MA2823: Introduction to Machine Learning **Fall 2017**

`http://tinyurl.com/ma2823-2017`

Syllabus

1 Course description

Machine learning lies at the heart of data science. It is essentially the intersection between statistics and computation, though the principles of machine learning have been rediscovered from many different traditions, including artificial intelligence, Bayesian statistics, and frequentist statistics. In this course, we view machine learning as the automatic learning of a prediction function given a training sample of data (labeled or not). Machine learning methods form the foundation of many successful companies and technologies in multiple domains. Their applications, to name a few, include search engines, robotics, bioinformatics analyses of genetic data, algorithmic trading, social network analysis, targeted advertising, computer vision, or machine translation. This course gives an overview of the most important trends in machine learning, with a particular focus on statistical risk and its minimization with respect to a prediction function. A substantial lab section (in Python) will let students apply the course to real-world data. Throughout the course, students will participate in a data science competition.

Course objectives: By the end of the course, students should be able to

- Identify problems that can be solved by machine learning;
- Formulate these problems in machine learning terms;
- Identify and apply the most appropriate classical algorithm(s);
- Implement some of these algorithms themselves;
- Fairly evaluate and compare machine learning algorithms for a particular task.

Course narrative: We will start this course by an overview of what is machine learning and of some of the most comon machine learning problems. We'll see how to formulate these problems mathematically as optimization problems (Chap 1) and will set up the bases of convex optimization that will be needed throughout the course to understand the various algorithms (Chap 2). Although most of our course will focus on supervised learning problems (making predictions), we will start by manipulating data and their representation with an introduction to dimensionality reduction (Chap 3 + Lab 1). Because knowing various supervised machine learning algorithms will not take you very far in practice unless you also know how to evaluate a model on your data and how to choose among several possible models, we will then study model selection (Chap 4). And because you'll need to be able to implement optimization algorithms to fit machine learning models, we'll dedicate Lab 2 to the `scipy.optimize` library.

At this juncture, we'll finally be able to start on what likely motivated you to take this course in the first part: predicting things! We will devote a few weeks to parametric models (in which you decide beforehand of the formula determining the data distribution or decision function), starting with some elements of Bayesian decision theory (Chap 5) and moving on to linear models (Chap 6) and their regularized variants (Chap 7). You will also start manipulating classification and regression problems in practice, both on the data science challenge you will be evaluated on and on other data sets that will be presented to you in the labs where you'll implement and apply algorithms such as Naive Bayes (Lab 3), linear and logistic regression (Lab 4), lasso, or ridge regression (Lab 5).

We will move away from linear models by means of non-parametric models. We'll start from nearest-neighbors approaches (Chap 8 + Lab 6) and move on to tree-based and ensemble methods to introduce random forests, which are one of the most powerful supervised learning algorithms to date (Chap 9 + Lab 7). Another approach to solve non-linear problems is to map them to linear problems, something we will see in details with an introduction to support vector machines and kernel methods (Chap 10 + Lab 8). Finally, we will move back to fully parametric models with neural networks (Chap 11), and a tutorial introduction by your TA Joseph Boyd on the famed deep learning you keep reading about in the newspapers. You will also get a taste of current topics in deep learning research and biomedical applications with a research talk by PhD student Peter Naylor.

Finally, because sometimes the labels you'd need to apply supervised machine learning techniques are not available, we'll conclude this course with an overview of clustering techniques (Chap 12 + Lab 9).

Prerequisites This course assumes working knowledge of

- linear algebra (keywords: matrix inversion, spectral theorem, eigendecompositions). Refresher: http://ocw.mit.edu/courses/mathematics/18-06-linear-algebra-spring-2010/video-lectures/
- basic probabilities (keywords: random variable, probability distribution, Bayes theorem). Refresher: http://jeremykun.com/2013/01/04/probability-theory-a-primer/ or https://www.youtube.com/playlist?list=PL17567A1A3F5DB5E4
- Python/NumPy (for a refresher, head to http://scipy-lectures.github.io/).

2 Teaching team

Instructor Chloé-Agathe Azencott (chloe-agathe.azencott@mines-paristech.fr).

Email policy Although you are encouraged to make use of office hours, you are welcome to ask me questions by email. I will endeavor to answer your email within 2 working days.

