

ML course, 2023 fall

What you should know:

Week 1:

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

(slides 1-10 from <https://profs.info.uaic.ro/~ciortuz/SLIDES/ml0.pdf>)

PART II (seminary): Revision: Basic issues in Probabilities¹

(slides 4-17 from <https://profs.info.uaic.ro/~ciortuz/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 (2023f) ch. *Foundations*, ex. 1-5 [6-7] 8, 89-93 [94-95] 96

¹Professors / teaching assistants who are in charge with the seminars 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.

²For a more concise / formal introductory text, see *Probability Theory Review for Machine Learning*, Samuel Jeong, 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 *linearity* 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, categorical 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, 121

Advanced issues (III):

Concepts/definitions:

- vector of random variables; covariance matrix for a vector of random variables
- positive [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, 134-135;

the central limit theorem (the i.i.d. case) and the law of large numbers: ex. 40.

Week 2.¹/₂: 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://profs.info.uaic.ro/~ciortuz/SLIDES/foundations.pdf>)

Theoretical results/formulas:

- $0 \leq H(X) \leq H(\underbrace{1/n, 1/n, \dots, 1/n}_{n \text{ times}}) = \log_2 n$

Concepts/definitions:

- entropy;
specific conditional entropy;
average conditional entropy;
information gain (mutual information)
joint entropy;
- $IG(X; Y) \stackrel{\text{def.}}{=} H(X) - H(X|Y) = H(Y) - H(Y|X)$
- $IG(X; Y) \geq 0$
- $IG(X; Y) = 0$ 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, $H(X_1, \dots, X_n) = H(X_1) + H(X_2 | X_1) + \dots + H(X_n | X_1, \dots, X_{n-1})$)
- $H(X, Y) = H(X) + H(Y)$ 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] 138-140;
- proof of some basic properties:
Ciortuz et al.'s exercise book: ch. *Foundations*, ex. [62] 141-144, 149.

Advanced issues:

- relative entropy (Kullback-Leibler divergence)
Ciortuz et al.'s exercise book: ch. *Foundations*, ex. 63, 146-147;
- cross-entropy
Ciortuz et al.'s exercise book: ch. *Foundations*, ex. 64-65, 145.

Weeks 2.²/₂, 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 2.²/₂:

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, 32-45 [57]

- **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

greedy: ex. 4, 21.a, 40

search algorithm (1-step look-ahead): ex. 3, 39

- **properties of ID3 trees:** ex. 2-4, 7-9, 21.a, 33, 39-40 [44] 55-56

- **implementation exercises:** ex. 35 [57]

- **revision:** ex. 21, 55-56

Important Note:

Some of the exercises listed above would be done in class (i.e., at seminars) in an easier / nicer way if students would priorly do at home the exercise 34, which asks for the **implementation** of the **information gain** (and also entropy, specific conditional entropy and average conditional entropy), starting from the counts (more precisely, from the data partitions) associated to the leaf nodes of a **decision stump**.³ Alternatively, the exercise 35 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, HW3⁴

- 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

³This implementation could be later extended to an implementation of ID3 algorithm (the basic form); see ex. 53.

⁴<https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/CMU.2012s.RR.HW3.DT+pruning.C-code.PlayTennis+mushrooms/>

⁵CMU, 2011 spring, T. Mitchell, A. Singh, HW1, pr. 3 — <https://profs.info.uaic.ro/~ciortuz/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, HW3⁶ – a similar problem to the above one — uses a *chess* dataset generated by Alen Shapiro (see *Structured Induction in Expert Systems*, 1983, 1987).

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 [46] 47-49 [51-52] 53-56

- **extensions of the ID3 algorithm**

- handling of continuous attributes: ex. 10-12, 46-49

- decision surfaces, decision boundaries: ex. 10, 47, and ch. *Instance-based learning*, ex. 11.b

- **other extensions to the ID3 algorithm**

- handling of attributes with many values: ex. 13, 51

- 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, 44-45, 47, 49.d

- ID3 as “eager” learner: ex. 16

- ID3 and [non-]robustness to noises, and *overfitting*: ex. 10, 21.bc, 48

- *pruning* strategies for decision trees: ex. 19 [20] [52] 53 [54]

Implementation exercises:

CMU, 2011 fall, T. Mitchell, A. Singh, HW1, pr. 2⁷

- 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*.

