

1. Introduction

- mathematical optimization
- least-squares and linear programming
- convex optimization
- example
- course goals and topics
- nonlinear optimization
- brief history of convex optimization

Mathematical optimization

(mathematical) optimization problem

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

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

optimal solution x^* has smallest value of f_0 among all vectors that satisfy the constraints

Examples

portfolio optimization

- variables: amounts invested in different assets
- constraints: budget, max./min. investment per asset, minimum return
- objective: overall risk or return variance

device sizing in electronic circuits

- variables: device widths and lengths
- constraints: manufacturing limits, timing requirements, maximum area
- objective: power consumption

data fitting

- variables: model parameters
- constraints: prior information, parameter limits
- objective: measure of misfit or prediction error

Solving optimization problems

general optimization problem

- very difficult to solve
- methods involve some compromise, *e.g.*, very long computation time, or not always finding the solution

exceptions: certain problem classes can be solved efficiently and reliably

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

Least-squares

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

solving least-squares problems

- analytical solution: $x^* = (A^T A)^{-1} A^T b$
- reliable and efficient algorithms and software
- computation time proportional to $n^2 k$ ($A \in \mathbf{R}^{k \times n}$); less if structured
- a mature technology

using least-squares

- least-squares problems are easy to recognize
- a few standard techniques increase flexibility (*e.g.*, including weights, adding regularization terms)

Linear programming

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

solving linear programs

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

using linear programming

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

Convex optimization problem

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

- objective and constraint functions are convex:

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y)$$

if $\alpha + \beta = 1$, $\alpha \geq 0$, $\beta \geq 0$

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

solving convex optimization problems

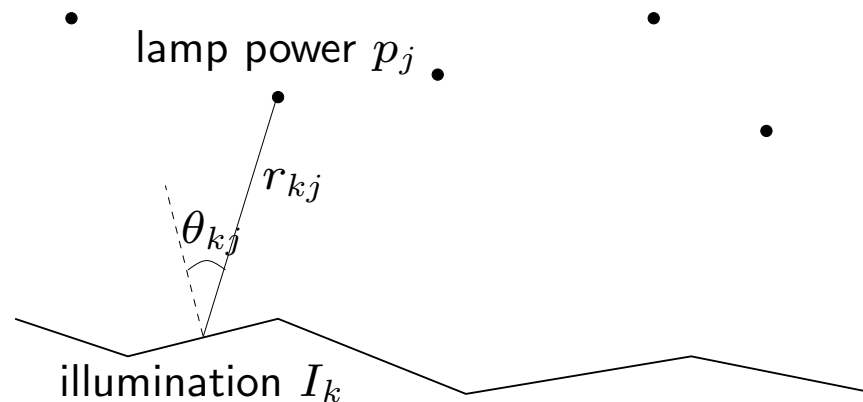
- no analytical solution
- reliable and efficient algorithms
- computation time (roughly) proportional to $\max\{n^3, n^2m, F\}$, where F is cost of evaluating f_i 's and their first and second derivatives
- almost a technology

using convex optimization

- often difficult to recognize
- many tricks for transforming problems into convex form
- surprisingly many problems can be solved via convex optimization

Example

m lamps illuminating n (small, flat) patches



intensity I_k at patch k depends linearly on lamp powers p_j :

$$I_k = \sum_{j=1}^m a_{kj} p_j, \quad a_{kj} = r_{kj}^{-2} \max\{\cos \theta_{kj}, 0\}$$

problem: achieve desired illumination I_{des} with bounded lamp powers

$$\begin{array}{ll} \text{minimize} & \max_{k=1, \dots, n} |\log I_k - \log I_{\text{des}}| \\ \text{subject to} & 0 \leq p_j \leq p_{\text{max}}, \quad j = 1, \dots, m \end{array}$$

how to solve?

1. use uniform power: $p_j = p$, vary p
2. use least-squares:

$$\text{minimize } \sum_{k=1}^n (I_k - I_{\text{des}})^2$$

round p_j if $p_j > p_{\text{max}}$ or $p_j < 0$

3. use weighted least-squares:

$$\text{minimize } \sum_{k=1}^n (I_k - I_{\text{des}})^2 + \sum_{j=1}^m w_j (p_j - p_{\text{max}}/2)^2$$

iteratively adjust weights w_j until $0 \leq p_j \leq p_{\text{max}}$

4. use linear programming:

$$\begin{array}{ll} \text{minimize} & \max_{k=1, \dots, n} |I_k - I_{\text{des}}| \\ \text{subject to} & 0 \leq p_j \leq p_{\text{max}}, \quad j = 1, \dots, m \end{array}$$

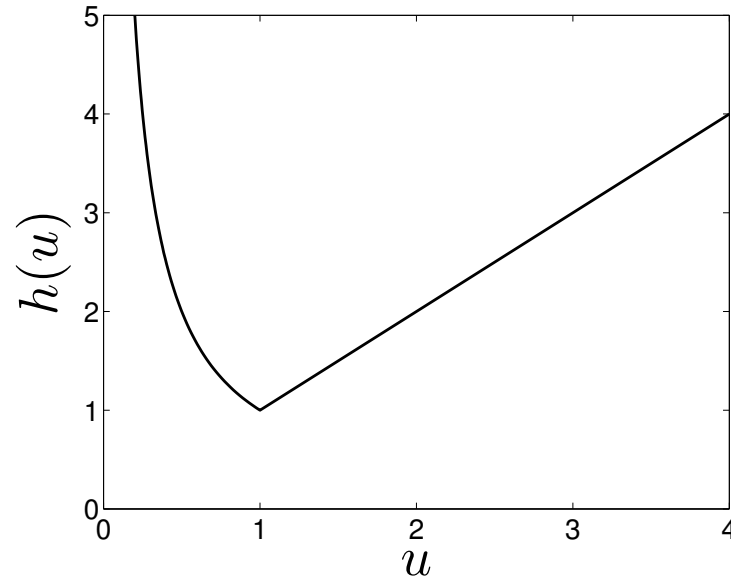
which can be solved via linear programming

of course these are approximate (suboptimal) ‘solutions’

5. use convex optimization: problem is equivalent to

$$\begin{array}{ll} \text{minimize} & f_0(p) = \max_{k=1,\dots,n} h(I_k/I_{\text{des}}) \\ \text{subject to} & 0 \leq p_j \leq p_{\max}, \quad j = 1, \dots, m \end{array}$$

with $h(u) = \max\{u, 1/u\}$



f_0 is convex because maximum of convex functions is convex

exact solution obtained with effort \approx modest factor \times least-squares effort

additional constraints: does adding 1 or 2 below complicate the problem?

1. no more than half of total power is in any 10 lamps

2. no more than half of the lamps are on ($p_j > 0$)

- answer: with (1), still easy to solve; with (2), extremely difficult
- moral: (untrained) intuition doesn't always work; without the proper background very easy problems can appear quite similar to very difficult problems

Course goals and topics

goals

1. recognize/formulate problems (such as the illumination problem) as convex optimization problems
2. develop code for problems of moderate size (1000 lamps, 5000 patches)
3. characterize optimal solution (optimal power distribution), give limits of performance, etc.

topics

1. convex sets, functions, optimization problems
2. examples and applications
3. algorithms

Nonlinear optimization

traditional techniques for general nonconvex problems involve compromises

local optimization methods (nonlinear programming)

- find a point that minimizes f_0 among feasible points near it
- fast, can handle large problems
- require initial guess
- provide no information about distance to (global) optimum

global optimization methods

- find the (global) solution
- worst-case complexity grows exponentially with problem size

these algorithms are often based on solving convex subproblems

Brief history of convex optimization

theory (convex analysis): ca1900–1970

algorithms

- 1947: simplex algorithm for linear programming (Dantzig)
- 1960s: early interior-point methods (Fiacco & McCormick, Dikin, . . .)
- 1970s: ellipsoid method and other subgradient methods
- 1980s: polynomial-time interior-point methods for linear programming (Karmarkar 1984)
- late 1980s–now: polynomial-time interior-point methods for nonlinear convex optimization (Nesterov & Nemirovski 1994)

applications

- before 1990: mostly in operations research; few in engineering
- since 1990: many new applications in engineering (control, signal processing, communications, circuit design, . . .); new problem classes (semidefinite and second-order cone programming, robust optimization)

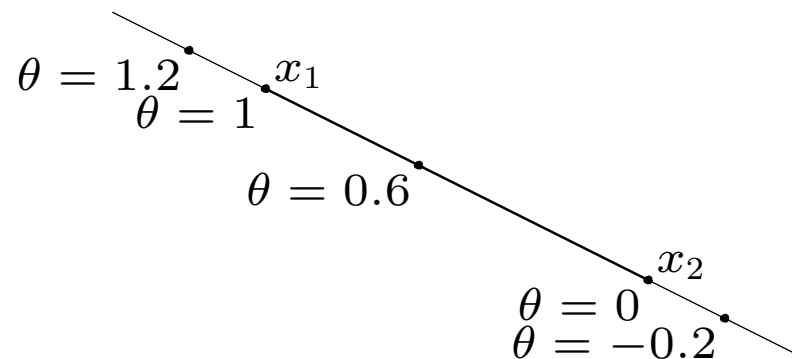
2. Convex sets

- affine and convex sets
- some important examples
- operations that preserve convexity
- generalized inequalities
- separating and supporting hyperplanes
- dual cones and generalized inequalities

Affine set

line through x_1, x_2 : all points

$$x = \theta x_1 + (1 - \theta)x_2 \quad (\theta \in \mathbf{R})$$



affine set: contains the line through any two distinct points in the set

example: solution set of linear equations $\{x \mid Ax = b\}$

(conversely, every affine set can be expressed as solution set of system of linear equations)

Convex set

line segment between x_1 and x_2 : all points

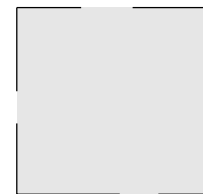
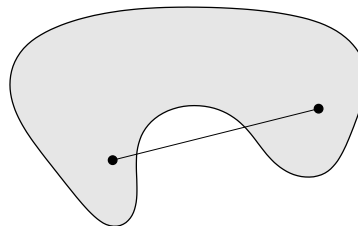
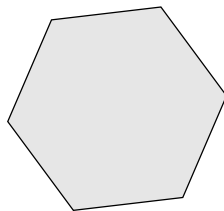
$$x = \theta x_1 + (1 - \theta)x_2$$

with $0 \leq \theta \leq 1$

convex set: contains line segment between any two points in the set

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

examples (one convex, two nonconvex sets)



