# Convex Optimization

**Lieven Vandenberghe**

University of California, Los Angeles

Tutorial lectures, Machine Learning Summer School

University of Cambridge, September 3-4, 2009

Sources:

- Boyd & Vandenberghe, *Convex Optimization*, 2004
- Courses EE236B, EE236C (UCLA), EE364A, EE364B (Stephen Boyd, Stanford Univ.)

# Introduction

- mathematical optimization, modeling, complexity
- convex optimization
- recent history

# Mathematical optimization

$$\begin{array}{ll}\text{minimize} & f_0(x_1, \ldots, x_n) \\ \text{subject to} & f_1(x_1, \ldots, x_n) \leq 0 \\ & \ldots \\ & f_m(x_1, \ldots, x_n) \leq 0\end{array}$$

- $x = (x_1, x_2, \ldots, x_n)$ are decision variables
- $f_0(x_1, x_2, \ldots, x_n)$ gives the cost of choosing $x$
- inequalities give constraints that $x$ must satisfy

a mathematical model of a decision, design, or estimation problem

# Limits of mathematical optimization

- how realistic is the model, and how certain are we about it?
- is the optimization problem tractable by existing numerical algorithms?

## Optimization research

- **modeling**

  generic techniques for formulating tractable optimization problems

- **algorithms**

  expand class of problems that can be efficiently solved

# Complexity of nonlinear optimization

- the general optimization problem is intractable
- even simple looking optimization problems can be very hard

**Examples**

- quadratic optimization problem with many constraints
- minimizing a multivariate polynomial

# The famous exception: Linear programming

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

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

# Solving linear programs

- no closed-form expression for solution
- widely available and reliable software
- for some algorithms, can prove polynomial time
- problems with over $10^5$ variables or constraints solved routinely

# Convex optimization problem

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_1(x) \leq 0 \\ & \cdots \\ & f_m(x) \leq 0 \end{array}$$

- objective and constraint functions are convex: for $0 \leq \theta \leq 1$

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

- includes least-squares problems and linear programs as special cases
- can be solved exactly, with similar complexity as LPs
- surprisingly many problems can be solved via convex optimization

# History

- 1940s: linear programming

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

- 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
- circuit design via geometric programming
- support vector machine learning via quadratic programming
- semidefinite programming relaxations in combinatorial optimization
- $\ell_1$-norm optimization for sparse signal reconstruction
- applications in structural optimization, statistics, signal processing, communications, image processing, computer vision, quantum information theory, finance, . . .

# Algorithms

**Interior-point methods**

- 1984 (Karmarkar): first practical polynomial-time algorithm
- 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

**First-order algorithms**

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

# Outline

- basic theory
  - convex sets and functions
  - convex optimization problems
  - linear, quadratic, and geometric programming
- cone linear programming and applications
  - second-order cone programming
  - semidefinite programming
- some recent developments in algorithms (since 1990)
  - interior-point methods
  - fast gradient methods

# Convex sets and functions

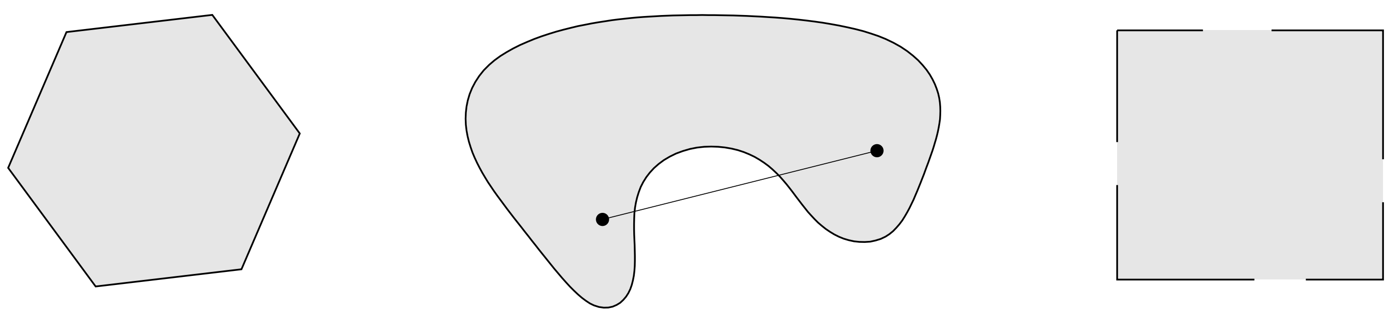
- definition
- basic examples and properties
- operations that preserve convexity

# Convex set

contains line segment between any two points in the set

$$x_1, x_2 \in C, \quad 0 \leq \theta \leq 1 \quad \Longrightarrow \quad \theta x_1 + (1-\theta) x_2 \in C$$

**Examples**: one convex, two nonconvex sets



# Examples and properties

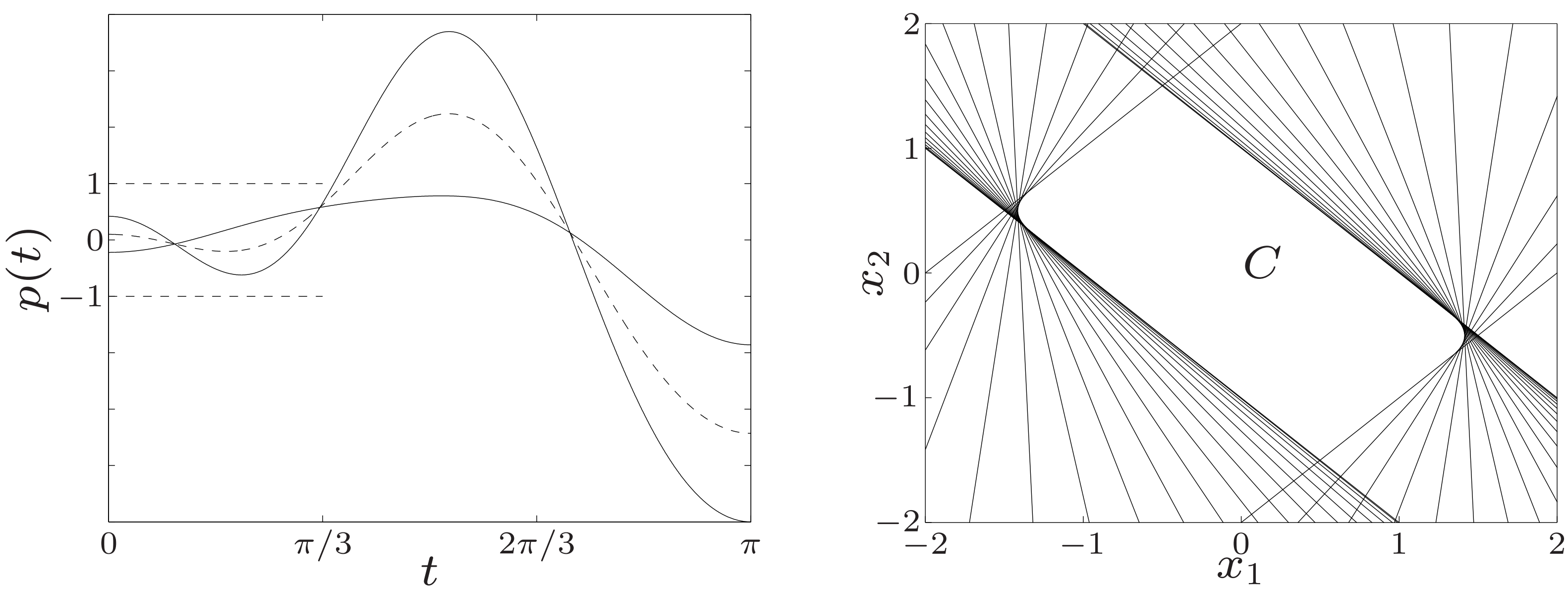
- solution set of linear equations $Ax = b$ (affine set)
- solution set of linear inequalities $Ax \preceq b$ (polyhedron)
- norm balls $\{x \mid \|x\| \leq R\}$ and norm cones $\{(x,t) \mid \|x\| \leq t\}$
- set of positive semidefinite matrices $\mathbf{S}^n_+ = \{X \in \mathbf{S}^n \mid X \succeq 0\}$

- image of a convex set under a linear transformation is convex
- inverse image of a convex set under a linear transformation is convex
- intersection of convex sets is convex

# Example of intersection property

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

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



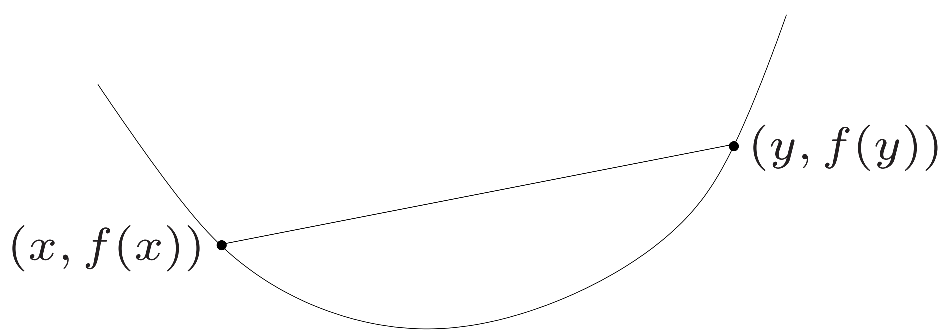

$C$ is intersection of infinitely many halfspaces, hence convex

# Convex function

domain $\mathbf{dom}\, f$ is a convex set and

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

for all $x, y \in \mathbf{dom}\, f$, $0 \leq \theta \leq 1$

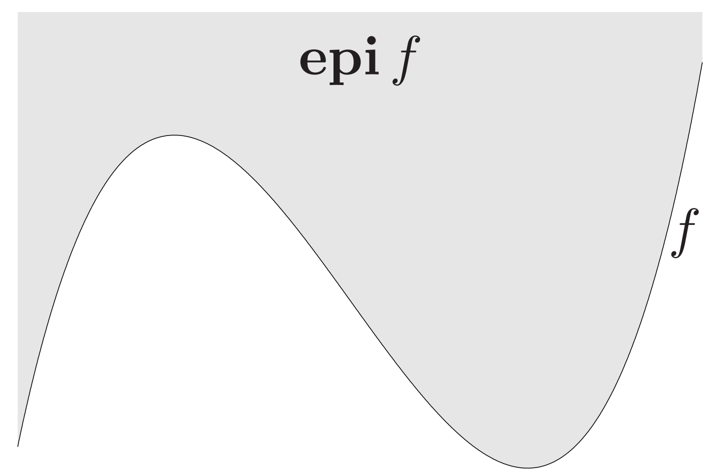



$f$ is concave if $-f$ is convex

# Epigraph and sublevel set

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

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




**Sublevel sets**: $C_\alpha = \{x \in \mathbf{dom}\, f \mid f(x) \leq \alpha\}$

the sublevel sets of a convex function are convex (converse is false)

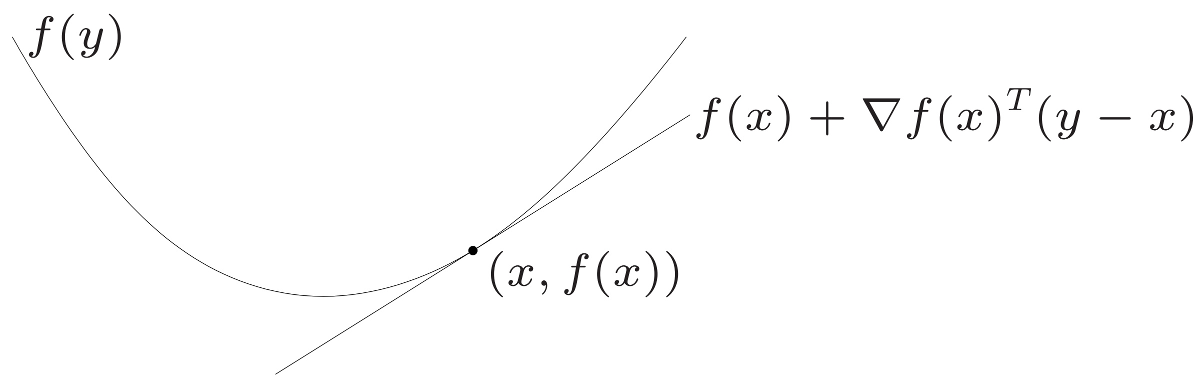
# Examples

- $\exp x$, $-\log x$, $x \log x$ are convex
- $x^\alpha$ is convex for $x > 0$ and $\alpha \geq 1$ or $\alpha \leq 0$; $|x|^\alpha$ is convex for $\alpha \geq 1$
- 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 $x \succeq 0$
- $\log \det X$ is concave on set of positive definite matrices
- $\log(e^{x_1} + \cdots e^{x_n})$ is convex
- linear and affine functions are convex and concave
- norms are convex

# Differentiable convex functions

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

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




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

$$\nabla^2 f(x) \succeq 0 \quad \text{for all } x \in \mathbf{dom}\, f$$

# Operations that preserve convexity

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
   - composition
   - minimization
   - 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**

- log 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 $f_1$, . . . , $f_m$ are convex

**Example:** sum of $r$ largest components of $x \in \mathbf{R}^n$

$$f(x) = x_{[1]} + x_{[2]} + \cdots + x_{[r]}$$

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

$$g(x) = \sup_{y \in \mathcal{A}} f(x, y)$$

is convex if $f(x, y)$ is convex in $x$ for each $y \in \mathcal{A}$

**Example:** maximum eigenvalue of symmetric matrix

$$\lambda_{\max}(X) = \sup_{\|y\|_2=1} y^T X y$$

# Composition with scalar functions

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

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

$f$ is convex if

$$\begin{array}{l} g \text{ convex, } h \text{ convex and nondecreasing} \\ g \text{ concave, } h \text{ convex and nonincreasing} \end{array}$$

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

**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 $g : \mathbf{R}^n \to \mathbf{R}^k$ and $h : \mathbf{R}^k \to \mathbf{R}$:

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

$f$ 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 $x \in \mathbf{dom}\, h$)

**Examples**

- $\sum_{i=1}^{m} \log g_i(x)$ is concave if $g_i$ are concave and positive
- $\log \sum_{i=1}^{m} \exp g_i(x)$ is convex if $g_i$ are convex

