Convex Math Programming: Optimization & Decomposition

MGMTPhD 291 - Spring 2016

Lectures:
Tues 9:00AM-11:50AM
Location: Paul Merage School of Business, Room SB2 111; see http://www.uci.edu/campusmaps.php

Instructor:
John Turner
Office: Paul Merage School of Business, Room SB2 338
Phone: 949-824-7941 E-mail: john.turner@uci.edu
Office Hours: Tues 4:30PM-5:30PM; other times by appointment

Course Website: https://eee.uci.edu/16s/39057
You can find the papers we will be reading here.

Message Board: https://eee.uci.edu/boards/s16/39057/
Please use the message board rather than email for all technical questions. This way, all students can share in the discussion and we can learn from each other.

Course Description:

This doctoral course introduces students to the field of mathematical programming through the lens of convex optimization. We will study the theory of convex optimization, and learn how to identify, formulate, transform, and solve convex optimization problems.

Convex programs are an important class of mathematical programs because (1) many problems can be formulated as convex programs, and (2) we have efficient techniques to find globally optimal solutions to convex programs. However, translating and formulating a given problem as a convex program is not always easy; in fact, it can require a high level of expertise to verify that a math program is indeed convex. In this class, we will introduce a methodology called *disciplined convex programming (DCP)*, which defines a set of rules derived from convex analysis If a math program is formulated following the DCP rules, it is guaranteed to be convex, eliminating the need to verify its convexity post-construction.

We will also study how classical decomposition techniques (e.g., column generation, Dantzig-Wolfe decomposition, Benders decomposition, and Lagrangian relaxation) can be helpful when solving large-scale convex optimization problems.

Students will choose a project which can be modeled as a convex optimization problem, and put to practice what they have learned using the modeling languages AMPL, MATLAB, and CVX. The techniques we will cover are applicable to a wide variety of business and engineering applications, and students are encouraged to choose a course project that is in line with their current research interests.

There are no formal prerequisites for this course beyond having a level of mathematical maturity which is expected of a PhD student at the Paul Merage School of Business. For example, it is expected that you know matrix algebra and multivariable calculus. Given that students may come from different backgrounds, I do not assume that students have a working knowledge of optimization theory. To get everyone up to speed, I will cover some background material in the first two lectures. But most importantly, if at some point during the course I start using terminology that you are unfamiliar with, please point this out so I can summarize any concepts which are unclear.

Course Materials (Required):

1. Boyd, S. and L. Vandenberghe. Convex Optimization. 2004. (ISBN: 0521833787). Available Online: http://www.stanford.edu/~boyd/cvxbook/
2. Resources on Disciplined Convex Programming. Available Online: http://dcp.stanford.edu/
3. The AMPL Book (2nd Edition) by Robert Fourer, David Gay, and Brian Kernighan, 2002 (ISBN: 0534388094). Available Online: http://www.ampl.com/BOOK/download.html
4. Papers and handouts posted on EEE

Useful Reading (Recommended):

1. Lasdon, L. Optimization Theory for Large Systems. 2002 (ISBN: 0486419991).
2. Wolsey, L. Integer Programming. 1998 (ISBN: 0471283665).
3. Desaulniers, G., J. Desrosiers, and M. Solomon. Column Generation. GERAD. 2005. (ISBN: 0387254854).
4. Martin, R. Large Scale Linear and Integer Optimization: A Unified Approach. 1999 (ISBN: 0792382021).
5. Conejo, A., E. Castillo, R. Minguez, and R. Garcia-Bertrand. Decomposition Techniques in Mathematical Programming. 2006 (ISBN: 3642066070).

Course Format:

You will be exposed to convex math programming through two main activities: reading and presenting papers, and working on a project that implements a convex math programming model and solution technique. This class will blend theory and practice, and so classes will be a mix of lectures, papers presented by me and by students, and tutorials on the use of the AMPL, MATLAB, and CVX modeling languages. By reading and presenting papers you will learn the theoretical foundations of convex math programming. Then, by working on a project you will be able to work through the details of a specific model and solution technique.

Grading Scheme:

Class Participation: 10%
Homework: 30%
Topic Presentation: 20%
Project: 40%, with breakdown: Proposal 5%, Presentation 15%, and Written Report 20%

Class Participation:

This is a technical course, and the best way to learn the subject matter is to be engaged in classroom discussions and to ask questions either in-class or on the message board. Please the read papers that will be presented in-class before the lecture. All students need to read all papers, not just the students that are presenting the topic. If it is apparent that you have not read the papers that other students are presenting, you will not get a full participation grade.