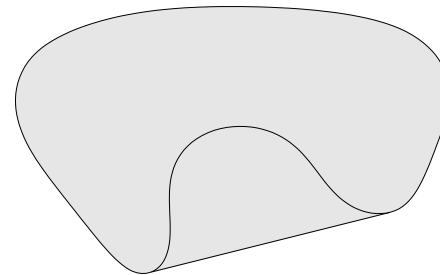
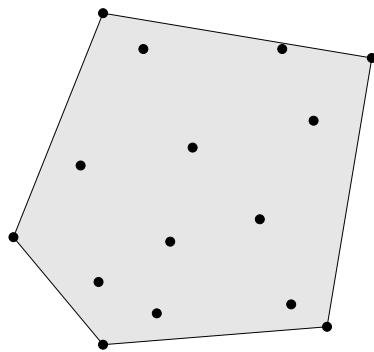
Convex combination and convex hull

convex combination of x_1, \dots, x_k : any point x of the form

$$x = \theta_1 x_1 + \theta_2 x_2 + \dots + \theta_k x_k$$

with $\theta_1 + \dots + \theta_k = 1$, $\theta_i \geq 0$

convex hull $\text{conv } S$: set of all convex combinations of points in S

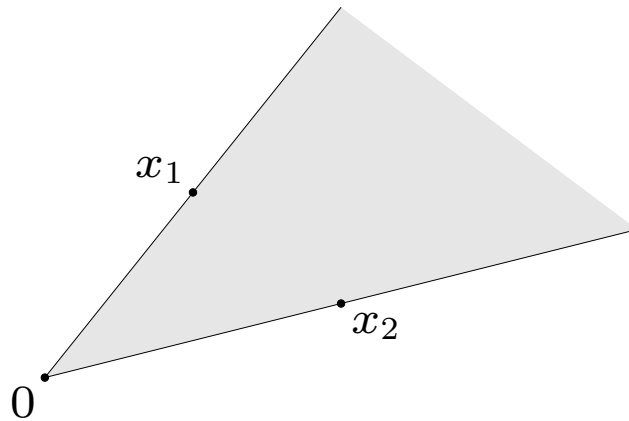


Convex cone

conic (nonnegative) combination of x_1 and x_2 : any point of the form

$$x = \theta_1 x_1 + \theta_2 x_2$$

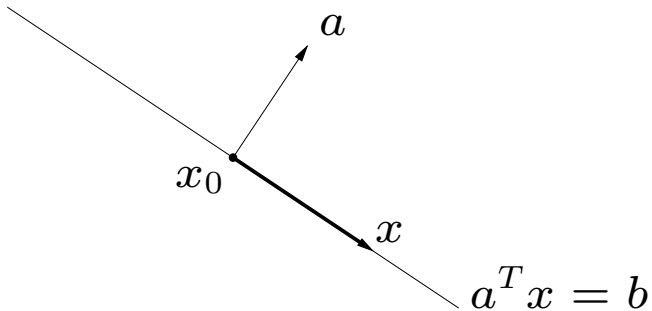
with $\theta_1 \geq 0$, $\theta_2 \geq 0$



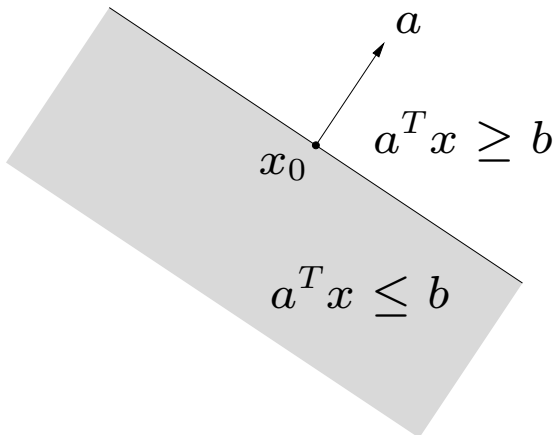
convex cone: set that contains all conic combinations of points in the set

Hyperplanes and halfspaces

hyperplane: set of the form $\{x \mid a^T x = b\}$ ($a \neq 0$)



halfspace: set of the form $\{x \mid a^T x \leq b\}$ ($a \neq 0$)



- a is the normal vector
- hyperplanes are affine and convex; halfspaces are convex

Euclidean balls and ellipsoids

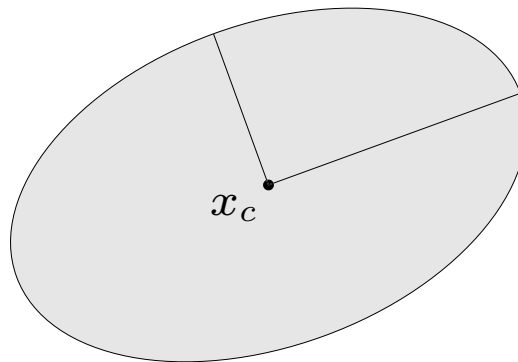
(Euclidean) ball with center x_c and radius r :

$$B(x_c, r) = \{x \mid \|x - x_c\|_2 \leq r\} = \{x_c + ru \mid \|u\|_2 \leq 1\}$$

ellipsoid: set of the form

$$\{x \mid (x - x_c)^T P^{-1} (x - x_c) \leq 1\}$$

with $P \in \mathbf{S}_{++}^n$ (*i.e.*, P symmetric positive definite)



other representation: $\{x_c + Au \mid \|u\|_2 \leq 1\}$ with A square and nonsingular

Norm balls and norm cones

norm: a function $\|\cdot\|$ that satisfies

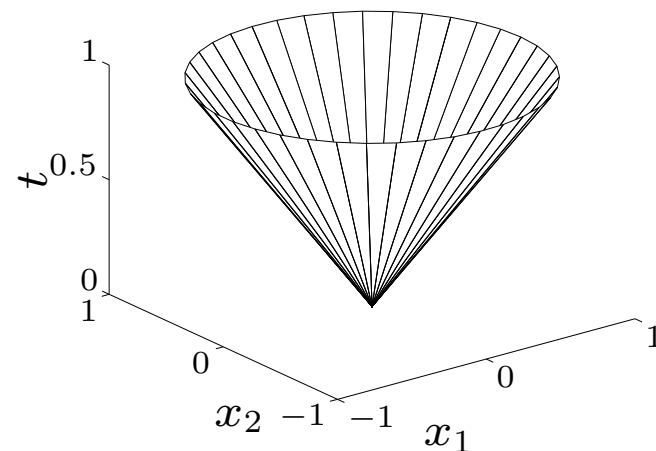
- $\|x\| \geq 0$; $\|x\| = 0$ if and only if $x = 0$
- $\|tx\| = |t| \|x\|$ for $t \in \mathbf{R}$
- $\|x + y\| \leq \|x\| + \|y\|$

notation: $\|\cdot\|$ is general (unspecified) norm; $\|\cdot\|_{\text{symb}}$ is particular norm

norm ball with center x_c and radius r : $\{x \mid \|x - x_c\| \leq r\}$

norm cone: $\{(x, t) \mid \|x\| \leq t\}$

Euclidean norm cone is called second-order cone



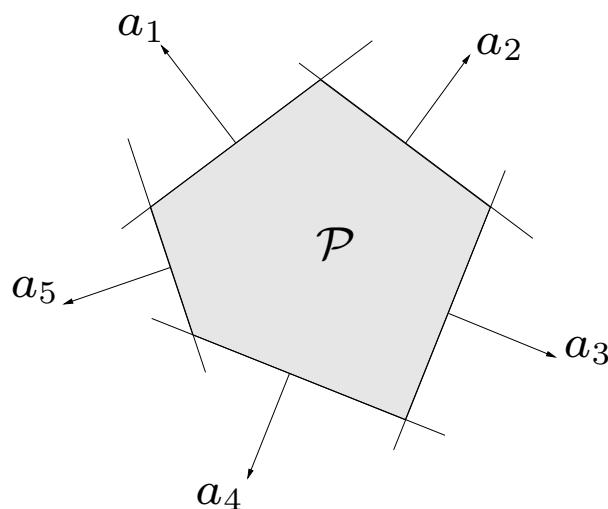
norm balls and cones are convex

Polyhedra

solution set of finitely many linear inequalities and equalities

$$Ax \preceq b, \quad Cx = d$$

($A \in \mathbf{R}^{m \times n}$, $C \in \mathbf{R}^{p \times n}$, \preceq is componentwise inequality)



polyhedron is intersection of finite number of halfspaces and hyperplanes

Positive semidefinite cone

notation:

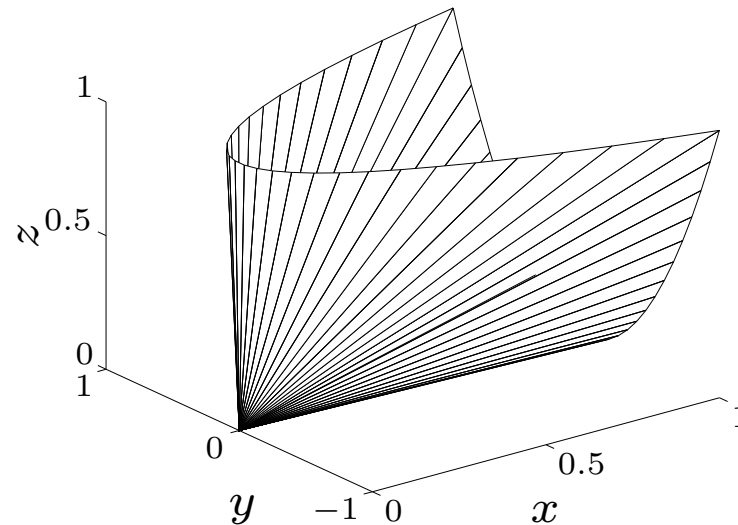
- \mathbf{S}^n is set of symmetric $n \times n$ matrices
- $\mathbf{S}_+^n = \{X \in \mathbf{S}^n \mid X \succeq 0\}$: positive semidefinite $n \times n$ matrices

$$X \in \mathbf{S}_+^n \iff z^T X z \geq 0 \text{ for all } z$$

\mathbf{S}_+^n is a convex cone

- $\mathbf{S}_{++}^n = \{X \in \mathbf{S}^n \mid X \succ 0\}$: positive definite $n \times n$ matrices

example: $\begin{bmatrix} x & y \\ y & z \end{bmatrix} \in \mathbf{S}_+^2$



Operations that preserve convexity

practical methods for establishing convexity of a set C

1. apply definition

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

2. show that C is obtained from simple convex sets (hyperplanes, halfspaces, norm balls, . . .) by operations that preserve convexity

