Convex Optimization

Lieven Vandenberghe

Electrical Engineering Department, UC Los Angeles

Tutorial lectures, 18th Machine Learning Summer School
September 13-14, 2011
Introduction

- mathematical optimization
- linear and convex optimization
- recent history
Mathematical optimization

\[
\begin{align*}
\text{minimize} & \quad f_0(x_1, \ldots, x_n) \\
\text{subject to} & \quad f_1(x_1, \ldots, x_n) \leq 0 \\
& \quad \ldots \\
& \quad f_m(x_1, \ldots, x_n) \leq 0
\end{align*}
\]

- a mathematical model of a decision, design, or estimation problem
- generally intractable
- even simple looking nonlinear optimization problems can be very hard
The famous exception: linear programming

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad a_i^T x \leq b_i, \quad i = 1, \ldots, m
\end{align*}
\]

- widely used since Dantzig introduced the simplex algorithm in 1948
- since 1950s, many applications in operations research, network optimization, finance, engineering, combinatorial optimization, . . .
- extensive theory (optimality conditions, sensitivity, . . .)
- there exist very efficient algorithms for solving linear programs
Convex optimization problem

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

• objective and constraint functions are convex: for \(0 \leq \theta \leq 1\)

\[
f_i(\theta x + (1 - \theta)y) \leq \theta f_i(x) + (1 - \theta)f_i(y)
\]

• can be solved globally, with similar (polynomial-time) complexity as LPs

• surprisingly many problems can be solved via convex optimization

• provides tractable heuristics and relaxations for non-convex problems
History

• 1940s: linear programming

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad a_i^T x \leq b_i, \quad i = 1, \ldots, m
\end{align*}
\]

• 1950s: quadratic programming

• 1960s: geometric programming

• 1990s: semidefinite programming, second-order cone programming, quadratically constrained quadratic programming, robust optimization, sum-of-squares programming, . . .
New applications since 1990

- linear matrix inequality techniques in control
- support vector machine training via quadratic programming
- semidefinite programming relaxations in combinatorial optimization
- circuit design via geometric programming
- $\ell_1$-norm optimization for sparse signal reconstruction
- applications in structural optimization, statistics, signal processing, communications, image processing, computer vision, quantum information theory, finance, power distribution, . . .
Advances in convex optimization algorithms

interior-point methods

• 1984 (Karmarkar): first practical polynomial-time algorithm for LP
• 1984-1990: efficient implementations for large-scale LPs
• around 1990 (Nesterov & Nemirovski): polynomial-time interior-point methods for nonlinear convex programming
• since 1990: extensions and high-quality software packages

fast first-order algorithms

• similar to gradient descent, but with better convergence properties
• based on Nesterov’s optimal-rate gradient methods from 1980s
• extend to certain nondifferentiable or constrained problems
Overview

1. Basic theory and convex modeling
   - convex sets and functions
   - common problem classes and applications

2. Interior-point methods for conic optimization
   - conic optimization
   - barrier methods
   - symmetric primal-dual methods

3. First-order methods
   - gradient algorithms
   - dual techniques
Convex sets and functions

• convex sets

• convex functions

• operations that preserve convexity
Convex set

contains the line segment between any two points in the set

\[ x_1, x_2 \in C, \quad 0 \leq \theta \leq 1 \quad \implies \quad \theta x_1 + (1 - \theta) x_2 \in C \]
Basic examples

**affine set:** solution set of linear equations $Ax = b$

**halfspace:** solution of one linear inequality $a^T x \leq b \ (a \neq 0)$

**polyhedron:** solution of finitely many linear inequalities $Ax \leq b$

**ellipsoid:** solution of quadratic inequality

$$(x - x_c)^T A (x - x_c) \leq 1 \quad (A \text{ positive definite})$$

**norm ball:** solution of $\|x\| \leq R$ (for any norm)

**positive semidefinite cone:** $S^n_+ = \{ X \in S^n \mid X \succeq 0 \}$

the **intersection** of any number of convex sets is convex
Example of intersection property

\[ C = \{ x \in \mathbb{R}^n \mid |p(t)| \leq 1 \text{ for } |t| \leq \pi/3 \} \]

where \( p(t) = x_1 \cos t + x_2 \cos 2t + \cdots + x_n \cos nt \)

\( C \) is intersection of infinitely many halfspaces, hence convex
Convex function

domain \( \text{dom} \ f \) is a convex set and Jensen’s inequality holds:

\[
f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)
\]

for all \( x, y \in \text{dom} \ f, \ 0 \leq \theta \leq 1 \)

\( f \) is concave if \( -f \) is convex
Examples

• linear and affine functions are convex and concave

• \( \exp x, -\log x, x \log x \) are convex

• \( x^\alpha \) is convex for \( x > 0 \) and \( \alpha \geq 1 \) or \( \alpha \leq 0 \); \( |x|^\alpha \) is convex for \( \alpha \geq 1 \)

• norms are convex

• quadratic-over-linear function \( x^T x / t \) is convex in \( x, t \) for \( t > 0 \)

• geometric mean \( (x_1 x_2 \cdots x_n)^{1/n} \) is concave for \( x \geq 0 \)

• \( \log \det X \) is concave on set of positive definite matrices

• \( \log(e^{x_1} + \cdots e^{x_n}) \) is convex
Epigraph and sublevel set

**epigraph:** $\text{epi } f = \{(x, t) \mid x \in \text{dom } f, \ f(x) \leq t\}$

A function is convex if and only if its epigraph is a convex set.

**sublevel sets:** $C_\alpha = \{x \in \text{dom } f \mid f(x) \leq \alpha\}$

The sublevel sets of a convex function are convex (converse is false).
Differentiable convex functions

differentiable $f$ is convex if and only if $\text{dom } f$ is convex and

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) \quad \text{for all } x, y \in \text{dom } f$$

![Graph showing convex function]

twice differentiable $f$ is convex if and only if $\text{dom } f$ is convex and

$$\nabla^2 f(x) \succeq 0 \quad \text{for all } x \in \text{dom } f$$
Methods for establishing convexity of a function

1. verify definition

2. for twice differentiable functions, show $\nabla^2 f(x) \succeq 0$

3. show that $f$ is obtained from simple convex functions by operations that preserve convexity
   - nonnegative weighted sum
   - composition with affine function
   - pointwise maximum and supremum
   - minimization
   - composition
   - perspective
Positive weighted sum & composition with affine function

**nonnegative multiple**: $\alpha f$ is convex if $f$ is convex, $\alpha \geq 0$

**sum**: $f_1 + f_2$ convex if $f_1, f_2$ convex (extends to infinite sums, integrals)

**composition with affine function**: $f(Ax + b)$ is convex if $f$ is convex

**examples**

- logarithmic barrier for linear inequalities

\[
f(x) = - \sum_{i=1}^{m} \log(b_i - a_i^T x)
\]

- (any) norm of affine function: $f(x) = \|Ax + b\|$
Pointwise maximum

\[ f(x) = \max\{f_1(x), \ldots, f_m(x)\} \]

is convex if \( f_1, \ldots, f_m \) are convex

**example:** sum of \( r \) largest components of \( x \in \mathbb{R}^n \)

\[ f(x) = x[1] + x[2] + \cdots + x[r] \]

is convex (\( x[i] \) is \( i \)th largest component of \( x \))

**proof:**

\[ f(x) = \max\{x_{i_1} + x_{i_2} + \cdots + x_{i_r} \mid 1 \leq i_1 < i_2 < \cdots < i_r \leq n\} \]
Pointwise supremum

\[ g(x) = \sup_{y \in A} f(x, y) \]

is convex if \( f(x, y) \) is convex in \( x \) for each \( y \in A \)

**example:** maximum eigenvalue of symmetric matrix

\[ \lambda_{\text{max}}(X) = \sup_{\|y\|_2 = 1} y^T X y \]
Minimization

\[ h(x) = \inf_{y \in C} f(x, y) \]

is convex if \( f(x, y) \) is convex in \((x, y)\) and \(C\) is a convex set

eamples

- distance to a convex set \(C\): \( h(x) = \inf_{y \in C} \|x - y\| \)
- optimal value of linear program as function of righthand side

\[ h(x) = \inf_{y : Ay \leq x} c^T y \]

follows by taking

\[ f(x, y) = c^T y, \quad \text{dom } f = \{(x, y) \mid Ay \leq x\} \]
Composition

composition of \( g : \mathbb{R}^n \to \mathbb{R} \) and \( h : \mathbb{R} \to \mathbb{R} \):

\[
f(x) = h(g(x))
\]

\( f \) is convex if

- \( g \) convex, \( h \) convex and nondecreasing
- \( g \) concave, \( h \) convex and nonincreasing

(if we assign \( h(x) = \infty \) for \( x \in \text{dom} \ h \))

examples

- \( \exp g(x) \) is convex if \( g \) is convex
- \( 1/g(x) \) is convex if \( g \) is concave and positive
Vector composition

composition of $g: \mathbb{R}^n \to \mathbb{R}^k$ and $h: \mathbb{R}^k \to \mathbb{R}$:

$$f(x) = h(g(x)) = h(g_1(x), g_2(x), \ldots, g_k(x))$$

$f$ is convex if

- $g_i$ convex, $h$ convex and nondecreasing in each argument
- $g_i$ concave, $h$ convex and nonincreasing in each argument

(if we assign $h(x) = \infty$ for $x \in \text{dom} h$)

example

$$\log \sum_{i=1}^m \exp g_i(x)$$
is convex if $g_i$ are convex
the **perspective** of a function $f : \mathbb{R}^n \to \mathbb{R}$ is the function $g : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$,

$$g(x, t) = tf(x/t)$$

$g$ is convex if $f$ is convex on $\text{dom } g = \{(x,t) | x/t \in \text{dom } f, t > 0\}$

**examples**

- perspective of $f(x) = x^T x$ is quadratic-over-linear function

  $$g(x, t) = \frac{x^T x}{t}$$

- perspective of negative logarithm $f(x) = -\log x$ is relative entropy

  $$g(x, t) = t \log t - t \log x$$
Conjugate function

the **conjugate** of a function $f$ is

$$f^*(y) = \sup_{x \in \text{dom } f} (y^T x - f(x))$$

$f^*$ is convex (even if $f$ is not)
Examples

convex quadratic function \((Q \succ 0)\)

\[
f(x) = \frac{1}{2} x^T Q x \quad f^*(y) = \frac{1}{2} y^T Q^{-1} y
\]

negative entropy

\[
f(x) = \sum_{i=1}^{n} x_i \log x_i \quad f^*(y) = \sum_{i=1}^{n} e^{y_i} - 1
\]

norm

\[
f(x) = \|x\| \quad f^*(y) = \begin{cases} 
0 & \|y\|_* \leq 1 \\
+\infty & \text{otherwise}
\end{cases}
\]

indicator function \((C \text{ convex})\)

\[
f(x) = I_C(x) = \begin{cases} 
0 & x \in C \\
+\infty & \text{otherwise}
\end{cases} \quad f^*(y) = \sup_{x \in C} y^T x
\]
Convex optimization problems

- linear programming
- quadratic programming
- geometric programming
- second-order cone programming
- semidefinite programming
Convex optimization problem

minimize \( f_0(x) \)
subject to \( f_i(x) \leq 0, \quad i = 1, \ldots, m \)
\( Ax = b \)

