification algorithms: ex. 11.b, 12.cd, 13, 26.b;

Shepard's algorithm: ex. 8.

Advanced issues: Some theoretical properties of the k-NN algorithm

the curse of [high] dimensionality: ex. 9;
the relationship between 1-NN and Optimal Bayes: ex. 10, 27;
kernelization of 1-NN with RBF has no effect: ex. 15.

Implementation exercises:

0. Stanford, 2013 fall, CS106L (Standard C++ Programming Lab) course, Cristian Cibils Bernardes, HW3
Implement *kd*-trees, in C++; use them in conjunction with k-NN.

1. CMU, ? spring, 10-711 (ML) course, HW1, pr. 5[15]
Implement k-NN in Matlab;
study the evolution of the test error with k, on a dataset from $\mathbb{R}^2$.

Advanced issues: Compare the performances of k-NN (where k is selected according to the previous task) and Gaussian Naive Bayes on the given dataset.

- MPI Informatik, Saarbrücken (Germany), 2005 spring, J. Rahnenführer and A. Alexa, HW4, pr. 11[16]
  Use k-NN from R, on the *Breast Cancer* gene expression dataset (full vs. reduced versions);
  implement a statistical *feature selection* filter (as specified in the problem), and by choosing a good value for k via cross-validation, make prediction for 3 test probes.

2. CMU, 2004 fall, Carlos Guestrin, HW4, pr. 3.2-8[17]
Implement k-NN in Matlab; apply it for *hand written-character recognition* (on the given dataset); explore different methods to choose a good value for k: the *train-test* method, the n-fold cross-vlidation method, and the CVLOO method.

- CMU, 2005 spring, Carlos Guestrin, Tom Mitchell, HW3, pr. 3.1-4[-7][18]
  Implement k-NN in Matlab;
  apply it for *text classification* (the dataset is provided), using the *cosine distance* (its implementation in Matlab is provided);[19]
  Implement n-fold CV, and apply it ($n = 10$) for different values of k.

  Advanced issues: Make comparisons between k-NN and SVM, using the *libSVM* implementation.[20]

[15] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.s.10-701.HW1.pr5.kNN-vs-GNB.data/
[16] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/MPI.Rahnenfuhrer+Alexa.2005.HW4.BreastCancer+kNN/
[17] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2004f.CGuestrin.HW4.pr3.2-8.k-NN.hand-written-char-reco.data/
[18] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2005s.CGuestrin+TMitchell.HW3.pr3.textClassif.data+code
[19] For the cosine distance, see http://en.wikipedia.org/wiki/Cosine_similarity.
[20] Matlab scripts for doing training and testing with SVM are provided.

3. CMU, 2010 spring, Eric Xing, Tom Mitchell, Aarti Singh, HW2, pr. 2.1[21]
Implement k-NN in Matlab, using the Euclidian distance;
apply it to *face recognition* on the provided ORL database;
perform 10-fold cross-validation to select a good value for k.[22]

[21] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2010s.EX+TM+AS.HW2.pr2.multinomial-logistic-regression.ORL-Faces-dataset.NO-code/

[22] Advanced issues: Another exercise — CMU, 2011 fall, Eric Xing, HW5, pr. 1.1-2, https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2011f.EricXing.HW5.pr1.1-NN+PCA.ORL-Face-data.sol.NO-code/ — uses the same (ORL Faces) database but first asks for *feature selection* via the application of PCA (Principal Component Analysis) from Matlab, and then requires the application of 1-NN.

Week 10: Logistic Regression — An Introduction.

Prerequisites:

– The sigmoid / logistic function: :
Ciortuz et al.'s exercise book, ch. *Regression Methods*, ex. 33.

– **Optimization methods:** the **gradient method**, and the **Newton method** (and its generalization, the **Newton-Raphson method**):
Ciortuz et al.'s exercise book, ch. *Foundations*, ex. 80, 128, 164.a, 166, 167.

Read:

– *Generative and Discriminative Classifiers: Naive Bayes and Logistic Regression*, additional chapter to the *Machine Learning* book by Tom Mitchell, 2017.
http://www.cs.cmu.edu/%7Etom/mlbook/NBayesLogReg.pdf

– *The Regression Problem*, ch. 6 in *Mathematical Foundations of Deep Learning: An introduction*, 2023, Leonid Beryland, Pierre-Emmanuel Jabin, 2023, Walter de Gruyter GmbH, Berlin/Boston.