⁶<https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/CMU.2011s.RoniRosenfeld.HW3.DT.C-code.chess/>

⁷<https://profs.info.uaic.ro/~ciortuz/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://profs.info.uaic.ro/~ciortuz/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 [98-99] 100-103;
- 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, 41;
- pseudo-code: ML book, page 177, and slides #12-14 in
<https://profs.info.uaic.ro/~ciortuz/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. 39-40;
- revision: ex. 42.

Advanced issues (I):

- an information-theoretic view on Naive Bayes algorithm [see the link with the equivalence relation $\max \text{likelihood} \Leftrightarrow \min \text{cross-entropy}$]: ex. 147 (and ex. 146) from the *Foundations* chapter.

Advanced issues (II): Gaussian Bayesian classification¹⁰

- applications: the *uni-variate* case: ex. 15-16, 43-44, 52;
the *bi-variate* case: GNB: ex. 42.c; GJB: ex. 20-21, 49.ef, 50-51;
- the nature of the *decision boundary*): GNB: ex. 17, 47; GJB: ex. 18-19, 49.ab;
- the relationship between Gaussian Bayesian classification and logistic regression: ex. 22, 23;
- the number of parameters to be estimated: GNB: ex. 45.b, GJB: ex. 45.d;
- GNB: the link between the estimation of parameters and the maximization of the log-verosimilarity of the data: ex. 46.

⁸This chapter is equally the source for all exercises listed below.

⁹The number of parameters indicated in parantheses refer to the case when both the input attributes and the output attribute are Bernoulli.

¹⁰I.e., Bayesian classifiers with continuous [numerical] input attributes — the case of Gaussian attributes.

Implementation exercises:

0. CMU, 2010 fall, Ziv Bar-Joseph, HW1, pr. 4¹¹

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. 6¹²

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. 3¹³

Implement the Naive Bayes classification algorithm and perform n -ary classification on the *20 Newsgroups* dataset;

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://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/CMU.2010f.ZBJ.HW1.pr4.NBayes.weatherPrediction.data/>

¹²<https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/Stanford.2012.OpenCourse.ANg.Ex6.NaiveBayes.spam-filtering.data.Matlab-code.sol/>

¹³<https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/CMU.2011s.TM.HW2.pr3.NB+featSelection.20newsgroups.DATA+CODE+sol/>

¹⁴An earlier exercise, CMU, 2009 spring, T. Mitchell, HW3, pr. 2 — <https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/CMU.2009s.TM.HW3.pr2.textClassif.NB+LR.hockey-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: hockey vs. baseball newsgroups. A similar exercise — CMU, 2014 fall, W. Cohen and Z. Bar-Joseph, HW2, pr. 6, <https://profs.info.uaic.ro/~ciortuz/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: 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. 5¹⁵

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. 11¹⁶

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. 3.2-8¹⁷

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-validation 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://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/CMU.s.10-701.HW1.pr5.kNN-vs-GNB.data/>

¹⁶<https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/MPI.Rahnenfuhrer+Alexa.2005.HW4.BreastCancer+kNN/>

¹⁷<https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/CMU.2004f.CGuestrin.HW4.pr3.2-8.k-NN.hand-written-char-reco.data/>

¹⁸<https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/CMU.2005s.CGuestrin+TMitchell.HW3.pr3.textClassif.data+code/>

¹⁹For the cosine distance, see http://en.wikipedia.org/wiki/Cosine_similarity.

²⁰Matlab scripts for doing training and testing with SVM are provided.

3. CMU, 2010 spring, Eric Xing, Tom Mitchell, Aarti Singh, HW2, pr. 2.1²¹
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

Week 8: midterm

²¹<https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/CMU.2010s.EX+TM+AS.HW2.pr2.multinomial-logistic-regression.ORL-Faces-dataset.NO-code/>

²²Advanced issues: Another exercise — CMU, 2011 fall, Eric Xing, HW5, pr. 1.1-2, <https://profs.info.uaic.ro/~ciortuz/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.