# Minimization

$$g(x) = \inf_{y \in C} f(x, y)$$

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

**Examples**

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

  $$g(x) = \inf_{y: Ay \preceq x} c^T y$$

  follows by taking

  $$f(x, y) = c^T y, \qquad \mathbf{dom}\, f = \{(x, y) \mid Ay \preceq x\}$$

# Perspective

the **perspective** of a function $f : \mathbf{R}^n \to \mathbf{R}$ is the function $g : \mathbf{R}^n \times \mathbf{R} \to \mathbf{R}$,

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

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

**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

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

# Convex optimization problems

- standard form
- linear, quadratic, geometric programming
- modeling languages

# Convex optimization problem

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

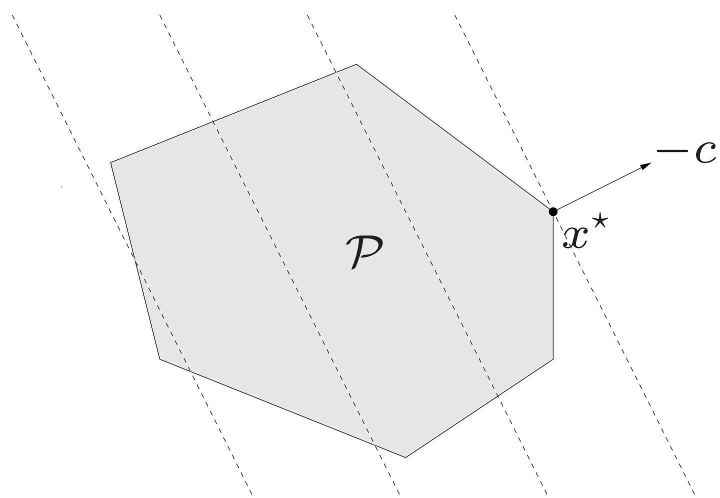
$f_0, f_1, \ldots, f_m$ are convex functions

- feasible set is convex
- locally optimal points are globally optimal
- tractable, both in theory and practice

# Linear program (LP)

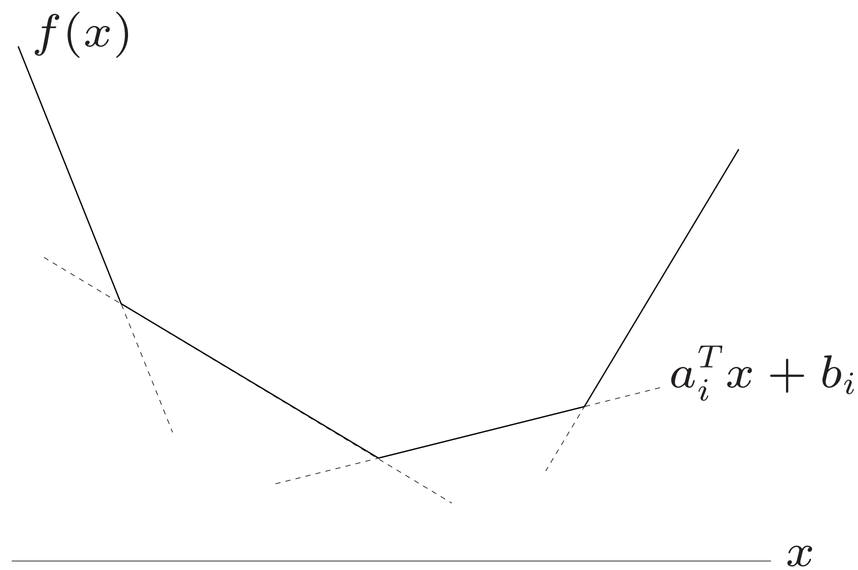
$$\begin{array}{ll} \text{minimize} & c^T x + d \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

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



# Piecewise-linear minimization

$$\text{minimize} \quad f(x) = \max_{i=1,\ldots,m} (a_i^T x + b_i)$$




## Equivalent linear program

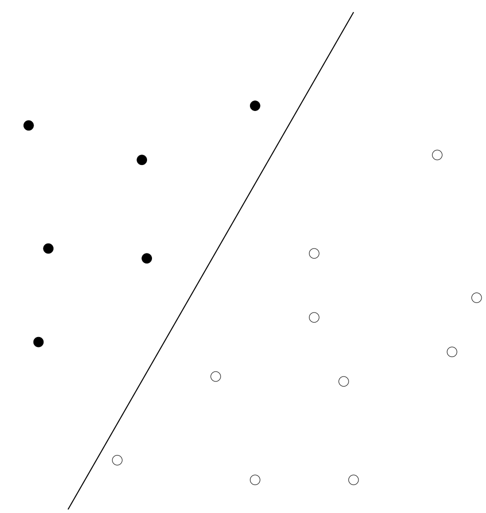
$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & a_i^T x + b_i \le t, \quad i = 1, \ldots, m \end{array}$$

an LP with variables $x$, $t \in \mathbf{R}$

# Linear discrimination

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

$$a^T x_i + b > 0, \quad i = 1, \ldots, N$$
$$a^T y_i + b < 0, \quad i = 1, \ldots, M$$

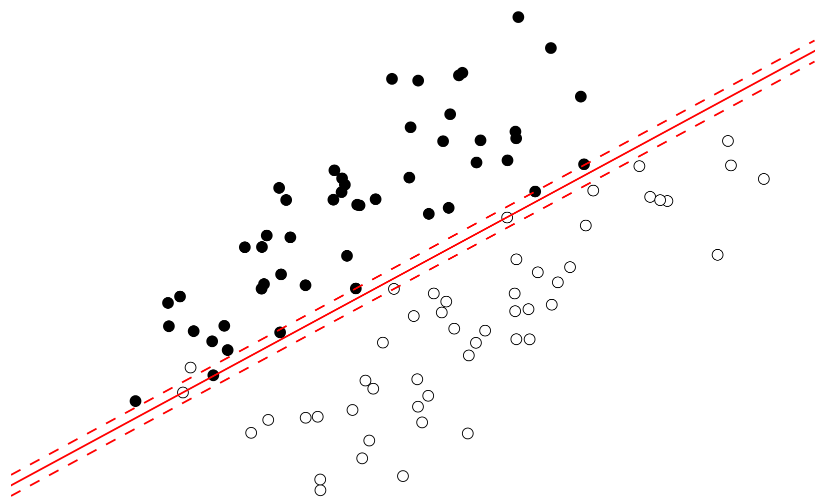


homogeneous in $a$, $b$, hence equivalent to the linear inequalities (in $a$, $b$)

$$a^T x_i + b \geq 1, \quad i = 1, \ldots, N, \qquad a^T y_i + b \leq -1, \quad i = 1, \ldots, M$$

# Approximate linear separation of non-separable sets

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



- a piecewise-linear minimization problem in $a$, $b$; equivalent to an LP
- can be interpreted as a heuristic for minimizing #misclassified points

# $\ell_1$-Norm and $\ell_\infty$-norm minimization

**$\ell_1$-Norm approximation and equivalent LP** ($\|y\|_1 = \sum_k |y_k|$)

$$\text{minimize} \quad \|Ax - b\|_1 \qquad\qquad \begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^n y_i \\ \text{subject to} & -y \preceq Ax - b \preceq y \end{array}$$

**$\ell_\infty$-Norm approximation** ($\|y\|_\infty = \max_k |y_k|$)

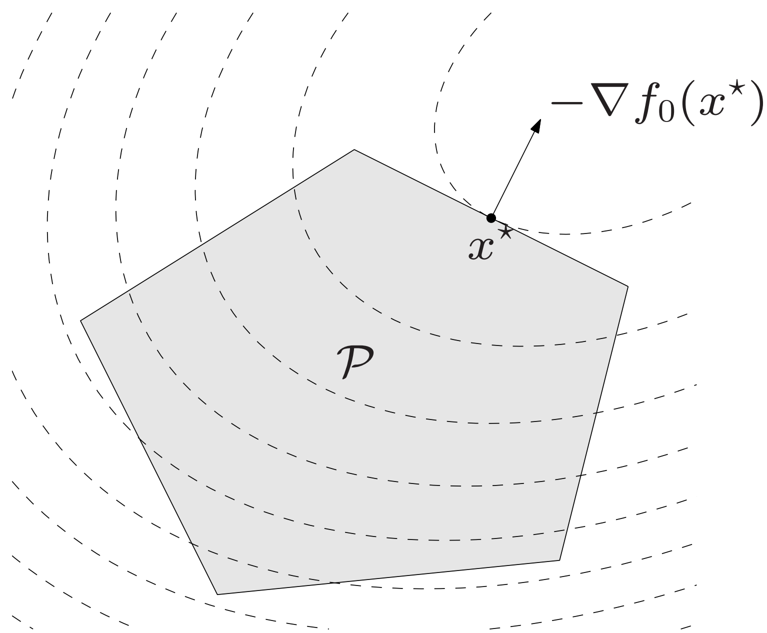
$$\text{minimize} \quad \|Ax - b\|_\infty \qquad\qquad \begin{array}{ll} \text{minimize} & y \\ \text{subject to} & -y\mathbf{1} \preceq Ax - b \preceq y\mathbf{1} \end{array}$$

($\mathbf{1}$ is vector of ones)

# Quadratic program (QP)

$$\begin{array}{ll} \text{minimize} & (1/2)x^TPx + q^Tx + r \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

- $P \in \mathbf{S}^n_+$, so objective is convex quadratic
- minimize a convex quadratic function over a polyhedron



# Linear program with random cost

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Gx \preceq h \end{array}$$

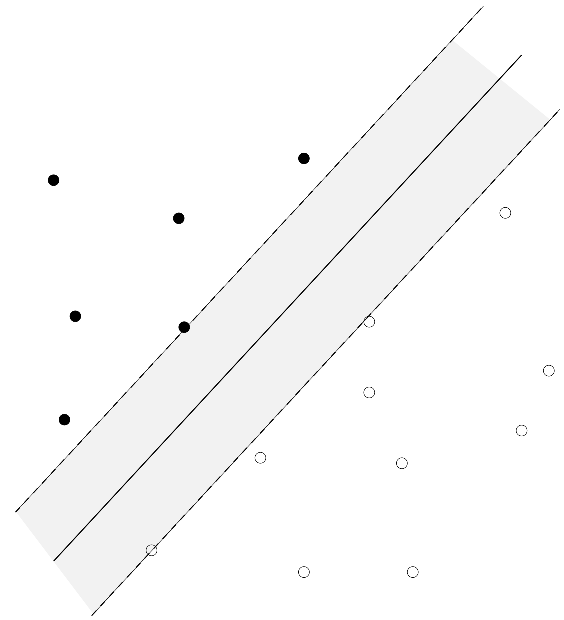
- $c$ is random vector with mean $\bar{c}$ and covariance $\Sigma$
- hence, $c^T x$ is random variable with mean $\bar{c}^T x$ and variance $x^T \Sigma x$

## Expected cost-variance trade-off

$$\begin{array}{ll} \text{minimize} & \mathbf{E}\, c^T x + \gamma\, \mathbf{var}(c^T x) = \bar{c}^T x + \gamma x^T \Sigma x \\ \text{subject to} & Gx \preceq h \end{array}$$

$\gamma > 0$ is risk aversion parameter

# Robust linear discrimination



$$\begin{aligned}\mathcal{H}_1 &= \{z \mid a^T z + b = 1\} \\ \mathcal{H}_2 &= \{z \mid a^T z + b = -1\}\end{aligned}$$

distance between hyperplanes is $2/\|a\|_2$

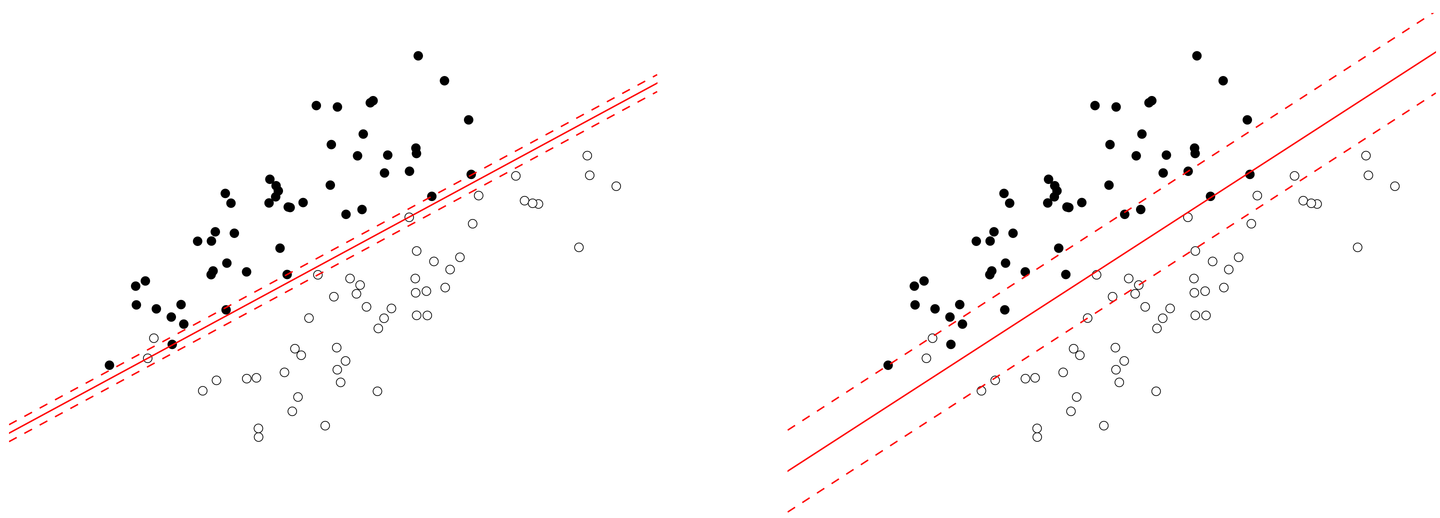
to separate two sets of points by maximum margin,