- intersection
- affine functions
- perspective function
- linear-fractional functions

Intersection

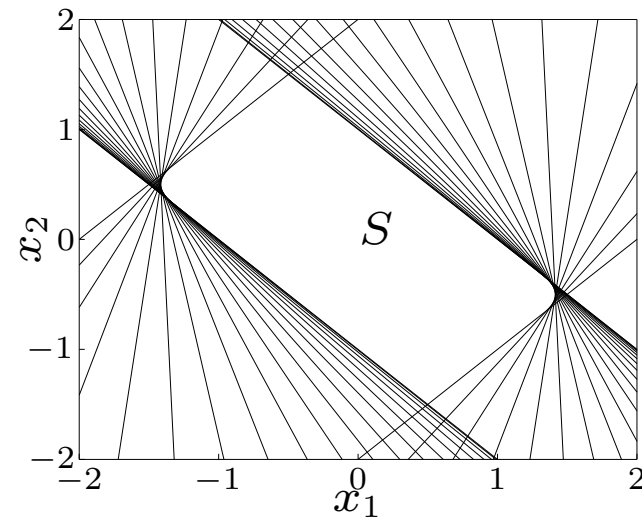
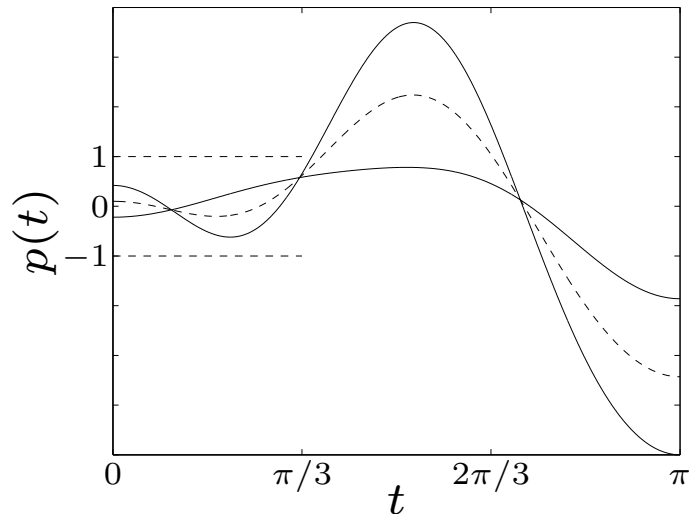
the intersection of (any number of) convex sets is convex

example:

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

where $p(t) = x_1 \cos t + x_2 \cos 2t + \cdots + x_m \cos mt$

for $m = 2$:



Affine function

suppose $f : \mathbf{R}^n \rightarrow \mathbf{R}^m$ is affine ($f(x) = Ax + b$ with $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$)

- the image of a convex set under f is convex

$$S \subseteq \mathbf{R}^n \text{ convex} \implies f(S) = \{f(x) \mid x \in S\} \text{ convex}$$

- the inverse image $f^{-1}(C)$ of a convex set under f is convex

$$C \subseteq \mathbf{R}^m \text{ convex} \implies f^{-1}(C) = \{x \in \mathbf{R}^n \mid f(x) \in C\} \text{ convex}$$

examples

- scaling, translation, projection
- solution set of linear matrix inequality $\{x \mid x_1 A_1 + \cdots + x_m A_m \preceq B\}$
(with $A_i, B \in \mathbf{S}^p$)
- hyperbolic cone $\{x \mid x^T P x \leq (c^T x)^2, c^T x \geq 0\}$ (with $P \in \mathbf{S}_+^n$)

Perspective and linear-fractional function

perspective function $P : \mathbf{R}^{n+1} \rightarrow \mathbf{R}^n$:

$$P(x, t) = x/t, \quad \text{dom } P = \{(x, t) \mid t > 0\}$$

images and inverse images of convex sets under perspective are convex

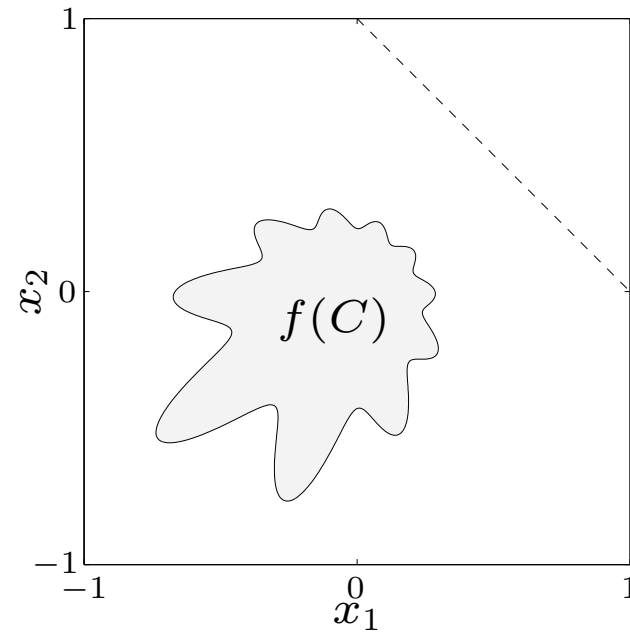
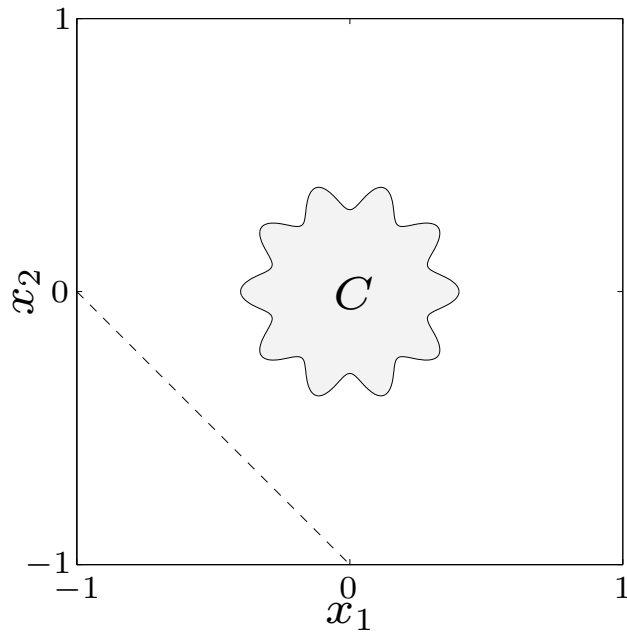
linear-fractional function $f : \mathbf{R}^n \rightarrow \mathbf{R}^m$:

$$f(x) = \frac{Ax + b}{c^T x + d}, \quad \text{dom } f = \{x \mid c^T x + d > 0\}$$

images and inverse images of convex sets under linear-fractional functions are convex

example of a linear-fractional function

$$f(x) = \frac{1}{x_1 + x_2 + 1}x$$



Generalized inequalities

a convex cone $K \subseteq \mathbf{R}^n$ is a **proper cone** if

- K is closed (contains its boundary)
- K is solid (has nonempty interior)
- K is pointed (contains no line)

examples

- nonnegative orthant $K = \mathbf{R}_+^n = \{x \in \mathbf{R}^n \mid x_i \geq 0, i = 1, \dots, n\}$
- positive semidefinite cone $K = \mathbf{S}_+^n$
- nonnegative polynomials on $[0, 1]$:

$$K = \{x \in \mathbf{R}^n \mid x_1 + x_2 t + x_3 t^2 + \dots + x_n t^{n-1} \geq 0 \text{ for } t \in [0, 1]\}$$

generalized inequality defined by a proper cone K :

$$x \preceq_K y \iff y - x \in K, \quad x \prec_K y \iff y - x \in \mathbf{int} K$$

examples

- componentwise inequality ($K = \mathbf{R}_+^n$)

$$x \preceq_{\mathbf{R}_+^n} y \iff x_i \leq y_i, \quad i = 1, \dots, n$$

- matrix inequality ($K = \mathbf{S}_+^n$)

$$X \preceq_{\mathbf{S}_+^n} Y \iff Y - X \text{ positive semidefinite}$$

these two types are so common that we drop the subscript in \preceq_K

properties: many properties of \preceq_K are similar to \leq on \mathbf{R} , *e.g.*,

$$x \preceq_K y, \quad u \preceq_K v \implies x + u \preceq_K y + v$$

Minimum and minimal elements

\preceq_K is not in general a *linear ordering*: we can have $x \not\preceq_K y$ and $y \not\preceq_K x$

$x \in S$ is **the minimum element** of S with respect to \preceq_K if

$$y \in S \implies x \preceq_K y$$

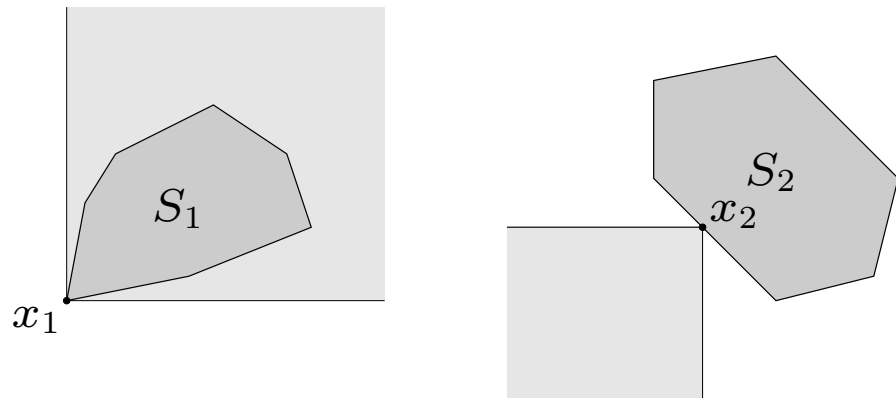
$x \in S$ is **a minimal element** of S with respect to \preceq_K if

$$y \in S, \quad y \preceq_K x \implies y = x$$

example ($K = \mathbf{R}_+^2$)