\( f_0, f_1, \ldots, f_m \) are convex functions

- feasible set is convex
- locally optimal points are globally optimal
- tractable, in theory and practice
Linear program (LP)

minimize \( c^T x + d \)
subject to
\[
\begin{align*}
Gx &\leq h \\
Ax &= b
\end{align*}
\]

- inequality is componentwise vector inequality
- convex problem with affine objective and constraint functions
- feasible set is a polyhedron
Piecewise-linear minimization

\[
\begin{align*}
\text{minimize} \quad & f(x) = \max_{i=1,\ldots,m} (a_i^T x + b_i) \\
\text{subject to} \quad & a_i^T x + b_i \leq t, \quad i = 1, \ldots, m
\end{align*}
\]

an LP with variables \( x, t \in \mathbb{R} \)
\( \ell_1 \)-Norm and \( \ell_\infty \)-norm minimization

\( \ell_1 \)-norm approximation and equivalent LP (\( \| y \|_1 = \sum_k |y_k| \))

\[
\begin{align*}
\text{minimize} & \quad \| Ax - b \|_1 \\
\text{subject to} & \quad -y \leq Ax - b \leq y
\end{align*}
\]

\( \ell_\infty \)-norm approximation (\( \| y \|_\infty = \max_k |y_k| \))

\[
\begin{align*}
\text{minimize} & \quad \| Ax - b \|_\infty \\
\text{subject to} & \quad -y \mathbf{1} \leq Ax - b \leq y \mathbf{1}
\end{align*}
\]

(\( \mathbf{1} \) is vector of ones)
**example:** histograms of residuals $Ax - b$ (with $A$ is $200 \times 80$) for

$$x_{ls} = \text{argmin} \|Ax - b\|_2, \quad x_{\ell_1} = \text{argmin} \|Ax - b\|_1$$

1-norm distribution is wider with a high peak at zero
• 42 points $t_i, y_i$ (circles), including two outliers
• function $f(t) = \alpha + \beta t$ fitted using 2-norm (dashed) and 1-norm
Linear discrimination

separate two sets of points \( \{x_1, \ldots, x_N\}, \{y_1, \ldots, y_M\} \) by a hyperplane

\[
\begin{align*}
  a^T x_i + b &> 0, \quad i = 1, \ldots, N \\
  a^T y_i + b &< 0, \quad i = 1, \ldots, M
\end{align*}
\]

homogeneous in \( a, b \), hence equivalent to the linear inequalities (in \( a, b \))

\[
\begin{align*}
  a^T x_i + b &\geq 1, \quad i = 1, \ldots, N, \\
  a^T y_i + b &\leq -1, \quad i = 1, \ldots, M
\end{align*}
\]
Approximate linear separation of non-separable sets

\[
\text{minimize } \sum_{i=1}^{N} \max\{0, 1 - a^T x_i - b\} + \sum_{i=1}^{M} \max\{0, 1 + a^T y_i + b\}
\]

- a piecewise-linear minimization problem in \(a, b\); equivalent to an LP
- can be interpreted as a heuristic for minimizing \#misclassified points
Quadratic program (QP)

minimize \((1/2)x^T Px + q^T x + r\)
subject to \(Gx \leq h\)

- \(P \in S^n_+\), so objective is convex quadratic
- minimize a convex quadratic function over a polyhedron
Linear program with random cost

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Gx \leq h
\end{align*}
\]

- \( c \) is random vector with mean \( \bar{c} \) and covariance \( \Sigma \)
- hence, \( c^T x \) is random variable with mean \( \bar{c}^T x \) and variance \( x^T \Sigma x \)

expected cost-variance trade-off

\[
\begin{align*}
\text{minimize} & \quad \mathbb{E}c^T x + \gamma \text{var}(c^T x) = \bar{c}^T x + \gamma x^T \Sigma x \\
\text{subject to} & \quad Gx \leq h
\end{align*}
\]

\( \gamma > 0 \) is risk aversion parameter
Robust linear discrimination

\[ \mathcal{H}_1 = \{ z \mid a^T z + b = 1 \} \]
\[ \mathcal{H}_2 = \{ z \mid a^T z + b = -1 \} \]

distance between hyperplanes is \( 2/\|a\|_2 \)

to separate two sets of points by maximum margin,

\[
\begin{align*}
\text{minimize} & \quad \|a\|_2^2 = a^T a \\
\text{subject to} & \quad a^T x_i + b \geq 1, \quad i = 1, \ldots, N \\
& \quad a^T y_i + b \leq -1, \quad i = 1, \ldots, M
\end{align*}
\]

a quadratic program in \( a, b \)
Support vector classifier

\[
\min \gamma \|a\|_2^2 + \sum_{i=1}^{N} \max\{0, 1 - a^T x_i - b\} + \sum_{i=1}^{M} \max\{0, 1 + a^T y_i + b\}
\]

\(\gamma = 0\)

\(\gamma = 10\)

equivalent to a QP

Convex optimization problems
Total variation signal reconstruction

\[
\text{minimize} \quad \|\hat{x} - x_{\text{cor}}\|_2^2 + \gamma \phi(\hat{x})
\]

- \(x_{\text{cor}} = x + v\) is corrupted version of unknown signal \(x\), with noise \(v\)
- variable \(\hat{x}\) (reconstructed signal) is estimate of \(x\)
- \(\phi : \mathbb{R}^n \to \mathbb{R}\) is quadratic or total variation smoothing penalty

\[
\phi_{\text{quad}}(\hat{x}) = \sum_{i=1}^{n-1} (\hat{x}_{i+1} - \hat{x}_i)^2, \quad \phi_{\text{tv}}(\hat{x}) = \sum_{i=1}^{n-1} |\hat{x}_{i+1} - \hat{x}_i|
\]
example: $x_{\text{cor}}$, and reconstruction with quadratic and t.v. smoothing

- quadratic smoothing smooths out noise and sharp transitions in signal
- total variation smoothing preserves sharp transitions in signal
Geometric programming

posynomial function

\[
f(x) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}}, \quad \text{dom } f = \mathbb{R}^n_{++}
\]

with \( c_k > 0 \)

geometric program (GP)

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

with \( f_i \) posynomial
Geometric program in convex form

change variables to

\[ y_i = \log x_i, \]

and take logarithm of cost, constraints

**geometric program** in convex form:

\[
\begin{align*}
\text{minimize} & \quad \log \left( \sum_{k=1}^{K} \exp \left( a_{0k}^T y + b_{0k} \right) \right) \\
\text{subject to} & \quad \log \left( \sum_{k=1}^{K} \exp \left( a_{ik}^T y + b_{ik} \right) \right) \leq 0, \quad i = 1, \ldots, m \\
\end{align*}
\]

\[ b_{ik} = \log c_{ik} \]
Second-order cone program (SOCP)

minimize \( f^T x \)
subject to \( \| A_i x + b_i \|_2 \leq c^T_i x + d_i, \quad i = 1, \ldots, m \)

- \( \| \cdot \|_2 \) is Euclidean norm \( \| y \|_2 = \sqrt{y_1^2 + \cdots + y_n^2} \)
- constraints are nonlinear, nondifferentiable, convex

constraints are inequalities w.r.t. second-order cone:

\[ \left\{ y \mid \sqrt{y_1^2 + \cdots + y_{p-1}^2} \leq y_p \right\} \]
**Robust linear program (stochastic)**

minimize \( c^T x \)
subject to \( \text{prob}(a_i^T x \leq b_i) \geq \eta, \quad i = 1, \ldots, m \)

- \( a_i \) random and normally distributed with mean \( \bar{a}_i \), covariance \( \Sigma_i \)
- we require that \( x \) satisfies each constraint with probability exceeding \( \eta \)

\[ \eta = 10\% \quad \eta = 50\% \quad \eta = 90\% \]
SOCP formulation

the ‘chance constraint’ $\text{prob}(a_i^T x \leq b_i) \geq \eta$ is equivalent to the constraint

$$\bar{a}_i^T x + \Phi^{-1}(\eta) \| \Sigma_i^{-1/2} x \|_2 \leq b_i$$

$\Phi$ is the (unit) normal cumulative density function

robust LP is a second-order cone program for $\eta \geq 0.5$
Robust linear program (deterministic)

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad a_i^T x \leq b_i \text{ for all } a_i \in \mathcal{E}_i, \quad i = 1, \ldots, m
\end{align*}
\]

- \(a_i\) uncertain but bounded by ellipsoid \(\mathcal{E}_i = \{\tilde{a}_i + P_i u \mid \|u\|_2 \leq 1\}\)
- we require that \(x\) satisfies each constraint for all possible \(a_i\)

**SOCP formulation**

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad \tilde{a}_i^T x + \|P_i^T x\|_2 \leq b_i, \quad i = 1, \ldots, m
\end{align*}
\]

follows from

\[
\sup_{\|u\|_2 \leq 1} (\tilde{a}_i + P_i u)^T x = \tilde{a}_i^T x + \|P_i^T x\|_2
\]
Examples of second-order cone constraints

**convex quadratic constraint** \( (A = LL^T \text{ positive definite}) \)

\[
x^T Ax + 2b^T x + c \leq 0
\]
\[
\Leftrightarrow
\]
\[
\|L^T x + L^{-1} b\|_2 \leq (b^T A^{-1} b - c)^{1/2}
\]

extends to positive semidefinite singular \( A \)

**hyperbolic constraint**

\[
x^T x \leq yz, \quad y, z \geq 0
\]
\[
\Leftrightarrow
\]
\[
\left\| \begin{bmatrix} 2x \\ y - z \end{bmatrix} \right\|_2 \leq y + z, \quad y, z \geq 0
\]
Examples of SOC-representable constraints

positive powers

\[ x^{1.5} \leq t, \quad x \geq 0 \]

\[ \exists z : \ x^2 \leq tz, \quad z^2 \leq x, \quad x, z \geq 0 \]

- two hyperbolic constraints can be converted to SOC constraints
- extends to powers \( x^p \) for rational \( p \geq 1 \)

negative powers

\[ x^{-3} \leq t, \quad x > 0 \]

\[ \exists z : \ 1 \leq tz, \quad z^2 \leq tx, \quad x, z \geq 0 \]

- two hyperbolic constraints on r.h.s. can be converted to SOC constraints
- extends to powers \( x^p \) for rational \( p < 0 \)
Semidefinite program (SDP)

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad x_1 A_1 + x_2 A_2 + \cdots + x_n A_n \preceq B
\end{align*}
\]

- \( A_1, A_2, \ldots, A_n, B \) are symmetric matrices
- Inequality \( X \preceq Y \) means \( Y - X \) is positive semidefinite, i.e.,
  \[
  z^T (Y - X) z = \sum_{i,j} (Y_{ij} - X_{ij}) z_i z_j \geq 0 \quad \text{for all} \quad z
  \]
- Includes many nonlinear constraints as special cases
• a nonpolyhedral convex cone

