Math 484: Topics Covered in Exam 3

Misha Lavrov

November 9, 2018

The second exam will, broadly speaking, cover all the material covered in class between Monday, October 15$^{\text{th}}$ and Friday, November 9$^{\text{th}}$, with some review of previous material. In this document, I go through each relevant section of the textbook and point out the things you should make sure to know from it.

To study for the exam, I suggest the following resources:

- Review the solution to the homework assignments. Moodle (https://learn.illinois.edu/) will have solution sets for each assignment on the day it is returned. It's worth reading over them even if you solved the problems correctly to see if you missed shortcuts or alternative approaches.
- When reviewing material from a specific section, look at the examples in the textbook, which are labeled "**(*x.y.z*) Example.**", and try to work through or understand them on your own before checking against the book's solution.
- The exercises at the end of each chapter are pretty good too. Some of them develop extra theory that wasn't covered in that chapter, which occasionally means they get deeper into the material than we do in this class. If you have questions about how to solve any of the exercises, I'm happy to answer them.

5 Convex Programming and the Karush–Kuhn–Tucker Conditions

5.1 Separation and Support Theorems for Convex Sets

Key definitions:

- interior, exterior, boundary, and closure of a set
- epigraph, subgradient, subdifferential of a convex function

Key results:

- The obtuse angle criterion (5.1.1)
- The basic separation theorem (5.1.5)
- The support theorem (5.1.9)

- Convex functions have subdifferentials at interior points (5.1.10)

In particular, you should be aware of how the other results follow from the obtuse angle criterion.

5.2 Convex Programming; The Karush–Kuhn–Tucker Theorem

Definitions:

- Convex program and superconsistent convex program.
- Value function $MP(\mathbf{z})$, sensitivity vector $\boldsymbol{\lambda}$, and Lagrangian $L(\mathbf{x}, \boldsymbol{\lambda})$ of a convex program.
- Tight and slack constraints.
- Complementary slackness.

You should be able to take any program, write down the KKT conditions, and solve them to get some points.

You should be very clear on the relationship between optimality, the saddle form of the KKT, and the gradient form of the KKT, and under what conditions each implies the other. This should let you determine when the points you get from KKT include the optimal solution.

5.3 The Karush–Kuhn–Tucker Theorem and Constrained Geometric Programming

You should be able to identify a constrained geometric program, put one into standard form, write down the dual program, solve it, and use the dual solution to find the primal solution.

The article "A tutorial on geometric programming" https://stanford.edu/~boyd/papers/pdf/gp_tutorial.pdf is another resource on geometric programming, which I personally found helpful for understanding what's going on in section 5.3 of our textbook, but I'm not holding your responsible for any of the contents of this article in particular.

5.4 Dual Convex Programs

Given a (not necessarily convex) program, you should be able to write down its KKT dual, explicitly solve for the dual objective function $h(\boldsymbol{\lambda})$, solve the dual problem, and if possible use this to find a solution for the primal program.

You should understand the general principle that we deduce constraints on the dual from finding cases where $h(\boldsymbol{\lambda}) = -\infty$ and excluding them. I won't ask you to do this in complicated cases, but arguments similar to the way we derived the dual of a linear program are fair game.

You should remember that sometimes there's a duality gap, but if there isn't (if we have a primal solution $\mathbf{x}$ and a dual solution $\boldsymbol{\lambda}$ with $f(\mathbf{x}) = h(\boldsymbol{\lambda})$) then this guarantees optimality.

5.5 Trust Regions

We skipped this section of the textbook, so you can ignore it.

6 Penalty Methods

6.1 Penalty Functions

For any program, you should be able to use either the absolute value penalty function (replacing $g(\mathbf{x})$ with $g^+(\mathbf{x})$) or the Courant–Beltrami penalty function (replacing $g(\mathbf{x})$ with $[g^+(\mathbf{x})]^2$) to write down an unconstrained optimization problem, and (in the case of Courant–Beltrami, assuming everything is continuously differentiable) solve it in terms of the penalty parameter.

6.2 The Penalty Method

When you apply the penalty method, you should be able to justify when you know that the result is an optimal solution for the original problem.

I'm not yet holding you responsible for conditions that guarantee the existence of an optimal solution ahead of time.