- theoretical foundations: Ciortuz et al.'s exercise book, ch. *Regression Methods*, ex. 13, 14, 34, 35.A;
- application: Ciortuz et al.'s exercise book, ch. *Regression Methods*, ex. 15, 36,
- logistic regression and Naive Bayes have the same (linear!) type of decision boundary: Ciortuz et al.'s exercise book, ch. *Bayesian Classification*, ex. 14.a, ex. 40;
- logistic regression is the *discriminative* correspondent of the *generative* Naive Bayes classifier: Ciortuz et al.'s exercise book, ch. *Regression Methods*, ex. 35.e;
- unlike Naive Bayes, logistic regression is not affected by the *duplication of attributes*: Ciortuz et al.'s exercise book, ch. *Regression Methods*, ex. 37;
- relationships between logistic regression and other classisiers: Ciortuz et al.'s exercise book, ch. *Regression Methods*, ex. 40.ab;
- True/False?: Ciortuz et al.'s exercise book, ch. *Regression Methods*, ex. 41.bc.

Advanced issues (I):

- coping with overfitting in logistic regression: the *regularization* of the w parameter: ex. 17, 35.B;
- logistic regression can be *kernelized*: ex. 18;
- logistic regression can be generalized to *multinomial calssification*: the softmax regression: ex. 19;
- logistic regression is the ,,probabilistic face" of the linear perceptron with logistic cost: ch. *Artificial Neural Networks*, ex. 16.

Implementation exercises

1. CMU, 2014 fall, William Cohen, Ziv Bar-Jospeh, HW3, pr. 1
Implement *Logistic Regression* and apply it to the task of hand-written character recognition.[23]

   ◦ CMU, 2014 fall, W. Cohen, Z. Bar-Joseph, HW3, pr. 1.4-6
   Implement *Multinomial Logistic Regression* and apply it to the *ORL Faces* dataset [while pr. 1.1-3,7 compares (M)LR with K-NN, Gaussian Naive Bayes and Gaussian Joint Bayes on this dataset].

2. CMU, 2010 fall, Ziv Bar-Jospeh, HW2, pr. 4
Study the regularization effect for logistic regression, using L2 and (especially) L1 norm, especially for *feature selection*. Work on the *communities and crime* dataset.

Advanced issues (II):

- The relationship between Logistic Regression and Naive Bayes

Read: Ciortuz et al.'s exercise book, ch. *Bayesian classification*, ex. 14.

Implementation exercises:

1. CMU, 2010 fall, Aarti Singh, HW1, pr. 5
Implement Naive Bayes (using Laplace's "Add-One" smoothing rule) and Logistic Regression, and compare them on the *20 Newsgroups* datset;
analyse whether Logistic Regression is affected by the problem of *unseen* words/data (as Naive Bayes is).

2. CMU, 2009 spring, Tom Mitchell, HW3, pr. 2
Similarly to the above exercise, it asks for the implementation of Naive Bayes and Logistic Regression (LR), evaluation on a smaller dataset (hockey and baseball newsgroups), extension with *feature selection* based on the norm of weights computed by LR, and finally the analysis the effect of *feature* (i.e. word) *duplication* on both NB and LR.

[23] A similar exercise, CMU, 2009 spring, Ziv Bar-Joseph, HW2, pr. 4.1-2, applies logistic regression (LR) on the *Breast Cancer* dataset [while pr. 4.3-4 compares LR with the Rosemblatt *perceptron* on this dataset].