Weeks 9-10: The AdaBoost Algorithm

- pseudocode + intro. to theoretical foundations:
Ciortuz et al.'s exercise book, ch. *Decision trees*, ex. 22, 23.abc;
applications: ex. 24, 25.a, 58-64;
- AdaBoost as an optimisation algorithm: ex. 23.d, 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, 66;
exemplifying situations when there is no *guarantee* for γ -weak learnability: ex. 25.b, 65
- AdaBoost can sometimes lead to overfitting: ex. 71
- using AdaBoost for feature selection: ex. 68
- a class of concepts which can be represented using linear combinations of decision stumps: ex. 69
- revision: ex. 31, 76

Advanced issues:

- *voting margins* (definition, characterization, and a property): ex. 27, 67
a sufficient condition for γ -weak learnability based on voting margins: ex. 28
- a class of γ -weak learnable concepts: any dataset made of distinct points in $\mathbb{R}$: ex. 70
- an upper bound for the generalization error produced by AdaBoost: ex. 30
- AdaBoost using confidence-rated “weak” classifiers: ex. 72
- AdaBoost can be seen an instance of a more general algorithm: ex. 73
- *generalizing* the AdaBoost algorithm with different *loss* / *cost functions*: ex. 29, 74
- SAMME: AdaBoost multi-class: ex. 75

Implementation exercises:

Stanford, 2016 fall, A. Ng, J. Duchi, HW2, pr. 6.d²³

- 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. 1.4²⁴

- 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.

²³<https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/Stanford.2016f.ANg+JDuchi.HW2.pr6.AdaBoost+HighEnergyPhysics.sol.data.Matlab-code/>

²⁴<https://profs.info.uaic.ro/~ciortuz/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. 2.3²⁵

- 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. 2.4²⁶

- 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.

²⁵<https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/CMU.2007s.CGuestrin.HW3.pr2.3.AdaBoost.BUPA-liver-Disorder.data/>

²⁶<https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/MIT.2009f.TJaakkola.HW3.AdaBoost.voting-margins.data.Matlab-code.sol/>

Weeks 11-12: Hierarchical and Partitional Clustering

Week 11 $\frac{1}{2}$: 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. 3.1-3²⁷

Hierarchical clustering applied on the *yeast gene expression* dataset

Weeks 11 $\frac{2}{2}$ -12: 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;

a “kernelized” version of k -Means: ex. 51;

comparison with the hierarchical clustering algorithms: ex. 14, 52;

comparison with classification algorithms: ex. 53;

implementation: ex. 55;

revision: ex. 54.

Implementation exercises:

0. CMU, 2012 fall, E. Xing, A. Singh, HW3, pr. 1²⁸

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.²⁹

1. Stanford, 2012 spring, Andrew Ng, HW9³⁰

Implement K -means in Matlab and apply it to image compression.

2. CMU, 2010 fall, Ziv Bar-Joseph, HW4, pr. 3.4-7³¹

k -means applied on the *yeast gene expression* dataset

²⁷<https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/CMU.2010f.ZBJ.HW4.hierarchicalClustering+K-means.geneExpression.data.Matlab-code/>

²⁸<https://profs.info.uaic.ro/~ciortuz/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/>

²⁹Another 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://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/CMU.2011f.EXing.HW2.pr4.imageClustering.DATAwithoutCode/>

³⁰<https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/Stanford.2012s.AndrewNg.HW9.K-means.imageCompression.DATA+CODE/>

³¹<https://profs.info.uaic.ro/~ciortuz/ML.ex-book/implementation-exercises/CMU.2010f.ZBJ.HW4.hierarchicalClustering+K-means.geneExpression.data.Matlab-code/>

Weeks 13-14: Logistic Regression.

Prerequisites:

– The sigmoid / logistic function: :

Ciortuz et al.’s exercise book, ch. *Regression Methods*, ex. 13.

– The **likelihood function, maximum likelihood estimation (MLE)**:

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

– The **Bernoulli distribution, MLE**:

Ciortuz et al.’s exercise book, ch. *Foundations*, ex. 42, 43.A, 123, 124.a, 125.a, 126.a.