x_1 is the minimum element of S_1

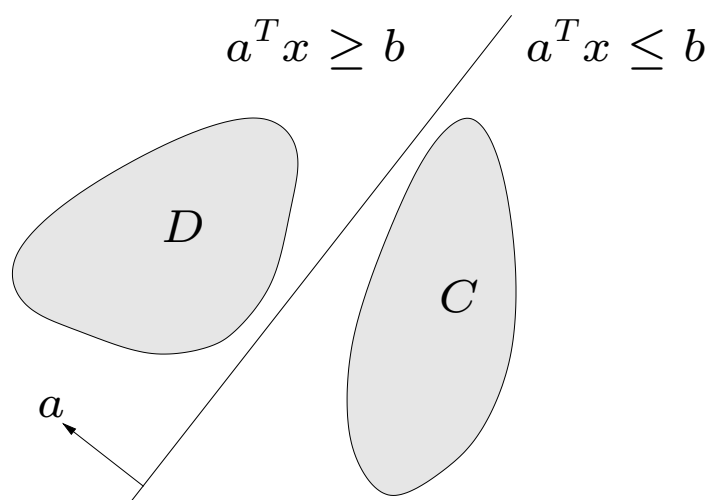
x_2 is a minimal element of S_2



Separating hyperplane theorem

if C and D are disjoint convex sets, then there exists $a \neq 0$, b such that

$$a^T x \leq b \text{ for } x \in C, \quad a^T x \geq b \text{ for } x \in D$$



the hyperplane $\{x \mid a^T x = b\}$ separates C and D

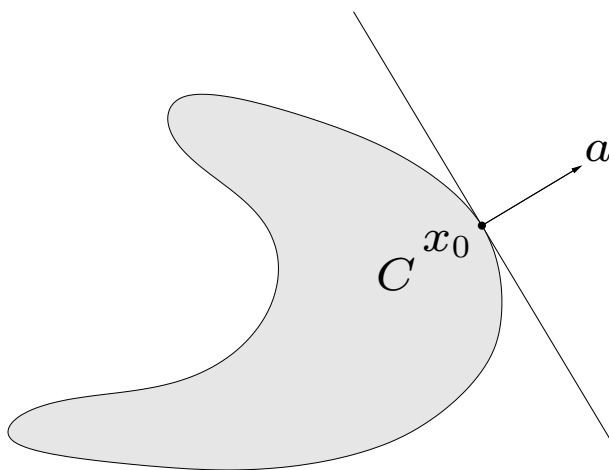
strict separation requires additional assumptions (*e.g.*, C is closed, D is a singleton)

Supporting hyperplane theorem

supporting hyperplane to set C at boundary point x_0 :

$$\{x \mid a^T x = a^T x_0\}$$

where $a \neq 0$ and $a^T x \leq a^T x_0$ for all $x \in C$



supporting hyperplane theorem: if C is convex, then there exists a supporting hyperplane at every boundary point of C

Dual cones and generalized inequalities

dual cone of a cone K :

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

examples

- $K = \mathbf{R}_+^n$: $K^* = \mathbf{R}_+^n$
- $K = \mathbf{S}_+^n$: $K^* = \mathbf{S}_+^n$
- $K = \{(x, t) \mid \|x\|_2 \leq t\}$: $K^* = \{(x, t) \mid \|x\|_2 \leq t\}$
- $K = \{(x, t) \mid \|x\|_1 \leq t\}$: $K^* = \{(x, t) \mid \|x\|_\infty \leq t\}$

first three examples are **self-dual** cones

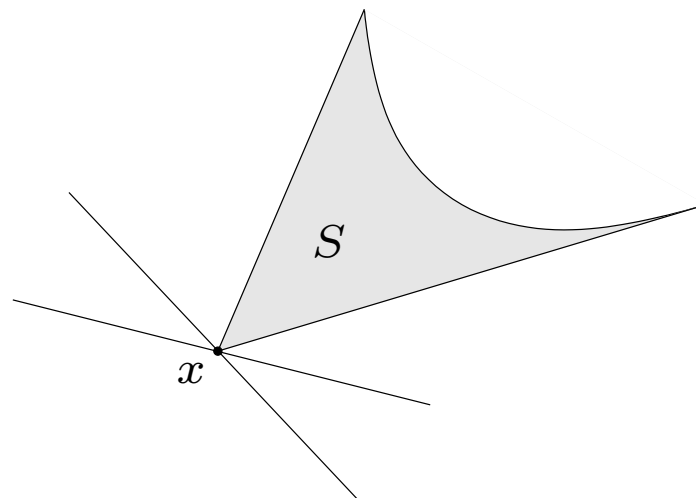
dual cones of proper cones are proper, hence define generalized inequalities:

$$y \succeq_{K^*} 0 \iff y^T x \geq 0 \text{ for all } x \succeq_K 0$$

Minimum and minimal elements via dual inequalities

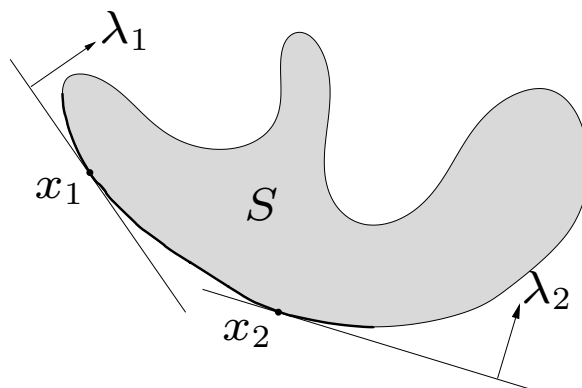
minimum element w.r.t. \preceq_K

x is minimum element of S iff for all $\lambda \succ_{K^*} 0$, x is the unique minimizer of $\lambda^T z$ over S



minimal element w.r.t. \preceq_K

- if x minimizes $\lambda^T z$ over S for some $\lambda \succ_{K^*} 0$, then x is minimal



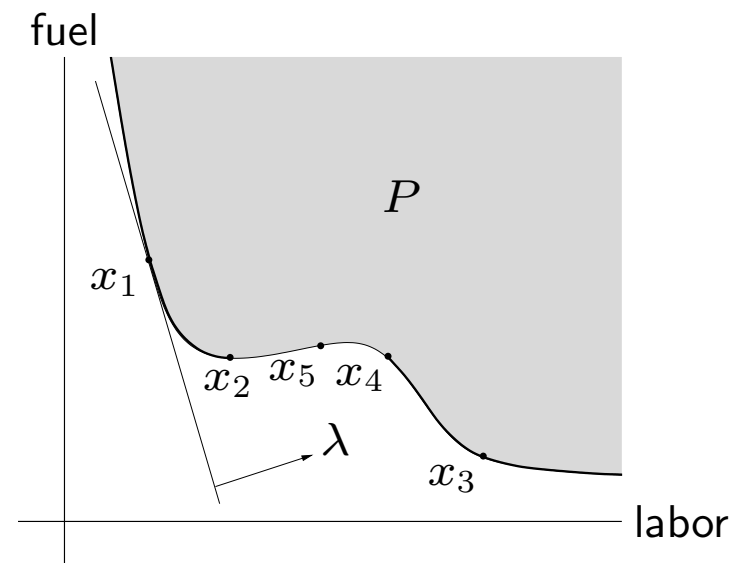
- if x is a minimal element of a *convex* set S , then there exists a nonzero $\lambda \succeq_{K^*} 0$ such that x minimizes $\lambda^T z$ over S

optimal production frontier

- different production methods use different amounts of resources $x \in \mathbf{R}^n$
- production set P : resource vectors x for all possible production methods
- efficient (Pareto optimal) methods correspond to resource vectors x that are minimal w.r.t. \mathbf{R}_+^n

example ($n = 2$)

x_1, x_2, x_3 are efficient; x_4, x_5 are not



3. Convex functions

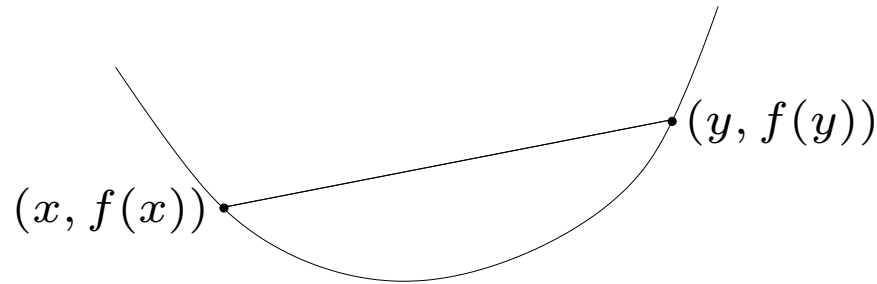
- basic properties and examples
- operations that preserve convexity
- the conjugate function
- quasiconvex functions
- log-concave and log-convex functions
- convexity with respect to generalized inequalities

Definition

$f : \mathbf{R}^n \rightarrow \mathbf{R}$ is convex if $\mathbf{dom} f$ 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
- f is strictly convex if $\mathbf{dom} f$ is convex and

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

for $x, y \in \mathbf{dom} f$, $x \neq y$, $0 < \theta < 1$

Examples on \mathbf{R}

convex:

- affine: $ax + b$ on \mathbf{R} , for any $a, b \in \mathbf{R}$
- exponential: e^{ax} , for any $a \in \mathbf{R}$
- powers: x^α on \mathbf{R}_{++} , for $\alpha \geq 1$ or $\alpha \leq 0$
- powers of absolute value: $|x|^p$ on \mathbf{R} , for $p \geq 1$
- negative entropy: $x \log x$ on \mathbf{R}_{++}

concave:

- affine: $ax + b$ on \mathbf{R} , for any $a, b \in \mathbf{R}$
- powers: x^α on \mathbf{R}_{++} , for $0 \leq \alpha \leq 1$
- logarithm: $\log x$ on \mathbf{R}_{++}

Examples on \mathbf{R}^n and $\mathbf{R}^{m \times n}$

affine functions are convex and concave; all norms are convex

examples on \mathbf{R}^n

- affine function $f(x) = a^T x + b$
- norms: $\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{1/p}$ for $p \geq 1$; $\|x\|_\infty = \max_k |x_k|$

examples on $\mathbf{R}^{m \times n}$ ($m \times n$ matrices)