• feasible set of a semidefinite program is the intersection of the positive semidefinite cone in high dimension with planes
Examples

\[ A(x) = A_0 + x_1 A_1 + \cdots + x_m A_m \quad (A_i \in \mathbb{S}^n) \]

eigenvalue minimization (and equivalent SDP)

\[
\begin{align*}
\text{minimize} & \quad \lambda_{\text{max}}(A(x)) & \text{minimize} & \quad t \\
\text{subject to} & \quad A(x) \preceq tI
\end{align*}
\]

matrix-fractional function

\[
\begin{align*}
\text{minimize} & \quad b^T A(x)^{-1} b & \text{minimize} & \quad t \\
\text{subject to} & \quad A(x) \succeq 0 & \text{subject to} & \quad \begin{bmatrix} A(x) & b \\ b^T & t \end{bmatrix} \succeq 0
\end{align*}
\]
Matrix norm minimization

\[ A(x) = A_0 + x_1 A_1 + x_2 A_2 + \cdots + x_n A_n \quad (A_i \in \mathbb{R}^{p \times q}) \]

**Matrix norm approximation** \( (\|X\|_2 = \max_k \sigma_k(X)) \)

- minimize \( \|A(x)\|_2 \)
- subject to \( \begin{bmatrix} tI & A(x)^T \\ A(x) & tI \end{bmatrix} \succeq 0 \)

**Nuclear norm approximation** \( (\|X\|_* = \sum_k \sigma_k(X)) \)

- minimize \( \|A(x)\|_* \)
- subject to \( \begin{bmatrix} U & A(x)^T \\ A(x) & V \end{bmatrix} \succeq 0 \)

Convex optimization problems
Semidefinite relaxations

Semidefinite programming is often used

- to find good bounds for nonconvex polynomial problems, via relaxation
- as a heuristic for good suboptimal points

example: Boolean least-squares

\[
\begin{align*}
\text{minimize} & \quad \|Ax - b\|_2^2 \\
\text{subject to} & \quad x_i^2 = 1, \quad i = 1, \ldots, n
\end{align*}
\]

- basic problem in digital communications
- could check all \(2^n\) possible values of \(x \in \{-1, 1\}^n\) \ldots
- an NP-hard problem, and very hard in general
Semidefinite lifting

Boolean least-squares problem

\[
\begin{align*}
\text{minimize} & \quad x^T A^T A x - 2b^T A x + b^T b \\
\text{subject to} & \quad x_i^2 = 1, \quad i = 1, \ldots, n
\end{align*}
\]

reformulation: introduce new variable \( Y = x x^T \)

\[
\begin{align*}
\text{minimize} & \quad \text{tr}(A^T A Y) - 2b^T A x + b^T b \\
\text{subject to} & \quad Y = x x^T \\
& \quad \text{diag}(Y) = 1
\end{align*}
\]

- cost function and second constraint are linear (in the variables \( Y, x \))
- first constraint is nonlinear and nonconvex

\ldots still a very hard problem
Semidefinite relaxation

replace $Y = xx^T$ with weaker constraint $Y \succeq xx^T$ to obtain relaxation

minimize $\text{tr}(A^TAY) - 2b^TAx + b^Tb$
subject to $Y \succeq xx^T$
$\text{diag}(Y) = 1$

- convex; can be solved as a semidefinite program

$Y \succeq xx^T \iff \begin{bmatrix} Y & x \\ x^T & 1 \end{bmatrix} \succeq 0$

- optimal value gives lower bound for Boolean LS problem
- if $Y = xx^T$ at the optimum, we have solved the exact problem
- otherwise, can use randomized rounding
  generate $z$ from $\mathcal{N}(x, Y - xx^T)$ and take $x = \text{sign}(z)$
Example

$n = 100$: feasible set has $2^{100} \approx 10^{30}$ points

• histogram of 1000 randomized solutions from SDP relaxation
Overview

1. Basic theory and convex modeling
   - convex sets and functions
   - common problem classes and applications

2. Interior-point methods for conic optimization
   - conic optimization
   - barrier methods
   - symmetric primal-dual methods

3. First-order methods
   - gradient algorithms
   - dual techniques
Conic optimization

- definitions and examples
- modeling
- duality
Generalized (conic) inequalities

**Conic inequality:** a constraint $x \in K$ with $K$ a convex cone in $\mathbb{R}^m$

we require that $K$ is a **proper** cone:

- closed
- pointed: $K \cap (-K) = \{0\}$
- with nonempty interior: $\text{int} K \neq \emptyset$; equivalently, $K + (-K) = \mathbb{R}^m$

**Notation**

$x \succeq_K y \iff x - y \in K,$

$x \succ_K y \iff x - y \in \text{int} K$

with subscript in $\succeq_K$ omitted if $K$ is clear from the context.
Cone linear program

\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad Ax \preceq_K b
\end{align*}

if $K$ is the nonnegative orthant, this reduces to regular linear program

widely used in recent literature on convex optimization

- **modeling**: a small number of ‘primitive’ cones is sufficient to express most convex constraints that arise in practice

- **algorithms**: a convenient problem format for extending interior-point algorithms for linear programming to convex optimization
Norm cones

\[ K = \{(x, y) \in \mathbb{R}^{m-1} \times \mathbb{R} \mid \|x\| \leq y\} \]

for the Euclidean norm this is the second-order cone (notation: \( Q^m \))
Second-order cone program

\[
\begin{align*}
\text{minimize} & \quad c^T x \\
\text{subject to} & \quad \|B_{k0}x + d_{k0}\|_2 \leq B_{k1}x + d_{k1}, \quad k = 1, \ldots, r
\end{align*}
\]

**cone LP formulation:** express constraints as \(Ax \preceq_K b\)

\[
K = Q^{m_1} \times \cdots \times Q^{m_r}, \quad A = \begin{bmatrix}
-B_{10} \\
-B_{11} \\
\vdots \\
-B_{r0} \\
-B_{r1}
\end{bmatrix}, \quad b = \begin{bmatrix}
d_{10} \\
d_{11} \\
\vdots \\
d_{r0} \\
d_{r1}
\end{bmatrix}
\]

(assuming \(B_{k0}, d_{k0}\) have \(m_k - 1\) rows)
Vector notation for symmetric matrices

- vectorized symmetric matrix: for $U \in \mathbb{S}^p$

$$\text{vec}(U) = \sqrt{2} \left( \frac{U_{11}}{\sqrt{2}}, U_{21}, \ldots, U_{p1}, \frac{U_{22}}{\sqrt{2}}, U_{32}, \ldots, U_{p2}, \ldots, \frac{U_{pp}}{\sqrt{2}} \right)$$

- inverse operation: for $u = (u_1, u_2, \ldots, u_n) \in \mathbb{R}^n$ with $n = p(p + 1)/2$

$$\text{mat}(u) = \frac{1}{\sqrt{2}} \begin{bmatrix} \sqrt{2}u_1 & u_2 & \cdots & u_p \\ u_2 & \sqrt{2}u_{p+1} & \cdots & u_{2p-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_p & u_{2p-1} & \cdots & \sqrt{2}u_{p(p+1)/2} \end{bmatrix}$$

coefficients $\sqrt{2}$ are added so that standard inner products are preserved:

$$\text{tr}(UV) = \text{vec}(U)^T \text{vec}(V), \quad u^Tv = \text{tr}(\text{mat}(u) \text{mat}(v))$$
Positive semidefinite cone

\[ S^p = \{ \text{vec}(X) \mid X \in S^p_+ \} = \{ x \in \mathbb{R}^{p(p+1)/2} \mid \text{mat}(x) \succeq 0 \} \]

\[ S^2 = \left\{ (x, y, z) \left| \begin{bmatrix} x & y/\sqrt{2} & z \\ y/\sqrt{2} & z \\ z \end{bmatrix} \succeq 0 \right. \right\} \]
Semidefinite program

minimize \( c^T x \)
subject to \( x_1 A_{11} + x_2 A_{12} + \cdots + x_n A_{1n} \preceq B_1 \)
\[ \vdots \]
\( x_1 A_{r1} + x_2 A_{r2} + \cdots + x_n A_{rn} \preceq B_r \)

\( r \) linear matrix inequalities of order \( p_1, \ldots, p_r \)

cone LP formulation: express constraints as \( Ax \preceq_K B \)

\[ K = S^{p_1} \times S^{p_2} \times \cdots \times S^{p_r} \]

\[ A = \begin{bmatrix} \text{vec}(A_{11}) & \text{vec}(A_{12}) & \cdots & \text{vec}(A_{1n}) \\ \text{vec}(A_{21}) & \text{vec}(A_{22}) & \cdots & \text{vec}(A_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{vec}(A_{r1}) & \text{vec}(A_{r2}) & \cdots & \text{vec}(A_{rn}) \end{bmatrix}, \quad b = \begin{bmatrix} \text{vec}(B_1) \\ \text{vec}(B_2) \\ \vdots \\ \text{vec}(B_r) \end{bmatrix} \]
Exponential cone

the epigraph of the perspective of $\exp x$ is a non-proper cone

$$K = \left\{ (x, y, z) \in \mathbb{R}^3 \mid ye^{x/y} \leq z, \ y > 0 \right\}$$

the exponential cone is $K_{\exp} = \text{cl} K = K \cup \{(x, 0, z) \mid x \leq 0, z \geq 0\}$
**Geometric program**

minimize \( c^T x \)

subject to \( \log \sum_{k=1}^{n_i} \exp(a_{ik}^T x + b_{ik}) \leq 0, \quad i = 1, \ldots, r \)

**cone LP formulation**

minimize \( c^T x \)

subject to \[
\begin{bmatrix}
a_{ik}^T x + b_{ik} \\
1 \\
z_{ik}
\end{bmatrix} \in K_{\exp}, \quad k = 1, \ldots, n_i, \quad i = 1, \ldots, r
\]

\[\sum_{k=1}^{n_i} z_{ik} \leq 1, \quad i = 1, \ldots, m\]
**Power cone**

**definition:** for $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m) > 0, \sum_{i=1}^{m} \alpha_i = 1$

$$K_\alpha = \{(x,y) \in \mathbb{R}_+^m \times \mathbb{R} \mid |y| \leq x_1^{\alpha_1} \cdots x_m^{\alpha_m}\}$$

**examples** for $m = 2$

$\alpha = (\frac{1}{2}, \frac{1}{2})$

$\alpha = (\frac{2}{3}, \frac{1}{3})$

$\alpha = (\frac{3}{4}, \frac{1}{4})$
Outline

• definition and examples

• modeling

• duality
Modeling software

modeling packages for convex optimization

- CVX, YALMIP (MATLAB)
- CVXMOD, CVXPY (Python)

assist in formulating convex problems by automating two tasks:
- verifying convexity from convex calculus rules
- transforming problem in input format required by standard solvers

related packages

general-purpose optimization modeling: AMPL, GAMS
**CVX example**

minimize $\|Ax - b\|_1$
subject to $0 \leq x_k \leq 1, \ k = 1, \ldots, n$

**MATLAB code**

cvx_begin
    variable x(3);
    minimize(norm(A*x - b, 1))
    subject to
        x >= 0;
        x <= 1;

    cvx_end

- between cvx_begin and cvx_end, x is a CVX variable
- after execution, x is MATLAB variable with optimal solution
convex modeling systems (CVX, YALMIP, CVXMOD, CVXPY, \ldots )