– **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, 127, 162, 163.ab, 164.A, 165.

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.

• **Ciortuz et al.’s exercise book**, ch. *Regression Methods*, ex. 13, 14, 18, 32.a-d, 33, 36.ab, 37.bc.

new exercise: Logistic Regression applied on the toy mushrooms dataset (ex. 2 from the *Decision Trees* chapter): https://piazza.com/class_profile/get_resource/lmux9ehsirmzw/lr968uwynnx3ms

Advanced issues:

- logistic regression is the „probabilistic face“ of the sigmoidal perceptron;
- coping with overfitting in logistic regression: the *regularization* of the w parameter;
- logistic regression is the *discriminative* correspondent of the *generative* Naive Bayes classifier;
- unlike, Naive Bayes, logistic regression is not affected by the *duplication of attributes*;
- logistic regression and Naive Bayes have the same (linear!) type of decision boundary;
- logistic regression can be *kernelized*;
- logistic regression can be generalized to *multinomial classification*: the softmax regression.

Weeks 15-18: [final] EXAM

Tabel centralizator

săptămăna	exerciții prezentate (ca exemple) la curs	exerciții rezolvate (sau date ca temă) la seminar (grupa A5)	exerciții prezentate ca demonstrații la „Seminarul Special“
1	4.2 slides #8-9, 32-33 in foundations.pdf slide #8,11-13 in ml3.pdf slides #12-22 in ML.ex-book.SLIDES.DT.pdf	4.2, 2.8 slide #14 in ml6.pdf slides #5-7 in foundations.pdf slide #9 in ml0.pdf	slides #4-7 in foundations.pdf 0.1.ab
2	[0.56] 0.55 (only the definitions), 4.36 4.3, 4.4, 4.33, [4.5] 4.8, 4.6 prop. (P0)-P(4), (P9), pag. 462, 464	0.89-93, 0.96-99, 0.138-140 4.38	slides #8-10, 14-17 in foundations.pdf 0.55.a
3	4.7, 4.12, 4.47, 3.11.b, 4.10 prop. (P5), (P7)-(P8), pag. 464 slide #26-27 in ml3.pdf	4.32-34, 4.37, 4.39-41, 4.43 4.45. 4.55-56.bc 0.111	4.36, 0.101, 0.58
4	4.15, 0.139.ab, 4.19 slides #26-31 in ml3.pdf 2.8, slides #4-6, 12-15 in ml6.pdf	4.38, 4.44, 4.48-51, 3.11.b	0.19, 0.23.c [0.38] 0.9-11, 0.99 [0.100]
5	2.3, 2.6-7, 2.10-11 prop. (P0)-(P4), (P6'), pag. 339-340	4.45, 4.49, 4.52-54 0.94-95 2.25, 2.27, 2.29-31	0.55, 0.79, 0.141-142, 0.144, 0.59
6	2.35.d, 2.36 0.42, 0.43.abc, 2.38 prop. (P5)-(P6), (P8), pag. 340 3.1, 3.3-4, 3.16 3.7, 3.11 la nivel de idee slide #4 in ml8.pdf prop. (P0)-(P2), (P5), pag. 419	2.32, 2.34, 2.36-37, 2.39-42	0.62, 4.49 0.82-83 la nivel de idee
7	2.9, 2.35, [3.25] 4.22, 4.24	0.123, 0.125 3.17-24, 3.28.cde	0.63-65
8	Partial examination (1) 21.11.2023, 16:00, and respectively 18:00	—	—
9	—	—	examen suplimentar
10	4.23-24 4.25, 4.70 la nivel de idee prop. (P0)-(P9), (P11), pag. 466-467 slides #6, 10-12, 15-16 in cluster.pdf 7.1-2, 7.5 prop. (P0)-(P2), pag. 809	4.58-64, 4.76	6.13, 6.15.a