- affine function

$$f(X) = \mathbf{tr}(A^T X) + b = \sum_{i=1}^m \sum_{j=1}^n A_{ij} X_{ij} + b$$

- spectral (maximum singular value) norm

$$f(X) = \|X\|_2 = \sigma_{\max}(X) = (\lambda_{\max}(X^T X))^{1/2}$$

Restriction of a convex function to a line

$f : \mathbf{R}^n \rightarrow \mathbf{R}$ is convex if and only if the function $g : \mathbf{R} \rightarrow \mathbf{R}$,

$$g(t) = f(x + tv), \quad \text{dom } g = \{t \mid x + tv \in \text{dom } f\}$$

is convex (in t) for any $x \in \text{dom } f$, $v \in \mathbf{R}^n$

can check convexity of f by checking convexity of functions of one variable

example. $f : \mathbf{S}^n \rightarrow \mathbf{R}$ with $f(X) = \log \det X$, $\text{dom } X = \mathbf{S}_{++}^n$

$$\begin{aligned} g(t) = \log \det(X + tV) &= \log \det X + \log \det(I + tX^{-1/2}VX^{-1/2}) \\ &= \log \det X + \sum_{i=1}^n \log(1 + t\lambda_i) \end{aligned}$$

where λ_i are the eigenvalues of $X^{-1/2}VX^{-1/2}$

g is concave in t (for any choice of $X \succ 0$, V); hence f is concave

Extended-value extension

extended-value extension \tilde{f} of f is

$$\tilde{f}(x) = f(x), \quad x \in \mathbf{dom} f, \quad \tilde{f}(x) = \infty, \quad x \notin \mathbf{dom} f$$

often simplifies notation; for example, the condition

$$0 \leq \theta \leq 1 \implies \tilde{f}(\theta x + (1 - \theta)y) \leq \theta \tilde{f}(x) + (1 - \theta) \tilde{f}(y)$$

(as an inequality in $\mathbf{R} \cup \{\infty\}$), means the same as the two conditions

- $\mathbf{dom} f$ is convex
- for $x, y \in \mathbf{dom} f$,

$$0 \leq \theta \leq 1 \implies f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$$

First-order condition

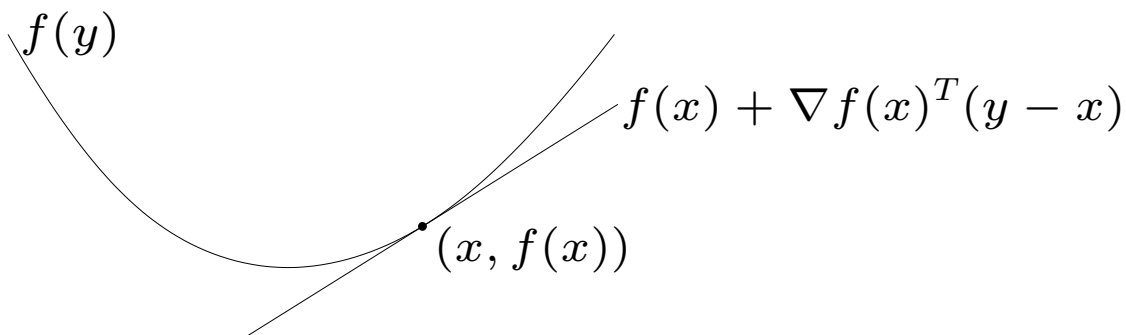
f is **differentiable** if $\text{dom } f$ is open and the gradient

$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \dots, \frac{\partial f(x)}{\partial x_n} \right)$$

exists at each $x \in \text{dom } f$

1st-order condition: differentiable f with convex domain is convex iff

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



first-order approximation of f is global underestimator

Second-order conditions

f is **twice differentiable** if $\text{dom } f$ is open and the Hessian $\nabla^2 f(x) \in \mathbf{S}^n$,

$$\nabla^2 f(x)_{ij} = \frac{\partial^2 f(x)}{\partial x_i \partial x_j}, \quad i, j = 1, \dots, n,$$

exists at each $x \in \text{dom } f$

2nd-order conditions: for twice differentiable f with convex domain

- f is convex if and only if

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

- if $\nabla^2 f(x) \succ 0$ for all $x \in \text{dom } f$, then f is strictly convex

Examples

quadratic function: $f(x) = (1/2)x^T P x + q^T x + r$ (with $P \in \mathbf{S}^n$)

$$\nabla f(x) = Px + q, \quad \nabla^2 f(x) = P$$

convex if $P \succeq 0$

least-squares objective: $f(x) = \|Ax - b\|_2^2$

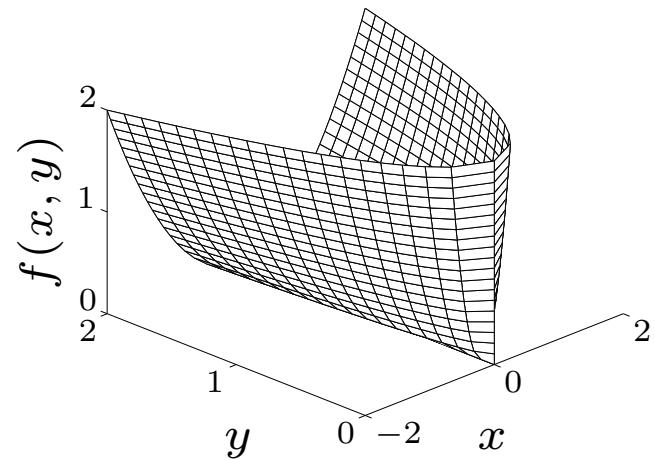
$$\nabla f(x) = 2A^T(Ax - b), \quad \nabla^2 f(x) = 2A^T A$$

convex (for any A)

quadratic-over-linear: $f(x, y) = x^2/y$

$$\nabla^2 f(x, y) = \frac{2}{y^3} \begin{bmatrix} y \\ -x \end{bmatrix} \begin{bmatrix} y \\ -x \end{bmatrix}^T \succeq 0$$

convex for $y > 0$



log-sum-exp: $f(x) = \log \sum_{k=1}^n \exp x_k$ is convex

$$\nabla^2 f(x) = \frac{1}{\mathbf{1}^T z} \mathbf{diag}(z) - \frac{1}{(\mathbf{1}^T z)^2} z z^T \quad (z_k = \exp x_k)$$

to show $\nabla^2 f(x) \succeq 0$, we must verify that $v^T \nabla^2 f(x) v \geq 0$ for all v :

$$v^T \nabla^2 f(x) v = \frac{(\sum_k z_k v_k^2)(\sum_k z_k) - (\sum_k v_k z_k)^2}{(\sum_k z_k)^2} \geq 0$$

since $(\sum_k v_k z_k)^2 \leq (\sum_k z_k v_k^2)(\sum_k z_k)$ (from Cauchy-Schwarz inequality)

geometric mean: $f(x) = (\prod_{k=1}^n x_k)^{1/n}$ on \mathbf{R}_{++}^n is concave

(similar proof as for log-sum-exp)

Epigraph and sublevel set

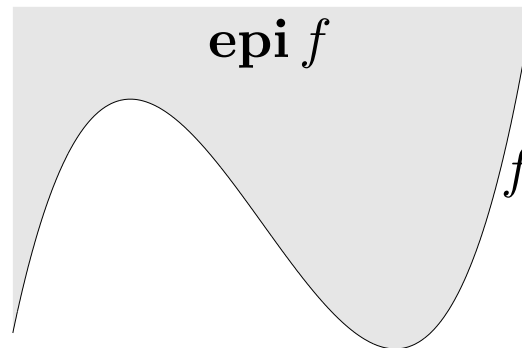
α -sublevel set of $f : \mathbf{R}^n \rightarrow \mathbf{R}$:

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

sublevel sets of convex functions are convex (converse is false)

epigraph of $f : \mathbf{R}^n \rightarrow \mathbf{R}$:

$$\mathbf{epi} f = \{(x, t) \in \mathbf{R}^{n+1} \mid x \in \mathbf{dom} f, f(x) \leq t\}$$



f is convex if and only if $\mathbf{epi} f$ is a convex set

Jensen's inequality

basic inequality: if f is convex, then for $0 \leq \theta \leq 1$,

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

extension: if f is convex, then

$$f(\mathbf{E} z) \leq \mathbf{E} f(z)$$

for any random variable z

basic inequality is special case with discrete distribution

$$\mathbf{prob}(z = x) = \theta, \quad \mathbf{prob}(z = y) = 1 - \theta$$

Operations that preserve convexity

practical methods for establishing convexity of a function

1. verify definition (often simplified by restricting to a line)
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: αf 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

$$f(x) = - \sum_{i=1}^m \log(b_i - a_i^T x), \quad \text{dom } f = \{x \mid a_i^T x < b_i, i = 1, \dots, m\}$$

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

Pointwise maximum

if f_1, \dots, f_m are convex, then $f(x) = \max\{f_1(x), \dots, f_m(x)\}$ is convex

examples

- piecewise-linear function: $f(x) = \max_{i=1,\dots,m}(a_i^T x + b_i)$ is convex
- sum of r largest components of $x \in \mathbf{R}^n$:

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

is convex ($x_{[i]}$ is i th largest component of x)

proof:

$$f(x) = \max\{x_{i_1} + x_{i_2} + \dots + x_{i_r} \mid 1 \leq i_1 < i_2 < \dots < i_r \leq n\}$$

Pointwise supremum

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

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

is convex

examples

- support function of a set C : $S_C(x) = \sup_{y \in C} y^T x$ is convex
- distance to farthest point in a set C :

$$f(x) = \sup_{y \in C} \|x - y\|$$

- maximum eigenvalue of symmetric matrix: for $X \in \mathbf{S}^n$,

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

Composition with scalar functions

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

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

f is convex if $\begin{array}{l} g \text{ convex, } h \text{ convex, } \tilde{h} \text{ nondecreasing} \\ g \text{ concave, } h \text{ convex, } \tilde{h} \text{ nonincreasing} \end{array}$

- proof (for $n = 1$, differentiable g, h)

$$f''(x) = h''(g(x))g'(x)^2 + h'(g(x))g''(x)$$

- note: monotonicity must hold for extended-value extension \tilde{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 $g : \mathbf{R}^n \rightarrow \mathbf{R}^k$ and $h : \mathbf{R}^k \rightarrow \mathbf{R}$:

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

f is convex if $\begin{array}{l} g_i \text{ convex, } h \text{ convex, } \tilde{h} \text{ nondecreasing in each argument} \\ g_i \text{ concave, } h \text{ convex, } \tilde{h} \text{ nonincreasing in each argument} \end{array}$

proof (for $n = 1$, differentiable g, h)

$$f''(x) = g'(x)^T \nabla^2 h(g(x)) g'(x) + \nabla h(g(x))^T g''(x)$$

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

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

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

is convex

examples

- $f(x, y) = x^T A x + 2x^T B y + y^T C y$ with

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succeq 0, \quad C \succ 0$$

minimizing over y gives $g(x) = \inf_y f(x, y) = x^T (A - B C^{-1} B^T) x$

g is convex, hence Schur complement $A - B C^{-1} B^T \succeq 0$

- distance to a set: $\mathbf{dist}(x, S) = \inf_{y \in S} \|x - y\|$ is convex if S is convex