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

a quadratic program in $a$, $b$

# Support vector classifier

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

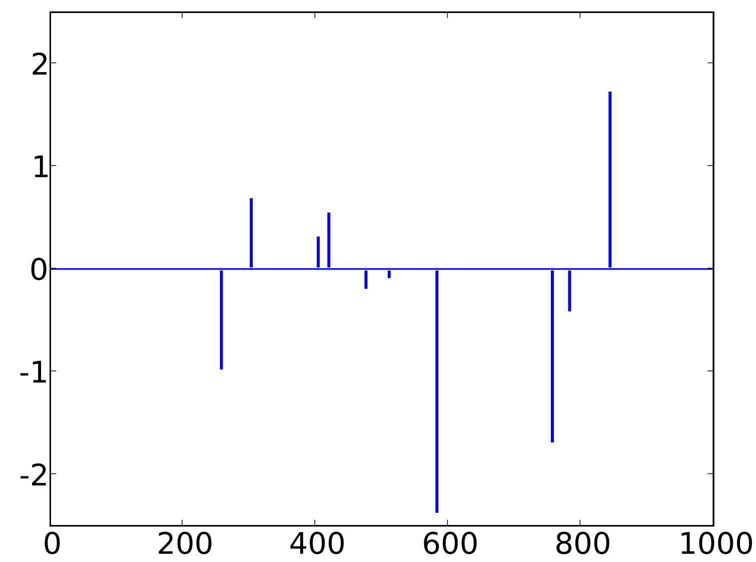


$\gamma = 0$ $\qquad$ $\gamma = 10$

equivalent to a QP

# Sparse signal reconstruction

- signal $\hat{x}$ of length $1000$
- ten nonzero components




reconstruct signal from $m = 100$ random noisy measurements

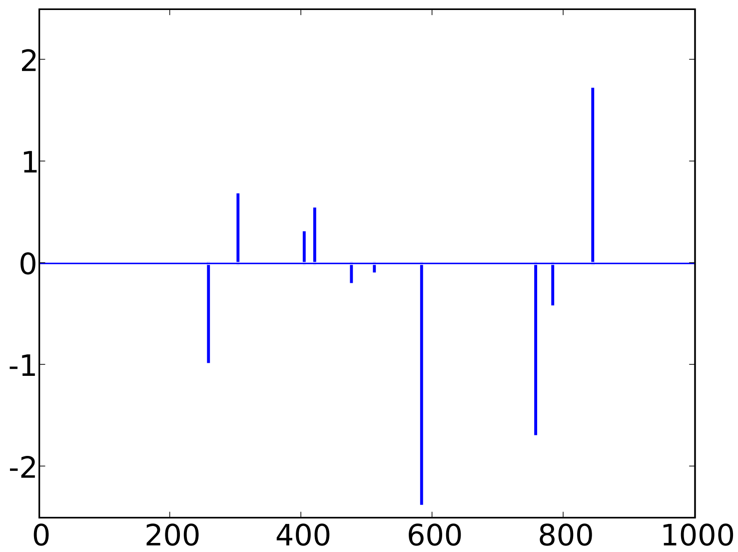
$$b = A\hat{x} + v$$

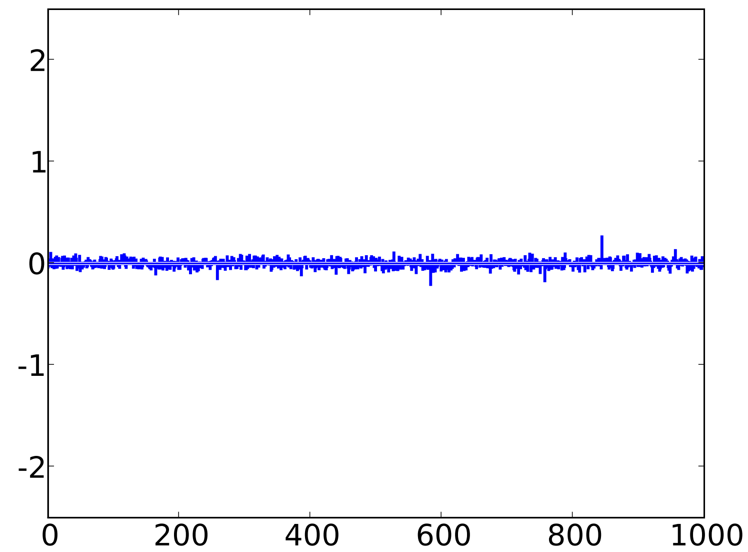
($A_{ij} \sim \mathcal{N}(0,1)$ i.i.d. and $v \sim \mathcal{N}(0, \sigma^2 I)$ with $\sigma = 0.01$)

# $\ell_2$-Norm regularization

$$\text{minimize} \quad \|Ax - b\|_2^2 + \gamma\|x\|_2^2$$

a least-squares problem



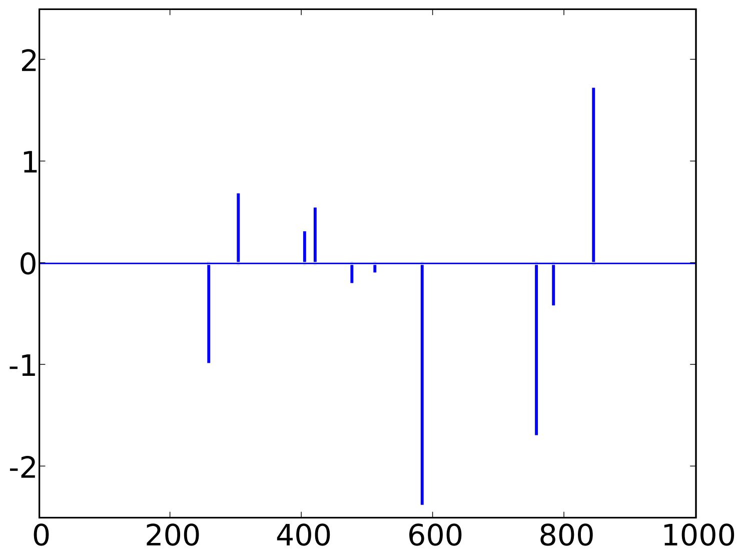



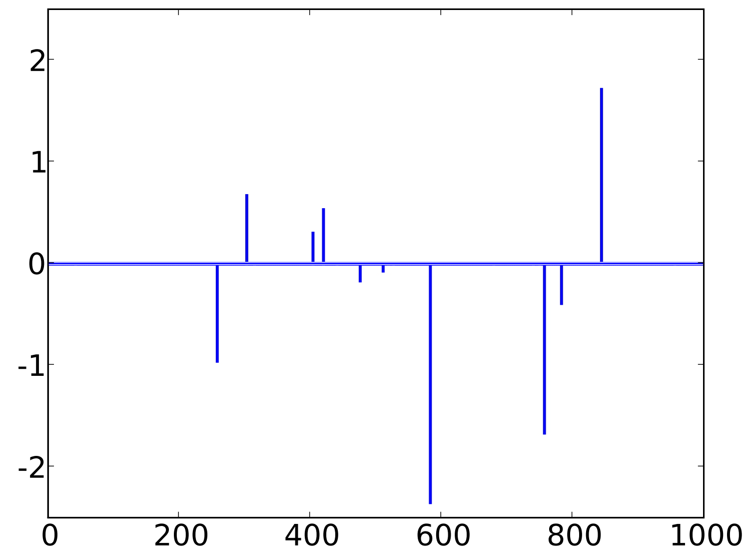

left: exact signal $\hat{x}$; right: 2-norm reconstruction

# $\ell_1$-Norm regularization

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

equivalent to a QP




left: exact signal $\hat{x}$; right: 1-norm reconstruction

# Geometric programming

**Posynomial function:**

$$f(x) = \sum_{k=1}^{K} c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}}, \qquad \mathbf{dom}\, f = \mathbf{R}^n_{++}$$

with $c_k > 0$

**Geometric program (GP)**

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

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:

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

$b_{ik} = \log c_{ik}$

# Modeling software

**Modeling packages for convex optimization**

- CVX, Yalmip (Matlab)
- CVXMOD (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

$$\begin{array}{ll}\text{minimize} & \|Ax-b\|_1 \\ \text{subject to} & -0.5 \leq x_k \leq 0.3, \quad k=1,\ldots,n\end{array}$$

## Matlab code

```
A = randn(5, 3);  b = randn(5, 1);
cvx_begin
    variable x(3);
    minimize(norm(A*x - b, 1))
    subject to
        -0.5 <= x;
         x <= 0.3;
cvx_end
```

- between `cvx_begin` and `cvx_end`, `x` is a CVX variable
- after execution, `x` is Matlab variable with optimal solution

# Cone programming

- generalized inequalities
- second-order cone programming
- semidefinite programming

# Cone linear program

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & Gx \preceq_K h \\ & Ax = b\end{array}$$

- $y \preceq_K z$ means $z - y \in K$, where $K$ is a proper convex cone
- extends linear programming ($K = \mathbf{R}^m_+$) to nonpolyhedral cones
- popular as standard format for nonlinear convex optimization
- theory and algorithms very similar to linear programming

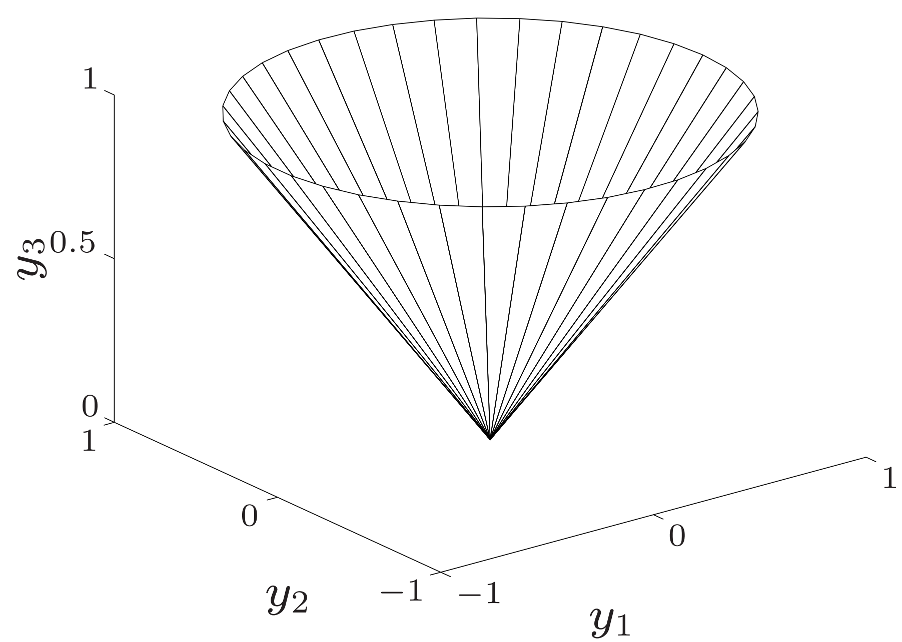
# Second-order cone program (SOCP)

$$\begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \ldots, m \end{array}$$

- $\|\cdot\|_2$ 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:

$$\left\{ y \,\middle|\, \sqrt{y_1^2 + \cdots + y_{p-1}^2} \leq y_p \right\}$$



# Examples of SOC-representable constraints

**Convex quadratic constraint** ($A = LL^T$ positive definite)

$$x^T A x + 2b^T x + c \leq 0 \qquad \Longleftrightarrow \qquad \left\| L^T x + L^{-1} b \right\|_2 \leq (b^T A^{-1} b - c)^{1/2}$$

also extends to positive semidefinite singular $A$

**Hyperbolic constraint**

$$x^T x \leq yz, \quad y, z \geq 0 \qquad \Longleftrightarrow \qquad \left\| \begin{bmatrix} 2x \\ y - z \end{bmatrix} \right\|_2 \leq y + z, \quad y, z \geq 0$$

# Examples of SOC-representable constraints

## Positive powers

$$x^{1.5} \leq t, \quad x \geq 0 \qquad \Longleftrightarrow \qquad \exists z: \quad x^2 \leq tz, \quad z^2 \leq x, \quad x, z \geq 0$$

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

## Negative powers

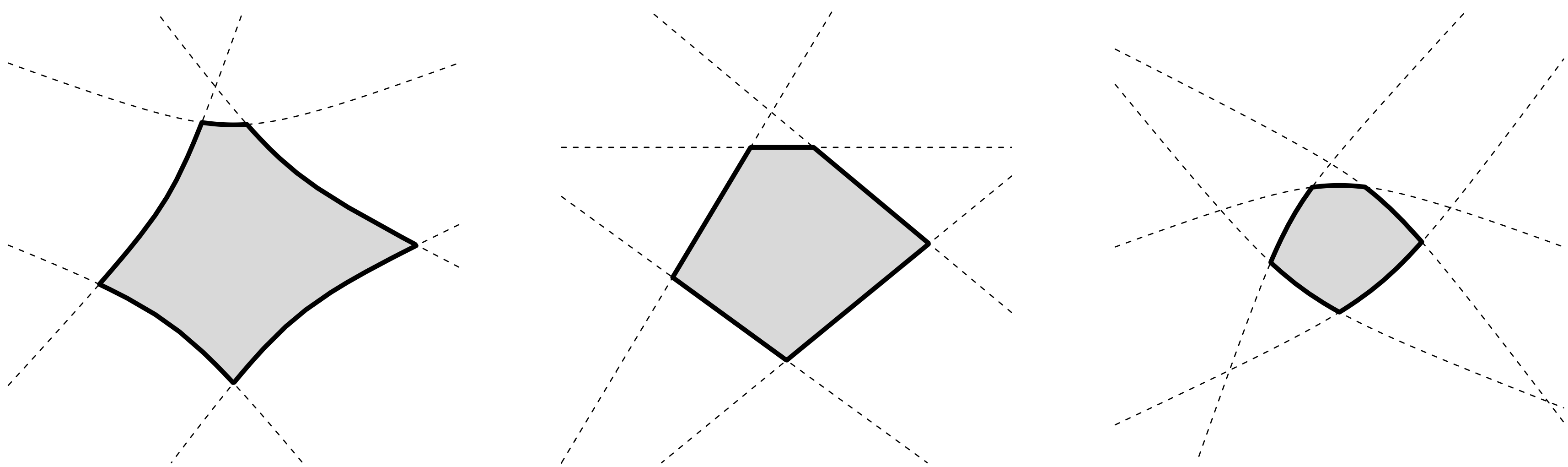
$$x^{-3} \leq t, \quad x > 0 \qquad \Longleftrightarrow \qquad \exists z: \quad 1 \leq tz, \quad z^2 \leq tx, \quad x, z \geq 0$$