- convert problems stated in standard mathematical notation to cone LPs
- in principle, any convex problem can be represented as a cone LP
- in practice, a small set of primitive cones is used ($\mathbb{R}_+^n$, $\mathbb{Q}^p$, $\mathbb{S}^p$)
- choice of cones is limited by available algorithms and solvers (see later)

modeling systems implement set of rules for expressing constraints

$$f(x) \leq t$$

as conic inequalities for the implemented cones
Examples of second-order cone representable functions

- convex quadratic

\[ f(x) = x^T P x + q^T x + r \quad (P \succeq 0) \]

- quadratic-over-linear function

\[ f(x, y) = \frac{x^T x}{y} \quad \text{with dom } f = \mathbb{R}^n \times \mathbb{R}_+ \quad (\text{assume } 0/0 = 0) \]

- convex powers with rational exponent

\[ f(x) = |x|^\alpha, \quad f(x) = \begin{cases} x^\beta & x > 0 \\ +\infty & x \leq 0 \end{cases} \]

for rational \( \alpha \geq 1 \) and \( \beta \leq 0 \)

- p-norm \( f(x) = \|x\|_p \) for rational \( p \geq 1 \)
Examples of SD cone representable functions

- matrix-fractional function
  \[ f(X, y) = y^T X^{-1} y \quad \text{with} \quad \text{dom} \ f = \{(X, y) \in S_+^n \times \mathbb{R}^n \mid y \in \mathcal{R}(X)\} \]

- maximum eigenvalue of symmetric matrix

- maximum singular value \( f(X) = \|X\|_2 = \sigma_1(X) \)

  \[ \|X\|_2 \leq t \iff \begin{bmatrix} tI & X \\ X^T & tI \end{bmatrix} \succeq 0 \]

- nuclear norm \( f(X) = \|X\|_* = \sum_i \sigma_i(X) \)

  \[ \|X\|_* \leq t \iff \exists U, V : \begin{bmatrix} U & X \\ X^T & V \end{bmatrix} \succeq 0, \quad \frac{1}{2}(\text{tr} \ U + \text{tr} \ V) \leq t \]
Functions representable with exponential and power cone

exponential cone

- exponential and logarithm

- entropy $f(x) = x \log x$

power cone

- increasing power of absolute value: $f(x) = |x|^p$ with $p \geq 1$

- decreasing power: $f(x) = x^q$ with $q \leq 0$ and domain $\mathbb{R}_{++}$

- $p$-norm: $f(x) = \|x\|_p$ with $p \geq 1$
Outline

• definition and examples
• modeling
• duality
Linear programming duality

primal and dual LP

\begin{align*}
\text{(P)} \quad & \text{minimize} & c^T x \\
& \text{subject to} & Ax \leq b
\end{align*}

\begin{align*}
\text{(D)} \quad & \text{maximize} & -b^T z \\
& \text{subject to} & A^T z + c = 0 \\
& & z \geq 0
\end{align*}

- primal optimal value is $p^*$ ($+\infty$ if infeasible, $-\infty$ if unbounded below)
- dual optimal value is $d^*$ ($-\infty$ if infeasible, $+\infty$ if unbounded below)

duality theorem

- weak duality: $p^* \geq d^*$, with no exception
- strong duality: $p^* = d^*$ if primal or dual is feasible
- if $p^* = d^*$ is finite, then primal and dual optima are attained
Dual cone

**definition**

\[ K^* = \{ y \mid x^T y \geq 0 \text{ for all } x \in K \} \]

a proper cone if \( K \) is a proper cone

**dual inequality:** \( x \succeq_* y \) means \( x \succeq_{K^*} y \) for generic proper cone \( K \)

note: dual cone depends on choice of inner product:

\[ H^{-1}K^* \]

is dual cone for inner product \( \langle x, y \rangle = x^T H y \)
Examples

• $\mathbb{R}^p_+, \mathbb{Q}^p, \mathbb{S}^p$ are self-dual: $K = K^*$

• dual of norm cone is norm cone for dual norm

• dual of exponential cone

$$K^*_{\text{exp}} = \{ (u, v, w) \in \mathbb{R}_- \times \mathbb{R} \times \mathbb{R}^+ | -u \log(-u/w) + u - v \leq 0 \}$$

(with $0 \log(0/w) = 0$ if $w \geq 0$)

• dual of power cone is

$$K^*_{\alpha} = \{ (u, v) \in \mathbb{R}^m_+ \times \mathbb{R} | |v| \leq (u_1/\alpha_1)^{\alpha_1} \cdots (u_m/\alpha_m)^{\alpha_m} \}$$
Primal and dual cone LP

**primal problem** (optimal value $p^*$)

minimize $c^T x$
subject to $Ax \leq b$

**dual problem** (optimal value $d^*$)

maximize $-b^T z$
subject to $A^T z + c = 0$
$z \succeq_0$

**weak duality**: $p^* \geq d^*$ (without exception)
Strong duality

\[ p^* = d^* \]

if primal or dual is strictly feasible

- slightly weaker than LP duality (which only requires feasibility)
- can have \( d^* < p^* \) with finite \( p^* \) and \( d^* \)

Other implications of strict feasibility

- if primal is strictly feasible, then dual optimum is attained (if \( d^* \) is finite)
- if dual is strictly feasible, then primal optimum is attained (if \( p^* \) is finite)
**Optimality conditions**

minimize \( c^T x \)  
subject to  
\( Ax + s = b \)  
\( s \geq 0 \)

maximize \( -b^T z \)  
subject to  
\( A^T z + c = 0 \)  
\( z \geq * 0 \)

**optimality conditions**

\[
\begin{bmatrix}
0 \\
s
\end{bmatrix} = 
\begin{bmatrix}
0 & A^T \\
-A & 0
\end{bmatrix} 
\begin{bmatrix}
x \\
z
\end{bmatrix} + 
\begin{bmatrix}
c \\
b
\end{bmatrix}
\]

\( s \geq 0, \quad z \geq * 0, \quad z^T s = 0 \)

**duality gap:** inner product of \((x, z)\) and \((0, s)\) gives

\[ z^T s = c^T x + b^T z \]
Barrier methods

• barrier method for linear programming

• normal barriers

• barrier method for conic optimization
History

• 1960s: Sequentially Unconstrained Minimization Technique (SUMT) solves nonlinear convex optimization problem

\[
\text{minimize } \quad f_0(x) \\
\text{subject to } \quad f_i(x) \leq 0, \quad i = 1, \ldots, m
\]

via a sequence of unconstrained minimization problems

\[
\text{minimize } \quad t f_0(x) - \sum_{i=1}^{m} \log(-f_i(x))
\]

• 1980s: LP barrier methods with polynomial worst-case complexity

• 1990s: barrier methods for non-polyhedral cone LPs
Logarithmic barrier function for linear inequalities

\[ \psi(x) = \phi(b - Ax), \quad \phi(s) = -\sum_{i=1}^{m} \log s_i \]

- a smooth convex function with \( \text{dom} \psi = \{ x \mid Ax < b \} \)
- \( \psi(x) \to \infty \) at boundary of \( \text{dom} \psi \)
- gradient and Hessian are

\[ \nabla \psi(x) = -A^T \nabla \phi(s), \quad \nabla^2 \psi(x) = A^T \nabla \phi^2(s) A \]

with \( s = b - Ax \)

\[ \nabla \phi(s) = -\left( \frac{1}{s_1}, \ldots, \frac{1}{s_m} \right), \quad \nabla \phi^2(s) = \text{diag} \left( \frac{1}{s_1^2}, \ldots, \frac{1}{s_m^2} \right) \]
Central path for linear program

**central path:** set of minimizers $x^*(t)$ (with $t > 0$) of

$$f_t(x) = tc^T x + \phi(b - Ax)$$

**optimality conditions:** $x = x^*(t)$ satisfies

$$\nabla f_t(x) = tc - A^T \nabla \phi(s) = 0, \quad s = b - Ax$$
Central path and duality

dual feasible point on central path

- for $x = x^*(t)$ and $s = b - Ax$,

$$z^*(t) = -\frac{1}{t} \nabla \phi(s) = \left( \frac{1}{ts_1}, \frac{1}{ts_2}, \ldots, \frac{1}{ts_m} \right)$$

is strictly dual feasible: $c + A^Tz = 0$ and $z > 0$

- can be modified to correct for inexact centering of $x$

duality gap between $x = x^*(t)$ and $z = z^*(t)$ is

$$c^T x + b^T z = s^T z = \frac{m}{t}$$

gives bound on suboptimality: $c^T x^*(t) - p^* \leq m/t$
Barrier method

starting with \( t > 0 \), strictly feasible \( x \), repeat until \( c^T x - p^* \leq \epsilon \)

- make one or more Newton steps to (approximately) minimize \( f_t \):
  \[
x^+ = x - \alpha \nabla^2 f_t(x)^{-1} \nabla f_t(x)
  \]

  step size \( \alpha \) is fixed or from line search

- increase \( t \)

**complexity**: with proper initialization, step size, update scheme for \( t \),

\[
\#\text{Newton steps} = O\left(\sqrt{m \log(1/\epsilon)}\right)
\]

result follows from convergence analysis of Newton’s method for \( f_t \)
Outline

- barrier method for linear programming
- normal barriers
- barrier method for conic optimization
Normal barrier for proper cone

\( \phi \) is a \( \theta \)-normal barrier for the proper cone \( K \) if it is

- a **barrier**: smooth, convex, domain \( \text{int} \ K \), blows up at boundary of \( K \)

- **logarithmically homogeneous** with parameter \( \theta \):

\[
\phi(tx) = \phi(x) - \theta \log t, \quad \forall x \in \text{int} \ K, \ t > 0
\]

- **self-concordant**: restriction \( g(\alpha) = \phi(x + \alpha v) \) to any line satisfies

\[
g'''(\alpha) \leq 2g''(\alpha)^{3/2}
\]

introduced by Nesterov and Nemirovski (1994)
Examples

nonnegative orthant: \( K = \mathbb{R}_+^m \)

\[
\phi(x) = - \sum_{i=1}^{m} \log x_i \quad (\theta = m)
\]

second-order cone: \( K = Q^p = \{(x, y) \in \mathbb{R}^{p-1} \times \mathbb{R} | \|x\|_2 \leq y\} \)

\[
\phi(x, y) = - \log(y^2 - x^T x) \quad (\theta = 2)
\]

semidefinite cone: \( K = S^m = \{x \in \mathbb{R}^{m(m+1)/2} | \text{mat}(x) \succeq 0\} \)

\[
\phi(x) = - \log \det \text{mat}(x) \quad (\theta = m)
\]
**exponential cone:** \( K_{\text{exp}} = \text{cl}\{(x, y, z) \in \mathbb{R}^3 \mid ye^{x/y} \leq z, \ y > 0\} \)

\[
\phi(x, y, z) = -\log(y \log(z/y) - x) - \log z - \log y \quad (\theta = 3)
\]

**power cone:** \( K = \{(x_1, x_2, y) \in \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R} \mid |y| \leq x_1^{\alpha_1} x_2^{\alpha_2}\} \)

\[
\phi(x, y) = -\log\left(x_1^{2\alpha_1} x_2^{2\alpha_2} - y^2\right) - \log x_1 - \log x_2 \quad (\theta = 4)
\]
Central path

cone LP (with inequality with respect to proper cone $K$)

\[
\text{minimize} \quad c^T x \\
\text{subject to} \quad Ax \preceq b
\]

barrier for the feasible set

\[
\phi(b - Ax)
\]

where $\phi$ is a $\theta$-normal barrier for $K$

central path: set of minimizers $x^*(t)$ (with $t > 0$) of

\[
f_t(x) = tc^T x + \phi(b - Ax)
\]
Newton step

centering problem

minimize \( f_t(x) = tc^T x + \phi(b - Ax) \)