Office hours My office is in Paris. You are welcome to schedule an appointment with me there by email. Please favor seeing me on the Gif-sur-Yvette campus on the Fridays when class meets, either between 1pm and 1:45pm (same room as where class meets) or during labs. You do not need to make an appointment for those.

Teaching assistants:

- Joseph Boyd joseph.boyd@mines-paristech.fr
- Benoît Playe benoit.playe@mines-paristech.fr
- Mihir Sahasrabudhe mihir.sahasrabudhe@centralesupelec.fr

3 Resources

Course webpage and other relevant URLs

- Course website: http://tinyurl.com/ma2823-2017
- GitHub page for labs: https://github.com/chagaz/ma2823_2017
- Turning in homeworks: http://tinyurl.com/ma2823-2017-hw

Textbooks The course material is (mostly) covered by the following textbooks, all available online. Pointers will be given in each lecture about which section(s) of which textbook(s) to refer to. Each of these textbooks go far beyond what will be covered in this course!

- *A Course in Machine Learning* by Hal Daumé III: http://ciml.info/dl/v0_99/ciml-v0_99-all.pdf
- *The Elements of Statistical Learning* by Trevor Hastie, Robert Tibshirani and Jerome Friedman: http://web.stanford.edu/~hastie/ElemStatLearn/
- *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond* by Bernhard Schölkopf and Alex Smola: http://agbs.kyb.tuebingen.mpg.de/lwk/
- *Convex Optimization* by Stephen Boyd and Lieven Vendenberghe: https://web.stanford.edu/~boyd/cvxbook/

Printable handouts A printable version of the slides will be posted to the course website the day before the course for those who wish to take notes on them.

4 Evaluation

Grades will be based on:

Homework assignments	10 pts
Project report	30 pts
Written exam (Dec. 22)	60 pts

Homework assignments: Homework will be assigned after each lecture but the last one and due one week later. Credit (1pt per assignment) will be given based on turning in a reasonable solution on time. A reasonable solution is one that shows you have attempted to solve the problem. Solutions will be provided on the course's website and will not be discussed in class due to time constraints, unless this proves necessary. Homework must be turned

in electronically at `http://tinyurl.com/ma2823-2017-hw`. Files should respect the following naming scheme: `HW<2-digits homework number>_<LastName>_<FirstName>.pdf`[1]. Please strip all accents from your name.

Project report: All details will be made available on Oct. 13.

Written exam: A written exam (pen and paper, closed book) will be given on Dec. 22. Exam problems will resemble homework assignments, and exams from previous years will be provided to you with solutions in due time.

5 Course schedule

Fr, Sep 29	13:45–15:15	Chap 1: **Introduction** *classification and regression, supervised and unsupervised learning, generalization, overfitting.*
	15:30–17:00	Chap 2: **Notions of convex optimization** *quadratic optimization, quadratic optimization with constraints, Lagrange multipliers, gradient descent.*
Mo, Oct 2	08:30–10:00	Chap 3: **Dimensionality reduction** *feature selection, wrapper approaches, principal component analysis.*
	10:15–11:45	Lab 1
Fr, Oct 6	13:45–15:15	Chap 4: **Model evaluation and selection** *training and test sets, cross-validation, bootstrap, measures of performance, measures of model complexity.*
	15:30–17:00	Lab 2
Fr, Oct 13	13:45–15:15	Chap 5: **Bayesian decision theory** *Bayes rule, losses and risks, Bayes risk, maximum a posteriori.*
	15:30–17:00	Lab 3
Fr, Oct 20	13:45–15:15	Chap 6: **Linear and logistic regressions** *parametric methods, maximum likelihood estimates, linear regression, logistic regression.*
	15:30–17:00	Lab 4
Fr, Nov 10	13:45–15:15	Chap 7: **Regularized linear regression** *Lasso, ridge regression, structured regularization.*
	15:30–17:00	Lab 5
Fr, Nov 17	13:45–15:15	Chap 8: **Nearest-neighbors methods** *non-parametric learning, k-nearest neighbors, instance-based learning, similarities, curse of dimensionality.*
	15:30–17:00	Lab 6
Fr, Nov 24	13:45–15:15	Chap 9: **Tree-based methods** *decision trees, ensemble methods, boosting, random forests.*
	15:30–17:00	Lab 7
Fr, Dec 1	13:45–15:15	Chap 10: **Support vector machines** *maximum margin, soft-margin SVM, non-linear data mapping, kernel trick, kernels.*
	15:30–17:00	Lab 8
Fr, Dec 8	13:45–15:15	Chap 11: **Neural networks** *perceptrons, multi-layer networks, backpropagation.*
	15:30–17:00	Tutorial: Deep learning by Joseph Boyd Research talk by Peter Naylor
Fr, Dec 15	13:45–15:15	Chap 12: **Clustering** *hierarchical clustering, k-means.*
	15:30–17:00	Lab 9
Fr, Dec 22	08:30–11:30	Written exam

