Introduction to Optimization Theory

MS&E 213 / CS 269O

Instructor: Aaron Sidford (sidford@stanford.edu / Huang 358)
Course Assistant: Xiaocheng Li (chengli1@stanford.edu)
Lectures: Monday / Wednesday - 1:30PM – 2:50PM in Building 380, Rm 380D
Office hours: TBD and by appointment (email me)
Website: Canvas (Discussions on Piazza)

Course Overview

This class will introduce the theoretical foundations of continuous optimization. Starting from first principles we show how to design and analyze simple iterative methods for efficiently solving broad classes of optimization problems. The focus of the course will be on achieving provable convergence rates for solving numerous complex large-scale problems.

While the field of optimization is vast, there is a much smaller set of techniques that appear repeatedly to achieve the best known theoretical guarantees, e.g. gradient descent, mirror descent, Newton's method, barrier methods, etc. Moreover, the efficacy of these techniques is well demonstrated by their performance on just a few prototypical optimization problems, e.g. smooth function minimization, convex function minimization, linear programming, etc. This class will walk through these classic results and more, providing a gateway to cutting edge research in the field. Emphasis will be on the following:

- **Continuous Optimization Algorithm Design**: the focus of this course is on designing and analyzing algorithms for solving broad classes of optimization problems. Typical lectures will either introduce a new algorithmic technique, analyze an old technique, or prove new properties of continuous optimization problems for later algorithmic use. By the end of the class you should be comfortable doing this on your own for new continuous optimization problems.

- **Iterative Methods**: the types of algorithms we consider in this course are primarily iterative. They consist of simple procedures for making local improvements that provably lead to global optimization guarantees. Over the past 5 years, advances in machine learning, data analysis, computer science, and operations research combined with rapidly growing data set sizes have made these methods incredibly popular in practice and well-studied in theory. By the end of the class you should feel comfortable further exploring this exciting research area.

- **Computational Complexity of Continuous Optimization**: our discussion of iterative methods will be centered on several prototypical optimization problems. Beyond designing fast algorithms we will probe the structure of these problems, proving lower bounds and limits of techniques for solving them. By the end of the class you should have a grasp of what to expect from optimization algorithms and what structure to look for to make optimization problems easier.

WARNING

This is primarily a theory class and our emphasis is on the design and provable analysis of algorithms for continuous optimization. While the class will (1) be an introduction to optimization, (2) analyze the structure of convex functions, and (3) provide a helpful lens on designing and applying optimization algorithms in practice, none of these are the focus of this course. We will provide little hands-on practical experience with optimization and while there will be some discussion of how to apply these procedures discussed in class in the real-world, this is not a central objective. If you are interested in any of these topics, there are other optimization courses at Stanford that target them more directly.

Nevertheless, the theory discussed in this class underlies recent progress in processing massive datasets and a firm grasp of the fundamentals taught in this class can be an invaluable tool for the increasingly complex task of optimizing in the real world.

Prerequisites

This class will assume that you understand linear algebra and multivariable calculus at the undergraduate level. We will re-introduce some mathematical concepts as needed, provide references when applicable, and are happy to help refresh material occasionally, but we will not teach these concepts in this class. If you have any questions about prerequisites or ever suspect that lectures are assuming too much prior knowledge, please do not hesitate to contact me. (sidford@stanford.edu).

Course Expectations

You are required to attend and participate in class, and complete problem sets and take-home exams.

- **Class Attendance**: I expect you to attend the lectures regularly. The class will move fairly quickly and each lecture will build upon previous ones. There will be lecture notes that contain a superset of the material, but the lectures will be the primary venue for disseminating content. Note that I will be regularly soliciting feedback to ensure the class is appropriately paced.

- **Class Participation**: I expect you to make efforts to participate during the lectures. Together we will motivate, derive, and analyze algorithms through class discussion. Your contributions are important, encouraged, and expected. Note that we will all likely make mathematical errors along the way and it is an essential part of the process to politely identify these and move forward together. Feedback is welcome and I look forward to working with you all.

- **Problem sets / take home exams**: I will assign problem sets regularly throughout the quarter and at least one take-home exam by the course’s termination. These are intended to give you hands-on experience, help you internalize the material, and push and test your understanding.

Grading: Grading will be based primarily on problem sets and take-home exams with in-class performance used to change grades on the boundary. Percentages on the problem sets and take-home exams will be designated based on their relative lengths and difficulties. Performance on problem sets will be the largest factor in determining your final grade.

More Course Expectations

I will pass out comment cards at the beginning of each lecture to get quick feedback on the course pace and ensure material is not lost. Please fill them out; it takes less than a minute and they are invaluable.

To listeners: You are welcome to get on Canvas (email me) and attend the lectures but please please try to attend regularly for the entirety of the course to ensure consistency of in-class discussion.

Course Material

There is no textbook or single pre-existing reference for the material of this course. Instead, I will provide lecture notes that coincide with the course material and give additional references to relevant literature. Feedback on the notes is welcome and highly encouraged. Please post comments, questions, and feedback on Canvas, or if something is sensitive, please email me (sidford@stanford.edu).

Tentative Schedule

- **Week 1: M 4/3**: Introduction – why optimization is hard (simple examples)
- **Week 1: W 4/5**: Introduction – why progress is easy (smoothness / gradient descent)

- **Week 2: M 4/10**: Smooth Convex Optimization – introduction to convex functions
- **Week 2: W 4/12**: Smooth Convex Optimization – gradient descent for convex functions

- **Week 3: M 4/17**: Smooth Convex Optimization – gradient descent for convex function
- **Week 3: W 4/19**: Smooth Convex Optimization – extensions (prox, coordinate, stochastic, etc.)

- **Week 4: M 4/24**: Nonsmooth Convex Optimization – subgradients and cutting plane methods
- **Week 4: W 4/26**: Nonsmooth Convex Optimization – subgradients and cutting plane methods

- **Week 5: M 5/1**: Nonsmooth Convex Optimization – mirror descent
- **Week 5: W 5/3**: Nonsmooth Convex Optimization – mirror descent / multiplicative weights

- **Week 6: M 5/8**: Nonsmooth Convex Optimization – online learning
- **Week 6: W 5/10**: Accelerated Gradient Descent

- **Week 7: M 5/15**: Accelerated Gradient Descent
- **Week 7: W 5/17**: Iterative Method Research

- **Week 8: M 5/22**: Newton's Method – non-convex convergence extension
- **Week 8: W 5/24**: Newton's Method – intro to self-concordance

- **Week 9: W 5/31**: Interior Point Methods - self-concordance (**note 5/29 is a holiday – no class**)

- **Week 10: M 6/5**: Interior Point Methods – intro to path finding (End-Quarter Period)
- **Week 10: W 6/7**: Linear Programming (End-Quarter Period)