Newton step at \( x \)

\[
\Delta x = -\nabla^2 f_t(x)^{-1} \nabla f_t(x)
\]

Newton decrement

\[
\lambda_t(x) = \left( \Delta x^T \nabla^2 f_t(x) \Delta x \right)^{1/2} = \left( -\nabla f_t(x)^T \Delta x \right)^{1/2}
\]

used to measure proximity of \( x \) to \( x^*(t) \)
Damped Newton method

minimize $f_t(x) = tc^T x + \phi(b - Ax)$

algorithm

select $\epsilon \in (0, 1/2)$, $\eta \in (0, 1/4]$, and a starting point $x \in \text{dom } f_t$

repeat:

1. compute Newton step $\Delta x$ and Newton decrement $\lambda_t(x)$
2. if $\lambda_t(x)^2 \leq \epsilon$, return $x$
3. otherwise, set $x := x + \alpha \Delta x$ with

$$\alpha = \frac{1}{1 + \lambda_t(x)} \quad \text{if } \lambda_t(x) \geq \eta, \quad \alpha = 1 \quad \text{if } \lambda_t(x) < \eta$$

alternatively, can use backtracking line search
Convergence results for damped Newton method

- **damped Newton phase**

  \[ f_t(x^+) - f_t(x) \leq -\gamma \quad \text{if } \lambda_t(x) \geq \eta \]

  where \( \gamma = \eta - \log(1 + \eta) \); \( f_t \) decreases by at least a positive constant \( \gamma \)

- **quadratically convergent phase**

  \[ 2\lambda_t(x^+) \leq (2\lambda_t(x))^2 \quad \text{if } \lambda_t(x) < \eta \]

  implies \( \lambda_t(x^+) \leq 2\eta^2 < \eta \), and Newton decrement decreases to zero

- **stopping criterion** \( \lambda_t(x)^2 \leq \epsilon \) implies

  \[ f_t(x) - \inf f_t(x) \leq \epsilon \]
Outline

• barrier method for linear programming

• normal barriers

• barrier method for conic optimization
Central path and duality

duality point on central path: \( x^*(t) \) defines a strictly dual feasible \( z^*(t) \)

\[
z^*(t) = -\frac{1}{t} \nabla \phi(s), \quad s = b - Ax^*(t)
\]

duality gap: gap between \( x = x^*(t) \) and \( z = z^*(t) \) is

\[
c^T x + b^T z = s^T z = \frac{\theta}{t}, \quad c^T x - p^* \leq \frac{\theta}{t}
\]

near central path: for inexact centrally centered \( x \)

\[
c^T x - p^* \leq \left(1 + \frac{\lambda_t(x)}{\sqrt{\theta}}\right) \frac{\theta}{t} \quad \text{if } \lambda_t(x) < 1
\]

(results follow from properties of normal barriers)
Short-step barrier method

**algorithm:** parameters $\epsilon \in (0, 1)$, $\beta = 1/8$

- select initial $x$ and $t$ with $\lambda_t(x) \leq \beta$
- repeat until $2\theta/t \leq \epsilon$:

$$t := \left(1 + \frac{1}{1 + 8\sqrt{\theta}}\right) t, \quad x := x - \nabla f_t(x)^{-1} \nabla f_t(x)$$

**properties**

- increase $t$ slowly so $x$ stays in region of quadratic region ($\lambda_t(x) \leq \beta$)
- iteration complexity

$$\text{#iterations} = O \left( \sqrt{\theta} \log \left( \frac{\theta}{\epsilon t_0} \right) \right)$$

- best known worst-case complexity; same as for linear programming
Predictor-corrector methods

short-step barrier methods

- stay in narrow neighborhood of central path (defined by limit on $\lambda_t$)
- make small, fixed increases $t^+ = \mu t$

as a result, quite slow in practice

predictor-corrector method

- select new $t$ using a linear approximation to central path (‘predictor’)
- re-center with new $t$ (‘corrector’)

allows faster and ‘adaptive’ increases in $t$; similar worst-case complexity
Primal-dual methods

- primal-dual algorithms for linear programming
- symmetric cones
- primal-dual algorithms for conic optimization
- implementation
Primal-dual interior-point methods

similarities with barrier method

• follow the same central path
• same linear algebra cost per iteration

differences

• more robust and faster (typically less than 50 iterations)
• primal and dual iterates updated at each iteration
• symmetric treatment of primal and dual iterates
• can start at infeasible points
• include heuristics for adaptive choice of central path parameter $t$
• often have superlinear asymptotic convergence
Primal-dual central path for linear programming

minimize \( c^T x \) \quad \text{maximize} \quad \( -b^T z \)
subject to \( Ax + s = b \) \quad \text{subject to} \quad \( A^T z + c = 0 \)
\( s \geq 0 \) \quad \( z \geq 0 \)

optimality conditions

\[ Ax + s = b, \quad A^T z + c = 0, \quad (s, z) \geq 0, \quad s \circ z = 0 \]

\( s \circ z \) is component-wise vector product

primal-dual parametrization of central path

\[ Ax + s = b, \quad A^T z + c = 0, \quad (s, z) \geq 0, \quad s \circ z = \frac{1}{t} \mathbf{1} \]

solution is \( x = x^*(t), \ z = z^*(t) \)
Primal-dual search direction

steps solve central path equations linearized around current iterates \( x, s, z \)

\[
A(x + \Delta x) + s + \Delta s = b, \quad A^T(z + \Delta z) + c = 0
\]

\[(s + \Delta z) \circ (z + \Delta z) = \sigma \mu 1\]

where \( \mu = (s^T z)/m \) and \( \sigma \in [0, 1] \)

- targets point on central path with \( 1/t = \sigma \mu, \ i.e., \ with \ gap \ \sigma s^T z \)
- different methods use different strategies for selecting \( \sigma \)

linearized equations: the two linear equations in (1) and

\[
z \circ \Delta s + s \circ \Delta z = \sigma \mu 1 - s \circ z
\]

after eliminating \( \Delta s, \Delta z \) this reduces to an equation

\[
A^T D A \Delta x = r, \quad D = \text{diag}(z_1/s_1, \ldots, z_m/s_m)
\]
Outline

- primal-dual algorithms for linear programming
- symmetric cones
- primal-dual algorithms for conic optimization
- implementation
Symmetric cones

Symmetric primal-dual solvers for cone LPs are limited to symmetric cones

- second-order cone
- positive semidefinite cone
- direct products of these ‘primitive’ symmetric cones (such as $\mathbb{R}^p_+$)

**definition:** cone of squares $x = y^2 = y \circ y$ for a product $\circ$ that satisfies

1. bilinearity ($x \circ y$ is linear in $x$ for fixed $y$ and vice-versa)
2. $x \circ y = y \circ x$
3. $x^2 \circ (y \circ x) = (x^2 \circ y) \circ x$
4. $x^T (y \circ z) = (x \circ y)^T z$

not necessarily associative
Vector product and identity element

**nonnegative orthant:** componentwise product

\[ x \circ y = \text{diag}(x)y \]

identity element is \( e = 1 = (1, 1, \ldots, 1) \)

**positive semidefinite cone:** symmetrized matrix product

\[ x \circ y = \frac{1}{2} \text{vec}(XY + YX) \quad \text{with} \quad X = \text{mat}(x), \; Y = \text{mat}(Y) \]

identity element is \( e = \text{vec}(I) \)

**second-order cone:** the product of \( x = (x_0, x_1) \) and \( y = (y_0, y_1) \) is

\[ x \circ y = \frac{1}{\sqrt{2}} \begin{bmatrix} x^T y \\ x_0 y_1 + y_0 x_1 \end{bmatrix} \]

identity element is \( e = (\sqrt{2}, 0, \ldots, 0) \)
Classification

- symmetric cones are studied in the theory of Euclidean Jordan algebras
- all possible symmetric cones have been characterized

**list of symmetric cones**

- the second-order cone
- the positive semidefinite cone of Hermitian matrices with real, complex, or quaternion entries
- $3 \times 3$ positive semidefinite matrices with octonion entries
- Cartesian products of these ‘primitive’ symmetric cones (such as $\mathbb{R}_+^p$)

**practical implication**

can focus on $\mathbb{Q}_+^p$, $\mathbb{S}_+^p$ and study these cones using elementary linear algebra
Spectral decomposition

with each symmetric cone/product we associate a ‘spectral’ decomposition

\[ x = \sum_{i=1}^{\theta} \lambda_i q_i, \quad \text{with} \quad \sum_{i=1}^{\theta} q_i = e \quad \text{and} \quad q_i \circ q_j = \begin{cases} q_i & i = j \\ 0 & i \neq j \end{cases} \]

semidefinite cone \((K = S^p)\): eigenvalue decomposition of \(\text{mat}(x)\)

\[ \theta = p, \quad \text{mat}(x) = \sum_{i=1}^{p} \lambda_i v_i v_i^T, \quad q_i = \text{vec}(v_i v_i^T) \]

second-order cone \((K = Q^p)\)

\[ \theta = 2, \quad \lambda_i = \frac{x_0 \pm \|x_1\|_2}{\sqrt{2}}, \quad q_i = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ \pm x_1/\|x_1\|_2 \end{bmatrix}, \quad i = 1, 2 \]
Applications

nonnegativity

\[ x \succeq 0 \iff \lambda_1, \ldots, \lambda_\theta \geq 0, \quad x \succ 0 \iff \lambda_1, \ldots, \lambda_\theta > 0 \]

powers (in particular, inverse and square root)

\[ x^\alpha = \sum_i \lambda_i^\alpha q_i \]

log-det barrier

\[ \phi(x) = -\log \det x = -\sum_{i=1}^{\theta} \log \lambda_i \]

- a \( \theta \)-normal barrier
- gradient is \( \nabla \phi(x) = -x^{-1} \)
Outline

- primal-dual algorithms for linear programming
- symmetric cones
- primal-dual algorithms for conic optimization
- implementation
Symmetric parametrization of central path

centering problem

\[ \text{minimize } tc^T x + \phi(b - Ax) \]

optimality conditions (using \( \nabla \phi(s) = -s^{-1} \))

\[ Ax + s = b, \quad A^T z + c = 0, \quad (s, z) \succ 0, \quad z = \frac{1}{t} s^{-1} \]

equivalent symmetric form

\[ Ax + b = s, \quad A^T z + c = 0, \quad (s, z) \succ 0, \quad s \circ z = \frac{1}{t} e \]
Scaling with Hessian

linear transformation with $H = \nabla^2 \phi(u)$ has several important properties

