Lecture 0: Introduction

Shi Pu

School of Data Science (SDS)
The Chinese University of Hong Kong, Shenzhen

Outline

1. Course Information
2. Formal Introduction

Outline

1. Course Information

2. Formal Introduction

Course Logistics

- Lectures: Monday and Wednesday 10:30 - 11:50 AM, Zhi Xin 101
- Office Hour: Tuesday 11:00 AM - 12:00 PM, Dao Yuan 506b
- Tutorials: Tuesday 6:00 - 6:50 PM, Zhi Xin 101
- TA: Runze You
- TA Office Hour: Tuesday 6:00 - 6:50 PM, Dao Yuan 224
- Course website: Blackboard and `https://ryou98.github.io/MAT3220-Optimization2/`

Course Logistics

- About 7 homework assignments.
  You can consult with other students, but all homework must be completed individually.
- Tentative exam schedule: Midterm: November 1 (in class), Final: December 13 (in class)
- Grading: homework 20%, midterm 40%, final 40%

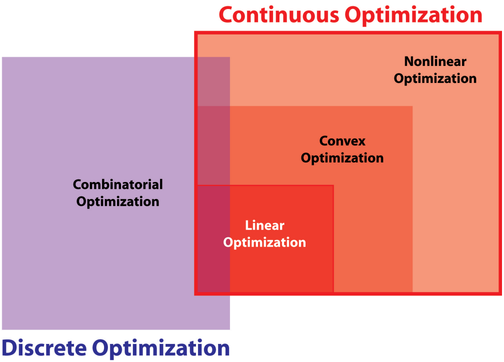
Course Logistics

References:

- Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB, by Amir Beck (electronic copy available online)
- Convex Optimization, by S. Boyd and L. Vandenberghe (electronic copy available online)
- Nonlinear Programming, by D. Bertsekas

What is the Course About?

A special class of optimization



Who Cares and Why?

Who?

- Anyone using or interested in computational aspects of nonlinear optimization

Why?

- To understand the underlying basic terminology, principles, and methodology (to efficiently use the existing software tools)
- To develop ability to modify tools when needed
- To develop ability to design new algorithms or improve the efficiency of the existing ones

The goal of this course is to provide you with working knowledge of nonlinear optimization, in particular, to provide you with skills and knowledge to

- recognize nonlinear problems
- model problems
- solve the problems

Nonlinear Optimization History

Optimization theory and analysis have being studied for a long time, mostly by mathematicians (History of Optimization)

Until late 1980's:

- Algorithmic development focused mainly on solving Linear Problems
  - Simplex Algorithm for linear programming (Dantzig, 1947)
  - Ellipsoid Method (Shor, 1970)
  - Interior-Point Methods for linear programming (Karmarkar, 1984)
- Applications in operations research and *few* in engineering

Nonlinear Optimization History

- Since late 1980's: A new interest in *convex optimization* emerges
- The recognition that Interior-Point Methods can efficiently solve certain classes of convex problems, including semi-definite programs and second-order cone programs, almost *as easily as linear programs*
- The new technologies and their applications created a need for new models (convex models are often suitable)
- Convex problems are prevalent in practice
  - Automatic Control Systems
  - Estimation, Signal and Image Processing
  - Communication and Data Networks
  - Data Analysis and Modeling
  - Statistics and Finance

Ongoing Research

- Optimization methods for *large-scale machine learning*
- Extending the methodology to *nonconvex problems* (training deep neural networks)
- *Distributed computations* for large-scale problems

Formal Introduction

- Mathematical Formulation of Optimization
- Some Examples of Optimization Problems
- Solving Optimization Problems
  - Least-Squares
  - Linear Optimization
  - Convex Optimization
  - Nonconvex Optimization

$$
\begin{array}{ll}
\text{minimize} & f(\mathbf{x}) \\
\text{subject to} & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\
& \mathbf{x} \in C
\end{array}
$$

- Vector $\mathbf{x} = (x_1, \ldots, x_n)^\top$ represents optimization (decision) variables
- Function $f : \mathbb{R}^n \to \mathbb{R}$ is an objective function
- Functions $g_i : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \ldots, m$ are constraint functions (representing inequality constraints)
- Set $C \subseteq \mathbb{R}^n$ is a constraint set

Optimal value: The smallest value of f among all vectors that satisfy the set and the inequality constraints
Optimal solution: A vector that achieves the optimal value of f and satisfies all the constraints

Some Examples of Optimization Problems

Communication Networks

- Variables: communication rates for users
- Constraints: link capacities
- Objective: user cost

Portfolio Optimization

- Variables: amounts invested in different assets
- Constraints: available budget, maximum/minimum investment per asset, minimum return, time constraints
- Objective: overall risk or return variance

Data Fitting

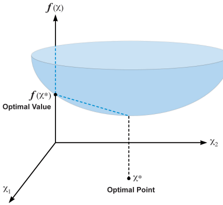
- Variables: model parameters
- Constraints: prior information, parameter limits
- Objective: measure of misfit or prediction error

