Convex Optimization

Lieven Vandenberghe

Electrical Engineering Department, UCLA

Joint work with **Stephen Boyd**, Stanford University

Ph.D. School in Optimization in Computer Vision

DTU, May 19, 2008

Introduction

Mathematical optimization

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

- $x = (x_1, \ldots, x_n)$: optimization variables
- $f_0 : \mathbf{R}^n \to \mathbf{R}$: objective function
- $f_i : \mathbf{R}^n \to \mathbf{R}$, $i = 1, \ldots, m$: constraint functions

Solving optimization problems

General optimization problem

- can be extremely difficult
- methods involve compromise: long computation time or local optimality

Exceptions: certain problem classes can be solved efficiently and reliably

- linear least-squares problems
- linear programming problems
- convex optimization problems

Least-squares

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

- analytical solution: $x^\star = (A^TA)^{-1}A^Tb$
- reliable and efficient algorithms and software
- computation time proportional to n^2p (for $A \in \mathbf{R}^{p\times n}$); less if structured
- a widely used technology

Using least-squares

- least-squares problems are easy to recognize
- standard techniques increase flexibility (weights, regularization, . . .)

Linear programming

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

- no analytical formula for solution; extensive theory
- reliable and efficient algorithms and software
- computation time proportional to $n^2 m$ if $m \geq n$; less with structure
- a widely used technology

Using linear programming

- not as easy to recognize as least-squares problems
- a few standard tricks used to convert problems into linear programs ($e.g.$, problems involving ℓ_1- or ℓ_∞-norms, piecewise-linear functions)

Convex optimization problem

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

- objective and constraint functions are convex:

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

  for all x, y, $0 \leq \theta \leq 1$

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

Solving convex optimization problems

- no analytical solution
- reliable and efficient algorithms
- computation time (roughly) proportional to $\max\{n^3, n^2m, F\}$, where F is cost of evaluating f_i's and their first and second derivatives
- almost a technology

Using convex optimization

- often difficult to recognize
- many tricks for transforming problems into convex form
- 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 pogramming relaxations in combinatorial optimization
- applications in structural optimization, statistics, signal processing, communications, image processing, quantum information theory, finance, . . .

Interior-point methods

Linear programming

- 1984 (Karmarkar): first practical polynomial-time algorithm
- 1984-1990: efficient implementations for large-scale LPs

Nonlinear convex optimization

- around 1990 (Nesterov & Nemirovski): polynomial-time interior-point methods for nonlinear convex programming
- since 1990: extensions and high-quality software packages

Traditional and new view of convex optimization

Traditional: special case of nonlinear programming with interesting theory

New: extension of LP, as tractable but substantially more general

reflected in notation: ‘cone programming’

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

‘$\preceq$’ is inequality with respect to non-polyhedral convex cone

Outline

- Convex sets and functions
- Modeling systems
- Cone programming
- Robust optimization
- Semidefinite relaxations
- ℓ_1-norm sparsity heuristics
- Interior-point algorithms

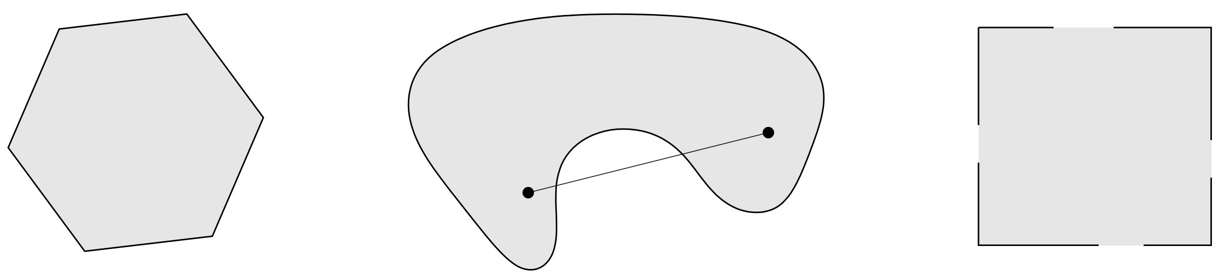
Convex Sets and Functions

Convex sets

Contains line segment between any two points in the set

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

example: one convex, two nonconvex sets:



Examples and properties

- solution set of linear equations
- solution set of linear inequalities
- norm balls $\{x \mid \|x\| \leq R\}$ and norm cones $\{(x,t) \mid \|x\| \leq t\}$
- set of positive semidefinite matrices

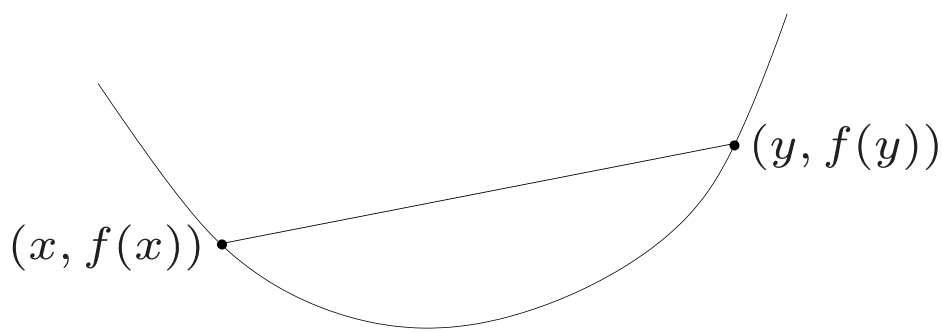
- image of a convex set under a linear transformation is convex
- inverse image of a convex set under a linear transformation is convex
- intersection of convex sets is convex

Convex functions

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

Examples

- $\exp x$, $-\log x$, $x \log x$ are convex
- x^α 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

Operations that preserve convexity

Pointwise maximum

if $f(x, y)$ is convex in x for fixed y, then

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

is convex in x

Composition rules

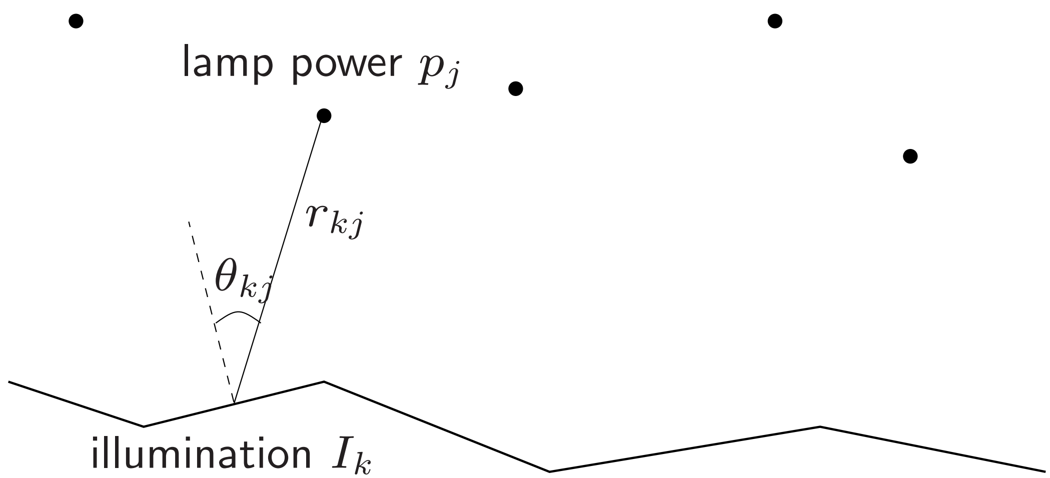
if h is convex and increasing and g is convex, then $h(g(x))$ is convex

Perspective

if $f(x)$ is convex then $tf(x/t)$ is convex in x, t for $t > 0$

Example

m lamps illuminating n (small, flat) patches



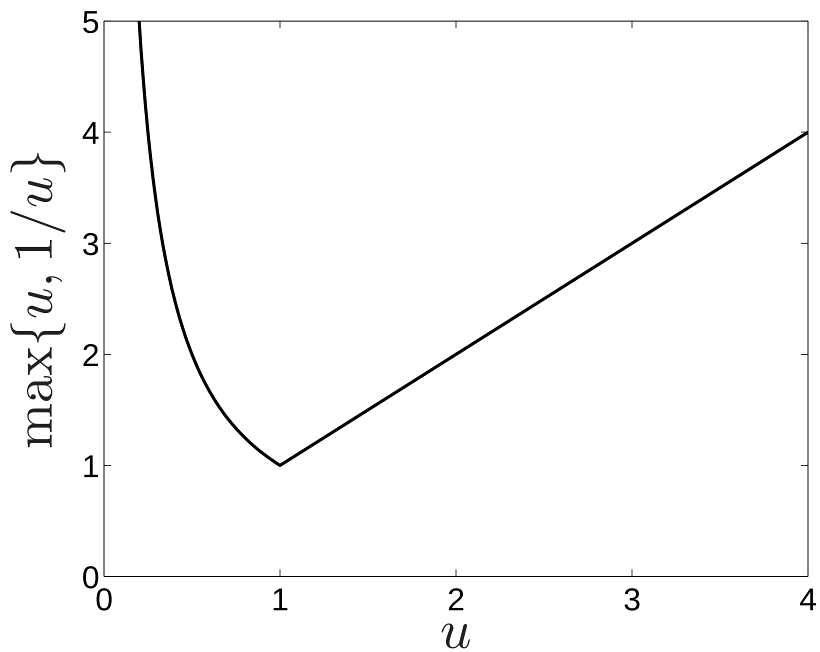
intensity I_k at patch k depends linearly on lamp powers p_j: $I_k = a_k^T p$