6 Setting up your Python environment

You are required to bring your own laptops to the lab sessions. Power plugs and Internet access will be provided. You are welcome to work on the labs in pairs. **Please let me know if this poses any issue.** The labs require Python 2.7 (Python 3.3 is also fine, but you might need to edit a few commands) with a number of packages installed. The easiest way to install all these packages is to install **Anaconda:** `https://www.anaconda.com/download/`
If you'd rather install packages via pip, you'll need to install:

[1] For instance, my first homework would be called `HW01_Azencott_ChloeAgathe.pdf`.

— Jupyter `http://jupyter.org/install.html`

— matplotlib `http://matplotlib.org/users/installing.html`

— seaborn `http://seaborn.pydata.org/installing.html`

— numpy and scipy `https://www.scipy.org/install.html`

— pandas `http://pandas.pydata.org/getpandas.html`

— scikit-learn `http://scikit-learn.org/stable/install.html`

To test your installation, try running `jupyter notebook` from a terminal. In a Python shell or notebook, you should be able to run the following commands: `import sklearn`, `import pandas`, `import matplotlib`.

7 How to use GitHub

GitHub (`http://github.com/`) is a web-based repository hosting services, allowing for version control and source code management. GitHub is based on the git (`https://git-scm.com/`) version control system. A version control system allows you to manage automatically different versions and draft of a document; in essence, it is the grownup version of `lab1_final_v2.2_chloe-copy-1.ipynb`[2]. Git (and GitHub) are widely used in tech nowadays and our labs will give you an opportunity to learn how to use it.

GitHub offers both private and public repositories, and supports free accounts for academics (`https://education.github.com/discount_requests/new`). You'll find a short tutorial below:

— Log into GitHub (sign up on `https://github.com` if you do not have an account)

— Create a fork of the ma2823_2017 repository. A fork is a copy you own and can experiment with without changing the project. To do so: navigate to `https://github.com/chagaz/ma2823_2017` and click "Fork" in the upper right corner.

— Download and install git if it is not installed on your computer. To do so, follow the instructions at `https://git-scm.com/downloads`. If you do not know whether Git is installed on your computer, try typing "git" in a terminal. If it returns a help message, then git is installed.

— Set up git, following instructions at `https://help.github.com/articles/set-up-git`

— Clone your fork. This means you'll get a local version on your computer (for now your fork only exists on GitHub's servers):

  — On the GitHub website, navigate to your fork of the ma2823_2017 repository. Its URL should be something like `https://github.com/<yourusername>/ma2823_2017`.
  — Click "Clone or download" (on the top right)
  — Copy the URL that was just displayed (should be something like `https://github.com/<yourusername>/ma2823_2017.git`)
  — In the terminal (I'm assuming Linux/MacOS), navigate to where you want your copy to be. For example, if you want it under "Desktop > Centrale_2017 > Machine_Learning", type
  `cd Desktop/Centrale_2017/Machine_Learning/`
  — Then type "`git clone <the URL you just copied>`". You should see a message telling you the repository is downloading.

  For more on creating forks, see `https://help.github.com/articles/fork-a-repo/`.

— On your computer, edit the file you want to make changes to (for example, our first lab).

— Push your changes (i.e. send them from your computer to your GitHub account). To do this, from your ma2823_2017 repository, do:

```
git add <name of the file you edited>
git commit -m "<Short message explaining your modifications>"
git push
```

  If "`git push`" gives you an error message, try "`git push origin master`".

[2] Read more about the benefits of version control systems here: `https://www.git-tower.com/learn/git/ebook/en/desktop-gui/basics/why-use-version-control`