1 Overview

The course provides an extensive account of the fundamental ideas underlying machine learning and the basic algorithms used in practice. The course first formalizes basic concepts used to establish the theory and language of machine learning. These concepts include PAC learnability, sample complexity, and the VC dimension. The course then covers the concepts of convexity, regularization, and stability as well as important algorithmic paradigms including stochastic gradient descent, boosting, support vector machines, kernel methods, feature selection, and neural networks.

2 Basic Information

- **Professor Yaron Singer**
  Email: yaron@seas.harvard.edu
- **Zev Nicolai-Scanio**
  Email: znicolaiscanio@college.harvard.edu
- **Annie Zhu**
  Email: szhu@college.harvard.edu
- **Dim Karev**
  Email: d_karev@college.harvard.edu
- **Jonathan Chu**
  Email: jonathanchu@college.harvard.edu
- **Gal Kaplun**
  Email: galkaplun@g.harvard.edu

Lectures: Thursdays 3:00 – 6:00pm, **Maxwell Dworkin G125**.

Sections: Fridays 1:00pm – 2pm, **Maxwell Dworkin 123** (subject to change).

TF office hours (subject to change):

- Monday 8pm to 9pm Lowell dining hall (Dim)
- Tuesday 5pm to 6pm MD second floor lobby (Gal)
- Wednesday 8pm to 9pm Adams dining hall (Annie and Zev)

- Thursday 8pm to 9pm Currier dining hall (Jonathan)
- Thursday 6pm to 7pm MD 239 (Prof. Singer)

Course homepage: Most information about the course can be found on the course Canvas site.

3 Prerequisites

Basic knowledge in linear algebra and competency with calculus are required (e.g. Math 23a or Math 25a and Math 25b) as well as basic probability (Stat 110). AM 121 is certainly helpful but is not a necessary prerequisite. An appreciation for aesthetics, as well as prior coursework in algorithms (e.g. CS 124), machine learning (CS 181), and statistics will be helpful but not necessary. There will be programming exercises (CS 50 and comfort with programming in a scripting language should suffice).

Diagnostic quiz. We prepared a diagnostic quiz that will be available on the course Canvas page. **Submission is mandatory.** Any submission (even an empty one) will be marked as complete. We encourage you to complete all the questions on the quiz. This will allow you to (a) make an honest self-assessment that will help you determine whether you have the sufficient background to take the class and (b) refresh basic concepts that will be used in class. You should, of course, feel free to discuss the quiz with the TF's.

4 Logistics

Problem sets. There will be 6 bi-weekly problem sets, not including the diagnostic quiz. Problem sets will typically be released on Thursday evenings and will be due *the Thursday two weeks later* at 11:59:00 AM **sharp**. The last problem set will be due one week after its release. Solutions to problem sets should be submitted via Canvas. Submission deadlines are **S T R I C T**, and solutions submitted after 11:59:00 AM will receive no credit. Except for unusual circumstances we will not accept late submissions. We will drop the lowest score of your problem sets (thus you can drop one problem set and still earn a perfect score on the problem sets).

Programming. All problem sets will include programming assignments. You're expected to have taken CS 50 or have similar background and experience. You are expected to know how to code in Python. We will not teach Python or programming related material.

Homework Canvas submission. Solutions to all non-coding problems should be submitted in a PDF file uploaded to the corresponding assignment on Canvas. For the coding problem, unless specified otherwise, you should include a short 1-2 paragraph write-up describing your results in your PDF file, along with your code (in PDF format). While you are encouraged to use Latex to write your solutions, the staff will also accept PDF files containing scanned images of your write-ups. Submissions that are deemed illegible or unacceptably messy will not receive any credit.

Sections. We will have sections on a weekly basis taught by the teaching fellows. Sections will include exercises relevant to the problem sets, and may also introduce coding concepts that will play a role in the programming assignment for that week or extra concepts that are interesting to know but there is no time to cover in class. Sections are not mandatory, but we strongly encourage you to participate.

Exams. There will be one final exam for this class scheduled on **April 23rd** 3-6pm EST. This is the last day of the class and there is no exam during exam period.

Grading. The grade of the class is determined as follows: 25% by the final exam, 75% by the average grade of the problem sets.

5 Additional Resources

Most of the material of the course is covered in the following textbook:

UNDERSTANDING MACHINE LEARNING: FROM THEORY TO ALGORITHMS
by Shalev-Shwartz and Ben-David
also available for free online:
`https://www.cs.huji.ac.il/~shais/UnderstandingMachineLearning/understanding-machine-learning-t`
`pdf`

6 Course outline

Prelude

- The statistical learning framework
- Empirical risk minimization
- Overfitting
- Learning with inductive bias

PAC Learnability

- PAC learning
- Agnostic PAC learning
- Uniform convergence and concentration bounds

The VC Dimension

- The VC Dimension

- Sauer's Lemma

Learning with Convex Objectives

- Computational complexity of learning
- Basic elements of convex analysis
- The class of halfspaces
- Linear classification
- Perceptron and neural networks

Stochastic Gradient Descent

- Gradient descent
- Stochastic gradient descent
- Online learning

Regularization and Stability

- Regularized loss minimization
- Ridge regression
- Tikhonov Regularization

Support Vector Machines

- Margin and hard-SVM
- Soft-SVM

Kernel Methods

- Embeddings into feature spaces
- The kernel trick

Multiclass classification

- All pairs and one-vs-all
- Generalized hinge loss
- Ranking problems

Boosting

- Weak learnability
- Adaboost

Feature Selection and Dimensionality Reduction

- Pearson's correlation coefficient
- Forward Stepwise Regression
- LASSO
- Singular value decomposition (SVD)
- Principal component analysis (PCA)

Adversarial Machine Learning

- Adversarial examples
- Adversarial training
- Generative Adversarial Networks (GANs)