Problem: achieve desired illumination $I_k \approx 1$ with bounded lamp powers

$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} \left|\log(a_k^T p)\right| \\ \text{subject to} & 0 \leq p_j \leq p_{\mathsf{max}}, \quad j = 1, \ldots, m \end{array}$$

Convex formulation: problem is equivalent to

$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} \max\{a_k^T p, 1/a_k^T p\} \\ \text{subject to} & 0 \leq p_j \leq p_{\mathsf{max}}, \quad j = 1, \ldots, m \end{array}$$



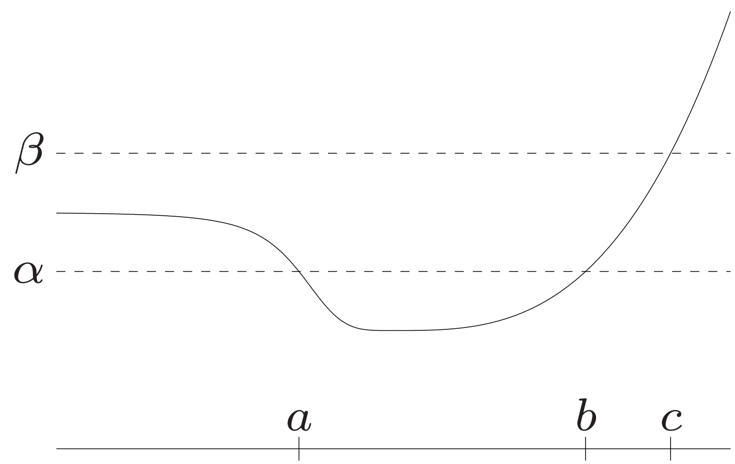
cost function is convex because maximum of convex functions is convex

Quasiconvex functions

domain $\mathbf{dom}\, f$ is convex and the sublevel sets

$$S_\alpha = \{x \in \mathbf{dom}\, f \mid f(x) \leq \alpha\}$$

are convex for all α



f is quasiconcave if $-f$ is quasiconvex

Examples

- $\sqrt{|x|}$ is quasiconvex on $\mathbf{R}$
- $\text{ceil}(x) = \inf\{z \in \mathbf{Z} \mid z \geq x\}$ is quasiconvex and quasiconcave
- $\log x$ is quasiconvex and quasiconcave on $\mathbf{R}_{++}$
- $f(x_1, x_2) = x_1 x_2$ is quasiconcave on $\mathbf{R}^2_{++}$
- linear-fractional function

$$f(x) = \frac{a^T x + b}{c^T x + d}, \qquad \mathbf{dom}\, f = \{x \mid c^T x + d > 0\}$$

  is quasiconvex and quasiconcave

- distance ratio

$$f(x) = \frac{\|x - a\|_2}{\|x - b\|_2}, \qquad \mathbf{dom}\, f = \{x \mid \|x - a\|_2 \leq \|x - b\|_2\}$$

  is quasiconvex

Quasiconvex optimization

Example

$$\begin{array}{ll} \text{minimize} & p(x)/q(x) \\ \text{subject to} & Ax \preceq b \end{array}$$

p convex, q concave, and $p(x) \geq 0$, $q(x) > 0$

Equivalent formulation (variables x, t)

$$\begin{array}{ll} \text{minimize} & t \\ \text{subjec to} & p(x) - tq(x) \leq 0 \\ & Ax \preceq b \end{array}$$

- for fixed t, constraint is a convex feasibility problem
- can determine optimal t via bisection

Modeling Systems

Convex optimization modeling systems

- allow simple specification of convex problems in natural form
  - declare optimization variables
  - form affine, convex, concave expressions
  - specify objective and constraints
- automatically transform problem to canonical form, call solver, transform back
- built using object-oriented methods and/or compiler-compilers

Example

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

variable $x \in \mathbf{R}^n$; parameters a_i, b_i, $w_i > 0$ are given

Specification in CVX (Grant, Boyd & Ye)

```
cvx_begin
    variable x(n)
    minimize ( -w' * log(b-A*x) )
cvx_end
```

Example

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2 + \lambda \|x\|_1 \\ \text{subject to} & Fx \preceq g + (\sum_{i=1} x_i) h \end{array}$$

variable $x \in \mathbf{R}^n$; parameters A, b, F, g, h given

CVX specification

```
cvx_begin
    variable x(n)
    minimize ( norm(A*x-b,2) + lambda*norm(x,1) )
    subject to
        F*x <= g + sum(x)*h
cvx_end
```

Illumination problem

$$\begin{array}{ll} \text{minimize} & \max_{k=1,\ldots,n} \max\{a_k^T x, 1/a_k^T x\} \\ \text{subject to} & 0 \preceq x \preceq \mathbf{1} \end{array}$$

variable $x \in \mathbf{R}^m$; parameters a_k given (and nonnegative)

CVX specification

```
cvx_begin
    variable x(m)
    minimize ( max( [ A*x; inv_pos(A*x) ] )
    subject to
        x >= 0
        x <= 1
cvx_end
```

History

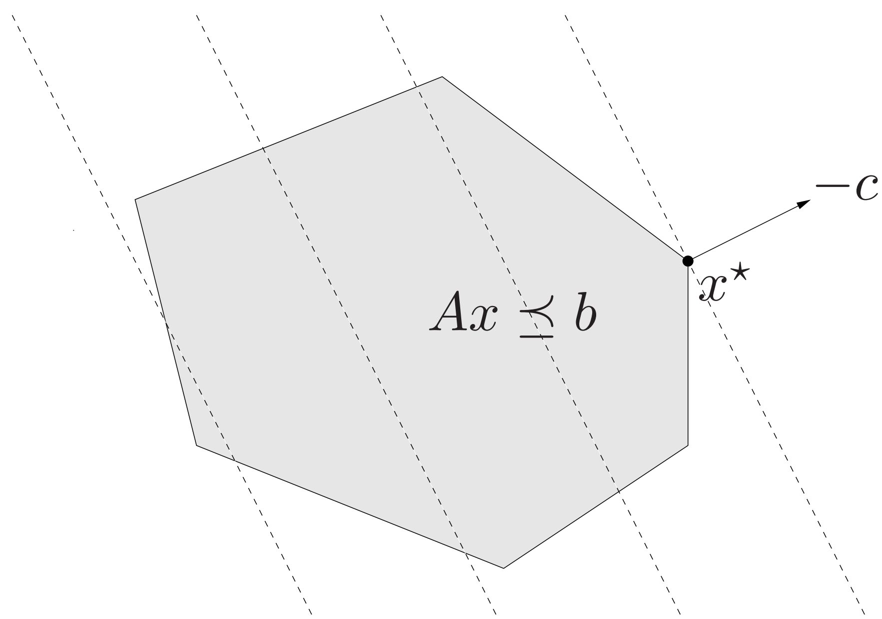
- general purpose optimization modeling systems AMPL, GAMS (1970s)
- systems for SDPs/LMIs (1990s): SDPSOL (Wu, Boyd), LMILAB (Gahinet, Nemirovski), LMITOOL (El Ghaoui)
- YALMIP (Löfberg 2000)
- automated convexity checking (Crusius PhD thesis 2002)
- disciplined convex programming (DCP) (Grant, Boyd, Ye 2004)
- CVX (Grant, Boyd, Ye 2005)
- CVXOPT (Dahl, Vandenberghe 2005)
- GGPLAB (Mutapcic, Koh, et al 2006)
- CVXMOD (Mattingley 2007)

