Optimization

Xiaojing Ye, Math & Stat, Georgia State University

Fall 2018

About Instructor

- **Instructor**: Dr. Xiaojing Ye
- **Email**: `xye@gsu.edu`
- **Webpage**: `https://math.gsu.edu/xye`
- **Office**: 25 Park Place, Room 1436
- **Office hour:** M 10:30–11:30AM, W 9:30–10:30AM (other time by appointment only)

About Course

- **Time**: MW 2:00–3:15PM
- **Location**: Langdale Hall 227
- **Textbook**: Introduction to Optimization, 4th edition, Edwin K. P. Chong and Stanislaw H. Zak, Wiley.
- **Prerequisit**:
  - Either MATH 3030 or both MATH 2641 (Formerly MATH 3435) and MATH 2215 with grades of C or higher.
  - The ability to program in a high-level language such as MATLAB or Python.

About Textbook and References

Textbook

- *Introduction to Optimization*, 4th edition, Edwin K. P. Chong and Stanislaw H. Zak, Wiley.

Reference books

- *Convex Analysis and Optimization*, Bertsekas, Nedic, and Ozdaglar, Athens.
- *Convex Optimization*, Boyd and Vandenberghe, Cambridge University Press.
- *Numerical Optimization*, Nocedal and Wright, Springer.

Homework

- Assigned every 2-2.5 weeks.
- 5 assignments. 6 point each. Credits given based on correctness and completeness.
- Due at beginning of class. **Late homework will not be accepted nor graded!**

Exams

- **Midterm 1 (20 pts)**: in class on September 9.
- **Midterm 2 (20 pts)**: in class on October 24.
- **Final (30 pts)**: 1:30-3:30PM on December 5.

Grading Policy

Grading points:

Homework	30 pts
Midterm 1	20 pts
Midterm 2	20 pts
Final	30 pts
Total	100 pts

Extra credit:

Attendance	1 missed	2 missed	3 missed	>3 missed
Extra credit	10 pts	5 pts	2 pts	0 pts

Grading scale:

Score	0–59	60–69	70–76	77–79	80–82	83–86	87–89	90–92	93–96	97–100
Grade	F	D	C	C+	B–	B	B+	A–	A	A+

Prerequisite

- Review uni- and multi-variable calculus and linear algebra (Part I Chapters 1-3 and 5 in textbook).
- MATLAB and Python programming

Some tips

- Before next class: preview materials by reading book, slides and notes;
- In class: try to understand as much as possible, take good notes, ask question if there's any!
- After class: review notes, and read book and slides again. Solve homework problems independently.
- If you stuck at a problem, think where you didn't fully understand and revisit that part of notes/book and try again.
- Come to my office hours if you need help.

1. Introduction

- mathematical optimization
- least-squares and linear programming
- convex optimization
- example
- course goals and topics
- nonlinear optimization
- brief history of convex optimization

Mathematical optimization

(mathematical) optimization problem

$$
\begin{array}{ll}
\text{minimize} & f_0(x) \\
\text{subject to} & f_i(x) \leq b_i, \quad i = 1, \ldots, m
\end{array}
$$

- $x = (x_1, \ldots, x_n)$: optimization variables
- $f_0 : \mathbf{R}^n \to \mathbf{R}$: objective function
- $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$: constraint functions

optimal solution $x^\star$ has smallest value of f_0 among all vectors that satisfy the constraints

Examples

portfolio optimization

- variables: amounts invested in different assets
- constraints: budget, max./min. investment per asset, minimum return
- objective: overall risk or return variance

device sizing in electronic circuits

- variables: device widths and lengths
- constraints: manufacturing limits, timing requirements, maximum area
- objective: power consumption

data fitting

- variables: model parameters
- constraints: prior information, parameter limits
- objective: measure of misfit or prediction error

Solving optimization problems

general optimization problem

- very difficult to solve
- methods involve some compromise, $e.g.$, very long computation time, or not always finding the solution

exceptions: certain problem classes can be solved efficiently and reliably

- least-squares problems
- linear programming problems
- convex optimization problems

Least-squares

$$\text{minimize} \quad \|Ax - b\|_2^2$$

solving least-squares problems

- analytical solution: $x^\star = (A^T A)^{-1} A^T b$
- reliable and efficient algorithms and software
- computation time proportional to $n^2 k$ ($A \in \mathbf{R}^{k \times n}$); less if structured
- a mature technology

using least-squares

- least-squares problems are easy to recognize
- a few standard techniques increase flexibility ($e.g.$, including weights, adding regularization terms)

Linear programming

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \dots, m \end{array}$$

solving linear programs

- no analytical formula for solution
- reliable and efficient algorithms and software
- computation time proportional to $n^2 m$ if $m \geq n$; less with structure
- a mature technology

using linear programming

- not as easy to recognize as least-squares problems
- a few standard tricks used to convert problems into linear programs ($e.g.$, problems involving ℓ_1- or ℓ_∞-norms, piecewise-linear functions)

Convex optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i, \quad i = 1, \ldots, m \end{array}$$

- objective and constraint functions are convex:

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$$

  if $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$

- includes least-squares problems and linear programs as special cases

solving convex optimization problems

- no analytical solution
- reliable and efficient algorithms
- computation time (roughly) proportional to $\max\{n^3, n^2m, F\}$, where F is cost of evaluating f_i's and their first and second derivatives
- almost a technology

using convex optimization

- often difficult to recognize
- many tricks for transforming problems into convex form
- surprisingly many problems can be solved via convex optimization