săptămăna	exerciții prezentate (ca exemple) la curs	exerciții rezolvate (sau date ca temă) la seminar (grupa A5)	exerciții prezentate ca demonstrații la „Seminarul Special“
11	7.3-4, 7.27, 7.35, 7.33-34, 7.6 prop. (P3)-(P5), pag. 810 slides #13, 23 in cluster.pdf 7.7.a, 7.10 prop. (P0)-(P1''), (P4), pag. 811	4.65-66, 4.67-69, 4.71 7.28-32, 7.35, 7.37	0.42, 0.43 2.38 1.13.a
12	7.7.b, 7.9, 7.11 7.12-13, 7.17.a, 7.42.a, 7.44 7.45 Companionul Practic, pr. 61, 62 prop. (P2)-(P3), (P5)-(P10), pag. 811	7.46, 7.49-50, 7.52	1.13.bc, 1.14, 0.80
13	0.42-43.A, 0.80, 1.32.a 1.13.ab ml-ex.pdf de pe site-ul Piazza	7.41-44, 7.47-48, 7.53-54	0.60.a, 0.31.a, 0.61.a 0.55.bc, 0.143
14	1.13.c, 1.14 0.164.A la nivel de idei: 1.33, 1.18, 0.64, 2.14	1.32.a-d, 1.36.ab, 1.37.bc 0.127 0.162, 0.163.ab, 0.164.A, 0.165	4.17-18 0.146-147

Notă explicativă:

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

Errata la
Exerciții de învățare automată, versiunea 2023f

1. *Cuvânt înainte*, pag. 10, rândul 2 de jos:
vor fi $\rightarrow$ va fi
(std. Ștefan Catîru)
2. *Cuvânt înainte*, pag. 12, rândul 12 de jos:
exercițiile $\rightarrow$ exercițiile
(std. Alina Duca)
3. *Cuvânt înainte*, pag. 12, rândul 15 de jos:
cunoștințele $\rightarrow$ cunoștințele
(std. Alina Duca)
4. *Cuvânt înainte*, pag. 12, rândul 20 de jos:
acestei acestei $\rightarrow$ acestei
(std. Ștefan Catîru)
5. Cap. *Fundamente*, ex. 16, pag. 41, rândul 15 de sus:
supra-rescripționat $\rightarrow$ supra-restricționat
(std. Alina Duca)
6. Cap. *Fundamente*, ex. 21, pag. 48, rândul 11 de sus:
minimizare $\rightarrow$ minizare
(std. Alina Duca)
7. Cap. *Fundamente*, ex. 31, pag. 67, rândurile 4 și 5 de jos:
 $p(x|\theta) > 0$ pentru orice x , fiindcă $\theta > 0$ și $e^{-\theta x}$ pentru orice x
 $\rightarrow p(x|\theta) \geq 0$ pentru orice x , fiindcă $\theta > 0$ și $e^{-\theta x} > 0$ pentru orice x
8. Cap. *Fundamente*, ex. 44, pag. 100, rândul 1 de sus:
cerrințelor $\rightarrow$ cerințelor
9. Cap. *Fundamente*, ex. 44, pag. 101, rândul 10 de sus:
aceasta $\rightarrow$ această
10. Cap. *Fundamente*, ex. 46, pag. 107, rândul 3 de jos:
din în enunț $\rightarrow$ din enunț
11. Cap. *Fundamente*, ex. 80, pag. 178, rândurile 15-16 de sus:
ierativă $\rightarrow$ iterativă
12. Cap. *Fundamente*, ex. 55, pag. 129, rândul 2 de jos:
evenual $\rightarrow$ eventual
13. Cap. *Fundamente*, ex. 75, pag. 166, rândul 4 de jos:
 $\sum_{i=1}^{\infty} \rightarrow \sum_{i=0}^{\infty}$
14. Cap. *Fundamente*, ex. 79, pag. 175, rândurile 2-3 de jos:
În fine, dacă se renunță la ambele condiții, suma se numește *combinație afină*.
 $\rightarrow$
Când se folosește doar cea de-a doua condiție ($\sum_i a_i = 1$), suma se numește *combinație afină*.
În fine, dacă se renunță la ambele condiții, suma se numește *combinație liniară*.