- two hyperbolic constraints can be converted to SOC constraints
- extends to powers $x^p$ for rational $p < 0$

# Robust linear program (stochastic)

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \mathbf{prob}(a_i^T x \le b_i) \ge \eta, \quad i = 1, \ldots, m \end{array}$$

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



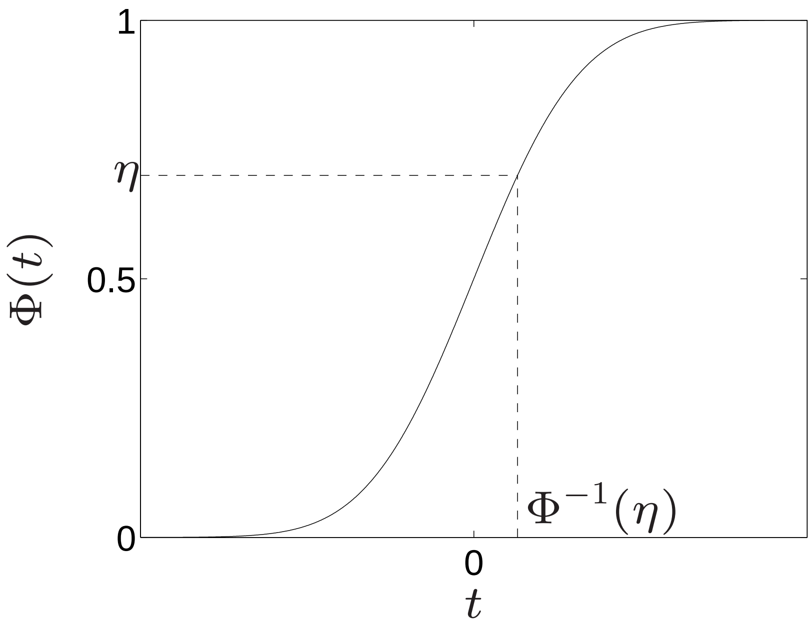
$\eta = 10\%$ $\eta = 50\%$ $\eta = 90\%$

# SOCP formulation

the 'chance constraint' $\mathbf{prob}(a_i^T x \leq b_i) \geq \eta$ 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)

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

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

## SOCP formulation

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \bar{a}_i^T x + \|P_i^T x\|_2 \leq b_i, \quad i = 1, \ldots, m \end{array}$$

follows from

$$\sup_{\|u\|_2 \leq 1} (\bar{a}_i + P_i u)^T x = \bar{a}_i^T + \|P_i^T x\|_2$$

# Semidefinite program (SDP)

$$
\begin{array}{ll}
\text{minimize} & c^T x \\
\text{subject to} & x_1 A_1 + x_2 A_2 + \cdots + x_n A_n \preceq B
\end{array}
$$

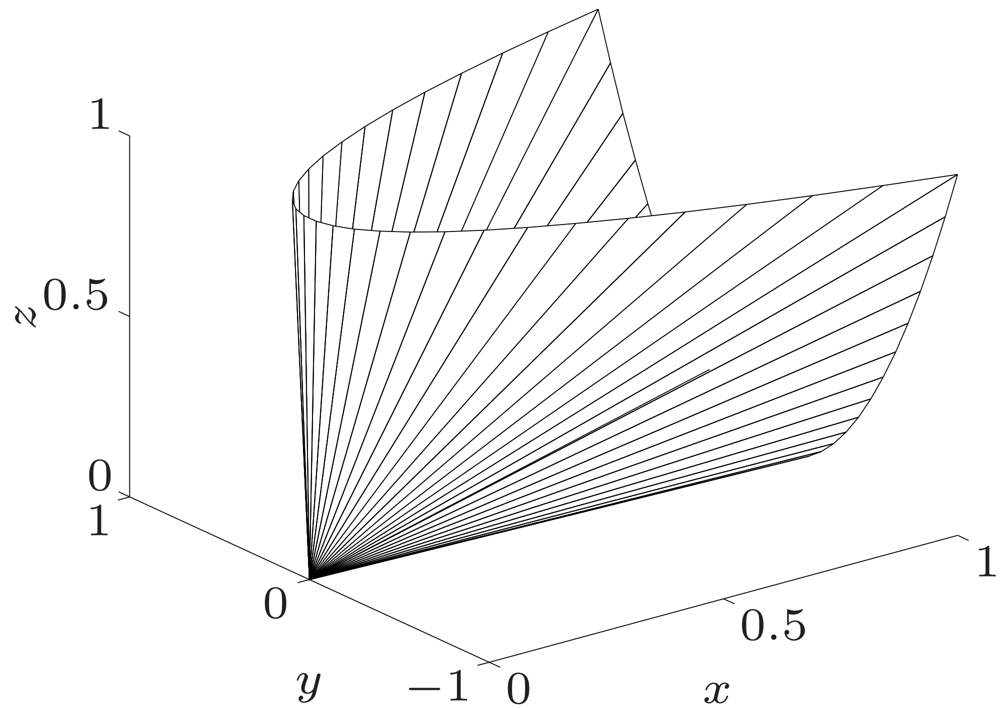
- $A_1$, $A_2$, . . . , $A_n$, $B$ are symmetric matrices
- inequality $X \preceq Y$ means $Y - X$ is *positive semidefinite*, $i.e.$,

$$
z^T(Y - X)z = \sum_{i,j}(Y_{ij} - X_{ij})z_i z_j \geq 0 \text{ for all } z
$$

- includes many nonlinear constraints as special cases

# Geometry

$$\begin{bmatrix} x & y \\ y & z \end{bmatrix} \succeq 0$$




- a nonpolyhedral convex cone
- feasible set of a semidefinite program is the intersection of the positive semidefinite cone in high dimension with planes

# Examples

$$A(x) = A_0 + x_1 A_1 + \cdots + x_m A_m \qquad (A_i \in \mathbf{S}^n)$$

**Eigenvalue minimization** (and equivalent SDP)

$$\begin{array}{ll} \text{minimize} & \lambda_{\max}(A(x)) \end{array} \qquad\qquad \begin{array}{ll} \text{minimize} & t \\ \text{subject to} & A(x) \preceq tI \end{array}$$

**Matrix-fractional function**

$$\begin{array}{ll} \text{minimize} & b^T A(x)^{-1} b \\ \text{subject to} & A(x) \succeq 0 \end{array} \qquad\qquad \begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \left[\begin{array}{cc} A(x) & b \\ b^T & t \end{array}\right] \succeq 0 \end{array}$$

# Matrix norm minimization

$$A(x) = A_0 + x_1 A_1 + x_2 A_2 + \cdots + x_n A_n \qquad (A_i \in \mathbf{R}^{p \times q})$$

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

$$\begin{array}{ll} \text{minimize} & \|A(x)\|_2 \end{array} \qquad \begin{array}{ll} \text{minimize} & t \\ \text{subject to} & \begin{bmatrix} tI & A(x)^T \\ A(x) & tI \end{bmatrix} \succeq 0 \end{array}$$

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

$$\begin{array}{ll} \text{minimize} & \|A(x)\|_* \end{array} \qquad \begin{array}{ll} \text{minimize} & (\mathbf{tr}\, U + \mathbf{tr}\, V)/2 \\ \text{subject to} & \begin{bmatrix} U & A(x)^T \\ A(x) & V \end{bmatrix} \succeq 0 \end{array}$$

# Semidefinite relaxations & randomization

semidefinite programming is increasingly used

- to find good bounds for hard (i.e., nonconvex) problems, via **relaxation**
- as a heuristic for good suboptimal points, often via **randomization**

**Example: Boolean least-squares**

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 \\ \text{subject to} & x_i^2 = 1, \quad i = 1, \ldots, n \end{array}$$

- basic problem in digital communications
- could check all $2^n$ possible values of $x \in \{-1, 1\}^n$ . . .
- an NP-hard problem, and very hard in practice

# Semidefinite lifting

with $P = A^T A$, $q = -A^T b$, $r = b^T b$

$$\|Ax - b\|_2^2 = \sum_{i,j=1}^{n} P_{ij} x_i x_j + 2 \sum_{i=1}^{n} q_i x_i + r$$

after introducing new variables $X_{ij} = x_i x_j$

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i,j=1}^{n} P_{ij} X_{ij} + 2 \sum_{i=1}^{n} q_i x_i + r \\ \text{subject to} & X_{ii} = 1, \quad i = 1, \ldots, n \\ & X_{ij} = x_i x_j, \quad i, j = 1, \ldots, n \end{array}$$

- cost function and first constraints are linear
- last constraint in matrix form is $X = xx^T$, nonlinear and nonconvex,

. . . still a very hard problem

# Semidefinite relaxation

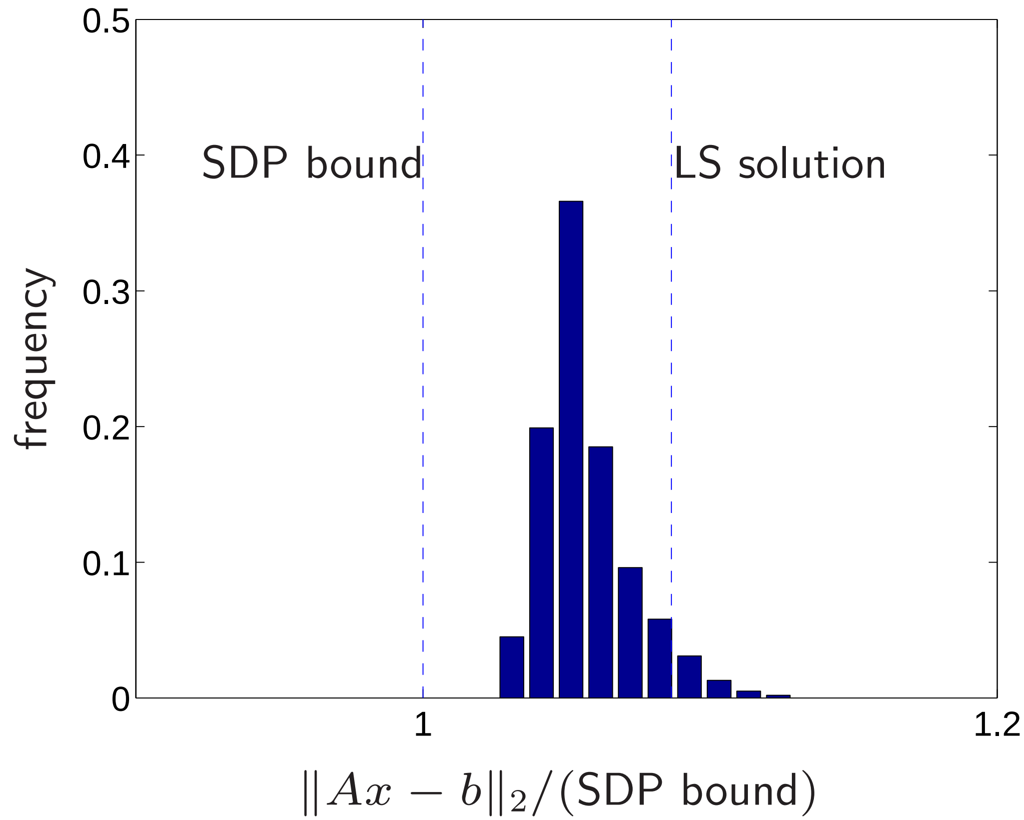
replace $X = xx^T$ with weaker constraint $X \succeq xx^T$, to obtain **relaxation**

$$\begin{array}{ll} \text{minimize} & \sum_{i,j=1}^{n} P_{ij}X_{ij} + 2\sum_{i=1}^{n} q_i x_i + r \\ \text{subject to} & X_{ii} = 1, \quad i = 1, \ldots, n \\ & X \succeq xx^T \end{array}$$

- convex; can be solved as an semidefinite program
- optimal value gives lower bound for BLS
- if $X = xx^T$ at the optimum, we have solved the exact problem
- otherwise, can use *randomized rounding*

  generate $z$ from $\mathcal{N}(x, X - xx^T)$ and take $x = \mathbf{sign}(z)$

# Example




- feasible set has $2^{100} \approx 10^{30}$ points
- histogram of 1000 randomized solutions from SDP relaxation

# Nonnegative polynomial on R

$$f(t) = x_0 + x_1 t + \cdots + x_{2m} t^{2m} \geq 0 \quad \text{for all } t \in \mathbf{R}$$

- a convex constraint on $x$
- holds if and only if $f$ is a sum of squares of (two) polynomials:

$$\begin{aligned} f(t) &= \sum_k (y_{k0} + y_{k1} t + \cdots + y_{km} t^m)^2 \\ &= \begin{bmatrix} 1 \\ \vdots \\ t^m \end{bmatrix}^T \sum_k y_k y_k^T \begin{bmatrix} 1 \\ \vdots \\ t^m \end{bmatrix} \\ &= \begin{bmatrix} 1 \\ \vdots \\ t^m \end{bmatrix}^T Y \begin{bmatrix} 1 \\ \vdots \\ t^m \end{bmatrix} \end{aligned}$$

where $Y = \sum_k y_k y_k^T \succeq 0$

# SDP formulation

$f(t) \geq 0$ if and only if for some $Y \succeq 0$,

$$f(t) = \begin{bmatrix} 1 \\ t \\ \vdots \\ t^{2m} \end{bmatrix}^T \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{2m} \end{bmatrix} = \begin{bmatrix} 1 \\ t \\ \vdots \\ t^m \end{bmatrix}^T Y \begin{bmatrix} 1 \\ t \\ \vdots \\ t^m \end{bmatrix}$$

this is an SDP constraint: there exists $Y \succeq 0$ such that

$$\begin{array}{rcl} x_0 & = & Y_{11} \\ x_1 & = & Y_{12} + Y_{21} \\ x_2 & = & Y_{13} + Y_{22} + Y_{32} \\ & \vdots & \\ x_{2m} & = & Y_{m+1,m+1} \end{array}$$

# General sum-of-squares constraints

$f(t) = x^T p(t)$ is a sum of squares if

$$x^T p(t) = \sum_{k=1}^{s} (y_k^T q(t))^2 = q(t)^T \left( \sum_{k=1}^{s} y_k y_k^T \right) q(t)$$