Cone Programming

Linear programming

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

'$\preceq$' is elementwise inequality between vectors

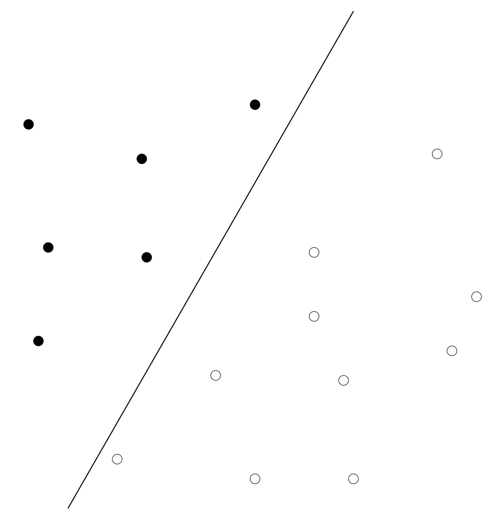


Linear discrimination

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

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

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

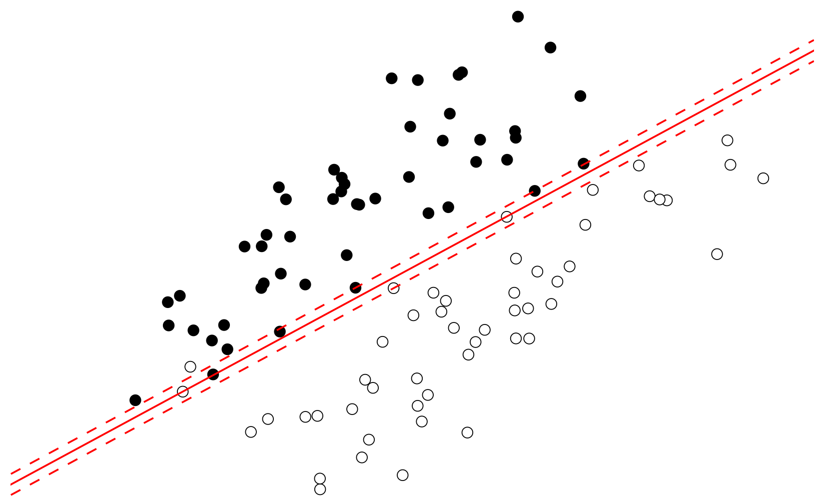


homogeneous in a, b, hence equivalent to the linear inequalities (in a, b)

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

Approximate linear separation of non-separable sets

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



can be interpreted as a heuristic for minimizing #misclassified points

Linear programming formulation

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

Equivalent LP

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^N u_i + \sum_{i=1}^M v_i \\ \text{minimize} & u_i \geq 1 - a^T x_i - b, \quad i = 1, \ldots, N \\ & v_i \geq 1 + a^T y_i + b, \quad i = 1, \ldots, M \\ & u \succeq 0, \quad v \succeq 0 \end{array}$$

variables a, b, $u \in \mathbf{R}^N$, $v \in \mathbf{R}^M$

Cone programming

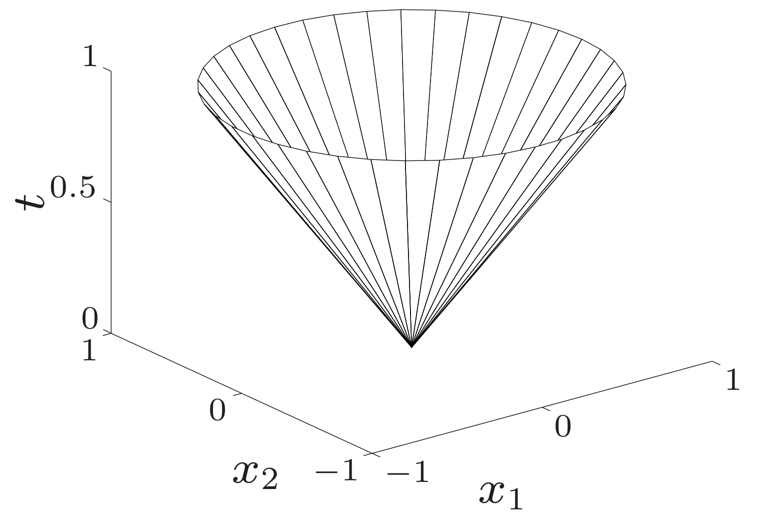
$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq_K b \end{array}$$

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

Second-order cone programming

Second-order cone

$C_{m+1} = \{(x,t) \in \mathbf{R}^m \times \mathbf{R} \mid \|x\| \leq t\}$



Second-order cone program

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

inequality constraints require $(A_i x + b_i, c_i^T x + d_i) \in C_{m_i+1}$

Linear program with chance constraints

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

a_i is Gaussian with mean $\bar{a}_i$, covariance Σ_i, and $\eta \geq 1/2$

Equivalent SOCP

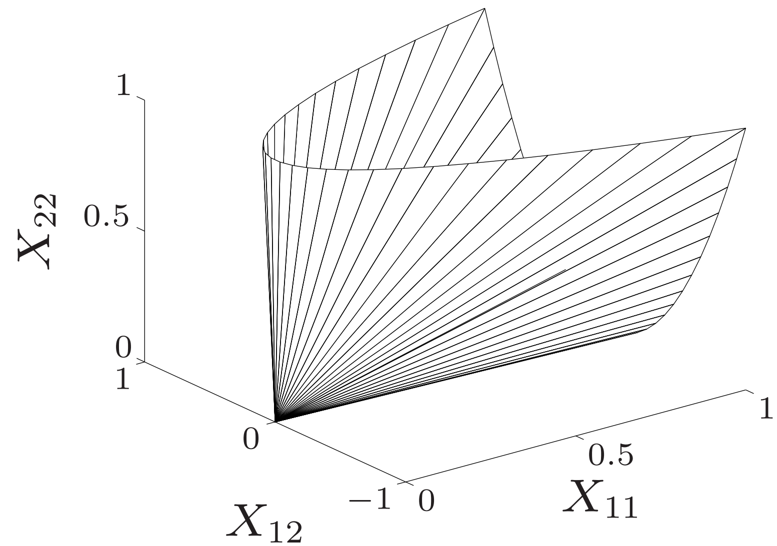
$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \bar{a}_i^T x + \Phi^{-1}(\eta) \|\Sigma_i^{1/2} x\|_2 \leq b_i, \quad i = 1, \ldots, m \end{array}$$

$\Phi(x)$ is zero-mean unit-variance Gaussian CDF

Semidefinite programming

Positive semidefinite cone

$$\mathbf{S}_{+}^{m} = \{X \in \mathbf{S}^{m} \mid X \succeq 0\}$$



Semidefinite programming

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

constraint requires $B - x_1 A_1 - \cdots - x_n A_n \in \mathbf{S}_{+}^{m}$

Eigenvalue minimization

$$\text{minimize} \quad \lambda_{\max}(A(x))$$

where $A(x) = A_0 + x_1 A_1 + \cdots + x_n A_n$ (with given $A_i \in \mathbf{S}^k$)

equivalent SDP

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

- variables $x \in \mathbf{R}^n$, $t \in \mathbf{R}$
- follows from

$$\lambda_{\max}(A) \leq t \quad \Longleftrightarrow \quad A \preceq tI$$

Matrix norm minimization

$$\text{minimize} \quad \|A(x)\|_2 = \left(\lambda_{\max}(A(x)^T A(x))\right)^{1/2}$$

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

equivalent SDP

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

- variables $x \in \mathbf{R}^n$, $t \in \mathbf{R}$
- constraint follows from

$$\|A\|_2 \leq t \quad \Longleftrightarrow \quad A^T A \preceq t^2 I, \; t \geq 0 \quad \Longleftrightarrow \quad \left[\begin{array}{cc} tI & A \\ A^T & tI \end{array}\right] \succeq 0$$

Chebyshev inequalities

Classical inequality: if X is a r.v. with $\mathbf{E}\,X = 0$, $\mathbf{E}\,X^2 = \sigma^2$, then

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

Generalized inequality: sharp lower bounds on $\mathbf{prob}(X \in C)$

- $X \in \mathbf{R}^n$ is a random variable with known moments

$$\mathbf{E}\,X = a, \qquad \mathbf{E}\,XX^T = S$$

- $C \subseteq \mathbf{R}^n$ is defined by quadratic inequalities

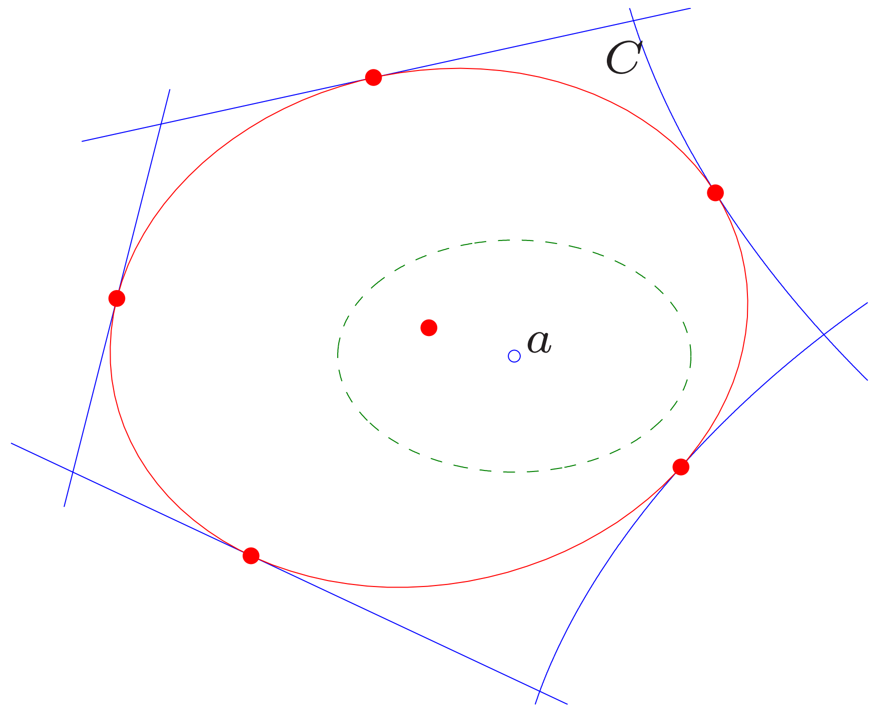
$$C = \{x \mid x^T A_i x + 2b_i^T x + c_i < 0,\ i = 1, \ldots, m\}$$