Perspective

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

$$g(x, t) = tf(x/t), \quad \text{dom } g = \{(x, t) \mid x/t \in \text{dom } f, t > 0\}$$

g is convex if f is convex

examples

- $f(x) = x^T x$ is convex; hence $g(x, t) = x^T x/t$ is convex for $t > 0$
- negative logarithm $f(x) = -\log x$ is convex; hence relative entropy $g(x, t) = t \log t - t \log x$ is convex on \mathbf{R}_{++}^2
- if f is convex, then

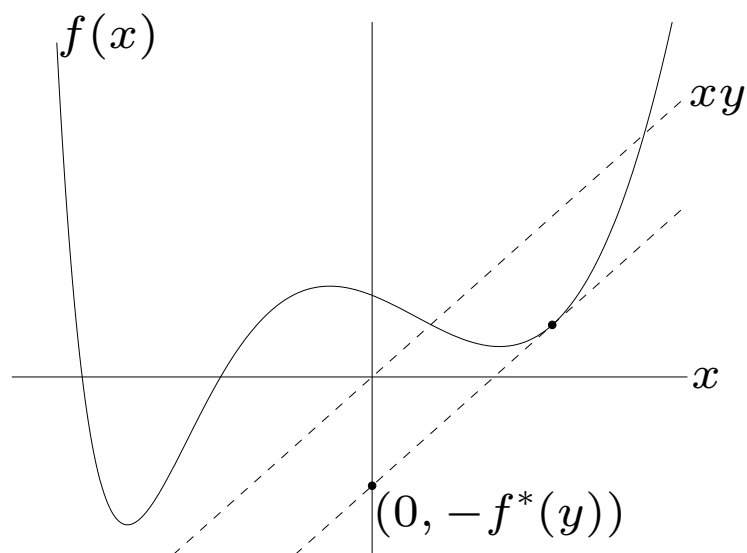
$$g(x) = (c^T x + d)f((Ax + b)/(c^T x + d))$$

is convex on $\{x \mid c^T x + d > 0, (Ax + b)/(c^T x + d) \in \text{dom } f\}$

The 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)
- will be useful in chapter 5

examples

- negative logarithm $f(x) = -\log x$

$$\begin{aligned} f^*(y) &= \sup_{x>0} (xy + \log x) \\ &= \begin{cases} -1 - \log(-y) & y < 0 \\ \infty & \text{otherwise} \end{cases} \end{aligned}$$

- strictly convex quadratic $f(x) = (1/2)x^T Qx$ with $Q \in \mathbf{S}_{++}^n$

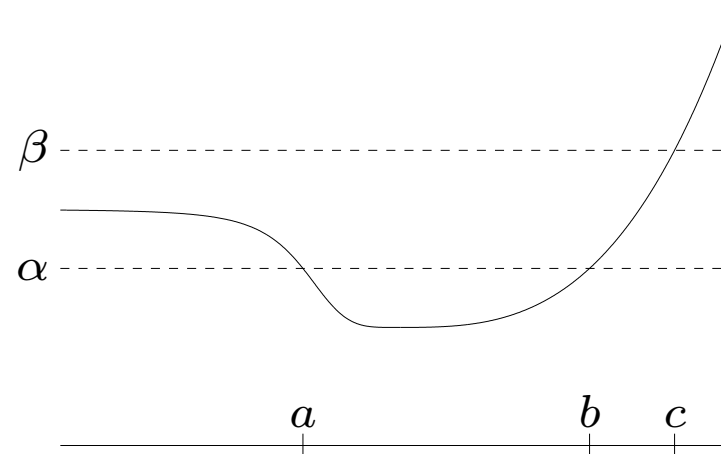
$$\begin{aligned} f^*(y) &= \sup_x (y^T x - (1/2)x^T Qx) \\ &= \frac{1}{2} y^T Q^{-1} y \end{aligned}$$

Quasiconvex functions

$f : \mathbf{R}^n \rightarrow \mathbf{R}$ is quasiconvex if $\mathbf{dom} f$ 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
- f is quasilinear if it is quasiconvex and quasiconcave

Examples

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

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

is quasilinear

- distance ratio

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

is quasiconvex

internal rate of return

- cash flow $x = (x_0, \dots, x_n)$; x_i is payment in period i (to us if $x_i > 0$)
- we assume $x_0 < 0$ and $x_0 + x_1 + \dots + x_n > 0$
- present value of cash flow x , for interest rate r :

$$\text{PV}(x, r) = \sum_{i=0}^n (1+r)^{-i} x_i$$

- internal rate of return is smallest interest rate for which $\text{PV}(x, r) = 0$:

$$\text{IRR}(x) = \inf\{r \geq 0 \mid \text{PV}(x, r) = 0\}$$

IRR is quasiconcave: superlevel set is intersection of halfspaces

$$\text{IRR}(x) \geq R \iff \sum_{i=0}^n (1+r)^{-i} x_i \geq 0 \text{ for } 0 \leq r \leq R$$

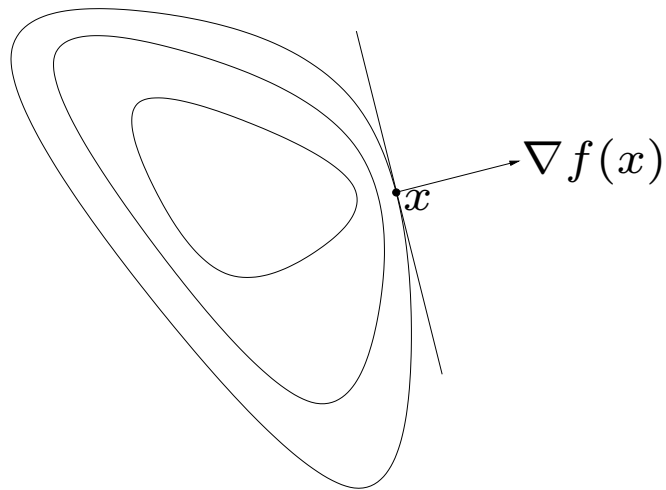
Properties

modified Jensen inequality: for quasiconvex f

$$0 \leq \theta \leq 1 \implies f(\theta x + (1 - \theta)y) \leq \max\{f(x), f(y)\}$$

first-order condition: differentiable f with cvx domain is quasiconvex iff

$$f(y) \leq f(x) \implies \nabla f(x)^T (y - x) \leq 0$$



sums of quasiconvex functions are not necessarily quasiconvex

Log-concave and log-convex functions

a positive function f is log-concave if $\log f$ is concave:

$$f(\theta x + (1 - \theta)y) \geq f(x)^\theta f(y)^{1-\theta} \quad \text{for } 0 \leq \theta \leq 1$$

f is log-convex if $\log f$ is convex

- powers: x^a on \mathbf{R}_{++} is log-convex for $a \leq 0$, log-concave for $a \geq 0$
- many common probability densities are log-concave, *e.g.*, normal:

$$f(x) = \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} e^{-\frac{1}{2}(x-\bar{x})^T \Sigma^{-1}(x-\bar{x})}$$

- cumulative Gaussian distribution function Φ is log-concave

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^x e^{-u^2/2} du$$

Properties of log-concave functions

- twice differentiable f with convex domain is log-concave if and only if

$$f(x)\nabla^2 f(x) \preceq \nabla f(x)\nabla f(x)^T$$

for all $x \in \text{dom } f$

- product of log-concave functions is log-concave
- sum of log-concave functions is not always log-concave
- integration: if $f : \mathbf{R}^n \times \mathbf{R}^m \rightarrow \mathbf{R}$ is log-concave, then

$$g(x) = \int f(x, y) \, dy$$

is log-concave (not easy to show)

consequences of integration property

- convolution $f * g$ of log-concave functions f, g is log-concave

$$(f * g)(x) = \int f(x - y)g(y)dy$$

- if $C \subseteq \mathbf{R}^n$ convex and y is a random variable with log-concave pdf then

$$f(x) = \mathbf{prob}(x + y \in C)$$

is log-concave

proof: write $f(x)$ as integral of product of log-concave functions

$$f(x) = \int g(x + y)p(y) dy, \quad g(u) = \begin{cases} 1 & u \in C \\ 0 & u \notin C, \end{cases}$$

p is pdf of y

example: yield function

$$Y(x) = \mathbf{prob}(x + w \in S)$$

- $x \in \mathbf{R}^n$: nominal parameter values for product
- $w \in \mathbf{R}^n$: random variations of parameters in manufactured product
- S : set of acceptable values

if S is convex and w has a log-concave pdf, then

- Y is log-concave
- yield regions $\{x \mid Y(x) \geq \alpha\}$ are convex

Convexity with respect to generalized inequalities

$f : \mathbf{R}^n \rightarrow \mathbf{R}^m$ is K -convex if $\text{dom } f$ is convex and

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

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

example $f : \mathbf{S}^m \rightarrow \mathbf{S}^m$, $f(X) = X^2$ is \mathbf{S}_+^m -convex

proof: for fixed $z \in \mathbf{R}^m$, $z^T X^2 z = \|Xz\|_2^2$ is convex in X , *i.e.*,

$$z^T (\theta X + (1 - \theta)Y)^2 z \leq \theta z^T X^2 z + (1 - \theta) z^T Y^2 z$$

for $X, Y \in \mathbf{S}^m$, $0 \leq \theta \leq 1$

therefore $(\theta X + (1 - \theta)Y)^2 \preceq \theta X^2 + (1 - \theta)Y^2$

4. Convex optimization problems

- optimization problem in standard form
- convex optimization problems
- quasiconvex optimization
- linear optimization
- quadratic optimization
- geometric programming
- generalized inequality constraints
- semidefinite programming
- vector optimization

Optimization problem in standard form

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

- $x \in \mathbf{R}^n$ is the optimization variable
- $f_0 : \mathbf{R}^n \rightarrow \mathbf{R}$ is the objective or cost function
- $f_i : \mathbf{R}^n \rightarrow \mathbf{R}$, $i = 1, \dots, m$, are the inequality constraint functions
- $h_i : \mathbf{R}^n \rightarrow \mathbf{R}$ are the equality constraint functions

optimal value:

$$p^* = \inf \{ f_0(x) \mid f_i(x) \leq 0, \ i = 1, \dots, m, \ h_i(x) = 0, \ i = 1, \dots, p \}$$

