AA/CSE/EE/ME 578: Convex Modeling and Optimization

Instructor: Maryam Fazel

University of Washington
Winter 2024

1. Introduction

- course logistics; goals; pre-requisites
- mathematical optimization
- least-squares; linear programming; convex optimization
- a sample result: infeasibility certificates

Course logistics

- Canvas page: all files, basic info
- main book: Convex Optimization, Boyd & Vandenberghe
- use Canvas 'Discussions' for questions (from me, TA, classmates)
- TA: Natalia Pavlasek, `pavlasek@uw`; Grader: Wenzheng (Wennie) Zhao , `wennie99@uw`
- lectures: Mon+Wed, TA sessions: Fri
- weekly homeworks (7-8 HW sets); submission on Gradescope
  midterm: short, in-class, closed-book exam; final: 24-hour take-home open-book exam the weekend before finals week.

see webpage for grading, dates, & more info

TA session this Fri Jan 5th: background & HW1

Course goals

1. understand convexity and its properties (convex analysis)
2. formulate problems as convex optimization problems:
   convex modeling
3. characterize optimality, duality theory, theoretical limits and tradeoffs
4. derive, examine the properties of, and solve problems using CVX in Matlab (or Python)
5. develop convex modeling skills to use in your research

Course prerequisites

1. EE 510, EE 547 (or equivalent). advanced linear algebra & matrix analysis: matrix norms; singular value decomposition; positive semidefinite forms; gradients and Hessians; etc.
2. basic analysis (open & closed sets, compactness)
3. basic probability (e.g., multivariate Gaussian)
4. background in optimization theory (e.g. linear programming) helps but not required
5. time commitment

See (i) appendix A of textbook and (ii) Background Review note.

Questions? come to TA session & office hours, read textbook chapter 2, start HW 1.

Mathematical optimization

optimization problem:

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

- $x = (x_1, \ldots, x_n) \in \mathbf{R}^n$: 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

communication networks

- variables: communication rates, bandwidth allocated to each flow
- constraints: link capacities
- objective: (maximize) total network utility

data fitting and machine learning

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

sparse signal recovery

- variables: unknown (discrete-time) sparse signal
- constraints: measurements of signal
- objective: recovery error, sparsity-promoting penalty

  (originally a hard combinatorial problem, but some solution guarantees exist for convex relaxations, e.g., compressed sensing theory)

Solving optimization problems

general optimization problem

- very difficult to solve
- methods involve some compromise, $e.g.$, extremely long (prohibitive) computation time, or no guarantee to find the solution

exceptions: certain problem classes can be solved globally and efficiently

- least-squares problems
- linear programming problems
- **convex** optimization problems (subject of this course)

"In fact, the great watershed in optimization is not between linearity and nonlinearity, but convexity and nonconvexity"

–R. T. Rockafellar, SIAM review, '93

Least-squares

$$\text{minimize} \quad \|Ax - b\|_2^2, \qquad x \in \mathbf{R}^n$$

solving least-squares problems

- analytical solution: $x^\star = (A^T A)^{-1} A^T b$
- reliable, efficient algorithms/software; computation time proportional to $n^2 m$ ($A \in \mathbf{R}^{m \times n}$, $n < m$); 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, regularization terms)

Linear programming

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

solving linear programs

- no analytical formula for solution
- reliable, 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 (roughly, have positive curvature):

  $$f(\lambda x + (1 - \lambda) y) \leq \lambda f(x) + (1 - \lambda) f(y)$$

  for $0 \leq \lambda \leq 1$

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

much more later. . .

solving convex problems

- no analytical solution
- reliable, efficient algorithms
- almost a technology

using convex optimization

- difficult to recognize convex problems
- surprisingly **many** applications
- important to learn skills to cast problems into convex forms
- important to distinguish convex and nonconvex problems

Nonlinear optimization

traditional techniques for general nonconvex problems involve compromises:

local optimization methods (nonlinear programming)

- fast, can handle large problems
- but: require initial guess, only local search (minimize f_0 among nearby feasible points); no information about distance to global opt

global optimization methods

- find the (global) solution
- worst-case complexity is exponential

insights from convex optimization can help with nonconvex problems:

- initialization for local opt methods; subproblems/bounds in global opt methods
- relaxations: nonconvex constraints replaced with looser convex ones

A sample result: infeasibility certificate

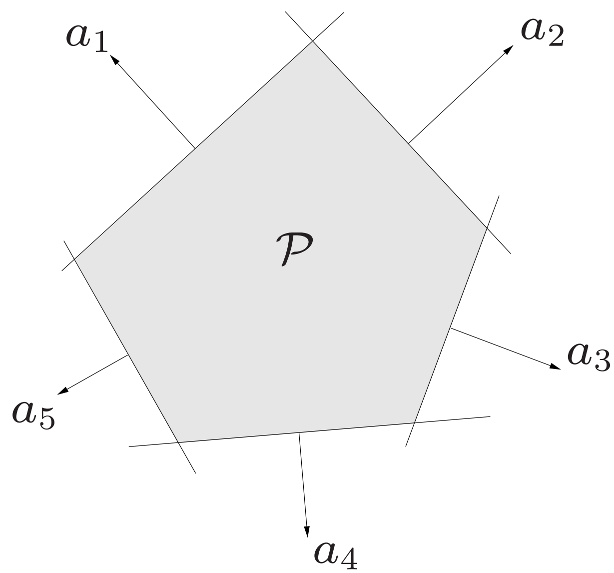
feasibility problem: find $x \in C$

convex case: feasibility algs. return **yes** or **no**

general case: feasibility algs. return **yes** or **maybe**

example:

$$\mathcal{P} = \left\{x \mid a_k^T x \leq b_k, k = 1, \ldots, m\right\}$$



how could you know $\mathcal{P} = \emptyset$?

suppose $\lambda_i \geq 0$, $\sum \lambda_i a_i = 0$, $\sum \lambda_i b_i < 0$

then $a_i^T x \leq b_i$, $i = 1, \ldots, m$, implies

$$0 \leq \sum_i \lambda_i (b_i - a_i^T x) = \sum_i \lambda_i b_i < 0$$

contradiction! we conclude:

$$\exists \lambda_i \geq 0, \ \sum \lambda_i a_i = 0, \ \sum \lambda_i b_i < 0 \Longrightarrow \mathcal{P} = \emptyset$$

we say λ_i's are a *certificate* or *proof* of infeasibility

fact (convexity): if $a_i^T x \leq b_i$ is infeasible, then there exists a certificate proving it!

certificate useful several ways:

- know for sure that $\mathcal{P} = \emptyset$
- can conclude that some 'relaxed' constraints are also infeasible:

$$a_i^T x \leq b_i + \epsilon$$

is infeasible for $\epsilon < (-\sum_i \lambda_i b_i)/(\sum_i \lambda_i)$

Recent advances and new interest

- algorithms that can efficiently solve broad classes of convex problems, including semidefinite programs, almost as easily as linear programs! (Nesterov & Nemirovski 1994)
- more problems recognized as convex; new settings (huge-scale, distributed, real-time, online); classes of algorithms (interior-point, 1st order, zeroth order,...)
- a major driver in the last 10 years: machine learning/AI, data-driven engineering, statistics

applications: convex optimization models appear in many areas:

- machine learning/AI/data science
- control systems; signal processing; communications
- finance; econ
- engineering design