AM 221: Advanced Optimization	Syllabus
Prof. Yaron Singer	Spring 2019

1 Overview

This is a rigorous graduate-level course on optimization. The course covers convex and combinatorial optimization for solving large-scale problems. In recent years optimization has had a profound impact on machine learning, data analysis, mathematical finance, signal processing, control, theoretical computer science, and many other areas. The first part of the course will be dedicated to the theory of convex optimization and its direct applications. The second part will focus on advanced techniques in combinatorial optimization using machinery developed in the first part. Throughout the course we will see applications such as linear classification, LASSO, boosting, portfolio selection, online learning, neural networks, Support Vector Machines (SVMs), influence in networks, clustering, Principal Component Analysis (PCA) and dimensionality reduction techniques, feature selection, Generative Adversarial Networks (GANs), and adversarial attacks in machine learning.

2 Basic Information

- **Professor Yaron Singer**
  Email: yaron@seas.harvard.edu
  Office hours: Wednesdays 4:30 – 5:30pm, Maxwell Dworkin 239
- **Dimitris Kalimeris**
  Email: kalimeris@g.harvard.edu
- **Dor Verbin**
  Email: dorverbin@g.harvard.edu
- **Sharon Qian**
  Email: sharonqian@g.harvard.edu

TF office hours: Mondays and Tuesdays 4:30 – 5:30pm Maxwell Dworkin 2nd floor lobby.

Lectures: **Mondays and Wednesdays 3:00 – 4:15pm**, **Maxwell Dworkin G125**.

Sections: **Wednesdays 4:30 – 5:30pm**, **Maxwell Dworkin 123** (subject to change).

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). The course is intended for graduate students, but advanced undergraduates are encouraged to attend as well.

Diagnostic quiz. We prepared a diagnostic quiz that is available on the course Canvas page. **Submission is mandatory.** Any submission (even an empty one) will receive full credit. 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

Lecture notes. Lecture notes will be released after class and will summarize definitions and main ideas. In general, we strongly encourage you to attend lecture, as the lecture notes are not necessarily self-contained and do not aim to substitute for class attendance.

Problem sets. There will be 12 weekly problem sets, not including the diagnostic quiz. Problem sets will typically be released on Wednesday evenings and will be due *the following Wednesday* at 11:59:00 AM **sharp**. Solutions to problem sets should be submitted via Canvas. Solutions to problem sets will be posted shortly after the deadline. Submission deadlines are **strict**, and submissions submited after 11:59:00 AM will receive no credit. Except for unusual circumstances we will not accept late submissions. We will drop the lowest two scores of your problem sets (thus you can drop two problem sets and still earn a perfect score on the problem sets).

Programming. Most problem sets will include programming assignments. The programming will be relatively light, where the idea is to use simple scripts to analyze real-world data sets and apply some of the algorithmic ideas you learn in class. You're expected to have taken CS 50 or have similar background and experience. You are expected to know how to code in a script programming language like 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. Section times and locations will be determined after taking a vote during the first week of the course to determine the best time. We will aim to schedule sections to take place on Wednesday or Thursdays.

Exams. There will be no exams for this class but students have to complete a final project.

Project. The project will be done in pairs (i.e. *two* (2) people, not three, not one). The project can be data-oriented or theoretically-focused, or (better) a combination of both. There is room for projects involving algorithms, data mining, machine learning, game theory and mechanism design, and statistics. You are expected to submit a project proposal in February and the will be a poster presentation in the last day of class. The project involves three milestones:

- **Milestone 1:** Due Wednesday, February 27th, 11:59am;
- **Milestone 2:** Due Wednesday, April 3rd, 11:59am;
- **Milestone 3:** Due Wednesday, May 1st, 11:59am.

We give clear guidelines throughout the course regarding what each milestone should include.

Grading. The grade of the class is determined 50% by your grade in the problem sets and 50% by your grade in the final project.

5 Additional Resources

There will be comprehensive lecture notes that will be released after class. In addition, students may find it useful to use relevant textbooks.

Textbooks. Most of the material of the course is covered in the following textbooks:

- *CONVEX OPTIMIZATION*
  by Boyd and Vandenberghe
  also available for free online:
  `http://web.stanford.edu/~boyd/cvxbook/bv_cvxbook.pdf`
- *ONLINE CONVEX OPTIMIZATION*
  by Hazan
  also available for free online:
  `http://ocobook.cs.princeton.edu/OCObook.pdf`

- *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-learni`
  `pdf`
- *INTRODUCTION TO LINEAR OPTIMIZATION*
  by Bertsimas and Tsitsiklis

6 Course outline

Prelude

- Optimization as a foundation for data science
- Convex sets, convex functions
- Projection, separating hyperplanes, polyhedral sets

Linear Optimization

- Farkas' lemma
- Duality theory
- The Simplex method
- Ellipsoid, and interior point methods

Convex Optimization

- Gradient descent
- Projected gradient descent, Conditional gradient method
- Online convex optimization, stochastic gradient descent
- Newton's method and coordinate descent
- Karush-Kuhn-Tucker conditions
- Lagrangian duality
- Semi-definite programming

Combinatorial Optimization

- Computational complexity
- Approximation algorithms

- Submodular functions
- Local search and greedy algorithms
- Dynamic Programming
- Matroids, multilinear extensions, convex and concave closures
- Rounding techniques
- Continuous approximation algorithms