Equivalent SDP

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

- an SDP with variables $P \in \mathbf{S}^n$, $q \in \mathbf{R}^n$, scalars r, τ_i
- optimal value is tight lower bound on $\mathbf{prob}(X \in C)$
- solution provides distribution that achieves lower bound

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 0.3992 achieved by distribution in red

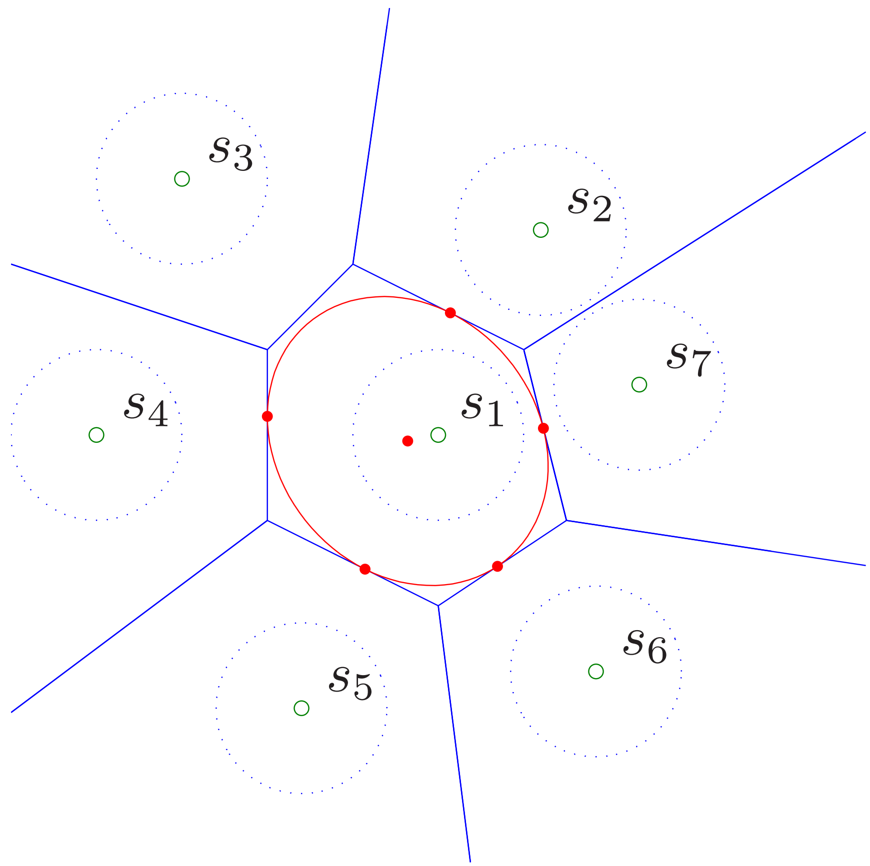
Detection example

$$x = s + v$$

- $x \in \mathbf{R}^n$: received signal
- s: transmitted signal $s \in \{s_1, s_2, \dots, s_N\}$ (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

Duality

Cone program

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

Dual cone program

$$\begin{array}{ll} \text{maximize} & -b^T z \\ \text{subject to} & A^T z + c = 0 \\ & z \succeq_{K^*} 0 \end{array}$$

- K^* is the dual cone: $K^* = \{z \mid z^T x \geq 0 \text{ for all } x \in K\}$
- nonnegative orthant, 2nd order cone, PSD cone are self-dual: $K = K^*$

Properties: optimal values are equal (if primal or dual is strictly feasible)

Robust Optimization

Robust optimization

(worst-case) robust convex optimization problem

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

- x is optimization variable; θ is an unknown parameter
- f_i convex in x for fixed θ
- tractability depends on $\mathcal{A}$

(Ben-Tal, Nemirovski, El Ghaoui, Bertsimas, . . .)

Robust linear programming

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

coefficients unknown but contained in ellipsoids $\mathcal{A}_i$:

$$\mathcal{A}_i = \{\bar{a}_i + P_i u \mid \|u\|_2 \leq 1\} \qquad (\bar{a}_i \in \mathbf{R}^n, \quad P_i \in \mathbf{R}^{n \times n})$$

center is $\bar{a}_i$, semi-axes determined by singular values/vectors of P_i

Equivalent SOCP

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

Robust least-squares

$$\text{minimize} \quad \sup_{\|u\|_2 \leq 1} \|(A_0 + u_1 A_1 + \cdots + u_p A_p)x - b\|_2$$

- coefficient matrix lies in ellipsoid;
- choose x to minimize worst-case residual norm

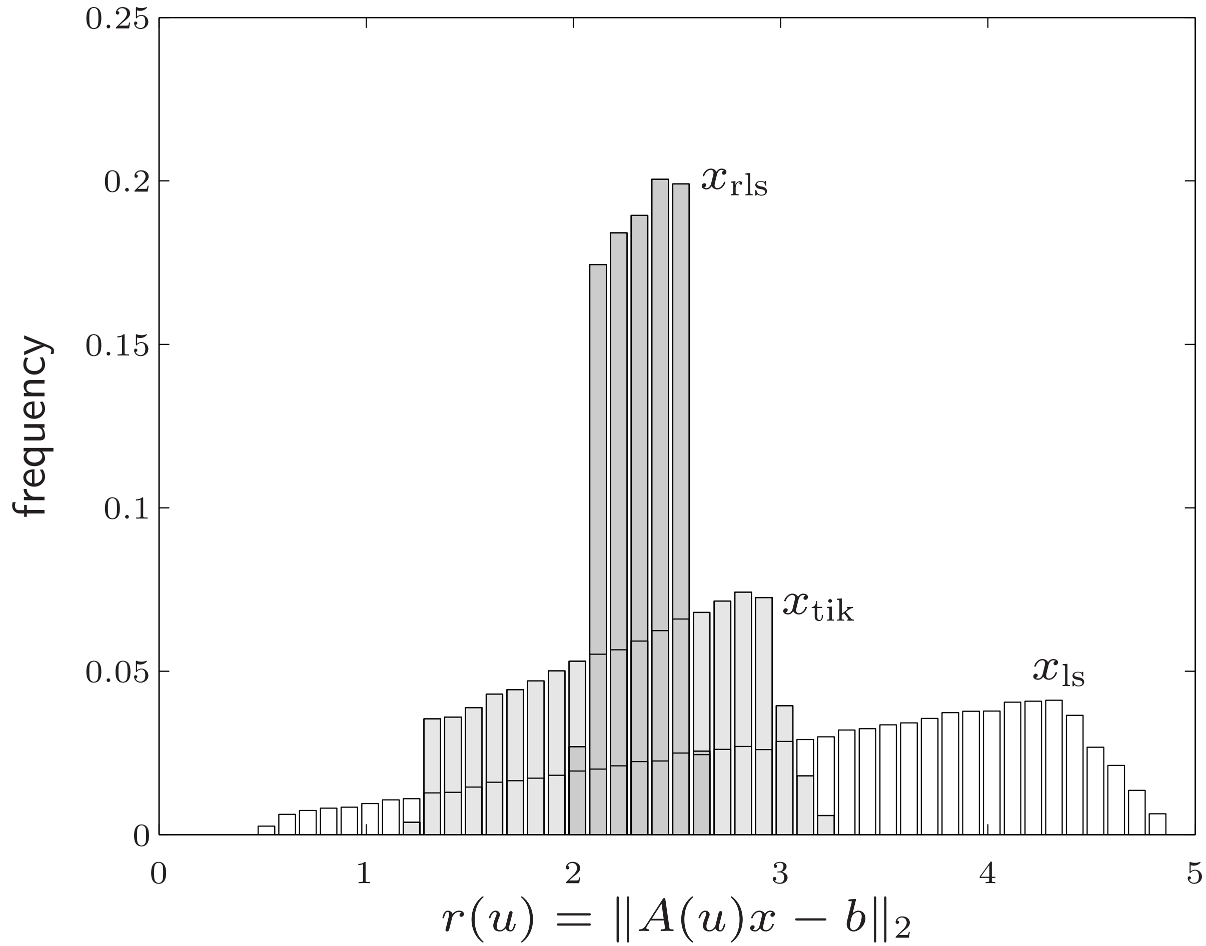
Equivalent SDP

$$\begin{array}{ll} \text{minimize} & t_1 + t_2 \\ \text{subject to} & \begin{bmatrix} I & P(x) & A_0 x - b \\ P(x)^T & t_1 I & 0 \\ (A_0 x - b)^T & 0 & t_2 \end{bmatrix} \succeq 0 \end{array}$$

where

$$P(x) = \begin{bmatrix} A_1 x & A_2 x & \cdots & A_p x \end{bmatrix}$$

Example ($p = 2$, u uniformly distributed in unit disk)