- preserves conic inequalities: $s \succ 0 \iff Hs \succ 0$

- if $s$ is invertible, then $Hs$ is invertible and $(Hs)^{-1} = H^{-1}s^{-1}$

- preserves central path:

$$s \circ z = \mu e \iff (Hs) \circ (H^{-1}z) = \mu e$$

- symmetric square root of $H$ is $H^{1/2} = \nabla^2 \phi(u^{1/2})$

example $(K = S^p)$:

$$\tilde{S} = U^{-1}SU^{-1} \quad S = \text{mat}(s), \quad U = \text{mat}(u)$$
Primal-dual search direction

steps solve central path equations linearized around current iterates $x, s, z$

\[
A(x + \Delta x) + s + \Delta s = b, \quad A^T(z + \Delta z) + c = 0 \quad (2)
\]

\[
(H(s + \Delta s)) \circ (H^{-1}(z + \Delta z)) = \sigma \mu e
\]

where $\mu = (s^T z)/m$, $\sigma \in [0, 1]$, and $H = \nabla^2 \phi(u)$

- different algorithms use different choices of $\sigma, u$
- Nesterov-Todd scaling: $H = \nabla^2 \phi(u)$ defined by $Hs = H^{-1} z$

**linearized equations:** the two linear equations (2) and

\[
(Hs) \circ (H^{-1} \Delta z) + (H^{-1} z) \circ (H \Delta s) = \sigma \mu e - (Hs) \circ (H^{-1} z)
\]

after eliminating $\Delta s, \Delta z$, reduces to an equation

\[
A^T \nabla^2 \phi(w) A \Delta x = r, \quad w = u^2
\]
Outline

- primal-dual algorithms for linear programming
- symmetric cones
- primal-dual algorithms for conic optimization
- implementation
Software implementations

general-purpose software for nonlinear convex optimization

- several high-quality packages (MOSEK, Sedumi, SDPT3, ...)  
- exploit sparsity to achieve scalability

customized implementations

- can exploit non-sparse types of problem structure  
- often orders of magnitude faster than general-purpose solvers
Example: $\ell_1$-regularized least-squares

\[
\text{minimize} \quad \| Ax - b \|_2^2 + \| x \|_1
\]

$A$ is $m \times n$ (with $m \leq n$) and dense

**quadratic program formulation**

\[
\text{minimize} \quad \| Ax - b \|_2^2 + 1^T u \\
\text{subject to} \quad -u \leq x \leq u
\]

- coefficient of Newton system in interior-point method is

\[
\begin{bmatrix}
A^T A & 0 \\
0 & 0
\end{bmatrix}
+ \begin{bmatrix}
D_1 + D_2 & D_2 - D_1 \\
D_2 - D_1 & D_1 + D_2
\end{bmatrix}
\quad (D_1, D_2 \text{ positive diagonal})
\]

- expensive ($O(n^3)$) for large $n$
customized implementation

• can reduce Newton equation to solution of a system

\[(AD^{-1}A^{T} + I)\Delta u = r\]

• cost per iteration is \(O(m^2n)\)

comparison (seconds on 2.83 Ghz Core 2 Quad machine)

<table>
<thead>
<tr>
<th>(m)</th>
<th>(n)</th>
<th>custom</th>
<th>general-purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>200</td>
<td>0.02</td>
<td>0.32</td>
</tr>
<tr>
<td>50</td>
<td>400</td>
<td>0.03</td>
<td>0.59</td>
</tr>
<tr>
<td>100</td>
<td>1000</td>
<td>0.12</td>
<td>1.69</td>
</tr>
<tr>
<td>100</td>
<td>2000</td>
<td>0.24</td>
<td>3.43</td>
</tr>
<tr>
<td>500</td>
<td>1000</td>
<td>1.19</td>
<td>7.54</td>
</tr>
<tr>
<td>500</td>
<td>2000</td>
<td>2.38</td>
<td>17.6</td>
</tr>
</tbody>
</table>

custom solver is CVXOPT; general-purpose solver is MOSEK
Overview

1. Basic theory and convex modeling
   - convex sets and functions
   - common problem classes and applications

2. Interior-point methods for conic optimization
   - conic optimization
   - barrier methods
   - symmetric primal-dual methods

3. First-order methods
   - gradient algorithms
   - dual techniques
Gradient methods

- gradient and subgradient method
- proximal gradient method
- fast proximal gradient methods
Classical gradient method

to minimize a convex differentiable function $f$: choose $x^{(0)}$ and repeat

$$x^{(k)} = x^{(k-1)} - t_k \nabla f(x^{(k-1)}), \quad k = 1, 2, \ldots$$

step size $t_k$ is constant or from line search

**advantages**

- every iteration is inexpensive
- does not require second derivatives

**disadvantages**

- often very slow; very sensitive to scaling
- does not handle nondifferentiable functions
Quadratic example

\[ f(x) = \frac{1}{2}(x_1^2 + \gamma x_2^2) \quad (\gamma > 1) \]

with exact line search and starting point \( x^{(0)} = (\gamma, 1) \)

\[
\frac{\|x^{(k)} - x^*\|_2}{\|x^{(0)} - x^*\|_2} = \left( \frac{\gamma - 1}{\gamma + 1} \right)^k
\]

Gradient methods
Nondifferentiable example

\[ f(x) = \begin{cases} \sqrt{x_1^2 + \gamma x_2^2} & (|x_2| \leq x_1), \\ \frac{x_1 + \gamma |x_2|}{\sqrt{1 + \gamma}} & (|x_2| > x_1) \end{cases} \]

with exact line search, \( x^{(0)} = (\gamma, 1) \), converges to non-optimal point
First-order methods

address one or both disadvantages of the gradient method

methods for nondifferentiable or constrained problems

• smoothing methods
• subgradient method
• proximal gradient method

methods with improved convergence

• variable metric methods
• conjugate gradient method
• accelerated proximal gradient method

we will discuss subgradient and proximal gradient methods
Subgradient

$g$ is a subgradient of a convex function $f$ at $x$ if

$$f(y) \geq f(x) + g^T(y - x) \quad \forall y \in \text{dom } f$$

This generalizes the basic inequality for convex differentiable $f$

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) \quad \forall y \in \text{dom } f$$
Subdifferential

the set of all subgradients of \( f \) at \( x \) is called the subdifferential \( \partial f(x) \)

**absolute value** \( f(x) = |x| \)

\[
f(x) = |x|
\]

**Euclidean norm** \( f(x) = \|x\|_2 \)

\[
\partial f(x) = \frac{1}{\|x\|_2} x \quad \text{if } x \neq 0, \quad \partial f(x) = \{g \mid \|g\|_2 \leq 1\} \quad \text{if } x = 0
\]
Subgradient calculus

weak calculus

rules for finding one subgradient

- sufficient for most algorithms for nondifferentiable convex optimization
- if one can evaluate $f(x)$, one can usually compute a subgradient
- much easier than finding the entire subdifferential

subdifferentiability

- convex $f$ is subdifferentiable on $\text{dom } f$ except possibly at the boundary
- example of a non-subdifferentiable function: $f(x) = -\sqrt{x}$ at $x = 0$
Examples of calculus rules

**nonnegative combination:** \( f = \alpha_1 f_1 + \alpha_2 f_2 \) with \( \alpha_1, \alpha_2 \geq 0 \)

\[ g = \alpha_1 g_1 + \alpha_2 g_2, \quad g_1 \in \partial f_1(x), \quad g_2 \in \partial f_2(x) \]

**composition with affine transformation:** \( f(x) = h(Ax + b) \)

\[ g = A^T \tilde{g}, \quad \tilde{g} \in \partial h(Ax + b) \]

**pointwise maximum** \( f(x) = \max \{ f_1(x), \ldots, f_m(x) \} \)

\[ g \in \partial f_i(x) \quad \text{where} \quad f_i(x) = \max_k f_k(x) \]

**conjugate** \( f^*(x) = \sup_y (x^T y - f(y)) \); take any maximizing \( y \)
Subgradient method

to minimize a nondifferentiable convex function $f$: choose $x^{(0)}$ and repeat

$$x^{(k)} = x^{(k-1)} - t_k g^{(k-1)}, \quad k = 1, 2, \ldots$$

$g^{(k-1)}$ is any subgradient of $f$ at $x^{(k-1)}$

step size rules

- fixed step size: $t_k$ constant
- fixed step length: $t_k \|g^{(k-1)}\|_2$ constant (i.e., $\|x^{(k)} - x^{(k-1)}\|_2$ constant)
- diminishing: $t_k \to 0, \sum_{k=1}^{\infty} t_k = \infty$
Some convergence results

**assumption:** $f$ is convex and Lipschitz continuous with constant $G > 0$:

$$|f(x) - f(y)| \leq G\|x - y\|_2 \quad \forall x, y$$

**results**

- fixed step size $t_k = t$
  
  converges to approximately $G^2 t/2$-suboptimal

- fixed length $t_k\|g^{(k-1)}\|_2 = s$
  
  converges to approximately $Gs/2$-suboptimal

- decreasing $\sum_k t_k \to \infty$, $t_k \to 0$: convergence
  
  rate of convergence is $1/\sqrt{k}$ with proper choice of step size sequence
Example: 1-norm minimization

\[
\text{minimize } \|Ax - b\|_1 \quad (A \in \mathbb{R}^{500 \times 100}, b \in \mathbb{R}^{500})
\]

subgradient is given by \(A^T \text{sign}(Ax - b)\)

\begin{align*}
\text{fixed steplength} & \quad s = 0.1, 0.01, 0.001 \\
\text{diminishing step size} & \quad t_k = 0.01/\sqrt{k}, t_k = 0.01/k
\end{align*}
Outline

• gradient and subgradient method

• proximal gradient method

• fast proximal gradient methods
**Proximal mapping**

the proximal mapping (prox-operator) of a convex function $h$ is

$$\text{prox}_h(x) = \arg\min_u \left( h(u) + \frac{1}{2} \| u - x \|^2 \right)$$

- $h(x) = 0$: $\text{prox}_h(x) = x$
- $h(x) = I_C(x)$ (indicator function of $C$): $\text{prox}_h$ is projection on $C$
  $$\text{prox}_h(x) = \arg\min_{u \in C} \| u - x \|^2 = P_C(x)$$
- $h(x) = \|x\|_1$: $\text{prox}_h$ is the ‘soft-threshold’ (shrinkage) operation
  $$\text{prox}_h(x)_i = \begin{cases} 
  x_i - 1 & x_i \geq 1 \\
  0 & |x_i| \leq 1 \\
  x_i + 1 & x_i \leq -1 
\end{cases}$$

---

Gradient methods 118
**Proximal gradient method**

**unconstrained problem** with cost function split in two components

\[ \text{minimize} \quad f(x) = g(x) + h(x) \]

- \( g \) convex, differentiable, with \( \text{dom} \ g = \mathbb{R}^n \)
- \( h \) convex, possibly nondifferentiable, with inexpensive prox-operator

**proximal gradient algorithm**

\[ x^{(k)} = \text{prox}_{t_k h} \left( x^{(k-1)} - t_k \nabla g(x^{(k-1)}) \right) \]

\( t_k > 0 \) is step size, constant or determined by line search
Interpretation

\[ x^+ = \text{prox}_{th}(x - t\nabla g(x)) \]

from definition of proximal operator:

\[
x^+ = \underset{u}{\text{argmin}} \left( h(u) + \frac{1}{2t} \|u - x + t\nabla g(x)\|^2_2 \right)
\]

\[
= \underset{u}{\text{argmin}} \left( h(u) + g(x) + \nabla g(x)^T (u - x) + \frac{1}{2t} \|u - x\|^2_2 \right)
\]

\[ x^+ \text{ minimizes } h(u) \text{ plus a simple quadratic local model of } g(u) \text{ around } x \]
Examples

\[ \text{minimize } g(x) + h(x) \]