Homework:
I will periodically assign a few homework problems for you to get practice with the material covered in class. A homework may be as simple as producing a question to ask at the beginning of the next class, or it could involve solving some textbook examples. It could also be more research-based, such as finding an example of a model with a particular property, or a counterexample which shows that a fundamental property does not hold in all cases. I expect to assign roughly 7-8 of these, over the quarter.

Topic Presentation:
Each student will be assigned a topic to present in-class. The list of topics this year are all decomposition techniques for solving convex math programs: column generation, Dantzig-Wolfe decomposition, Lagrangian relaxation, Bender's decomposition, and the L-shaped method. For a given topic, I will provide you with some guidance about the important points that I would like you to cover in your presentation, and provide you with a short-list of papers and resources that you can use to begin your reading to understand the topic. Learning how to read and summarize academic papers is an important skill, especially when the subject matter is notation-heavy as will often be the case in this class. You will be graded on how well you communicate the main concepts of the topic you have been assigned. Please strive for clarity, correctness, and completeness (within the time allowed). You may use any combination of PowerPoint or scribbling on the whiteboard that you wish. If you choose to present your papers using just the whiteboard, I strongly suggest that you have the full presentation written down beforehand on paper, so that you can copy from your notes onto the whiteboard without making mistakes. You should plan your presentation to be about 30 minutes in length. I will aim to assign you a topic that is somewhat connected to your project proposal.

Project:

Throughout the course each student will work on a project. Projects are individual work (one project per student). If possible, it is a good idea to choose a topic that is directly related to your thesis. If your current research does not directly require optimization, then perhaps there is a related problem (i.e. something tangential to your current research) that would be good for you to explore that does require convex optimization techniques. But feel free to choose any topic that interests you, even if it is not part of your current research program.

Once you have a chosen a topic, your next step is to write a 3-page project proposal. The proposal should describe the problem that you wish to solve, where you are going to get the input data (randomly generated is fine; real data is usually better), which modeling aspects you expect to be difficult and/or challenging, and a description of your first course of action (i.e. a very high-level description of a solution technique that you might like to try). Proposals are due: **Tuesday, April 5th**.

I will review your proposals, and provide feedback to help you get started. After this point, we will reserve some class time for technical questions regarding the projects that I may address on a one-on-one basis (please bring your laptops to class!). Of course, you may also post technical questions to the newsgroup and see me during office hours for help with your projects. Although projects are individual work and each student will turn in one project, you may discuss your project with other students since it is very useful to share technical knowledge of how to get AMPL or CVX to do what you want it to do and how to implement a specific optimization-based technique.

By the end of May, your project should be nearing completion. You will prepare a 30-minute presentation to be presented in-class on **Tuesday, June 7th**. Your presentation should clearly describe the problem you are solving, the optimization formulation you adopted, where the input data came from, the solution optimization technique that you used to solve the problem and how you adapted this technique to your particular problem, and your computational results.

Your final deliverable for the project is a written report, to be turned in to me by email no later than **Tuesday, June 7th**. In your report, you should describe in detail the same things that you presented more broadly in your presentation (the problem you are solving, the optimization formulation you adopted, where the input data came from, the solution technique you used to solve the problem and how you adapted this technique to your particular problem, and your computational results). Please include the AMPL or CVX code that you used to solve your problem as an Appendix (very important!), and describe in your report the AMPL or CVX commands that were important in your project (e.g. `option relax_integrality`) and why. As a rough guideline, your report should be about 15 double-spaced 12-point font pages in length, excluding the Appendix (the code).

Academic Honesty and Student Conduct Policy:
Please refer to UCI's policies: http://www.editor.uci.edu/catalogue/appx/appx.2.htm.

Course Drop Deadline:
Friday, April 8th at 5PM

Course Outline:
The following is a *tentative* schedule for the course which may be updated as the course progresses.

March 29 – Class 1: Convex Sets
April 5 – Class 2: Convex Functions
April 12 – Class 3: Disciplined Convex Programming
April 19 – Class 4: Duality Theory
April 26 – Class 5: Column Generation (Yiwei)
May 3 – Class 6: Dantzig-Wolfe Decomposition (Ali H.K.)
May 10 – Class 7: Lagrangian Relaxation (Vahid)
May 17 – Class 8: Bender's Decomposition (Yuhan)
May 24 – Class 9: L-Shaped Method (Ali E.)
May 31 – NO CLASS (John's away)
June 7 – Project Presentations