$x_{\rm tik}$ minimizes $\|A_0 x - b\|_2^2 + \|x\|_2^2$

Semidefinite Relaxations

Relaxation and randomization

convex optimization is increasingly used

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

Semidefinite relaxations

Boolean least-squares

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

- a basic problem in digital commununciations
- non-convex, very hard to solve exactly

Equivalent formulation

$$\begin{array}{ll} \text{minimize} & \mathbf{tr}(A^T A Z) - 2b^T A z + b^T b \\ \text{subject to} & Z_{ii} = 1, \quad i = 1, \ldots, n \\ & Z = zz^T \end{array}$$

follows from $\|Az - b\|_2^2 = \mathbf{tr}(A^T A Z) - 2b^T A z + b^T b$ if $Z = zz^T$

Semidefinite relaxation

replace constraint $Z = zz^T$ with $Z \succeq zz^T$

$$
\begin{array}{ll}
\text{minimize} & \mathbf{tr}(A^TAZ) - 2b^TAz + b^Tb \\
\text{subject to} & Z_{ii} = 1, \quad i = 1, \ldots, n \\
& \left[\begin{array}{cc} Z & z \\ z^T & 1 \end{array}\right] \succeq 0
\end{array}
$$

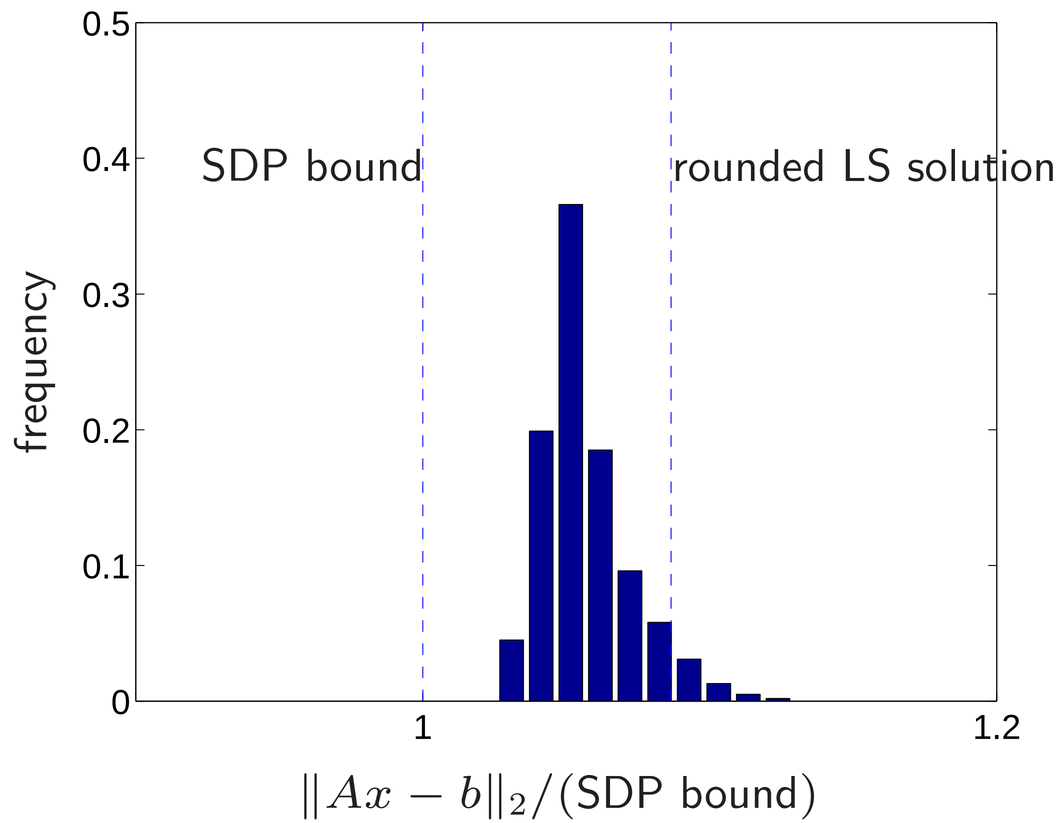
- an SDP with variables Z, z
- optimal value is a lower bound for Boolean LS optimal value
- rounding Z, z gives suboptimal solution for Boolean LS

Randomized rounding

- generate vector from $\mathcal{N}(z, Z - zz^T)$
- round components to ± 1

Example

- (randomly chosen) parameters $A \in \mathbf{R}^{150\times 100}$, $b \in \mathbf{R}^{150}$
- $x \in \mathbf{R}^{100}$, so feasible set has $2^{100} \approx 10^{30}$ points



distribution of randomized solutions based on SDP solution

Sums of squares and semidefinite programming

Sum of squares: a function of the form

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

$q(t)$: vector of basis functions (polynomial, trigonometric, . . .)

SDP parametrization:

$$f(t) = q(t)^T X q(t), \quad X \succeq 0$$

- a **sufficient** condition for nonnegativity of f, useful in nonconvex polynomial optimization (Parrilo, Lasserre, Henrion, De Klerk . . .)
- in some important special cases, **necessary and sufficient**

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

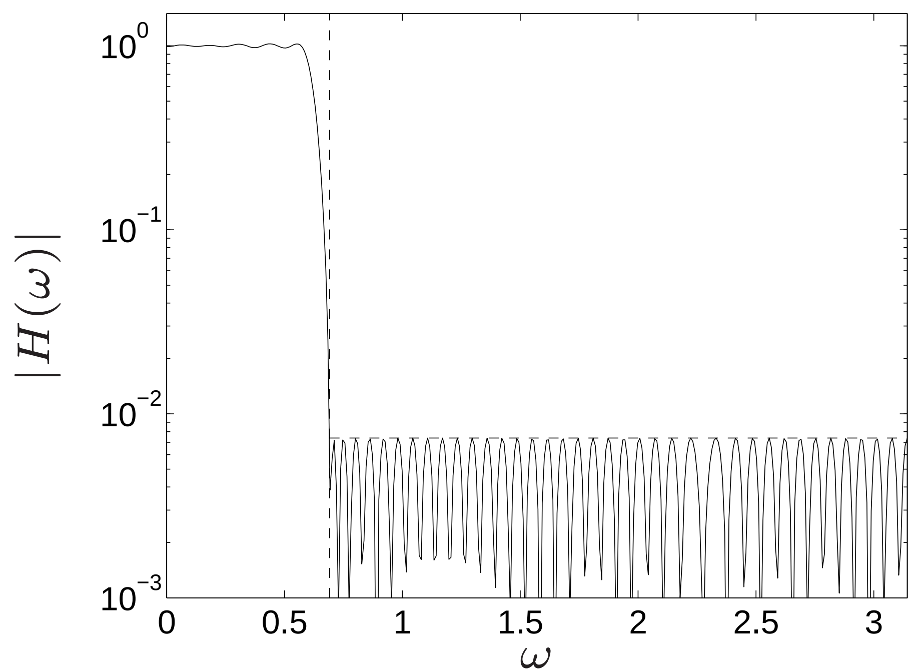
$$x^T p(\omega) = q_1(\omega)^T X_1 q_1(\omega) + s(\omega) q_2(\omega)^T X_2 q_2(\omega), \quad X_1 \succeq 0, \quad X_2 \succeq 0$$

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

$$\text{minimize} \quad \sup_{\omega \geq \omega_\mathrm{s}} |h_0 + h_1 \cos\omega + \cdots + h_n \cos n\omega|$$

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



(Example with $n = 50$, $M = 5$, $\omega_\mathrm{s} = 0.69$)

SDP formulation

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

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 $(i \neq kM)$, 4 matrices X_i of size roughly n

