Introduction

Goran Banjac

Large-Scale Convex Optimization
ETH Zurich

February 18, 2020

Outline

Mathematical optimization

Least-squares and linear programming

Convex optimization

Solvers and modeling languages

Course organization

Mathematical optimization

(mathematical) optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \end{array}$$

- $x \in \mathbb{R}^n$ is (vector) variable to be chosen
- f_0 is the *objective function* to be minimized
- $f_1, \ldots, f_m$ are *constraint functions*

- *solution* or *minimizer* $x^\star$ has the smallest value of f_0 among all vectors that satisfy the constraints
- the problem is *infeasible* if no x satisfies the constraints

Finding good (or best) actions

- x represents some *action*, *e.g.*,
  - trades in a portfolio
  - airplane control surface deflections
  - schedule or assignment
  - resource allocation
  - transmitted signal
- constraints limit actions or impose conditions on outcome
- the smaller the objective $f_0(x)$, the better
  - total cost (or negative profit)
  - deviation from desired or target outcome
  - fuel used
  - risk

Engineering design

- x represents a *design* (of a circuit, device, structure, . . .)
- constraints come from
  - manufacturing process
  - performance requirements
- objective $f_0(x)$ is combination of cost, weight, power, . . .

Finding good models

- x represents the *parameters* in a model
- constraints impose requirements on model parameters (*e.g.*, nonnegativity)
- objective $f_0(x)$ is the prediction error on some observed data (and possibly a term that penalizes model complexity)

Estimation

- x is something we want to estimate/reconstruct, given some measurement y
- constraints come from prior knowledge about x
- objective $f_0(x)$ measures deviation between predicted and actual measurements

Worst-case analysis

- variables are actions or parameters out of our control
  (and possibly under the control of an adversary)
- constraints limit the possible values of the parameters
- minimizing $-f_0(x)$ finds *worst possible parameter values*

- if the worst possible value of $f_0(x)$ is tolerable, you're OK
- it's good to know what the worst possible scenario can be

Optimization-based models

- model an entity as taking actions that solve an optimization problem
  - an individual makes choices that maximize expected utility
  - an organism acts to maximize its reproductive success
  - reaction rates in a cell maximize growth
  - currents in an electric circuit minimize total power
- (except the last) these are *very crude* models
- and yet, they often work very well

Summary

- **summary:** optimization arises *everywhere*
- **the bad news:** most optimization problems are *intractable*
  i.e., *we cannot solve them*
- **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: $x^\star = (A^TA)^{-1}A^Tb$
- reliable and efficient algorithms and software
- 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

linear program

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

solving linear programs

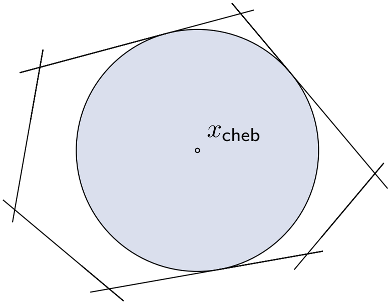
- no analytical formula for solution
- reliable and efficient algorithms and software
- a mature technology

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-affine functions)

Example – Chebyshev center of a polyhedron

- finding the largest Euclidean ball that lies in a polyhedron
- the problem can be converted into a linear program



Convex optimization

convex optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m \end{array}$$

- $f_0, \ldots, f_m$ are **convex**: for $\theta \in [0, 1]$,

  $$f_i(\theta x + (1 - \theta) y) \leq \theta f_i(x) + (1 - \theta) f_i(y)$$

  i.e., f_i have nonnegative (upward) curvature

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

Why

- beautiful, nearly complete theory
  - duality, optimality conditions, ...
- effective algorithms (in theory and practice)
  - get **global solution** (and optimality certificate)
  - polynomial complexity
- conceptual unification of many methods
- **lots of applications** (many more than previously thought)

Application areas

- machine learning, statistics
- finance
- supply chain, revenue management, advertising
- control
- signal and image processing, vision
- networking
- circuit design
- combinatorial optimization
- quantum mechanics
- flux-based analysis

Brief history of convex optimization

theory (convex analysis): 1900–1970

algorithms

- 1947: simplex algorithm for linear programming (Dantzig)
- 1970s: ellipsoid method and other subgradient methods
- 1980s & 90s: polynomial-time interior-point methods for convex optimization (Karmarkar 1984, Nesterov & Nemirovski 1994)
- since 2000s: many methods for large-scale convex optimization

applications

- before 1990: mostly in operations research, a few in engineering
- since 1990: many applications in engineering (control, signal processing, communications, circuit design, . . .)
- since 2000s: statistics, machine learning, . . .

