Convex Optimization

Stephen Boyd  Lieven Vandenberghe

Revised slides by Stephen Boyd, Lieven Vandenberghe, and Parth Nobel
1. Introduction
Outline

Mathematical optimization

Convex optimization
Optimization problem

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq 0, \quad i = 1, \ldots, m \\
& \quad g_i(x) = 0, \quad i = 1, \ldots, p
\end{align*}
\]

- \( x \in \mathbb{R}^n \) is (vector) variable to be chosen (\( n \) scalar variables \( x_1, \ldots, x_n \))
- \( f_0 \) is the \textbf{objective function}, to be minimized
- \( f_1, \ldots, f_m \) are the \textbf{inequality constraint functions}
- \( g_1, \ldots, g_p \) are the \textbf{equality constraint functions}

- variations: maximize objective, multiple objectives, \ldots
Finding good (or best) actions

- $x$ represents some action, e.g.,
  - trades in a portfolio
  - airplane control surface deflections
  - schedule or assignment
  - resource allocation

- 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
  - risk
  - fuel use
Finding good models

- $x$ represents the **parameters** in a model
- constraints impose requirements on model parameters (e.g., nonnegativity)
- objective $f_0(x)$ is sum of two terms:
  - a prediction error (or loss) on some observed data
  - a (regularization) term that penalizes model complexity
Worst-case analysis (pessimization)

- 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 a circuit minimize total power

- (except the last) these are very crude models
- and yet, they often work very well
Basic use model for mathematical optimization

▶ instead of saying how to choose (action, model) $x$
▶ you articulate what you want (by stating the problem)
▶ then let an algorithm decide on (action, model) $x$
Can you solve it?

- generally, no
- but you can try to solve it approximately, and it often doesn’t matter

- the exception: **convex optimization**
  - includes linear programming (LP), quadratic programming (QP), many others
  - we can solve these problems reliably and efficiently
  - come up in many applications across many fields
Nonlinear optimization

traditional techniques for general nonconvex problems involve compromises

local optimization methods (nonlinear programming)
- find a point that minimizes $f_0$ among feasible points near it
- can handle large problems, e.g., neural network training
- require initial guess, and often, algorithm parameter tuning
- provide no information about how suboptimal the point found is

global optimization methods
- find the (global) solution
- worst-case complexity grows exponentially with problem size
- often based on solving convex subproblems
Outline

Mathematical optimization

Convex optimization
Convex optimization

convex optimization problem:

\[
\begin{align*}
\text{minimize} & \quad f_0(x) \\
\text{subject to} & \quad f_i(x) \leq 0, \quad i = 1, \ldots, m \\
& \quad Ax = b
\end{align*}
\]

- variable \( x \in \mathbb{R}^n \)
- equality constraints are linear
- \( f_0, \ldots, f_m \) 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
When is an optimization problem hard to solve?

- classical view:
  - linear (zero curvature) is easy
  - nonlinear (nonzero curvature) is hard

- the classical view is **wrong**

- the correct view:
  - convex (nonnegative curvature) is easy
  - nonconvex (negative curvature) is hard
Solving convex optimization problems

- many different algorithms (that run on many platforms)
  - interior-point methods for up to 10000s of variables
  - first-order methods for larger problems
  - do not require initial point, babysitting, or tuning

- can develop and deploy quickly using modeling languages such as CVXPY

- solvers are reliable, so can be embedded

- code generation yields real-time solvers that execute in milliseconds
  (e.g., on Falcon 9 and Heavy for landing)
Modeling languages for convex optimization

- domain specific languages (DSLs) for convex optimization
  - describe problem in high level language, close to the math
  - can automatically transform problem to standard form, then solve

- enables rapid prototyping
- it’s now much easier to develop an optimization-based application
- ideal for teaching and research (can do a lot with short scripts)

- gets close to the basic idea: **say what you want, not how to get it**
**CVXPY example: non-negative least squares**

**math:**

\[
\begin{align*}
\text{minimize} & \quad \|Ax - b\|_2^2 \\
\text{subject to} & \quad x \geq 0
\end{align*}
\]

- variable is \(x\)
- \(A, b\) given
- \(x \geq 0\) means \(x_1 \geq 0, \ldots, x_n \geq 0\)

**CVXPY code:**

```python
import cvxpy as cp

A, b = ...

x = cp.Variable(n)
obj = cp.norm2(A @ x - b)**2
constr = [x >= 0]
prob = cp.Problem(cp.Minimize(obj), constr)
prob.solve()
```

Convex Optimization Boyd and Vandenberghe 1.15
Brief history of convex optimization

▶ **theory (convex analysis):** 1900–1970

▶ **algorithms**
  – 1947: simplex algorithm for linear programming (Dantzig)
  – 1960s: early interior-point methods (Fiacco & McCormick, Dikin, …)
  – 1970s: ellipsoid method and other subgradient methods
  – 1980s & 90s: interior-point methods (Karmarkar, Nesterov & Nemirovski)
  – 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: machine learning and statistics, finance
Summary

convex optimization problems
▶ are optimization problems of a special form
▶ arise in many applications
▶ can be solved effectively
▶ are easy to specify using DSLs