Week 11: The AdaBoost Algorithm — An Introduction

- pseudocode + intro. to theoretical foundations:
  Ciortuz et al.'s exercise book, ch. *Decision trees*, ex. 22;
  applications: ex. 24, 25, 59-66, 71, 73;
- revision: ex. 31, 78

Advanced issues:

- AdaBoost as an optimisation algorithm: ex. 23, 26
- γ-*weak learnability* and *convergence* of [the training error $err_S(H_T)$ produced by the combined hypothesis of] the AdaBoost algorithm: ex. 23.de, 68;
  exemplifying situations when there is no *guarantee* for γ-weak learnability: ex. 25.b, 67
- a class of concepts which can be represented using linear combinations of decision stumps: ex. 71
- a class of γ-weak learnable concepts: any dataset made of distinct points in $\mathbb{R}$: ex. 72
- *voting margins* (definition, characterization, and a property): ex. 27, 69
  a sufficient condition for γ-weak learnability based on voting margins: ex. 28
- AdaBoost can sometimes lead to overfitting: ex. 73
- using AdaBoost for feature selection: ex. 70
- an upper bound for the generalization error produced by AdaBoost: ex. 30
- AdaBoost using confidence-rated "weak" classifiers: ex. 74
- AdaBoost can be seen an instance of a more general algorithm: ex. 75
- *generalizing* the AdaBoost algorithm with different *loss* / cost *functions*: ex. 29, 76
- SAMME: AdaBoost multi-class: ex. 77

Implementation exercises:

Stanford, 2016 fall, A. Ng, J. Duchi, HW2, pr. 6.d[24]
- Complete a Matlab implementation of AdaBoost and apply it on a real *dataset* related to the Higgs boson.
- Plot on the same graph the *training* and the *test errors* as a function of the number of iterations when working with the *best* and respectively *random* decision stumps.

MIT, 2001 fall, Tommi Jaakkola, HW3, pr. 1.4[25]
- Complete a Matlab implementation of AdaBoost and apply it on a handwritten digit *dataset*.
- Plot on the same graph the *training* and the *test errors* as a function of the number of iterations, as well as the *weighted training error* of the selected decision stumps.
- Plot the *empirical cumulative distribution* of the *voting margins* $y_i f_T(x_i)$ after 4 and 16 iterations respectively. Explain the differences.

[24] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/Stanford.2016f.ANg+JDuchi.HW2.pr6.AdaBoost+HighEnergyPhysics.sol.data.Matlab-code/

[25] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/MIT.2001f.TJaakkola.HW3.AdaBoost.confidence-rated-classifiers.digit-DATA.Matlab-code.sol/

CMU, 2007 spring, Carlos Guestrin, HW2, pr. 2.3[26]

• Use a Matlab implementation of AdaBoost and apply it on a real *dataset*: Bupa Liver Disorder.
• Perform 10-fold cross-validation.
• Identify the best 10 features / decision stumps.
• Plot the *empirical cumulative distribution* of the *voting margins* $y_i f_T(x_i)$ after 10, 50 and 100 iterations respectively.

MIT, 2009 fall, Tommi Jaakkola, HW3, pr. 2.4[27]

• Study [with the help of some plots] the *evolution of voting margins* during the execution of the AdaBoost algorithm on a 2-dimensional dataset.
• Understand the *link* between the evolution of voting margins and the *sequential minimization* of function representing an *upper bound* to the training error rate achieved by the *combined hypothesis* calculated by AdaBoost.

[26] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2007s.CGuestrin.HW3.pr2.3.AdaBoost.BUPA-liver-Disorder.data/

[27] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/MIT.2009f.TJaakkola.HW3.AdaBoost.voting-margins.data.Matlab-code.sol/

Weeks 12-14: Hierarchical and Partitional Clustering

Week 12: Hierarchical Clustering:

Read: Chapter 14 from Manning and Schütze' *Foundations of Statistical Natural Language Processing* book.

Ciortuz et al.'s exercise book, ch. *Clustering*, ex. 1-6, 27-35 [36-39]

Implementation exercises:

◦ CMU, 2010 fall, Ziv Bar-Joseph, HW4, pr. 3.1-3[28]
Hierarchical clustering applied on the *yeast gene expression* dataset

Weeks 13-14: Partitional Clustering: **The k-Means Algorithm**

See section 2.2 in the *Overview* of the *Clustering* chapter in Ciortuz et al.'s exercise book.

- application: ex. 7-11, 15.a, 17.a, 21.a, 22.a, 40-41;
- properties (convergence, optimality and other issues): ex. 12-13, 42-47;
- k-Means for image compression: ex. 48;
- using another distance metric (than the Euclidian one): ex. 49;
- k-Means++: ex. 50;
- comparison with the hierarchical clustering algorithms: ex. 14, 52;
- comparison with classification algorithms: ex. 53;
- implementation: ex. 55;
- revision: ex. 54.