Multivariate trigonometric sums of squares

$$h(\omega) = \sum_{\mathbf{k}=-\mathbf{n}}^{\mathbf{n}} x_{\mathbf{k}} e^{-j\mathbf{k}^T\omega} = \sum_i |g_i(\omega)|^2, \qquad (x_{\mathbf{k}} = x_{-\mathbf{k}}, \ \ \omega \in \mathbf{R}^d)$$

- g_i is a polynomial in $e^{-j\mathbf{k}^T\omega}$; can have degree higher than $\mathbf{n}$
- necessary for positivity of R
- restricting the degrees of g_i gives a sufficient condition for nonnegativity

Spectral mask constraints defined by trigonometric polynomials d_i

$$h(\omega) = s_0(\omega) + \sum_i d_i(\omega) s_i(\omega), \qquad s_i \text{ is s.o.s.}$$

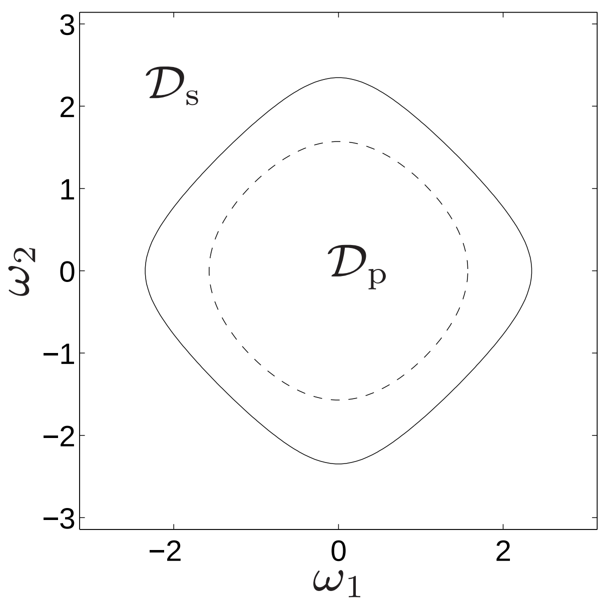
guarantees $h(\omega) \geq 0$ on $\{\omega \mid d_i(\omega \geq 0\}$

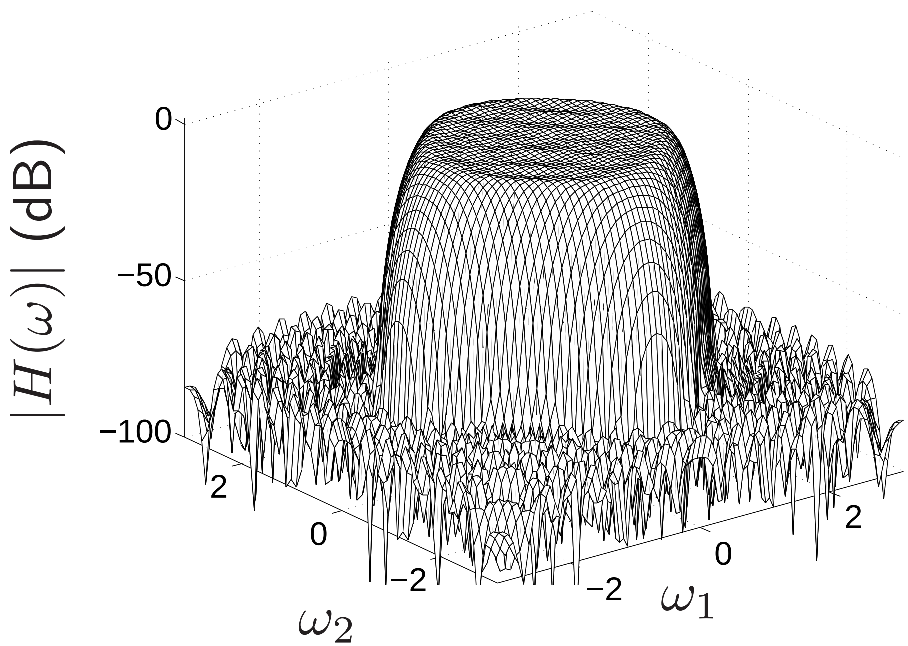
(B. Dumitrescu)

Two-dimensional FIR filter design

$$
\begin{array}{ll}
\text{minimize} & \delta_s \\
\text{subject to} & |1 - H(\omega)| \le \delta_p, \quad \omega \in \mathcal{D}_p \\
& |H(\omega)| \le \delta_s, \quad \omega \in \mathcal{D}_s,
\end{array}
$$

where $H(\omega) = \sum_{i=0}^{n} \sum_{k=0}^{n} h_{ik} \cos i\omega_1 \cos k\omega_2$





1-Norm Sparsity Heuristics

1-**Norm heuristics**

use ℓ_1-norm $\|x\|_1$ as convex approximation of the ℓ_0-'norm' $\mathbf{card}(x)$

- sparse regressor selection (Tibshirani, Hastie, . . .)

$$\text{minimize} \quad \|Ax - b\|_2 + \rho\|x\|_1$$

- sparse signal representation (basis pursuit, sparse compression) (Donoho, Candes, Tao, Romberg, . . .)

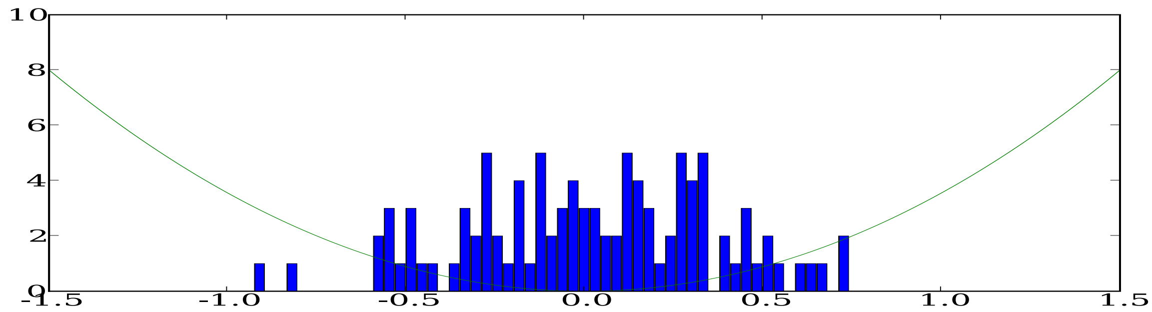
$$\begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & Ax = b \end{array} \qquad \begin{array}{ll} \text{minimize} & \|x\|_1 \\ \text{subject to} & \|Ax - b\|_2 \leq \epsilon \end{array}$$

Norm approximation

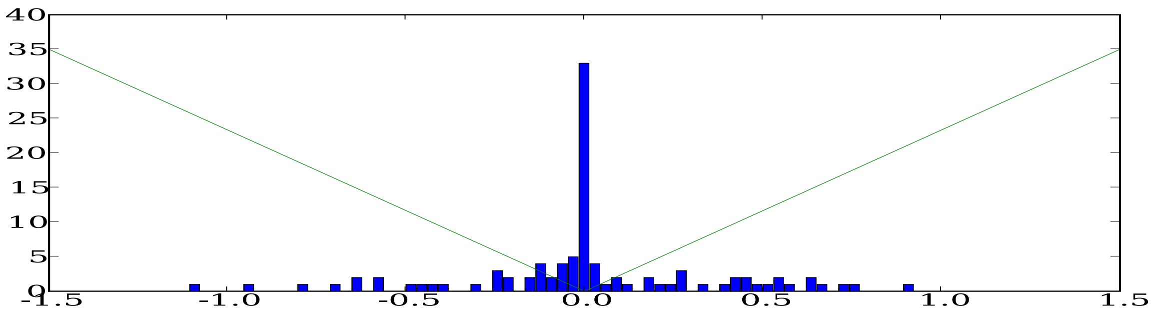
minimize $\|Ax - b\|_2$ minimize $\|Ax - b\|_1$