- $p^* = \infty$ if problem is infeasible (no x satisfies the constraints)
- $p^* = -\infty$ if problem is unbounded below

Optimal and locally optimal points

x is **feasible** if $x \in \text{dom } f_0$ and it satisfies the constraints

a feasible x is **optimal** if $f_0(x) = p^*$; X_{opt} is the set of optimal points

x is **locally optimal** if there is an $R > 0$ such that x is optimal for

$$\begin{array}{ll} \text{minimize (over } z) & f_0(z) \\ \text{subject to} & f_i(z) \leq 0, \quad i = 1, \dots, m, \quad h_i(z) = 0, \quad i = 1, \dots, p \\ & \|z - x\|_2 \leq R \end{array}$$

examples (with $n = 1$, $m = p = 0$)

- $f_0(x) = 1/x$, $\text{dom } f_0 = \mathbf{R}_{++}$: $p^* = 0$, no optimal point
- $f_0(x) = -\log x$, $\text{dom } f_0 = \mathbf{R}_{++}$: $p^* = -\infty$
- $f_0(x) = x \log x$, $\text{dom } f_0 = \mathbf{R}_{++}$: $p^* = -1/e$, $x = 1/e$ is optimal
- $f_0(x) = x^3 - 3x$, $p^* = -\infty$, local optimum at $x = 1$

Implicit constraints

the standard form optimization problem has an **implicit constraint**

$$x \in \mathcal{D} = \bigcap_{i=0}^m \text{dom } f_i \cap \bigcap_{i=1}^p \text{dom } h_i,$$

- we call \mathcal{D} the **domain** of the problem
- the constraints $f_i(x) \leq 0$, $h_i(x) = 0$ are the explicit constraints
- a problem is **unconstrained** if it has no explicit constraints ($m = p = 0$)

example:

$$\text{minimize } f_0(x) = -\sum_{i=1}^k \log(b_i - a_i^T x)$$

is an unconstrained problem with implicit constraints $a_i^T x < b_i$

Feasibility problem

$$\begin{array}{ll}\text{find} & x \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \dots, m \\ & h_i(x) = 0, \quad i = 1, \dots, p\end{array}$$

can be considered a special case of the general problem with $f_0(x) = 0$:

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

- $p^* = 0$ if constraints are feasible; any feasible x is optimal
- $p^* = \infty$ if constraints are infeasible

Convex optimization problem

standard form convex optimization problem

$$\begin{array}{ll}\text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \dots, m \\ & a_i^T x = b_i, \quad i = 1, \dots, p\end{array}$$

- f_0, f_1, \dots, f_m are convex; equality constraints are affine
- problem is *quasiconvex* if f_0 is quasiconvex (and f_1, \dots, f_m convex)

often written as

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

important property: feasible set of a convex optimization problem is convex

example

$$\begin{array}{ll}\text{minimize} & f_0(x) = x_1^2 + x_2^2 \\ \text{subject to} & f_1(x) = x_1/(1 + x_2^2) \leq 0 \\ & h_1(x) = (x_1 + x_2)^2 = 0\end{array}$$

- f_0 is convex; feasible set $\{(x_1, x_2) \mid x_1 = -x_2 \leq 0\}$ is convex
- not a convex problem (according to our definition): f_1 is not convex, h_1 is not affine
- equivalent (but not identical) to the convex problem

$$\begin{array}{ll}\text{minimize} & x_1^2 + x_2^2 \\ \text{subject to} & x_1 \leq 0 \\ & x_1 + x_2 = 0\end{array}$$

Local and global optima

any locally optimal point of a convex problem is (globally) optimal

proof: suppose x is locally optimal and y is optimal with $f_0(y) < f_0(x)$

x locally optimal means there is an $R > 0$ such that

$$z \text{ feasible, } \|z - x\|_2 \leq R \implies f_0(z) \geq f_0(x)$$

consider $z = \theta y + (1 - \theta)x$ with $\theta = R/(2\|y - x\|_2)$

- $\|y - x\|_2 > R$, so $0 < \theta < 1/2$
- z is a convex combination of two feasible points, hence also feasible
- $\|z - x\|_2 = R/2$ and

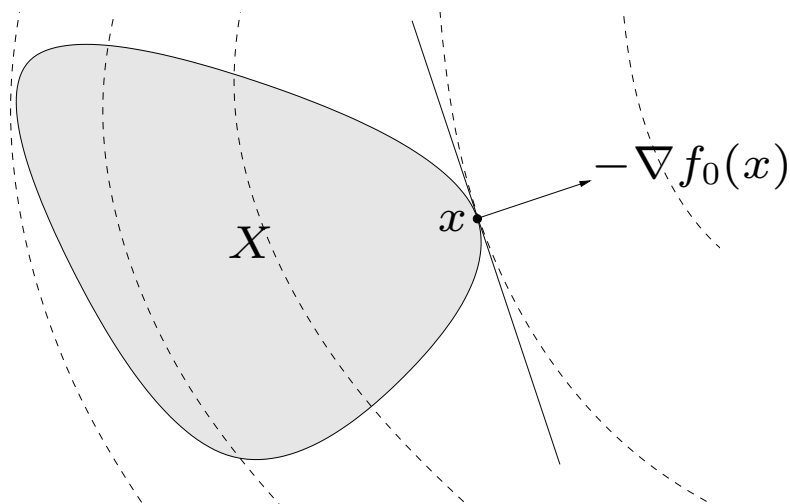
$$f_0(z) \leq \theta f_0(y) + (1 - \theta)f_0(x) < f_0(x)$$

which contradicts our assumption that x is locally optimal

Optimality criterion for differentiable f_0

x is optimal if and only if it is feasible and

$$\nabla f_0(x)^T(y - x) \geq 0 \quad \text{for all feasible } y$$



if nonzero, $\nabla f_0(x)$ defines a supporting hyperplane to feasible set X at x

- **unconstrained problem:** x is optimal if and only if

$$x \in \mathbf{dom} f_0, \quad \nabla f_0(x) = 0$$

- **equality constrained problem**

$$\text{minimize } f_0(x) \quad \text{subject to } Ax = b$$

x is optimal if and only if there exists a ν such that

$$x \in \mathbf{dom} f_0, \quad Ax = b, \quad \nabla f_0(x) + A^T \nu = 0$$

- **minimization over nonnegative orthant**

$$\text{minimize } f_0(x) \quad \text{subject to } x \succeq 0$$

x is optimal if and only if

$$x \in \mathbf{dom} f_0, \quad x \succeq 0, \quad \begin{cases} \nabla f_0(x)_i \geq 0 & x_i = 0 \\ \nabla f_0(x)_i = 0 & x_i > 0 \end{cases}$$

Equivalent convex problems

two problems are (informally) **equivalent** if the solution of one is readily obtained from the solution of the other, and vice-versa

some common transformations that preserve convexity:

- **eliminating equality constraints**

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

is equivalent to

$$\begin{array}{ll}\text{minimize (over } z) & f_0(Fz + x_0) \\ \text{subject to} & f_i(Fz + x_0) \leq 0, \quad i = 1, \dots, m\end{array}$$

where F and x_0 are such that

$$Ax = b \iff x = Fz + x_0 \text{ for some } z$$

- **introducing equality constraints**

$$\begin{array}{ll}\text{minimize} & f_0(A_0x + b_0) \\ \text{subject to} & f_i(A_ix + b_i) \leq 0, \quad i = 1, \dots, m\end{array}$$

is equivalent to

$$\begin{array}{ll}\text{minimize (over } x, y_i) & f_0(y_0) \\ \text{subject to} & f_i(y_i) \leq 0, \quad i = 1, \dots, m \\ & y_i = A_ix + b_i, \quad i = 0, 1, \dots, m\end{array}$$

- **introducing slack variables for linear inequalities**

$$\begin{array}{ll}\text{minimize} & f_0(x) \\ \text{subject to} & a_i^T x \leq b_i, \quad i = 1, \dots, m\end{array}$$

is equivalent to

$$\begin{array}{ll}\text{minimize (over } x, s) & f_0(x) \\ \text{subject to} & a_i^T x + s_i = b_i, \quad i = 1, \dots, m \\ & s_i \geq 0, \quad i = 1, \dots, m\end{array}$$

- **epigraph form:** standard form convex problem is equivalent to

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

- **minimizing over some variables**

$$\begin{array}{ll} \text{minimize} & f_0(x_1, x_2) \\ \text{subject to} & f_i(x_1) \leq 0, \quad i = 1, \dots, m \end{array}$$

is equivalent to

$$\begin{array}{ll} \text{minimize} & \tilde{f}_0(x_1) \\ \text{subject to} & f_i(x_1) \leq 0, \quad i = 1, \dots, m \end{array}$$

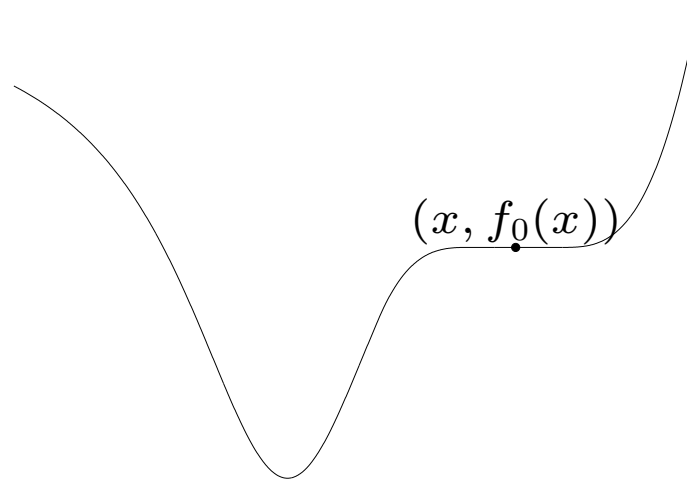
where $\tilde{f}_0(x_1) = \inf_{x_2} f_0(x_1, x_2)$

Quasiconvex optimization

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

with $f_0 : \mathbf{R}^n \rightarrow \mathbf{R}$ quasiconvex, f_1, \dots, f_m convex

can have locally optimal points that are not (globally) optimal



convex representation of sublevel sets of f_0

if f_0 is quasiconvex, there exists a family of functions ϕ_t such that:

- $\phi_t(x)$ is convex in x for fixed t
- t -sublevel set of f_0 is 0-sublevel set of ϕ_t , i.e.,

$$f_0(x) \leq t \iff \phi_t(x) \leq 0$$

example

$$f_0(x) = \frac{p(x)}{q(x)}$$

with p convex, q concave, and $p(x) \geq 0$, $q(x) > 0$ on $\text{dom } f_0$

can take $\phi_t(x) = p(x) - tq(x)$:

- for $t \geq 0$, ϕ_t convex in x
- $p(x)/q(x) \leq t$ if and only if $\phi_t(x) \leq 0$

quasiconvex optimization via convex feasibility problems

$$\phi_t(x) \leq 0, \quad f_i(x) \leq 0, \quad i = 1, \dots, m, \quad Ax = b \quad (1)$$

- for fixed t , a convex feasibility problem in x
- if feasible, we can conclude that $t \geq p^*$; if infeasible, $t \leq p^*$

Bisection method for quasiconvex optimization

given $l \leq p^*$, $u \geq p^*$, tolerance $\epsilon > 0$.

repeat

1. $t := (l + u)/2$.