**gradient method:** \( h(x) = 0, \text{ i.e., minimize } g(x) \)

\[ x^+ = x - t\nabla g(x) \]

**gradient projection method:** \( h(x) = I_C(x), \text{ i.e., minimize } g(x) \text{ over } C \)

\[ x^+ = P_C(x - t\nabla g(x)) \]
iterative soft-thresholding: \( h(x) = \|x\|_1 \), i.e., minimize \( g(x) + \|x\|_1 \)

\[
x^+ = \text{prox}_{th}(x - t \nabla g(x))
\]

and

\[
\text{prox}_{th}(u)_i = \begin{cases} 
  u_i - t & u_i \geq t \\
  0 & -t \leq u_i \leq t \\
  u_i + t & u_i \geq t 
\end{cases}
\]
**Some properties of proximal mappings**

\[
\text{prox}_h(x) = \arg\min_u \left( h(u) + \frac{1}{2}\|u - x\|^2 \right)
\]

assume \( h \) is closed and convex (i.e., convex with closed epigraph)

- \( \text{prox}_h(x) \) is uniquely defined for all \( x \)
- \( \text{prox}_h \) is nonexpansive

\[
\|\text{prox}_h(x) - \text{prox}_h(y)\|_2 \leq \|x - y\|_2
\]

- Moreau decomposition

\[
x = \text{prox}_h(x) + \text{prox}_{h^*}(x)
\]

cf., properties of Euclidean projection on convex sets
**example:** $h$ is indicator function of subspace $L$

$$h(u) = I_L(u) = \begin{cases} 
0 & u \in L \\
+\infty & \text{otherwise}
\end{cases}$$

- conjugate $h^*$ is indicator function of the orthogonal complement $L^\perp$

$$h^*(v) = \sup_{u \in L} v^T u = \begin{cases} 
0 & v \in L^\perp \\
+\infty & \text{otherwise}
\end{cases} = I_{L^\perp}(v)$$

- Moreau decomposition is orthogonal decomposition

$$x = P_L(x) + P_{L^\perp}(x)$$
Examples of inexpensive prox-operators

projection on simple sets

- hyperplanes and halfspaces
- rectangles \( \{ x \mid l \leq x \leq u \} \)
- probability simplex \( \{ x \mid 1^T x = 1, x \geq 0 \} \)
- norm ball for many norms (Euclidean, 1-norm, \ldots)
- nonnegative orthant, second-order cone, positive semidefinite cone

Euclidean norm: \( h(x) = \|x\|_2 \)

\[
\text{prox}_{th}(x) = \left( 1 - \frac{t}{\|x\|_2} \right) x \quad \text{if} \quad \|x\|_2 \geq t, \quad \text{prox}_{th}(x) = 0 \quad \text{otherwise}
\]
logarithmic barrier

\[
h(x) = - \sum_{i=1}^{n} \log x_i, \quad \text{prox}_{th}(x)_i = \frac{x_i + \sqrt{x_i^2 + 4t}}{2}, \quad i = 1, \ldots, n
\]

Euclidean distance: \(d(x) = \inf_{y \in C} \|x - y\|_2\) (\(C\) closed convex)

\[
\text{prox}_{td}(x) = \theta P_C(x) + (1 - \theta)x, \quad \theta = \frac{t}{\max\{d(x), t\}}
\]

squared Euclidean distance: \(h(x) = d(x)^2/2\)

\[
\text{prox}_{th}(x) = \frac{1}{1 + t} x + \frac{t}{1 + t} P_C(x)
\]
**Prox-operator of conjugate**

\[
\text{prox}_{th^*}(x) = x - t \, \text{prox}_{h/t}(x/t)
\]

- follows from Moreau decomposition
- of interest when prox-operator of \( h \) is inexpensive

**example: norms**

\[
h(x) = I_C(x), \quad h^*(y) = \|y\|^*
\]

where \( C \) is unit norm ball for \( \| \cdot \| \) and \( \| \cdot \|^* \) is dual norm of \( \| \cdot \| \)

- \( \text{prox}_h \) is projection on \( C \)
- formula useful for prox-operator of \( \| \cdot \|^* \) if projection on \( C \) is inexpensive
many convex functions can be expressed as \textbf{support functions}

\[
h(x) = S_C(x) = \sup_{y \in C} x^T y
\]

with \( C \) closed, convex

- conjugate is indicator function of \( C \): \( h^*(y) = I_C(y) \)
- hence, can compute \( \text{prox}_{th} \) via projection on \( C \)

**example:** \( h(x) \) is sum of largest \( r \) components of \( x \)

\[
h(x) = x_1 + \cdots + x_r = S_C(x), \quad C = \{ y \mid 0 \leq y \leq 1, 1^T y = r \}
\]
Convergence of proximal gradient method

\begin{align*}
\text{minimize} \quad f(x) &= g(x) + h(x) \\
\text{assumptions} \\
&\bullet \nabla g \text{ is Lipschitz continuous with constant } L > 0 \\
&\quad \|\nabla g(x) - \nabla g(y)\|_2 \leq L \|x - y\|_2 \quad \forall x, y \\
&\bullet \text{optimal value } f^* \text{ is finite and attained at } x^* \text{ (not necessarily unique)} \\
\text{result:} \quad \text{with fixed step size } t_k = 1/L \\
&\quad f(x^{(k)}) - f^* \leq \frac{L}{2k} \|x^{(0)} - x^*\|_2^2 \\
&\bullet \text{compare with } 1/\sqrt{k}: \text{rate of subgradient method} \\
&\bullet \text{can be extended to include line searches}
\end{align*}
Outline

- gradient and subgradient method
- proximal gradient method
- fast proximal gradient methods
Fast (proximal) gradient methods

- Beck & Teboulle (2008): FISTA, a proximal gradient version of Nesterov’s 1983 method
- several recent variations and extensions

**this lecture**: FISTA (Fast Iterative Shrinkage-Thresholding Algorithm)
FISTA

unconstrained problem with composite objective

\[
\text{minimize} \quad f(x) = g(x) + h(x)
\]

- \( g \) convex differentiable with \( \text{dom} \ g = \mathbb{R}^n \)
- \( h \) convex with inexpensive prox-operator

algorithm: choose \( x^{(0)} = y^{(0)} \in \text{dom} \ h \); for \( k \geq 1 \)

\[
x^{(k)} = \text{prox}_{t_k h} \left( y^{(k-1)} - t_k \nabla g(y^{(k-1)}) \right)
\]

\[
y^{(k)} = x^{(k)} + \frac{k - 1}{k + 2} (x^{(k)} - x^{(k-1)})
\]
Interpretation

- first iteration \( (k = 1) \) is a proximal gradient step at \( x^{(0)} \)
- next iterations are proximal gradient steps at extrapolated points \( y^{(k-1)} \)

\[
x^{(k)} = \text{prox}_{t_k h}(y^{(k-1)} - t_k \nabla g(y^{(k-1)}))
\]

sequence \( x^{(k)} \) remains feasible (in \( \text{dom } h \)); sequence \( y^{(k)} \) not necessarily
Convergence of FISTA

minimize $f(x) = g(x) + h(x)$

assumptions

• optimal value $f^*$ is finite and attained at $x^*$ (not necessarily unique)
• $\text{dom } g = \mathbb{R}^n$ and $\nabla g$ is Lipschitz continuous with constant $L > 0$
• $h$ is closed (implies $\text{prox}_{th}(u)$ exists and is unique for all $u$)

result: with fixed step size $t_k = 1/L$

$$f(x^{(k)}) - f^* \leq \frac{2L}{(k + 1)^2} \|x^{(0)} - f^*\|^2_2$$

• compare with $1/k$ convergence rate for gradient method
• can be extended to include line searches
Example

minimize \[ \log \sum_{i=1}^{m} \exp(a_i^T x + b_i) \]

randomly generated data with \( m = 2000, n = 1000 \), same fixed step size

FISTA is not a descent method
Dual methods

• Lagrange duality
• dual decomposition
• dual proximal gradient method
• multiplier methods
**Dual function**

**convex problem** (with linear constraints for simplicity)

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad Gx \leq h \\
& \quad Ax = b
\end{align*}
\]

optimal value \( p^* \)

**Lagrangian**

\[
L(x, \lambda, \nu) = f(x) + \lambda^T(Gx - h) + \nu^T(Ax - b)
\]

\[
= f(x) + (G^T \lambda + A^T \nu)^T x - h^T \lambda - b^T \nu
\]

**dual function**

\[
g(\lambda, \nu) = \inf_x L(x, \lambda, \nu) = -f^*(-G^T \lambda - A^T \nu) - h^T \lambda - b^T \nu
\]

(with \( f^*(y) = \sup_x (y^T x - f(x)) \) the conjugate of \( f \))
Dual problem

maximize \( g(\lambda, \nu) \)
subject to \( \lambda \geq 0 \)

optimal value \( d^* \)
a convex optimization problem in \( \lambda, \nu \)

**weak duality:** \( p^* \geq d^* \), without exception

**strong duality:** \( p^* = d^* \) if a constraint qualification holds
(for example, primal problem is feasible and \( \text{dom} f \) open)
Least-norm solution of linear equations

\[
\begin{align*}
\text{minimize} & \quad f(x) = \|x\| \\
\text{subject to} & \quad Ax = b
\end{align*}
\]

recall that \(f^*\) is indicator function of unit dual norm ball

dual problem

\[
\begin{align*}
\text{maximize} & \quad -b^T \nu - f^*(-A^T \nu) = \begin{cases} 
-b^T \nu & \|A^T \nu\|_* \leq 1 \\
-\infty & \text{otherwise}
\end{cases}
\end{align*}
\]

reformulated dual problem

\[
\begin{align*}
\text{maximize} & \quad b^T z \\
\text{subject to} & \quad \|A^T z\|_* \leq 1
\end{align*}
\]
Norm approximation

minimize \[ \|Ax - b\| \]

reformulated problem

minimize \[ \|y\| \]
subject to \[ y = Ax - b \]

dual function

\[ g(\nu) = \inf_{x,y} \left( \|y\| + \nu^T y - \nu^T Ax + b^T \nu \right) \]
\[ = \begin{cases} b^T \nu & A^T \nu = 0, \|\nu\|_* \leq 1 \\ -\infty & \text{otherwise} \end{cases} \]

dual problem

maximize \[ b^T z \]
subject to \[ A^T z = 0, \|z\|_* \leq 1 \]
Karush-Kuhn-Tucker optimality conditions

if strong duality holds, then \( x, \lambda, \nu \) are optimal if and only if

1. **primal feasibility:**

\[
x \in \text{dom} \ f, \quad Gx \leq h, \quad Ax = b
\]

2. \( \lambda \geq 0 \)

3. **complementary slackness:**

\[
\lambda^T(h - Gx) = 0
\]

4. \( x \) minimizes

\[
L(x, \lambda, \nu) = f(x) + \lambda^T(Gx - h) + \nu^T(Ax - b)
\]

for differentiable \( f \), condition 4 can be expressed as

\[
\nabla f(x) + G^T \lambda + A^T \nu = 0
\]
Outline

- Lagrange dual
- dual decomposition
- dual proximal gradient method
- multiplier methods
Dual methods