- $p$, $q$: basis functions (of polynomials, trigonometric polynomials, . . . )
- independent variable $t$ can be one- or multidimensional
- a *sufficient* condition for nonnegativity of $x^T p(t)$, useful in nonconvex polynomial optimization in several variables
- in some nontrivial cases ($e.g.$, polynomial on **R**), *necessary and sufficient*

**Equivalent SDP constraint** (on the variables $x$, $X$)

$$x^T p(t) = q(t)^T X q(t), \qquad X \succeq 0$$

# Example: Cosine polynomials

$$f(\omega) = x_0 + x_1 \cos\omega + \cdots + x_{2n} \cos 2n\omega \geq 0$$

**Sum of squares theorem:** $f(\omega) \geq 0$ for $\alpha \leq \omega \leq \beta$ if and only if

$$f(\omega) = g_1(\omega)^2 + s(\omega) g_2(\omega)^2$$

- $g_1$, $g_2$: cosine polynomials of degree $n$ and $n-1$
- $s(\omega) = (\cos\omega - \cos\beta)(\cos\alpha - \cos\omega)$ is a given weight function

**Equivalent SDP formulation:** $f(\omega) \geq 0$ for $\alpha \leq \omega \leq \beta$ if and only if

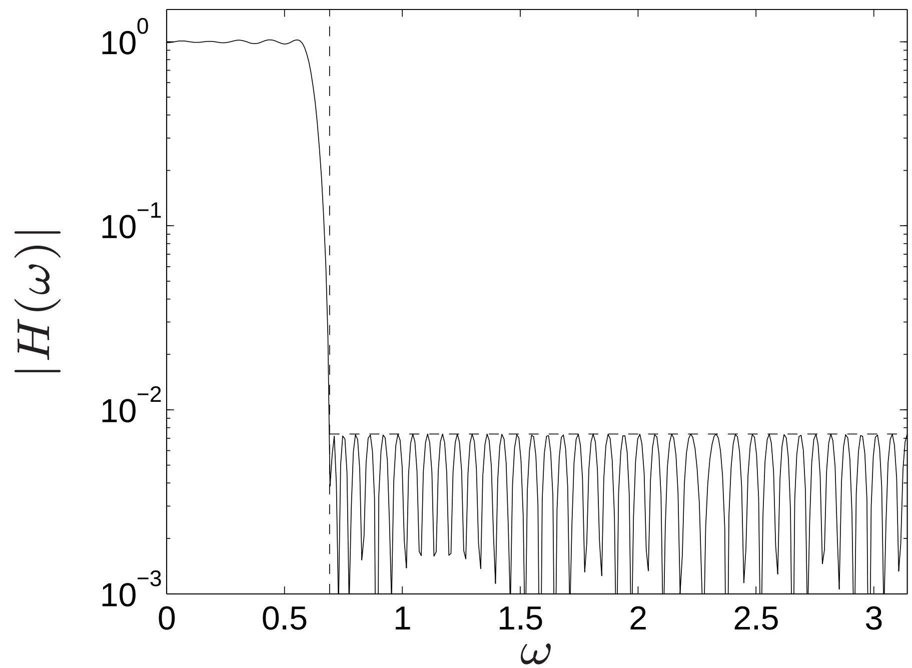
$$x^T p(\omega) = q_1(\omega)^T X_1 q_1(\omega) + s(\omega) q_2(\omega)^T X_2 q_2(\omega), \quad X_1 \succeq 0, \quad X_2 \succeq 0$$

$p$, $q_1$, $q_2$: basis vectors $(1, \cos\omega, \cos(2\omega), \ldots)$ up to order $2n$, $n$, $n-1$

# Example: Linear-phase Nyquist filter

$$\text{minimize} \quad \sup_{\omega \geq \omega_s} |h_0 + h_1 \cos\omega + \cdots + h_{2n} \cos 2n\omega|$$

with $h_0 = 1/M$, $h_{kM} = 0$ for positive integer $k$




(Example with $n = 25$, $M = 5$, $\omega_s = 0.69$)

# SDP formulation

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -t \le H(\omega) \le t, \quad \omega_{\rm s} \le \omega \le \pi \end{array}$$

where $H(\omega) = h_0 + h_1 \cos\omega + \cdots + h_{2n} \cos 2n\omega$

## Equivalent SDP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & t - H(\omega) = q_1(\omega)^T X_1 q_1(\omega) + s(\omega) q_2(\omega)^T X_2 q_2(\omega) \\ & t + H(\omega) = q_1(\omega)^T X_3 q_1(\omega) + s(\omega) q_2(\omega)^T X_3 q_2(\omega) \\ & X_1 \succeq 0, \quad X_2 \succeq 0, \quad X_3 \succeq 0, \quad X_4 \succeq 0 \end{array}$$

Variables $t$, $h_i$ $(i \ne kM)$, 4 matrices $X_i$ of size roughly $n$

# Chebyshev inequalities

## Classical (two-sided) Chebyshev inequality

$$\mathbf{prob}(|X| < 1) \geq 1 - \sigma^2$$

- holds for all random $X$ with $\mathbf{E}\,X = 0$, $\mathbf{E}\,X^2 = \sigma^2$
- there exists a distribution that achieves the bound

## Generalized Chebyshev inequalities

give lower bound on $\mathbf{prob}(X \in C)$, given moments of $X$

# Chebyshev inequality for quadratic constraints

- $C$ is defined by quadratic inequalities

$$C = \{x \in \mathbf{R}^n \mid x^T A_i x + 2b_i^T x + c_i \leq 0,\ i = 1, \ldots, m\}$$

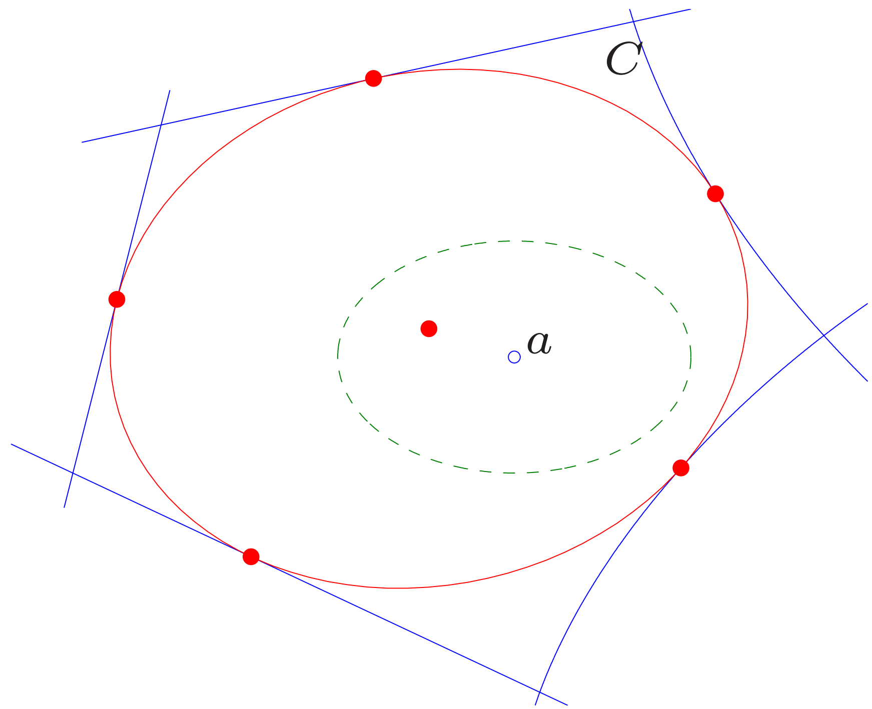
- $X$ is random vector with $\mathbf{E}\, X = a$, $\mathbf{E}\, XX^T = S$

**SDP formulation** (variables $P \in \mathbf{S}^n$, $q \in \mathbf{R}^n$, $r, \tau_1, \ldots, \tau_m \in \mathbf{R}$)

$$\begin{array}{ll} \text{maximize} & 1 - \mathbf{tr}(SP) - 2a^T q - r \\ \text{subject to} & \left[\begin{array}{cc} P & q \\ q^T & r-1 \end{array}\right] \succeq \tau_i \left[\begin{array}{cc} A_i & b_i \\ b_i^T & c_i \end{array}\right], \quad \tau_i \geq 0 \quad i = 1, \ldots, m \\ & \left[\begin{array}{cc} P & q \\ q^T & r \end{array}\right] \succeq 0 \end{array}$$

optimal value is tight lower bound on $\mathbf{prob}(X \in S)$

# Example




- $a = \mathbf{E}\,X$; dashed line shows $\{x \mid (x-a)^T(S - aa^T)^{-1}(x-a) = 1\}$
- lower bound on $\mathbf{prob}(X \in C)$ is achieved by distribution shown in red
- ellipse is defined by $x^TPx + 2q^Tx + r = 1$

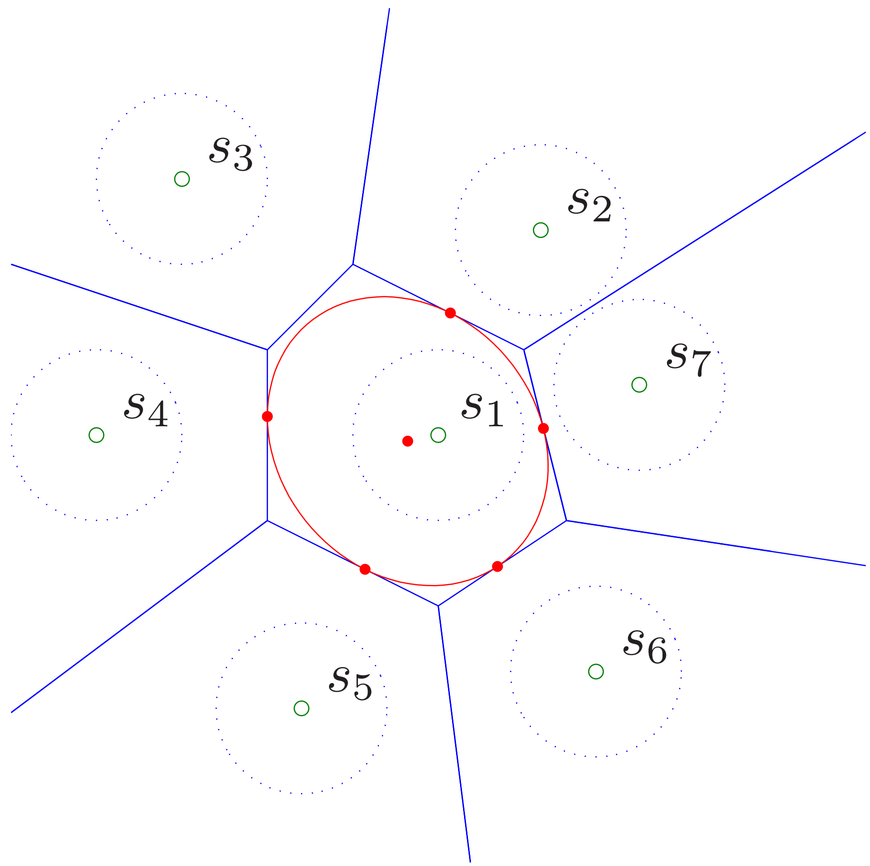
# Detection example

$$x = s + v$$

- $x \in \mathbf{R}^n$: received signal
- $s$: transmitted signal $s \in \{s_1, s_2, \ldots, s_N\}$ (one of $N$ possible symbols)
- $v$: noise with $\mathbf{E}\, v = 0$, $\mathbf{E}\, vv^T = \sigma^2 I$

**Detection problem**: given observed value of $x$, estimate $s$

**Example** $(N = 7)$: bound on probability of correct detection of $s_1$ is $0.205$




dots: distribution with probability of correct detection $0.205$

# Cone programming duality

**Primal and dual cone program**

| | | | | | |
|---|---|---|---|---|---|
| P: | minimize | $c^T x$ | D: | maximize | $-b^T z$ |
| | subject to | $Ax \preceq_K b$ | | subject to | $A^T z + c = 0$ |
| | | | | | $z \succeq_{K^*} 0$ |

- optimal values are equal (if primal or dual is strictly feasible)
- dual inequality is with respect to the dual cone

$$K^* = \{z \mid x^T z \geq 0 \text{ for all } x \in K\}$$

- $K = K^*$ for linear, second-order cone, semidefinite programming

**Applications:** optimality conditions, sensitivity analysis, algorithms, . . .

# Interior-point methods

- Newton's method
- barrier method
- primal-dual interior-point methods
- problem structure

# Equality-constrained convex optimization

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & Ax = b \end{array}$$

$f$ twice continuously differentiable and convex

**Optimality (Karush-Kuhn-Tucker or KKT) condition**

$$\nabla f(x) + A^T y = 0, \qquad Ax = b$$

**Example:** $f(x) = (1/2)x^T P x + q^T x + r$ with $P \succeq 0$

$$\left[\begin{array}{cc} P & A^T \\ A & 0 \end{array}\right] \left[\begin{array}{c} x \\ y \end{array}\right] = \left[\begin{array}{c} -q \\ b \end{array}\right]$$

a symmetric indefinite set of equations, known as a KKT system

# Newton step

replace $f$ with second-order approximation $f_{\mathrm{q}}$ at feasible $\hat{x}$:

$$\begin{array}{ll} \text{minimize} & f_{\mathrm{q}}(x) \stackrel{\Delta}{=} f(\hat{x}) + \nabla f(\hat{x})^T(x - \hat{x}) + \frac{1}{2}(x - \hat{x})^T \nabla^2 f(\hat{x})(x - \hat{x}) \\ \text{subject to} & Ax = b \end{array}$$