15. Cap. *Fundamente*, ex. 109, pag. 220, rândul 4 de sus:
oarice $\rightarrow$ orice
16. Cap. *Fundamente*, ex. 109, pag. 235, rândul 12 de sus:
 $\text{Gamma}(r, \beta) \rightarrow \text{Gamma}(r, \beta)$
17. Cap. *Fundamente*, ex. 139, pag. 238, rândul 3 de jos:
indepdendentă $\rightarrow$ independența
(Sebastian Ciobanu)
18. Cap. *Metode de regresie*, ex. 14, pag. 302, rândul 10 de jos:
 $\ln \prod_{i=1}^n P(Y = y^{(i)} | X = x^{(i)}) = \sum_{i=1}^n \ln P(Y = y^{(i)} | X = x^{(i)})$
 $\rightarrow \ln \prod_{i=1}^n P(Y = y^{(i)} | X = x^{(i)}) = \sum_{i=1}^n \ln P(Y = y^{(i)} | X = x^{(i)})$
19. Cap. *Clasificare bayesiană*, *Sumar*, pag. 339, rândul 11 de jos:
 $\mathcal{O}(dn) \rightarrow \mathcal{O}(dK)$
20. Cap. *Clasificare bayesiană*, *Sumar*, pag. 339, rândul 8 de jos:
 $\mathcal{O}(d') \rightarrow \mathcal{O}(dK)$
21. Cap. *Clasificare bayesiană*, *Sumar*, pag. 340, rândul 2 de sus:
 d' este numărul de atribute de intrare din instanța de test $\rightarrow K$ este numărul de de clase
22. Cap. *Clasificare bayesiană*, *Sumar*, pag. 340, rândul 21 de jos:
Foundamente $\rightarrow$ Fundamente
23. Cap. *Clasificare bayesiană*, ex. 11, pag. 367, rândul 17 de sus:
în 50% din cazuri $\rightarrow$ cu o probabilitate de 0.5
24. Cap. *Clasificare bayesiană*, ex. 25, pag. 398, rândul 11 de sus:
înțelesă $\rightarrow$ înțealessă
(Sebastian Ciobanu)
25. Cap. *Clasificare bayesiană*, ex. 25, pag. 398, rândul 23 de sus:
 $P(X = x | Y = y) \rightarrow P(Y = y | X = x)$
(Sebastian Ciobanu)
26. Cap. *Clasificare bayesiană*, ex. 39, pag. 408, rândul 10 de sus:
decît $\rightarrow$ decât
(std. Alexandra Pal)
27. Cap. *Învățare bazată pe memorare*, ex. 23, pag. 456, rândul 3 de jos:
Distanță $\rightarrow$ Distanța
28. Cap. *Învățare bazată pe memorare*, ex. 7, pag. 432, rândul 11 de sus:
pentru orice două puncte A și B din $\mathbb{R}^2$
 $\rightarrow$ pentru orice două puncte A și B din $\mathbb{R}^2$ care au aceeași *ordonată*
29. Cap. *Învățare bazată pe memorare*, ex. 23, pag. 457, rândurile 6-7 de jos:
mai mult decât o singură instanță de antrenament $\rightarrow$ instanțe având etichete diferite
30. Cap. *Arbori de decizie*, ex. 12, pag. 493, rândul 15 de sus:
nivelul 0 $\rightarrow$ nivelul 1
31. Cap. *Arbori de decizie*, ex. 23, pag. 523, rândul 3 de jos:
 $\varepsilon_t \cdot e^\alpha - (1 - \varepsilon_t) \cdot e^{-\alpha} \rightarrow \varepsilon_t \cdot e^\alpha + (1 - \varepsilon_t) \cdot e^{-\alpha}$
(std. Gheorghică Ștefana)

32. Cap. *Arbori de decizie*, ex. 62, pag. 525, rândul 5 de jos:
pentru orice pentru orice $\rightarrow$ pentru orice
(std. Diana Carcea)
33. Cap. *Arbori de decizie*, ex. 62, pag. 590, rândul 15 de sus:
and $\rightarrow$ și
34. Cap. *Rețele neuronale artificiale*, ex. 15, pag. 756, rândul 12 de jos:
12 $\rightarrow$ 14
35. Cap. *Clusterizare*, ex. 42, pag. 905, rândul 3 de jos:
iterație respectivă $\rightarrow$ iterația respectivă
(std. Ștefan Catîru)