example (A is 100×30): histograms of residuals

2-norm

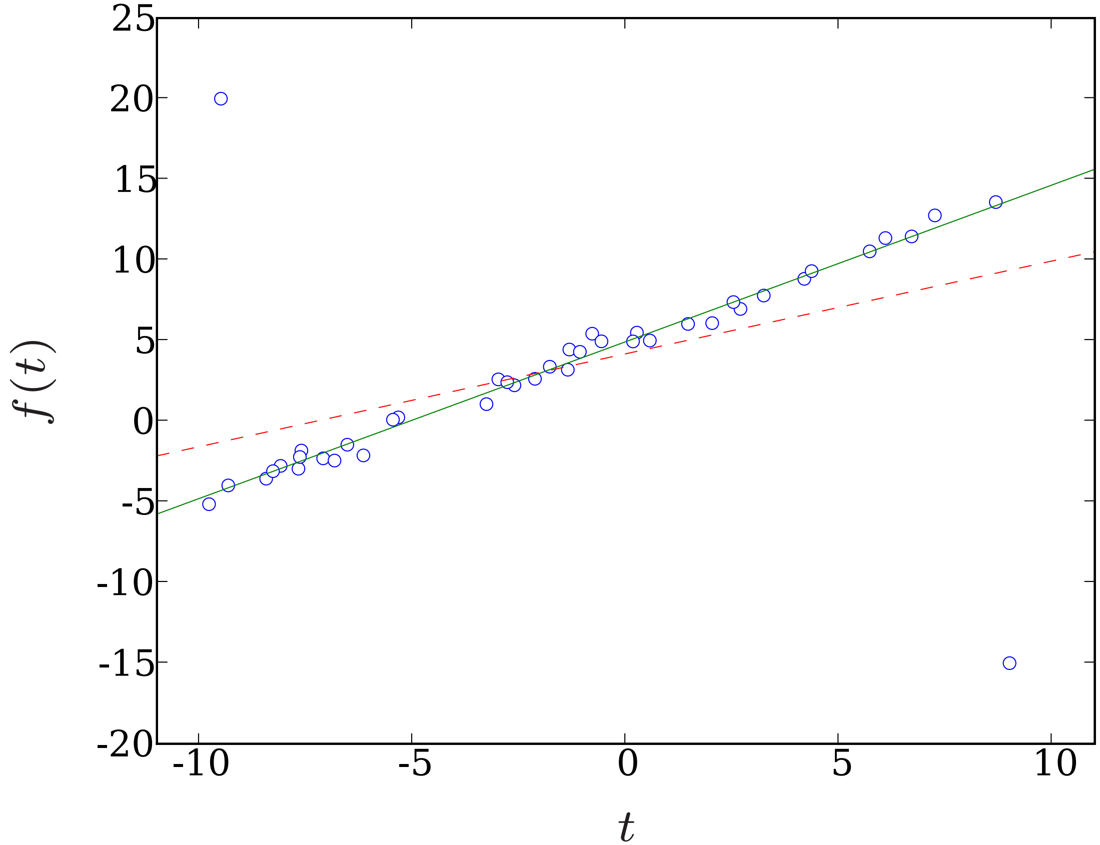


1-norm



note large number of zero residuals in 1-norm solution

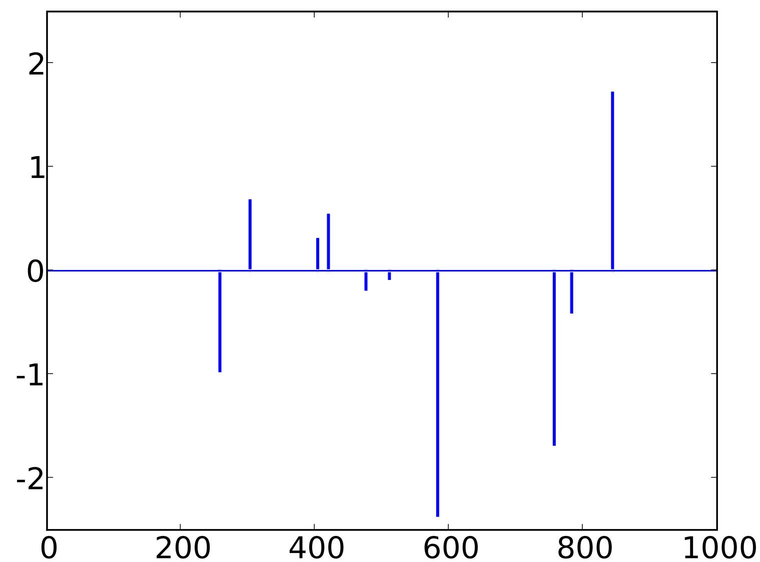
Robust regression



- 42 points t_i, y_i (circles), including two outliers
- function $f(t) = \alpha + \beta t$ fitted using 2-norm (dashed) and 1-norm

Sparse reconstruction

signal $\hat{x} \in \mathbf{R}^n$ with $n = 1000$, 10 nonzero components



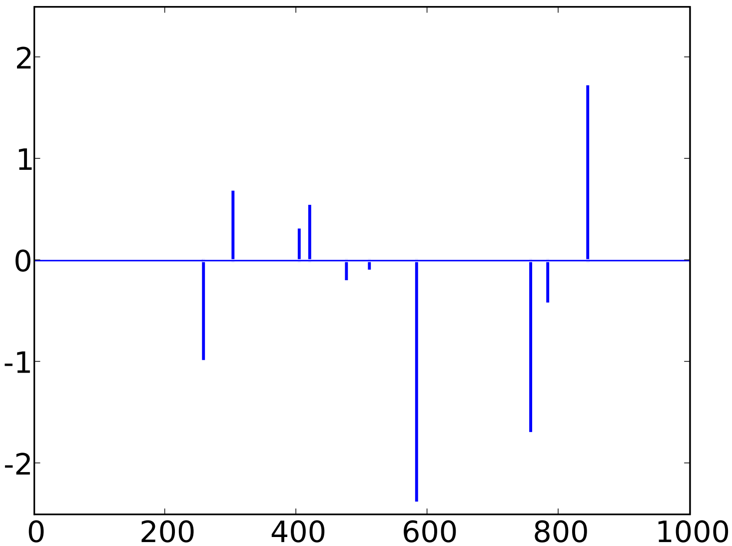
$m = 100$ random noisy measurements

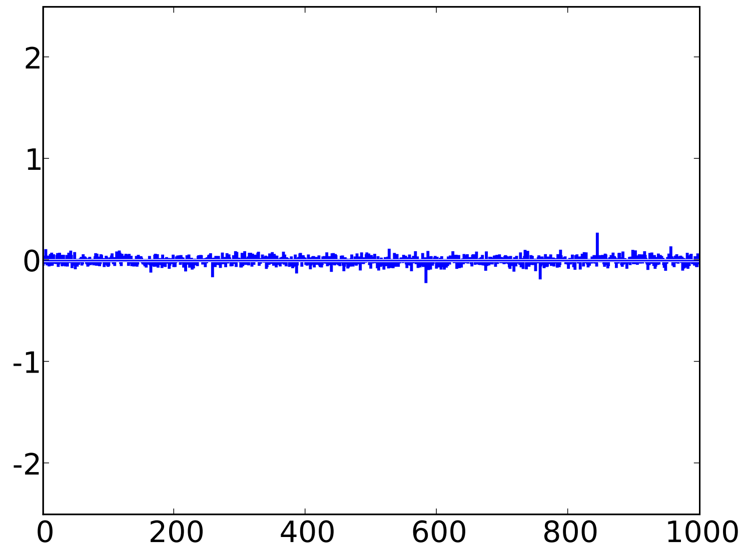
$$b = A\hat{x} + v$$

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

ℓ_2-Norm reconstruction

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

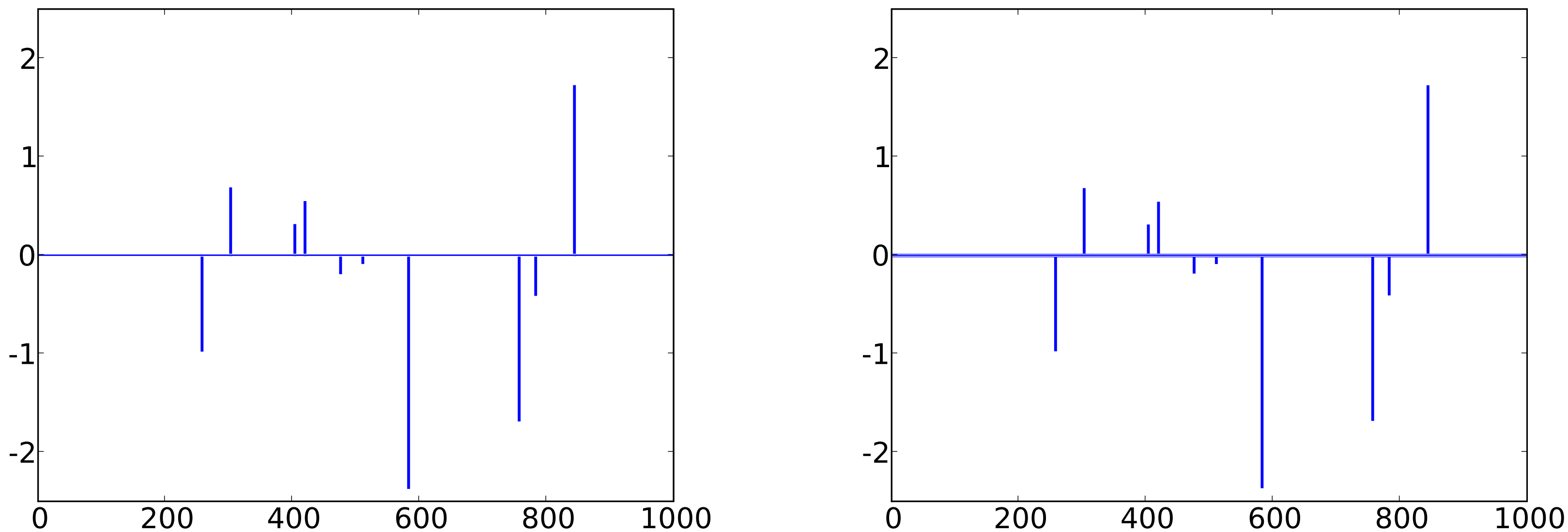




left: exact signal $\hat{x}$; right: ℓ_2 reconstruction

ℓ_1-Norm reconstruction

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



left: exact signal $\hat{x}$; right: ℓ_1 reconstruction

Interior-Point Algorithms

Interior-point algorithms

- handle linear and nonlinear convex problems
- follow **central path** as guide to the solution (using Newton's method)
- worst-case complexity theory: # Newton iterations $\sim \sqrt{\text{problem size}}$
- in practice: # Newton steps between 10 and 50
- performance is similar across wide range of problem dimensions, problem data, problem classes
- controlled by a small number of easily tuned algorithm parameters