General Optimization Problem

- Very difficult to solve
- Existing methods involve trade offs between "time" and "accuracy", eg., very long computation time, or finding a sub-optimal solution

Exceptions: Certain problem classes can be solved efficiently and reliably

- Least-Squares Problems
- Linear Programming Problems
- Some classes of Convex/Nonconvex Optimization Problems



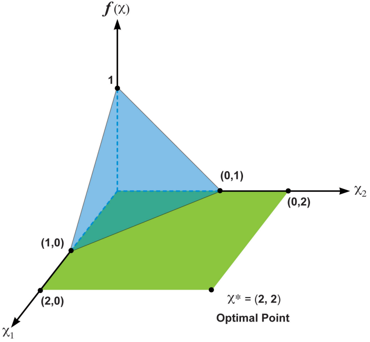
minimize $\quad \|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$

Solving Least-Squares Problems

- Analytical solution: $\mathbf{x}^* = (\mathbf{A}^\top \mathbf{A})^{-1}\mathbf{A}^\top \mathbf{b}$
- Reliable and efficient algorithms and software
- A mature technology
- Computation time proportional to $n^2 m$ ($\mathbf{A} \in \mathbb{R}^{m\times n}$); less if structured

Using Least-Squares

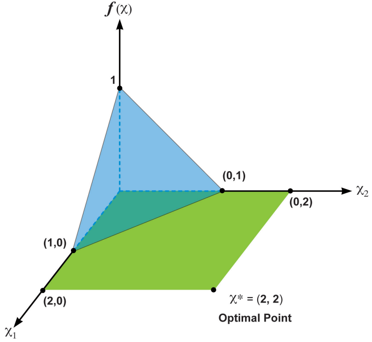
- Least-squares problems are easy to recognize
- In regression analysis, optimal control, parameter estimation
- A few standard techniques increase its flexibility in applications (eg., including weights, regularization terms)



$$
\begin{array}{ll}
\text{minimize} & \mathbf{c}^\top \mathbf{x} \\
\text{subject to} & \mathbf{a}_i^\top \mathbf{x} \le b_i, \quad 1 \le i \le m
\end{array}
$$

Solving Linear Programs

- No analytical solution
- Reliable and efficient algorithms and software
- A mature technology



$$\begin{array}{ll} \text{minimize} & \mathbf{c}^\top \mathbf{x} \\ \text{subject to} & \mathbf{a}_i^\top \mathbf{x} \leq b_i, \quad 1 \leq i \leq m \end{array}$$

Using Linear Programming

- Not as easy to recognize as least-squares problems (linear formulation possible but not always obvious)
- A few standard tricks used to convert problems into linear programs (eg., problems involving maximum norm, piecewise-linear functions)

$$
\begin{array}{ll}
\text{minimize} & f(\mathbf{x}) \\
\text{subject to} & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\
& \mathbf{x} \in C
\end{array}
$$

- Objective and constraint functions are convex
- Constraint set is convex
- Includes least-squares problems and linear programs as special cases

Solving Convex Optimization Problems

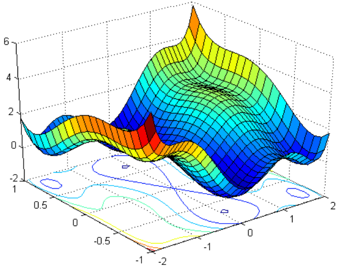
- No analytical solution
- Reliable and efficient algorithms for some classes
- Computation time (roughly) proportional to $\max\{n^3, n^2 m, G\}$, where G is a cost of evaluating g_i's and their first and second derivatives
- Almost a technology
- Modern methods for *large-scale* problems

$$
\begin{array}{ll}
\text{minimize} & f(\mathbf{x}) \\
\text{subject to} & g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, m \\
& \mathbf{x} \in C
\end{array}
$$

Using Convex Optimization

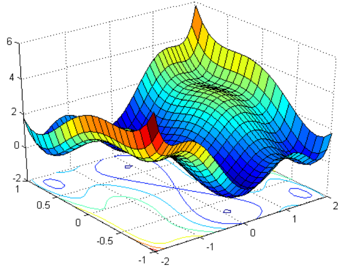
- Often difficult to recognize
- Many tricks for transforming problems into convex form
- Many practical problems can be modeled as convex optimization

Nonconvex Optimization



Traditional techniques for general nonconvex problems involve compromises

- Local optimization methods (nonlinear programming)
  - find a point that minimizes f among feasible points near it
  - fast, can handle large problems
  - require initial guess
  - provide no information about distance to (global) optimum



Traditional techniques for general nonconvex problems involve compromises

- Global optimization methods
  - find the (global) solution
  - worst-case complexity grows exponentially with problem size

Algorithms are often based on solving convex subproblems

- Mathematical Preliminaries
- Optimality conditions for unconstrained optimization
- Gradient method and Newton's method
- Convex optimization
- KKT conditions and Duality
- Subgradients and subgradient method
- (optional) Proximal gradient method
- (optional) Randomized algorithms
- (optional) Accelerated gradient method