Convex Optimization: Spring 2015

Machine Learning 10-725/Statistics 36-725

Instructor: Ryan Tibshirani, ryantibs@cmu.edu

Teaching assistants:
Mattia Ciollaro, ciollaro@cmu.edu
Junier Oliva, joliva.cs.cmu.edu
Nicole Rafidi, nrafidi@cs.cmu.edu
Veeranjaneyulu Sadhanala, vsadhana@cs.cmu.edu
Yu-Xiang Wang, yuxiangw@cs.cmu.edu

Lectures: Mondays and Wednesdays 10:30-11:50am, Hamburg Hall 1000

Overview and objectives

Nearly every problem in machine learning and statistics can be formulated in terms of the optimization of some function, possibly under some set of constraints. As we obviously cannot solve every problem in machine learning or statistics, this means that we cannot generically solve every optimization problem (at least not efficiently). Fortunately, many problems of interest in statistics and machine learning can be posed as optimization tasks that have special properties—such as convexity, smoothness, separability, sparsity, etc.—permitting standardized, efficient solution techniques.

This course is designed to give a graduate-level student a thorough grounding in these properties and their role in optimization, and a broad comprehension of algorithms tailored to exploit such properties. The main focus will be on convex optimization problems, though we will also discuss nonconvex problems at the end. We will visit and revisit important applications in statistics and machine learning. Upon completing the course, students should be able to approach an optimization problem (often derived from a statistics or machine learning context) and:

1. identify key properties such as convexity, smoothness, sparsity, etc., and/or possibly reformulate the problem so that it possesses such desirable properties;

2. select an algorithm for this optimization problem, with an understanding of the advantages and disadvantages of applying one method over another, given the problem and properties at hand;

3. implement this algorithm or use existing software to efficiently compute the solution.

Outline of material

Here is an rough outline of the course topics.

Core topics: Split into 4 parts, as follows.

- Theory I: Basics of convex analysis
- Algorithms I: First-order methods
- Theory II: Duality and the KKT conditions
- Algorithms II: Second-order methods

Advanced topics: We will choose a subset of the following, depending on time/interest.

- Dual methods and ADMM
- Coordinate-based methods
- Conditional gradient methods
- Proximal and projected Newton methods
- Exact path-following methods
- Screening rules
- Large-scale optimization

Logistics

Prerequisities: Students entering the class are expected to have a pre-existing working knowledge of basic algorithms and data structures; however, the class has been designed to allow students that do not have this background, but have strong analytical skills, to catch up and fully participate. Students are expected to be comfortable with rigorous mathematical arguments. Though not required, having taken 10-701 or an equivalent machine learning or statistics class will be very helpful, since we will frequently use applications in machine learning and statistics to demonstrate the concepts we learn in class.

Class website: The class website is `http://www.stat.cmu.edu/~ryantibs/convexopt/`. The class schedule, lecture notes, homeworks, etc., will be posted there.

Office hours: The weekly schedule for office hours can be found on the course website.

Discussion group: We will use Piazza for class discussions.

Textbook: Lectures are intended to be mostly self-contained. But a terrific reference for most of the material covered in class is *Convex Optimization* by Boyd and Vandenberghe, which is available online for free (`http://www.stanford.edu/~boyd/cvxbook/`). We will often draw examples or problems from this book. In addition, a wonderfully thorough mathematical reference for much of the underlying convex analysis we cover is *Convex Analysis* by Rockafellar.

Quizzes: There will be a short, easy quiz due at midnight on the day of each lecture, consisting of multiple choice or true/false questions. The quizzes will be taken online, and the links will be given on the course website.

Homework: There will be 6 homework assignments, approximately one every two weeks. The assignments will be posted on the course website, and your homeworks will be submitted according to the instructions given there. **Late homework will not be accepted.**

The homeworks are structured to give you experience in both written mathematical exercises and programming exercises. As we will often reuse problems from previous versions of the course, or problems covered by papers and webpages, we expect you not to copy, refer to, or look at solutions in preparing your answers.

Also, while it is completely acceptable for you to collaborate with other students in order to solve the problems, we assume that you will be taking **full responsibility in terms of writing up your own solutions and implementing your own code**. You must indicate on each homework the students with whom you collaborated.

Midterm: There will be one in-class midterm, scheduled for the end of March, with the precise date on the course website. This will cover about the first 2/3 of the course material (the core topics, before the advanced topics).

Little test: There will be one final little test, scheduled for the end of April, with the precise date on the course website. This will consist of multiple choice and true/false questions, and will cover everything learned in the course.

Class project: You have the option to complete a class project. If you choose to do the class project, you should enroll in the 12 unit option for the course; those not doing the project should enroll in the 9 unit option. Details on the project to come.

Scribing: Students can sign up for scribing lectures, with no more than three students signed up for a particular lecture. While scribing is not strictly required, it bump up your grade if you are on the border (e.g., B+ to an A-). The sign up sheet can be found on the course website.

Evaluation: The grading breakdown is as follows.

Homework	45%
Quizzes	5%
Midterm	15%
Little test	10%
Project	25%

(For those of you enrolled in the 9 unit option with no project, just subtract the 25% for the project and renormalize.)