Cone program

Primal and dual cone program

minimize	$c^T x$		maximize	$-b^T y$
subject to	$Ax + s = b$		subject to	$A^T z + c = 0$
	$s \succeq_K 0$			$z \succeq_{K^*} 0$

- $s \succeq_K 0$ means $s \in K$ (convex cone)
- $z \succeq_{K^*} 0$ means $z \in K^*$ (dual cone $K^* = \{z \mid s^T z \geq 0 \ \forall s \in K\}$)

Examples (of self-dual cones: $K = K^*$)

- linear program: K is nonnegative orthant
- second order cone program: K is second order cone $\{(t, x) \mid \|x\|_2 \leq t\}$
- semidefinite program: K is cone of positive semidefinite matrices

Central path

solution $\{(x(t), s(t)) \mid t > 0\}$ of

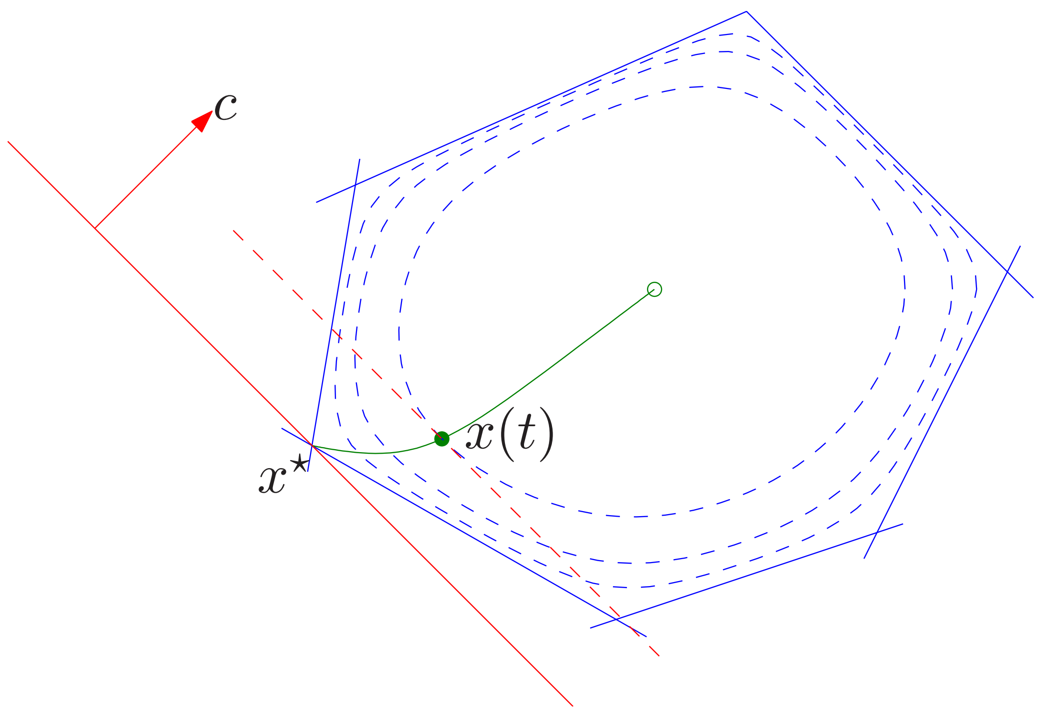
$$\begin{array}{ll} \text{minimize} & tc^T x + \phi(s) \\ \text{subject to} & Ax + s = b \end{array}$$

ϕ is a **logarithmic barrier** for primal cone K

- nonnegative orthant: $\phi(u) = -\sum_k \log u_k$
- second order cone: $\phi(u, v) = -\log(u^2 - v^T v)$
- positive semidefinite cone: $\phi(V) = -\log\det V$

Example: central path for linear program

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



Newton equation

Central path optimality conditions

$$Ax + s = b, \qquad A^T z + c = 0, \qquad z + \frac{1}{t}\nabla\phi(s) = 0$$

Newton equation: linearize optimality conditions

$$\begin{aligned} \left[\begin{array}{c} 0 \\ \Delta s \end{array}\right] + \left[\begin{array}{cc} 0 & A^T \\ A & 0 \end{array}\right]\left[\begin{array}{c} \Delta x \\ \Delta z \end{array}\right] &= \left[\begin{array}{c} -c - A^T z \\ b - Ax - s \end{array}\right] \\ \Delta z + \frac{1}{t}\nabla^2\phi(s)\Delta s &= -z - \frac{1}{t}\nabla\phi(s) \end{aligned}$$

- gives search directions Δx, Δs, Δz
- many variations ($e.g.$, primal-dual symmetric linearizations)

Computational effort per Newton step

- Newton step effort dominated by solving linear equations to find search direction
- equations inherit structure from underlying problem
- equations same as for weighted LS problem of similar size and structure

Conclusion

we can solve a convex problem with about the same effort as solving 30 least-squares problems

Direct methods for exploiting sparsity

- well developed, since late 1970s
- based on (heuristic) variable orderings, sparse factorizations
- standard in general purpose LP, QP, GP, SOCP implementations
- can solve problems with up to 10^5 variables, constraints (depending on sparsity pattern)

Some convex optimization solvers

primal-dual, interior-point, exploit sparsity

- many for LP, QP (GLPK, CPLEX, . . .)
- SeDuMi, SDPT3 (open source; Matlab; LP, SOCP, SDP)
- DSDP, CSDP, SDPA (open source; C; SDP)
- MOSEK (commercial; C with Matlab interface; LP, SOCP, GP, . . .)
- solver.com (commercial; excel interface; LP, SOCP)
- GPCVX (open source; Matlab; GP)
- CVXOPT (open source; Python/C; LP, SOCP, SDP, GP, . . .)

. . . and many others

Problem structure beyond sparsity

- state structure
- Toeplitz, circulant, Hankel; displacement rank
- fast transform (DFT, wavelet, . . .)
- Kronecker, Lyapunov structure
- symmetry

can exploit for efficiency, but not in most generic solvers

Example: 1-norm approximation

$$\text{minimize} \quad \|Ax - b\|_1$$

Equivalent LP

$$\begin{array}{ll} \text{minimize} & \sum_k y_k \\ \text{subject to} & -y \preceq Ax - b \preceq y \end{array}$$

Newton equation (D_1, D_2 positive diagonal)

$$\begin{bmatrix} 0 & 0 & -A^T & A^T \\ 0 & 0 & -I & -I \\ -A & -I & -D_1 & 0 \\ A & -I & 0 & -D_2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z_1 \\ \Delta z_2 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix}$$

- reduces to equation of the form $A^T D A \Delta x = r$
- cost = cost of (weighted) least squares problem

Iterative methods

- conjugate-gradient (and variants like LSQR) exploit general structure
- rely on fast methods to evaluate Ax and $A^T y$, where A is huge
- can terminate early, to get truncated-Newton interior-point method
- can solve huge problems (10^7 variables, constraints), with
  - good preconditioner
  - proper tuning
  - some luck

Solving specific problems

in developing custom solver for specific application, we can

- exploit structure very efficiently
- determine ordering, memory allocation beforehand
- cut corners in algorithm, e.g., terminate early
- use warm start

to get **very fast** solver

opens up possibility of **real-time embedded** convex optimization

Conclusions

Convex optimization

Fundamental theory

recent advances include new problem classes, robust optimization, semidefinite relaxations of nonconvex problems, ℓ_1-norm heuristics . . .

Applications

Recent applications in wide range of areas; many more to be discovered

Algorithms and software

- High-quality general-purpose implementations of interior-point methods
- Customized implementations can be orders of magnitude faster
- Good modeling systems
- With the right software, suitable for embedded applications