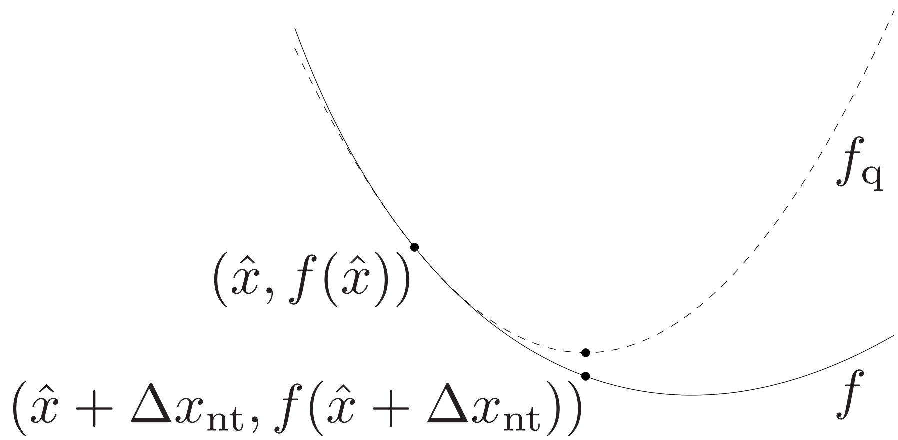
solution is $x = \hat{x} + \Delta x_{\mathrm{nt}}$ with $\Delta x_{\mathrm{nt}}$ defined by

$$\left[ \begin{array}{cc} \nabla^2 f(\hat{x}) & A^T \\ A & 0 \end{array} \right] \left[ \begin{array}{c} \Delta x_{\mathrm{nt}} \\ w \end{array} \right] = \left[ \begin{array}{c} -\nabla f(\hat{x}) \\ 0 \end{array} \right]$$

$\Delta x_{\mathrm{nt}}$ is called the **Newton step** at $\hat{x}$

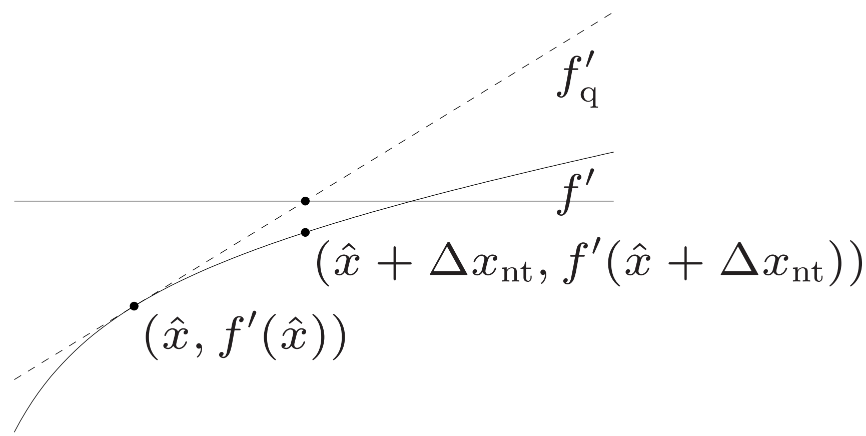
# Interpretation (for unconstrained problem)

$\hat{x} + \Delta x_{\rm nt}$ minimizes 2nd-order approximation $f_{\rm q}$




$\hat{x} + \Delta x_{\rm nt}$ solves linearized optimality condition

$$\begin{aligned} \nabla f_{\rm q}(x) &\\ &= \nabla f(\hat{x}) + \nabla^2 f(\hat{x})(x - \hat{x}) \\ &= 0 \end{aligned}$$



# Newton's algorithm

given starting point $x^{(0)} \in \mathbf{dom}\, f$ with $Ax^{(0)} = b$, tolerance $\epsilon$
repeat for $k = 0, 1, \ldots$

1. compute Newton step $\Delta x_{\mathrm{nt}}$ at $x^{(k)}$ by solving

$$\left[\begin{array}{cc} \nabla^2 f(x^{(k)}) & A^T \\ A & 0 \end{array}\right] \left[\begin{array}{c} \Delta x_{\mathrm{nt}} \\ w \end{array}\right] = \left[\begin{array}{c} -\nabla f(x^{(k)}) \\ 0 \end{array}\right]$$

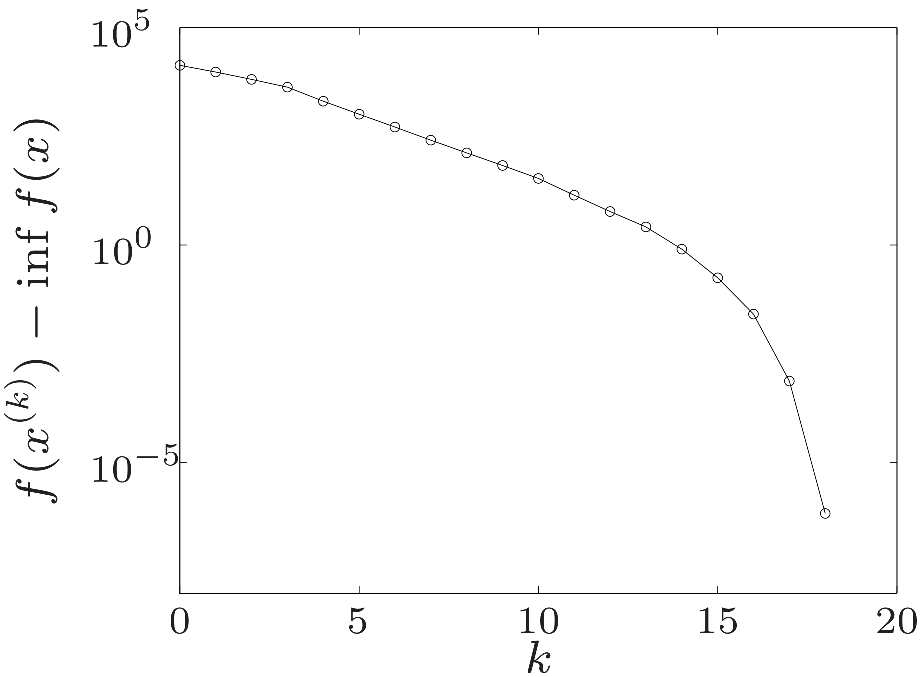
2. terminate if $-\nabla f(x^{(k)})^T \Delta x_{\mathrm{nt}} \leq \epsilon$
3. $x^{(k+1)} = x^{(k)} + t\Delta x_{\mathrm{nt}}$, with $t$ determined by line search

## Comments

- $\nabla f(x^{(k)})^T \Delta x_{\mathrm{nt}}$ is directional derivative at $x^{(k)}$ in Newton direction
- line search needed to guarantee $f(x^{(k+1)}) < f(x^{(k)})$, global convergence

# Example

$$f(x) = -\sum_{i=1}^{n} \log(1 - x_i^2) - \sum_{i=1}^{m} \log(b_i - a_i^T x) \qquad \text{(with } n = 10^4,\ m = 10^5\text{)}$$




- high accuracy after small number of iterations
- fast asymptotic convergence

# Classical convergence analysis

**Assumptions** ($m$, $L$ are positive constants)

- $f$ strongly convex: $\nabla^2 f(x) \succeq mI$
- $\nabla^2 f$ Lipschitz continuous: $\|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq L\|x - y\|_2$

**Summary:** two regimes

- damped phase ($\|\nabla f(x)\|_2$ large): for some constant $\gamma > 0$

$$f(x^{(k+1)}) - f(x^{(k)}) \leq -\gamma$$

- quadratic convergence ($\|\nabla f(x)\|_2$ small)

$$\|\nabla f(x^{(k)})\|_2 \text{ decreases quadratically}$$

# Self-concordant functions

**Shortcomings of classical convergence analysis**

- depends on unknown constants ($m$, $L$, ...)
- bound is not affinely invariant, although Newton's method is

**Analysis for self-concordant functions** (Nesterov and Nemirovski, 1994)

- a convex function of one variable is self-concordant if

$$|f'''(x)| \leq 2f''(x)^{3/2} \quad \text{for all } x \in \mathbf{dom}\, f$$

  a function of several variables is s.c. if its restriction to lines is s.c.

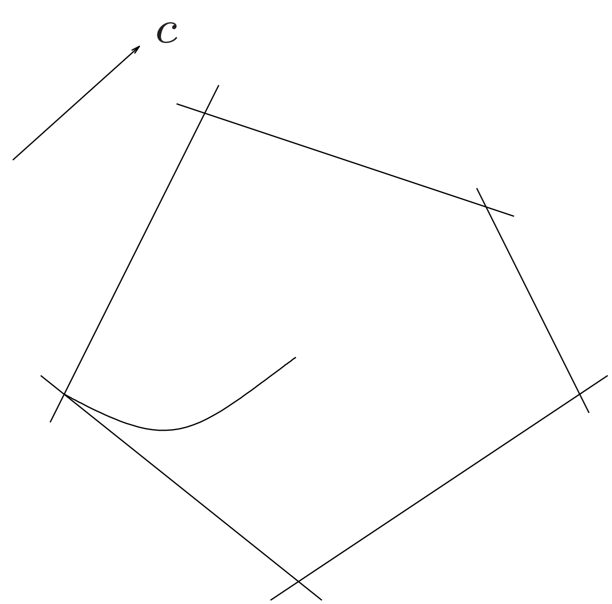
- analysis is affine-invariant, does not depend on unknown constants
- developed for complexity theory of interior-point methods

# Interior-point methods

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

functions $f_i$, $i = 0, 1, \ldots, m$, are convex

**Basic idea:** follow 'central path' through interior feasible set to solution



# General properties

- path-following mechanism relies on Newton's method
- every iteration requires solving a set of linear equations (KKT system)
- number of iterations small (10–50), fairly independent of problem size
- some versions known to have polynomial worst-case complexity

## History

- introduced in 1950s and 1960s
- used in polynomial-time methods for linear programming (1980s)
- polynomial-time algorithms for general convex optimization (ca. 1990)

# Reformulation via indicator function

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

## Reformulation

$$
\begin{array}{ll}
\text{minimize} & f_0(x) + \sum_{i=1}^m I_-(f_i(x)) \\
\text{subject to} & Ax = b
\end{array}
$$

where $I_-$ is indicator function of $\mathbf{R}_-$:

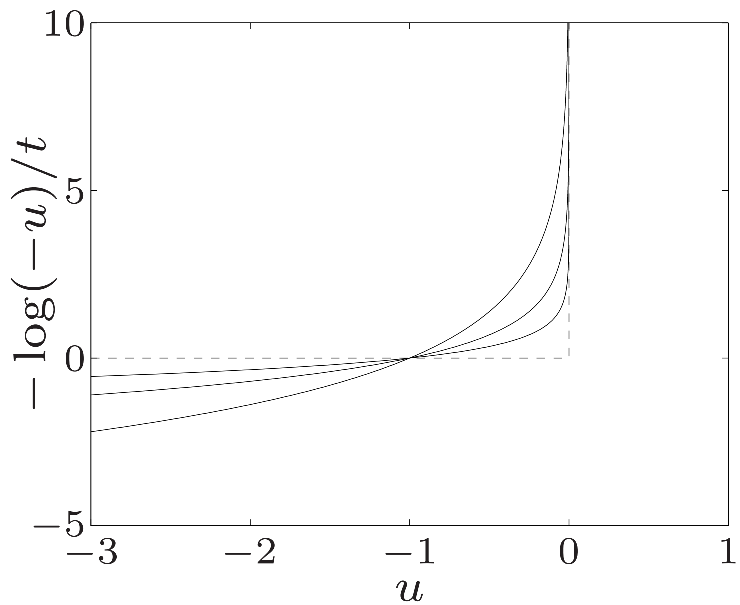
$$
I_-(u) = 0 \quad \text{if } u \leq 0, \qquad I_-(u) = \infty \quad \text{otherwise}
$$

- reformulated problem has no inequality constraints
- however, objective function is not differentiable

# Approximation via logarithmic barrier

$$\begin{array}{ll} \text{minimize} & f_0(x) - \dfrac{1}{t}\displaystyle\sum_{i=1}^{m} \log(-f_i(x)) \\ \text{subject to} & Ax = b \end{array}$$

- for $t > 0$, $-(1/t)\log(-u)$ is a smooth approximation of $I_-$
- approximation improves as $t \to \infty$



# Logarithmic barrier function

$$\phi(x) = -\sum_{i=1}^{m} \log(-f_i(x))$$

with $\mathbf{dom}\, \phi = \{x \mid f_1(x) < 0, \ldots, f_m(x) < 0\}$

- convex (follows from composition rules and convexity of $f_i$)
- twice continuously differentiable, with derivatives

$$\begin{aligned}
\nabla \phi(x) &= \sum_{i=1}^{m} \frac{1}{-f_i(x)} \nabla f_i(x) \\
\nabla^2 \phi(x) &= \sum_{i=1}^{m} \frac{1}{f_i(x)^2} \nabla f_i(x) \nabla f_i(x)^T + \sum_{i=1}^{m} \frac{1}{-f_i(x)} \nabla^2 f_i(x)
\end{aligned}$$

# Central path

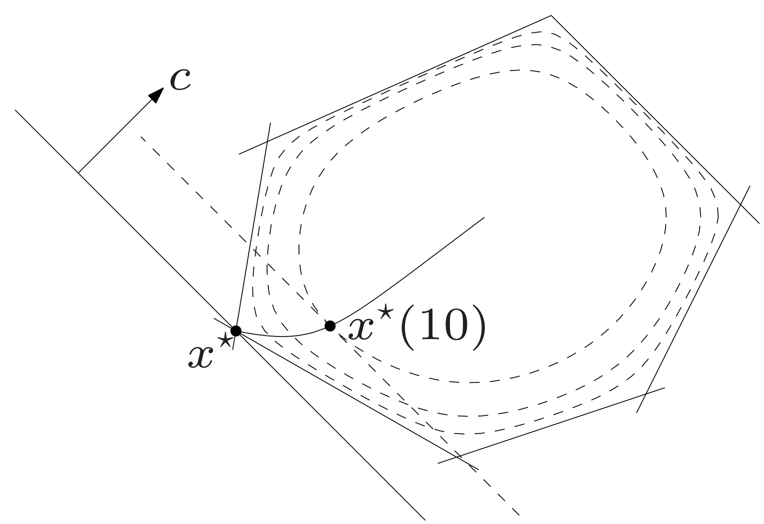
central path is $\{x^\star(t) \mid t > 0\}$, where $x^\star(t)$ is the solution of

$$\begin{array}{ll} \text{minimize} & t f_0(x) + \phi(x) \\ \text{subject to} & Ax = b \end{array}$$

**Example: central path for an LP**

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \ldots, 6 \end{array}$$

hyperplane $c^T x = c^T x^\star(t)$ is tangent to level curve of $\phi$ through $x^\star(t)$



# Barrier method

given strictly feasible $x$, $t := t^{(0)} > 0$, $\mu > 1$, tolerance $\epsilon > 0$
repeat:

1. *Centering step.* Compute $x^\star(t)$ and set $x := x^\star(t)$
2. *Stopping criterion.* Terminate if $m/t < \epsilon$
3. *Increase* $t$*.* $t := \mu t$

- stopping criterion $m/t \leq \epsilon$ guarantees

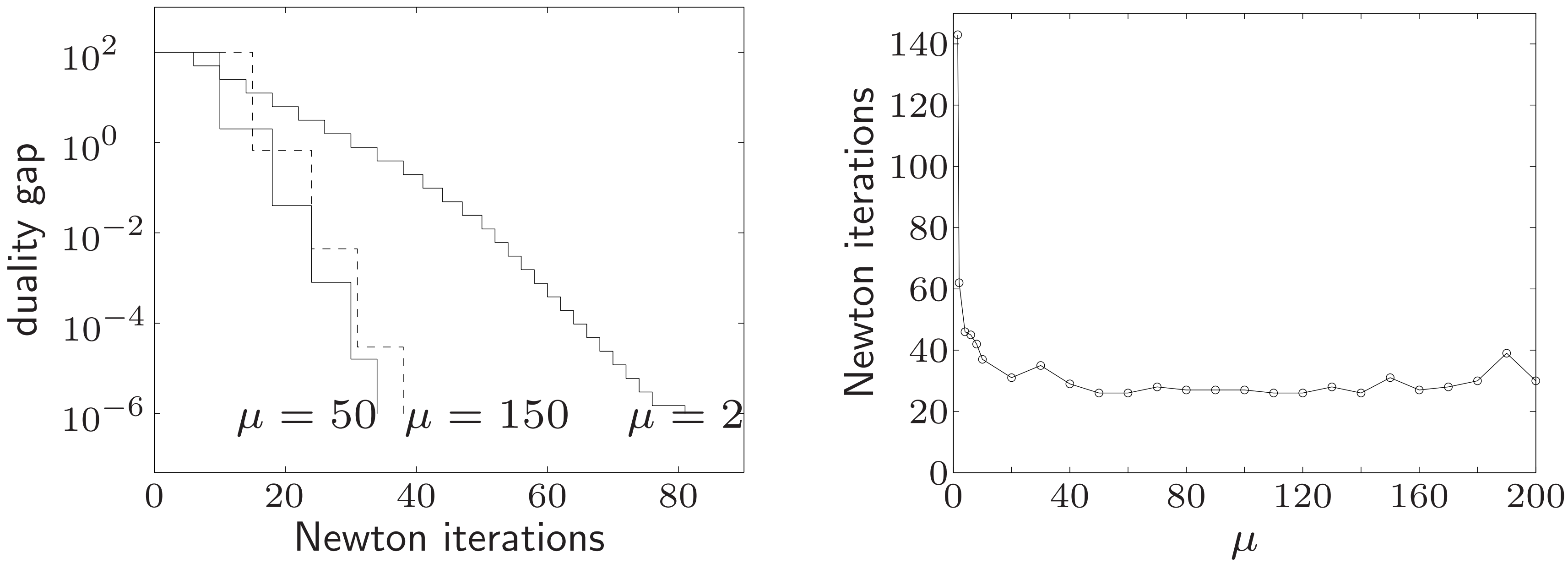
$$f_0(x) - \text{optimal value} \leq \epsilon$$

  (follows from duality)

- typical value of $\mu$ is $10$–$20$
- several heuristics for choice of $t^{(0)}$
- centering usually done using Newton's method, starting at current $x$

# Example: Inequality form LP

$m = 100$ inequalities, $n = 50$ variables

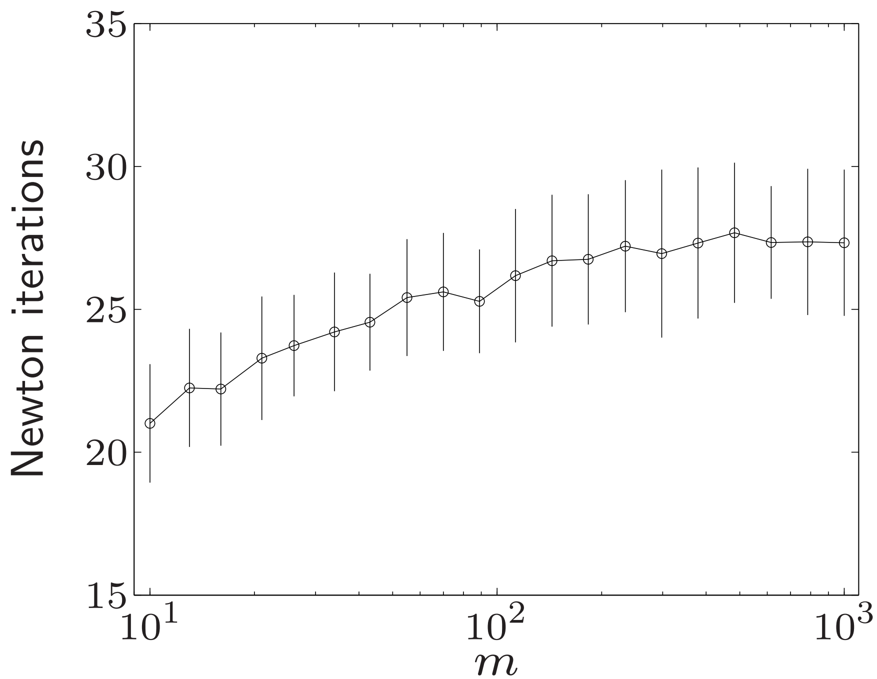



- starts with $x$ on central path ($t^{(0)} = 1$, duality gap $100$)
- terminates when $t = 10^8$ (gap $m/t = 10^{-6}$)
- total number of Newton iterations not very sensitive for $\mu \geq 10$

# Family of standard LPs

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b, \quad x \succeq 0 \end{array}$$

$A \in \mathbf{R}^{m \times 2m}$; for each $m$, solve 100 randomly generated instances




number of iterations grows very slowly as $m$ ranges over a $100 : 1$ ratio

# Second-order cone programming

$$
\begin{array}{ll}
\text{minimize} & f^T x \\
\text{subject to} & \|A_i x + b_i\|_2 \leq c_i^T x + d_i, \quad i = 1, \ldots, m
\end{array}
$$

**Logarithmic barrier function**

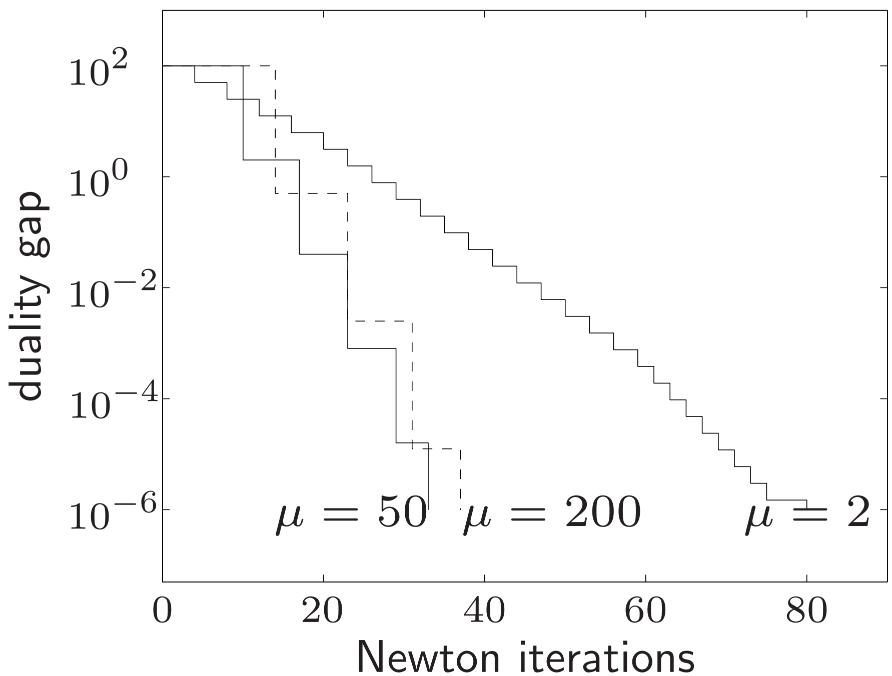
$$
\phi(x) = -\sum_{i=1}^{m} \log\left( (c_i^T x + d_i)^2 - \|A_i x + b_i\|_2^2 \right)
$$

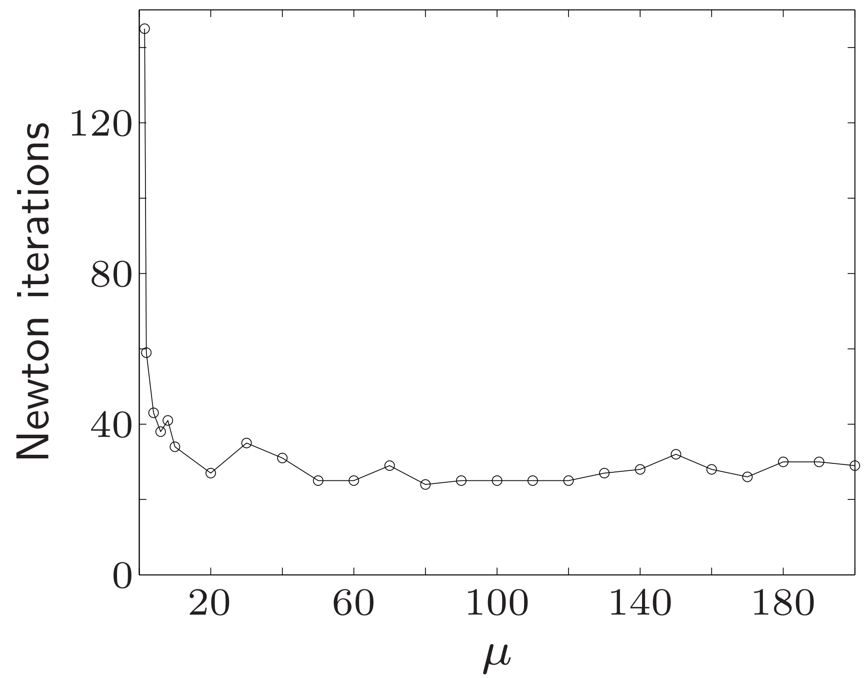
- a convex function
- $\log(v^2 - u^T u)$ is 'logarithm' for 2nd-order cone $\{(u, v) \mid \|u\|_2 \leq v\}$

**Barrier method:** follows central path $x^\star(t) = \mathrm{argmin}(t f^T x + \phi(x))$

# Example

$50$ variables, $50$ second-order cone constraints in $\mathbf{R}^6$



# Semidefinite programming

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & x_1 A_1 + \cdots + x_n A_n \preceq B \end{array}$$

**Logarithmic barrier function**

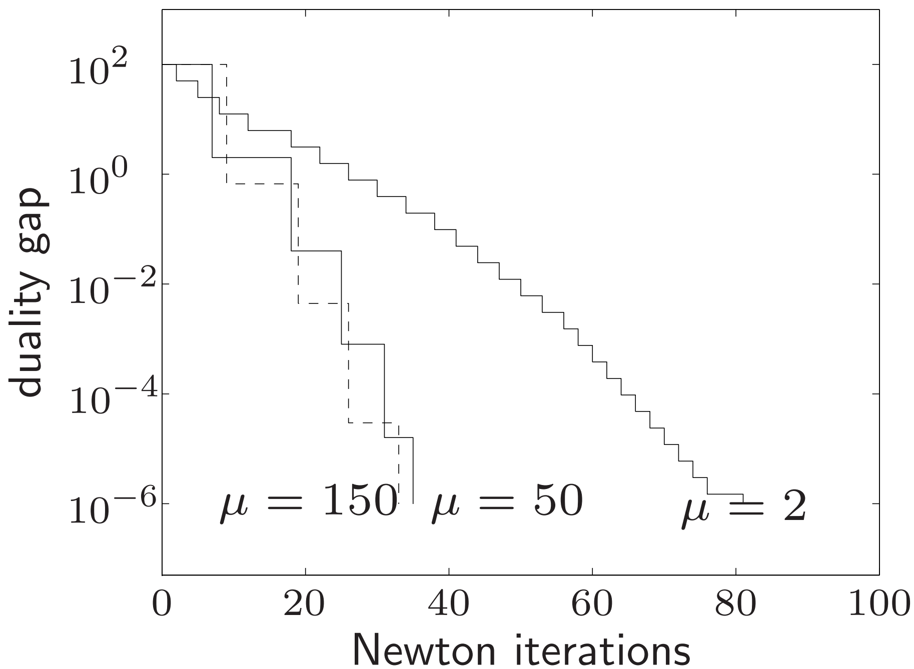
$$\phi(x) = -\log\det(B - x_1 A_1 - \cdots - x_n A_n)$$

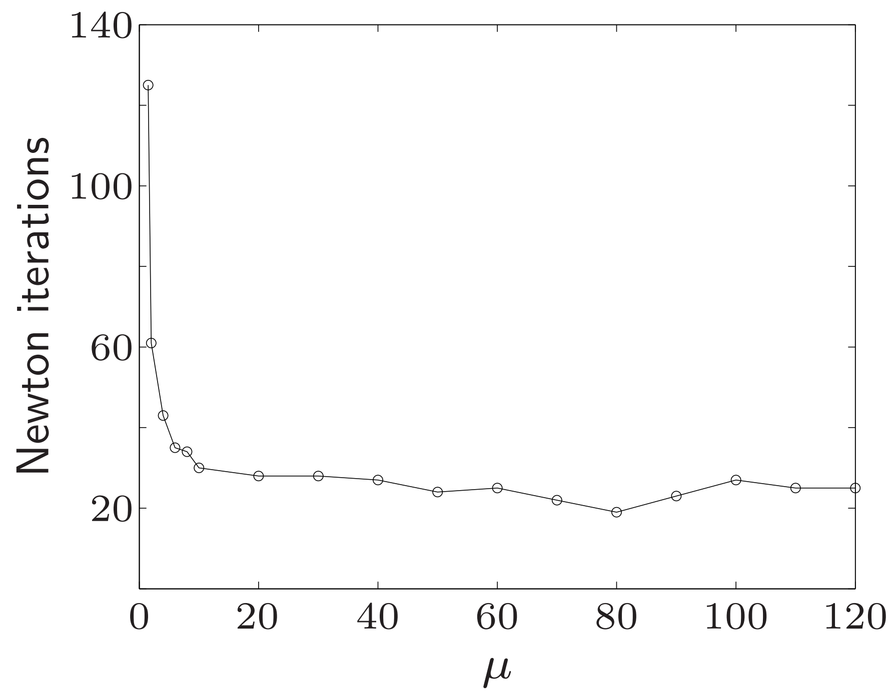
- a convex function
- $\log\det X$ is 'logarithm' for p.s.d. cone