Advanced issues:

- a "kernelized" version of k-Means: ex. 51.

Implementation exercises:

0. CMU, 2012 fall, E. Xing, A. Singh, HW3, pr. 1[29]
implement K-means, apply it on data from $\mathbb{R}^2$;
the importance of choosing the initial centroids: the K-means++ algorithm;
a good heuristics for choosing the right value of K;
the importance of scaling the data on different dimensions.[30]

1. Stanford, 2012 spring, Andrew Ng, HW9[31]
Implement K-means in Matlab and apply it to image compression.

[28] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2010f.ZBJ.HW4.hierarchicalClustering+K-means.geneExpression.data.Matlab-code/

[29] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2012f.EXing+ASingh.HW3.pr1.K-means.HOW-TO-select-K-and-initial-centroids.the-importance-of-scaling-the-data.DATA.R-code/

[30] Another exercise — CMU, 2011 fall, Eric Xing, HW2, pr. 4.2-3 — explores the first 2 of these strategies on a *face recognition* dataset. See
https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2011f.EXing.HW2.pr4.imageClustering.DATAwithoutCode/

[31] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/Stanford.2012s.AndrewNg.HW9.K-means.imageCompression.DATA+CODE/

2. CMU, 2010 fall, Ziv Bar-Joseph, HW4, pr. 3.4-7[32]
k-means applied on the *yeast gene expression* dataset

Weeks 15-18: [final] EXAM

[32] https://edu.info.uaic.ro/invatare-automata/ML.ex-book/implementation-exercises/CMU.2010f.ZBJ.HW4.hierarchicalClustering+K-means.geneExpression.data.Matlab-code/

Tabel centralizator

săp- tă- mâna	exerciţii prezentate (ca exemple) la curs	exerciţii rezolvate (sau date ca temă) la seminar (grupa A5)	exerciţii prezentate ca demonstraţii la „Seminarul Special“
1	0.55 (only the definitions), 0.56, 4.34 4.2 slides #32-33, 35 in foundations.pdf slide #8,10-13 in ml3.pdf slides #2-4, 12-22 in DT.pdf	0.89-93, 0.96-99 slides #4-9 in foundations.pdf	slides #4-7 in foundations.pdf
2	0.55 4.33, 4.1, 4.3-4, 4.36 prop. (P0)-(P4), (P11) pages 486, 488 slide #10 in ml0.pdf, slide #10 in ml3.pdf	0.140-142 4.32, 4.37-38, 4.40-41	0.99 slides #8-10 and 14-17 in foundations.pdf 0.4, 0.9-11, 0.23.bcd
3	4.5-7, 4.13-15 (only the ideas) 4.8, 4.12, 4.48, 3.11.b prop. (P5)-(P9) page 488 slides #20-24 in ml3.pdf	0.78 (only the definitions) 4.39, 4.42-43, 4.44 [4.37.ac, 4.38.abcd, 4.40, 4.41 using 4.36]	0.79 0.144-145 0.58
4	4.10, prop. (P10) page 488 slides #26-31 in ml3.pdf 4.46, 4.19 (only the ideas) 2.3, 2.8, slides #4-6, 12-14 in ml6.pdf slides #90, 93 in Bayes.pdf	4.44, 4.46, 4.51-52, 4.54.abc completare ex. 4.19-20 3.11.b	0.23.a 0.100, 0.19 0.143
5	prop. (P0)-(P7) pages 355-356 2.4.cd, 2.5-7, 2.11, 2.33	4.49, 4.50, 4.54.d, 4.56 0.94-95, 0.103 2.27-31	0.146 0.16, 0.101

Notă explicativă:

Cifra [urmată de punct] scrisă în faţa numărului unui exerciţiu corespunde capitolului din culegere din care face parte exerciţiul respectiv. Spre exemplu, am desemnat prin 0.10 exerciţiul 10 de la capitolul 0 (*Fundamente*) şi prin 4.8 exerciţiul 8 de la capitolul 4 (*Arbori de decizie*).