**primal problem**

minimize $f(x)$

subject to $Gx \leq h$

$Ax = b$

**dual problem**

maximize $-h^T\lambda - b^T\nu - f^*(-G^T\lambda - A^T\nu)$

subject to $\lambda \geq 0$

possible advantages of solving the dual when using first-order methods

- dual problem is unconstrained or has simple constraints
- dual problem can be decomposed into smaller problems
(Sub-)gradients of conjugate function

\[ f^*(y) = \sup_x \left( y^T x - f(x) \right) \]

- subgradient: \( x \) is a subgradient at \( y \) if it maximizes \( y^T x - f(x) \)
- if maximizing \( x \) is unique, then \( f^* \) is differentiable
  this is the case, for example, if \( f \) is strictly convex

**strongly convex function:** \( f \) is strongly convex with parameter \( \mu > 0 \) if

\[ f(x) - \frac{\mu}{2} x^T x \quad \text{is convex} \]

implies that \( \nabla f^*(x) \) is Lipschitz continuous with parameter \( 1/\mu \)
Dual gradient method

primal problem with equality constraints and dual

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad Ax = b
\end{align*}
\]

dual ascent: use (sub-)gradient method to minimize

\[
-g(\nu) = b^T \nu + f^*(-A^T \nu) = \sup_x \left( (b - Ax)^T \nu - f(x) \right)
\]

algorithm

\[
\begin{align*}
x^+ &= \arg\min_x \left( f(x) + \nu^T Ax \right) \\
\nu^+ &= \nu + t(Ax^+ - b)
\end{align*}
\]

of interest if calculation of \(x^+\) is inexpensive (for example, separable)
Dual decomposition

\begin{align*}
\text{minimize} & \quad f_1(x_1) + f_2(x_2) \\
\text{subject to} & \quad G_1 x_1 + G_2 x_2 \leq h
\end{align*}

objective is separable; constraint is \textit{complicating} (or \textit{coupling}) constraint

dual problem \textit{('master' problem)}

\begin{align*}
\text{maximize} & \quad -h^T \lambda - f_1^*(-G_1^T \lambda) - f_2^*(-G_2^T \lambda) \\
\text{subject to} & \quad \lambda \geq 0
\end{align*}

can be solved by (sub-)gradient projection if \( \lambda \geq 0 \) is the only constraint

\textbf{subproblems:} for \( j = 1, 2 \), evaluate

\[ f_j^*(-G_j^T \lambda) = - \inf_{x_j} \left( f_j(x_j) + \lambda^T G_j x_j \right) \]

maximizer \( x_j \) gives subgradient \(-G_j x_j\) of \( f_j^*(-G_j^T \lambda)\) w.r.t. \( \lambda \)
dual subgradient projection method

• solve two unconstrained (and independent) subproblems

\[ x_j^+ = \arg\min_{x_j} (f_j(x_j) + \lambda^T G_j x_j), \quad j = 1, 2 \]

• make projected subgradient update of \( \lambda \)

\[ \lambda^+ = (\lambda + t(G_1 x_1^+ + G_2 x_2^+ - h))_+ \]

interpretation: price coordination between two units in a system

• constraints are limits on shared resources; \( \lambda_i \) is price of resource \( i \)

• dual update \( \lambda_i^+ = (\lambda_i - ts_i)_+ \) depends on slacks \( s = h - G_1 x_1 - G_2 x_2 \)
  
  - increases price \( \lambda_i \) if resource is over-utilized \( (s_i < 0) \)
  
  - decreases price \( \lambda_i \) if resource is under-utilized \( (s_i > 0) \)
  
  - never lets prices get negative

Dual methods
Outline

- Lagrange dual
- dual decomposition
- dual proximal gradient method
- multiplier methods
First-order dual methods

\[
\begin{align*}
\text{minimize} & \quad f(x) & \quad \text{maximize} & \quad -f^*(-G^T\lambda - A^T\nu) \\
\text{subject to} & \quad Gx \geq h & \quad \text{subject to} & \quad \lambda \geq 0 \\
 & \quad Ax = b
\end{align*}
\]

**subgradient method:** slow, step size selection difficult

**gradient method:** faster, requires differentiable \( f^* \)

- in many applications \( f^* \) is not differentiable, has a nontrivial domain
- \( f^* \) can be smoothed by adding a small strongly convex term to \( f \)

**proximal gradient method (this section):** dual costs split in two terms

- first term is differentiable
- second term has an inexpensive prox-operator
Composite structure in the dual

primal problem with separable objective

\[
\begin{align*}
\text{minimize} & \quad f(x) + h(y) \\
\text{subject to} & \quad Ax + By = b
\end{align*}
\]

dual problem

\[
\begin{align*}
\text{maximize} & \quad -f^*(A^T z) - h^*(B^T z) + b^T z
\end{align*}
\]

has the composite structure required for the proximal gradient method if

- \( f \) is strongly convex; hence \( \nabla f^* \) is Lipschitz continuous
- prox-operator of \( h^*(B^T z) \) is cheap (closed form or efficient algorithm)
Regularized norm approximation

\[ \text{minimize } f(x) + \|Ax - b\| \]

\(f\) strongly convex with modulus \(\mu\); \(\| \cdot \|\) is any norm

reformulated problem and dual

\[ \begin{align*}
\text{minimize } & \quad f(x) + \|y\| \\
\text{subject to } & \quad y = Ax - b \\
\text{maximize } & \quad b^T z - \text{f}^*(A^T z) \\
\text{subject to } & \quad \|z\|_* \leq 1
\end{align*} \]

- gradient of dual cost is Lipschitz continuous with parameter \(\|A\|_2^2/\mu\)

\[ \nabla \text{f}^*(A^T z) = \arg\min_x \left( f(x) - z^T Ax \right) \]

- for most norms, projection on dual norm ball is inexpensive
**problem:** minimize $f(x) + \|Ax - b\|$

**dual gradient projection algorithm:** choose initial $z$ and repeat

$$
\hat{x} := \underset{x}{\text{argmin}} \left( f(x) - z^T A x \right) \\
z := P_C \left( z + t \left( b - A \hat{x} \right) \right)
$$

- $P_C$ is projection on $C = \{ y \mid \|y\|_* \leq 1 \}$
- step size $t$ is constant or from backtracking line search
- can use accelerated gradient projection algorithm (FISTA) for $z$-update
- first step decouples if $f$ is separable
Outline

- Lagrange dual
- dual decomposition
- dual proximal gradient method
- multiplier methods
Moreau-Yosida regularization of the dual

a general technique for smoothing the dual of

\[
\begin{align*}
\text{minimize} & \quad f(x) \\
\text{subject to} & \quad Ax = b
\end{align*}
\]

- maximizing \( g(\nu) = \inf_x (f(x) + \nu^T(Ax - b)) \) is equivalent to maximizing

\[
g_t(\nu) = \sup_z \left( g(z) - \frac{1}{2t} \|\nu - z\|^2 \right)
\]

- from duality, \( g_t(\nu) = \inf_x L_t(x, \nu) \) where

\[
L_t(x, \nu) = f(x) + \nu^T(Ax - b) + (t/2)\|Ax - b\|^2
\]

- \( g_t \) is concave, differentiable with Lipschitz cont. gradient (constant \( 1/t \))

\[
\nabla g_t(\nu) = A\hat{x} - b, \quad \hat{x} = \arg\min_x L_t(x, \nu)
\]
Augmented Lagrangian method

**algorithm:** choose initial $\nu$ and repeat

$$x^+ = \text{argmin } L_t(x, \nu)$$

$$\nu^+ = \nu + t(Ax^+ - b)$$

- maximizes Moreau-Yosida regularization $g_t$ via gradient method
- $L_t$ is the augmented Lagrangian (Lagrangian plus quadratic penalty)

$$L_t(x, \nu) = f(x) + \nu^T(Ax - b) + \frac{t}{2}\|Ax - b\|_2^2$$

- method can be extended to problems with inequality constraints
Dual decomposition

convex problem with separable objective

\[
\begin{align*}
\text{minimize} & \quad f(x) + h(y) \\
\text{subject to} & \quad Ax + By = b
\end{align*}
\]

augmented Lagrangian

\[
L_t(x, y, \nu) = f(x) + h(y) + \nu^T(Ax + By - b) + \frac{t}{2} \|Ax + By - b\|_2^2
\]

- difficulty: quadratic penalty destroys separability of Lagrangian
- solution: replace minimization over \((x, y)\) by alternating minimization
Alternating direction method of multipliers

apply one cycle of alternating minimization steps to augmented Lagrangian

1. minimize augmented Lagrangian over $x$:

$$x^{(k)} = \arg\min_x L_t(x, y^{(k-1)}, \nu^{(k-1)})$$

2. minimize augmented Lagrangian over $y$:

$$y^{(k)} = \arg\min_y L_t(x^{(k)}, y, \nu^{(k-1)})$$

3. dual update:

$$\nu^{(k)} := \nu^{(k-1)} + t \left( Ax^{(k)} + By^{(k)} - b \right)$$

can be shown to converge under weak assumptions
Example: sparse covariance selection

minimize \( \text{tr}(CX) - \log \det X + \|X\|_1 \)

variable \( X \in S^n \); \( \|X\|_1 \) is sum of absolute values of \( X \)

reformulation

minimize \( \text{tr}(CX) - \log \det X + \|Y\|_1 \)
subject to \( X - Y = 0 \)

augmented Lagrangian

\[
L_t(X, Y, Z) = \text{tr}(CX) - \log \det X + \|Y\|_1 + \text{tr}(Z(X - Y)) + \frac{t}{2} \|X - Y\|_F^2
\]
**ADMM steps:** alternating minimization of augmented Lagrangian

\[
\text{tr}(CX) - \log \det X + \|Y\|_1 + \text{tr}(Z(X - Y)) + \frac{t}{2} \|X - Y\|_F^2
\]

- minimization over \( X \):

\[
\hat{X} = \arg\min_X \left( -\log \det X + \frac{t}{2} \|X - Y + \frac{1}{t}(C + Z)\|_F^2 \right)
\]

follows easily from eigenvalue decomposition of \( Y - (1/t)(C + Z) \)

- minimization over \( Y \):

\[
\hat{Y} = \arg\min_Y \left( \|Y\|_1 + \frac{t}{2} \|Y - \hat{X} - \frac{1}{t}Z\|_F^2 \right)
\]

apply element-wise soft-thresholding to \( \hat{X} - (1/t)Z \)

- dual update \( Z := Z + t(\hat{X} - \hat{Y}) \)

cost per iteration dominated by cost of eigenvalue decomposition
Sources and references

these lectures are based on the courses

- EE364A (S. Boyd, Stanford), EE236B (UCLA), Convex Optimization
  www.stanford.edu/class/ee364a
  www.ee.ucla.edu/ee236b/

- EE236C (UCLA) Optimization Methods for Large-Scale Systems
  www.ee.ucla.edu/~vandenbe/ee236c

- EE364B (S. Boyd, Stanford University) Convex Optimization II
  www.stanford.edu/class/ee364b

see the websites for expanded notes, references to literature and software