**Barrier method:** follows central path $x^\star(t) = \operatorname{argmin}(tf^T x + \phi(x))$

# Example

$100$ variables, one linear matrix inequality in $\mathbf{S}^{100}$



# Complexity of barrier method

**Iteration complexity**

- can be bounded by polynomial function of problem dimensions (with correct formulation, barrier function)
- examples: $O(\sqrt{m})$ iteration bound for LP or SOCP with $m$ inequalities, SDP with constraint of order $m$
- proofs rely on theory of Newton's method for self-concordant functions
- in practice: #iterations roughly constant as a function of problem size

**Linear algebra complexity**

dominated by solution of Newton system

# Primal-dual interior-point methods

**Similarities with barrier method**

- follow the same central path
- linear algebra (KKT system) per iteration is similar

**Differences**

- faster and more robust
- update primal and dual variables in each step
- no distinction between inner (centering) and outer iterations
- include heuristics for adaptive choice of barrier parameter $t$
- can start at infeasible points
- often exhibit superlinear asymptotic convergence

# 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**

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 + \mathbf{1}^T u \\ \text{subject to} & -u \preceq x \preceq u \end{array}$$

- coefficient of Newton system in interior-point method is

$$\left[\begin{array}{cc} A^T A & 0 \\ 0 & 0 \end{array}\right] + \left[\begin{array}{cc} D_1 + D_2 & D_2 - D_1 \\ D_2 - D_1 & D_1 + D_2 \end{array}\right] \qquad (D_1,\ D_2 \text{ positive diagonal})$$

- very 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 $O(m^2 n)$

**Comparison** (seconds on 3.2Ghz machine)

| $m$ | $n$ | custom | general-purpose |
|---|---|---|---|
| 50 | 100 | 0.02 | 0.05 |
| 50 | 200 | 0.03 | 0.17 |
| 100 | 1000 | 0.32 | 10.6 |
| 100 | 2000 | 0.71 | 76.9 |
| 500 | 1000 | 2.5 | 11.2 |
| 500 | 2000 | 5.5 | 79.8 |

general-purpose solver is MOSEK

# First-order methods

- gradient method
- Nesterov's gradient methods
- extensions

# Gradient method

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

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

$t_k$ is step size (fixed or determined by backtracking line search)

## Classical convergence result

- assume $\nabla f$ Lipschitz continuous ($\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2$)
- error decreases as $1/k$, hence

$$O\left(\frac{1}{\epsilon}\right) \text{ iterations}$$

needed to reach accuracy $f(x^{(k)}) - f^\star \leq \epsilon$

# Nesterov's gradient method

choose $x^{(0)}$; take $x^{(1)} = x^{(0)} - t_1 \nabla f(x^{(0)})$ and for $k \geq 2$

$$\begin{aligned} y^{(k)} &= x^{(k-1)} + \frac{k-2}{k+1}(x^{(k-1)} - x^{(k-2)}) \\ x^{(k)} &= y^{(k)} - t_k \nabla f(y^{(k)}) \end{aligned}$$

- gradient method with 'extrapolation'
- if $f$ has Lipschitz continuous gradient, error decreases as $1/k^2$; hence

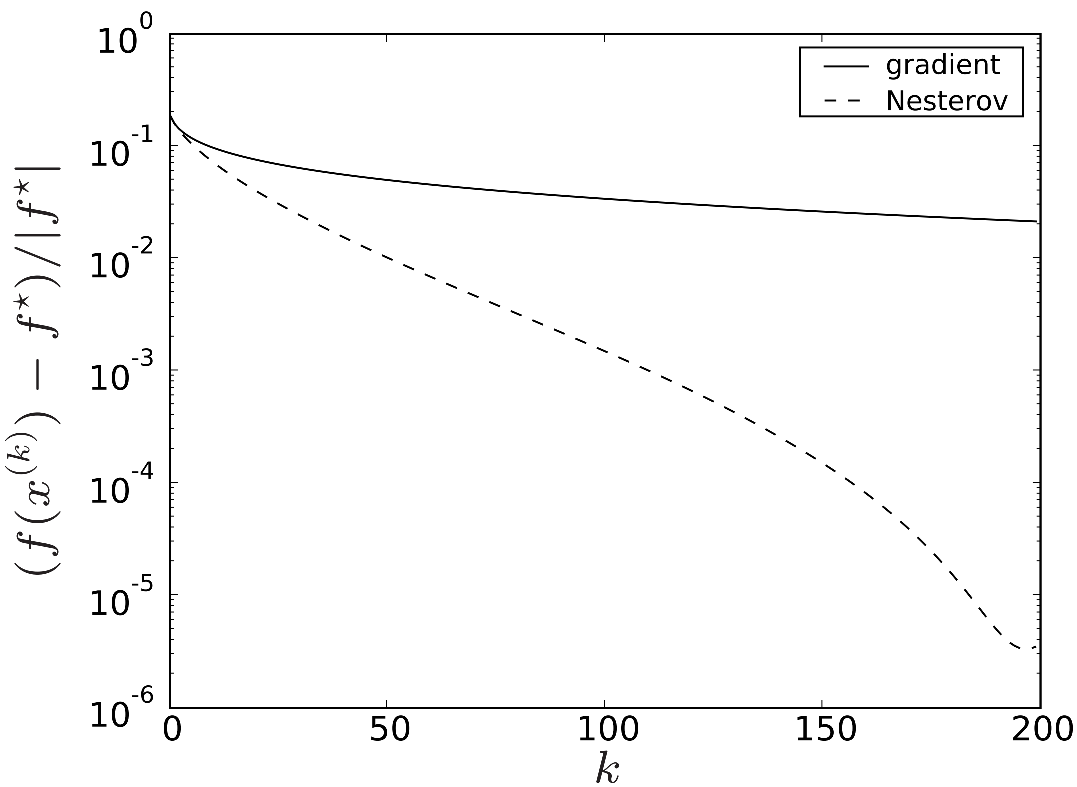
  $$O\left(\frac{1}{\sqrt{\epsilon}}\right) \quad \text{iterations}$$

  needed to reach accuracy $f(x^{(k)}) - f^\star \leq \epsilon$
- many variations; first one published in 1983

# Example

$$\text{minimize} \quad \log \sum_{i=1}^{m} \exp(a_i^T x + b_i)$$

randomly generated data with $m = 2000$, $n = 1000$, fixed step size



# Interpretation of gradient update

$$
\begin{aligned}
x^{(k)} &= x^{(k-1)} - t_k \nabla f(x^{(k-1)}) \\
&= \operatorname*{argmin}_z \left( \nabla f(x^{(k-1)})^T z + \frac{1}{t_k} \|z - x^{(k-1)}\|_2^2 \right)
\end{aligned}
$$

## Interpretation

$x^{(k)}$ minimizes

$$
f(x^{(k-1)}) + \nabla f(x^{(k-1)})^T (z - x^{(k-1)}) + \frac{1}{t_k} \|z - x^{(k-1)}\|_2^2
$$

a simple quadratic model of $f$ at $x^{(k-1)}$

# Projected gradient method

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in C \end{array}$$

$f$ convex, $C$ a closed convex set

$$\begin{aligned} x^{(k)} &= \operatorname*{argmin}_{z \in C} \left( \nabla f(x^{(k-1)})^T z + \frac{1}{t_k} \|z - x^{(k-1)}\|_2^2 \right) \\ &= P_C\left( x^{(k-1)} - t_k \nabla f(x^{(k-1)}) \right) \end{aligned}$$

- useful if projection $P_C$ on $C$ is inexpensive ($e.g.$, box constraints)
- similar convergence result as for basic gradient algorithm
- can be used in fast Nesterov-type gradient methods

# Nonsmooth components

$$\text{minimize} \quad f(x) + g(x)$$

$f$, $g$ convex, with $f$ differentiable, $g$ nondifferentiable

$$\begin{aligned}
x^{(k)} &= \operatorname*{argmin}_z \left( \nabla f(x^{(k-1)})^T z + g(x) + \frac{1}{t_k} \|z - x^{(k-1)}\|_2^2 \right) \\
&= \operatorname*{argmin}_z \left( \frac{1}{2t_k} \left\| z - x^{(k-1)} + t_k \nabla f(x^{(k-1)}) \right\|_2^2 + g(z) \right) \\
&\triangleq S_{t_k}\left( x^{(k-1)} - t_k \nabla f(x^{(k-1)}) \right)
\end{aligned}$$

- gradient step for $f$ followed by 'thresholding' operation $S_t$
- useful if thresholding is inexpensive ($e.g.$, because $g$ is separable)
- similar convergence result as basic gradient method

# Example: $\ell_1$-norm regularization

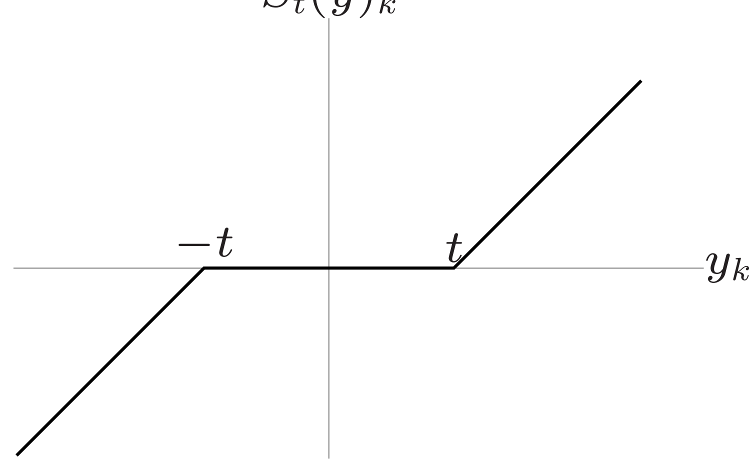
$$\text{minimize} \quad f(x) + \|x\|_1$$

$f$ convex and differentiable

**Thresholding operator**

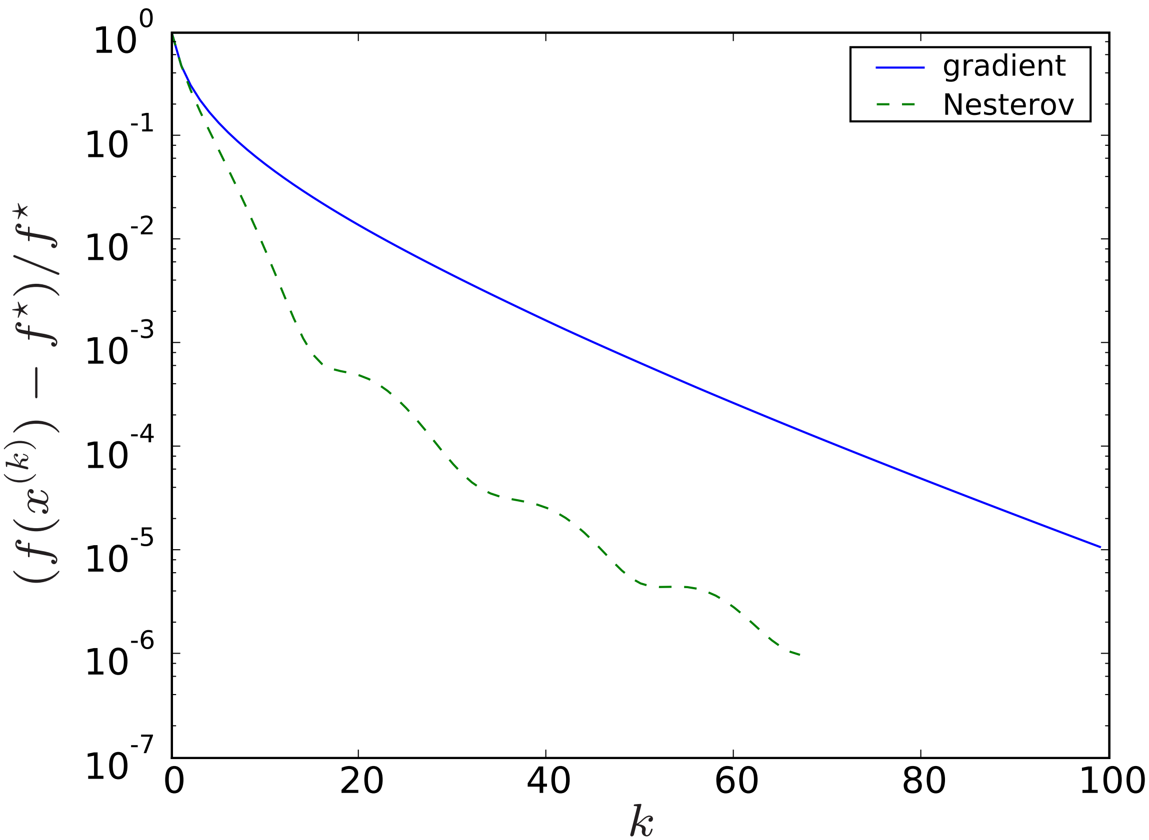
$$S_t(y) = \operatorname*{argmin}_z \left( \frac{1}{2t} \|z - y\|_2^2 + \|z\|_1 \right)$$

$$S_t(y)_k = \begin{cases} y_k - t & y_k \geq t \\ 0 & -t \leq y_k \leq t \\ y_k + t & y_k \leq -t \end{cases}$$



# $\ell_1$-Norm regularized least-squares

$$\text{minimize} \quad \frac{1}{2}\|Ax - b\|_2^2 + \|x\|_1$$




randomly generated $A \in \mathbf{R}^{2000 \times 1000}$; fixed step

# Summary: Advances in convex optimization

**Theory**

new problem classes, robust optimization, convex relaxations, . . .

**Applications**

new applications in different fields; surprisingly many discovered recently

**Algorithms and software**

- high-quality general-purpose implementations of interior-point methods
- software packages for convex modeling
- new first-order methods