Medium-scale solvers

- 1000s – 10000s variables, constraints
- reliably solved by interior-point methods on single machine (especially for problems in standard cone form)
- exploit problem sparsity
- not quite a technology, but getting there
- used in control, finance, engineering design, signal processing, . . .

Example – Optimal spacecraft landing

https://youtu.be/2t15vP1PyoA

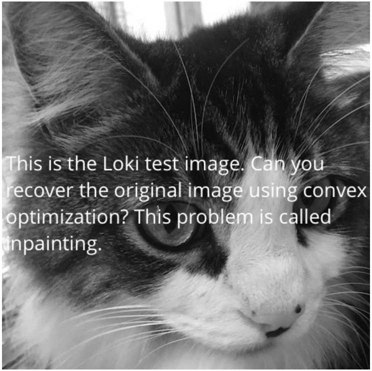
Large-scale solvers

- 100k – 1B variables, constraints
- solved using custom (often problem specific) methods
  - limited memory BFGS
  - stochastic subgradient
  - block coordinate descent
  - operator splitting methods
- require custom implementation, tuning for each problem
- drift from physics-based towards information-based applications
  - data is super plentiful
  - storage, transmission of data is easy
  - computers are super fast (and many are super cheap)
  - high-level programming languages make it easy to do complex stuff
- used in machine learning, image processing, medical imaging, . . .

Example – Image in-painting

512×512 grayscale image ($n \approx 300$k variables)

Corrupted image



In-painted image

Modeling languages

- high-level language support for convex optimization
  - describe problem in high-level language
  - description automatically transformed to a standard form
  - solved by standard solver, transformed back to original form
- implementations:
  - YALMIP, CVX (Matlab)
  - CVXPY (Python)
  - Convex.jl (Julia)
  - CVXR (R)
- enable rapid prototyping
- ideal for teaching (can do a lot with short scripts)

CVXPY

(Diamond & Boyd, 2013); (Agrawal et al., 2018)

$$\begin{array}{ll} \text{minimize} & \|Ax-b\|_2^2 + \gamma\|x\|_1 \\ \text{subject to} & \|x\|_\infty \le 1 \end{array}$$

```
import cvxpy as cp

x = cp.Variable(n)
cost = cp.sum_squares(A*x-b) + gamma*cp.norm(x,1)
prob = cp.Problem(cp.Minimize(cost),
                  [cp.norm(x,"inf") <= 1])
opt_val = prob.solve()
solution = x.value
```

- `A`, `b`, `gamma` are constants (`gamma` nonnegative)
- `solve` method converts problem to standard form, solves, assigns `value` attributes

What the course is about

- we will take you from zero to functional in the big world of modern convex optimization
- you'll learn about
  - convex sets, functions, optimization problems
  - duality theory
  - state-of-the-art methods for large-scale optimization
- and, *you'll actually do stuff with it*
  - data fitting and classification
  - image processing
  - optimal control
  - portfolio optimization

  (to mention just a few things)
- we'll de-mystify some things that (might) look like magic to you now

Your instructors this semester

- Goran Banjac
- Mathias Hudoba de Badyn
- Andrea Iannelli
- Angeliki Kamoutsi
- Ilnura Usmanova

Course organization

- all official course info on ETH Moodle platform
  - Lectures: Tuesdays, 16:15 – 18:00, CAB G 61
  - Exercise sessions: Fridays, 10:15 – 12:00, CAB G 61
  - Office hours: Wednesdays, 13:30 – 14:30, ETL I 10.1
  - special dates will be announced via Moodle
- you should know:
  - basic calculus
  - linear algebra
  - minimal programming (*e.g.*, Python or Matlab)
- requirements:
  - attendance at lecture
  - attendance at weekly exercise session
  - weekly homework (30% of the grade)
  - exam (70% of the grade)

Recommended reading

- S. Boyd and L. Vandenberghe: *Convex Optimization*. 2004.
- R. T. Rockafellar: *Convex Analysis*. 1970.
- H. H. Bauschke and P. L. Combettes: *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. 2011.
- J.-B. Hiriart-Urruty and C. Lemaréchal: *Fundamentals of Convex Analysis*. 2001.

- other relevant literature will be suggested during the course

References

- these lecture notes are based to a large extent on those for the Stanford EE364a class developed by Stephen Boyd
- the original slides can be downloaded from

  `https://web.stanford.edu/class/ee364a/lectures.html`

  `https://web.stanford.edu/~boyd/papers/cvx_short_course.html`