2. Solve the convex feasibility problem (1).

3. **if** (1) is feasible, $u := t$; **else** $l := t$.

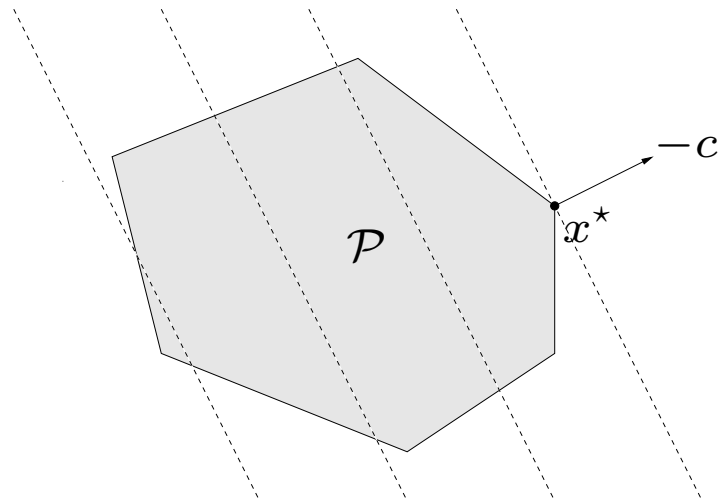
until $u - l \leq \epsilon$.

requires exactly $\lceil \log_2((u - l)/\epsilon) \rceil$ iterations (where u, l are initial values)

Linear program (LP)

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

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



Examples

diet problem: choose quantities x_1, \dots, x_n of n foods

- one unit of food j costs c_j , contains amount a_{ij} of nutrient i
- healthy diet requires nutrient i in quantity at least b_i

to find cheapest healthy diet,

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

piecewise-linear minimization

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

equivalent to an LP

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

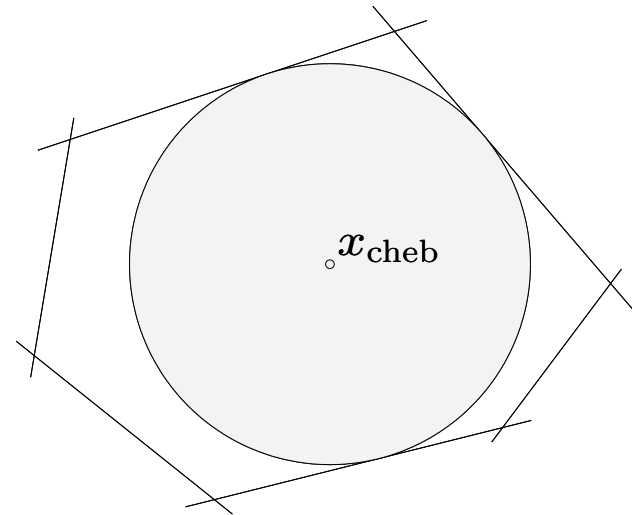
Chebyshev center of a polyhedron

Chebyshev center of

$$\mathcal{P} = \{x \mid a_i^T x \leq b_i, \ i = 1, \dots, m\}$$

is center of largest inscribed ball

$$\mathcal{B} = \{x_c + u \mid \|u\|_2 \leq r\}$$



- $a_i^T x \leq b_i$ for all $x \in \mathcal{B}$ if and only if

$$\sup\{a_i^T (x_c + u) \mid \|u\|_2 \leq r\} = a_i^T x_c + r\|a_i\|_2 \leq b_i$$

- hence, x_c, r can be determined by solving the LP

$$\begin{array}{ll} \text{maximize} & r \\ \text{subject to} & a_i^T x_c + r\|a_i\|_2 \leq b_i, \quad i = 1, \dots, m \end{array}$$

(Generalized) linear-fractional program

$$\begin{array}{ll}\text{minimize} & f_0(x) \\ \text{subject to} & Gx \preceq h \\ & Ax = b\end{array}$$

linear-fractional program

$$f_0(x) = \frac{c^T x + d}{e^T x + f}, \quad \text{dom } f_0(x) = \{x \mid e^T x + f > 0\}$$

- a quasiconvex optimization problem; can be solved by bisection
- also equivalent to the LP (variables y, z)

$$\begin{array}{ll}\text{minimize} & c^T y + dz \\ \text{subject to} & Gy \preceq hz \\ & Ay = bz \\ & e^T y + fz = 1 \\ & z \geq 0\end{array}$$

generalized linear-fractional program

$$f_0(x) = \max_{i=1,\dots,r} \frac{c_i^T x + d_i}{e_i^T x + f_i}, \quad \text{dom } f_0(x) = \{x \mid e_i^T x + f_i > 0, i = 1, \dots, r\}$$

a quasiconvex optimization problem; can be solved by bisection

example: Von Neumann model of a growing economy

$$\begin{array}{ll} \text{maximize (over } x, x^+) & \min_{i=1,\dots,n} x_i^+ / x_i \\ \text{subject to} & x^+ \succeq 0, \quad Bx^+ \preceq Ax \end{array}$$

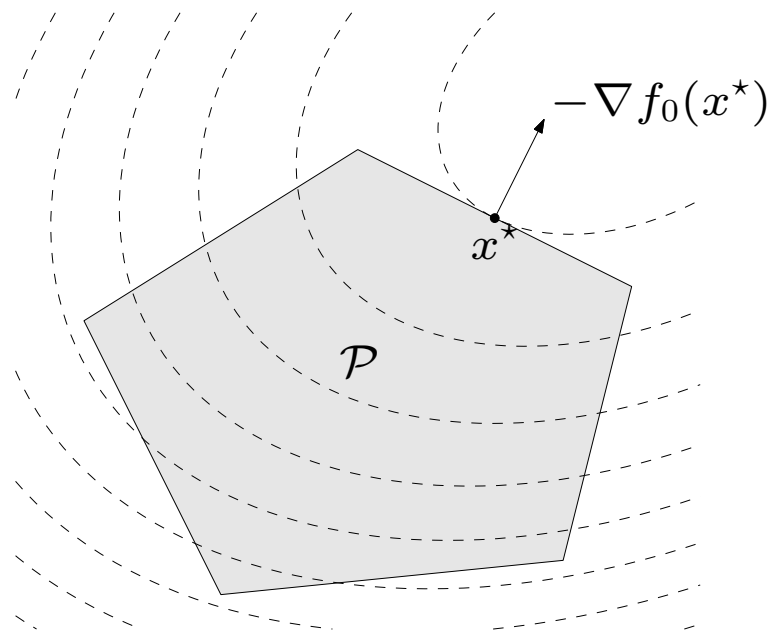
- $x, x^+ \in \mathbf{R}^n$: activity levels of n sectors, in current and next period
- $(Ax)_i, (Bx^+)_i$: produced, resp. consumed, amounts of good i
- x_i^+ / x_i : growth rate of sector i

allocate activity to maximize growth rate of slowest growing sector

Quadratic program (QP)

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

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



Examples

least-squares

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

- analytical solution $x^* = A^\dagger b$ (A^\dagger is pseudo-inverse)
- can add linear constraints, *e.g.*, $l \preceq x \preceq u$

linear program with random cost

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

- c is random vector with mean \bar{c} and covariance Σ
- hence, $c^T x$ is random variable with mean $\bar{c}^T x$ and variance $x^T \Sigma x$
- $\gamma > 0$ is risk aversion parameter; controls the trade-off between expected cost and variance (risk)

Quadratically constrained quadratic program (QCQP)

$$\begin{array}{ll}\text{minimize} & (1/2)x^T P_0 x + q_0^T x + r_0 \\ \text{subject to} & (1/2)x^T P_i x + q_i^T x + r_i \leq 0, \quad i = 1, \dots, m \\ & Ax = b\end{array}$$

- $P_i \in \mathbf{S}_+^n$; objective and constraints are convex quadratic
- if $P_1, \dots, P_m \in \mathbf{S}_{++}^n$, feasible region is intersection of m ellipsoids and an affine set

Second-order cone programming

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

$$(A_i \in \mathbf{R}^{n_i \times n}, F \in \mathbf{R}^{p \times n})$$

- inequalities are called second-order cone (SOC) constraints:

$$(A_i x + b_i, c_i^T x + d_i) \in \text{second-order cone in } \mathbf{R}^{n_i+1}$$

- for $n_i = 0$, reduces to an LP; if $c_i = 0$, reduces to a QCQP
- more general than QCQP and LP

Robust linear programming

the parameters in optimization problems are often uncertain, *e.g.*, in an LP

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

there can be uncertainty in c , a_i , b_i

two common approaches to handling uncertainty (in a_i , for simplicity)

- deterministic model: constraints must hold for all $a_i \in \mathcal{E}_i$

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

- stochastic model: a_i is random variable; constraints must hold with probability η

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

deterministic approach via SOCP

- choose an ellipsoid as \mathcal{E}_i :

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

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

- robust LP

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

is equivalent to the SOCP

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

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

stochastic approach via SOCP

- assume a_i is Gaussian with mean \bar{a}_i , covariance Σ_i ($a_i \sim \mathcal{N}(\bar{a}_i, \Sigma_i)$)
- $a_i^T x$ is Gaussian r.v. with mean $\bar{a}_i^T x$, variance $x^T \Sigma_i x$; hence

$$\mathbf{prob}(a_i^T x \leq b_i) = \Phi \left(\frac{b_i - \bar{a}_i^T x}{\|\Sigma_i^{1/2} x\|_2} \right)$$

where $\Phi(x) = (1/\sqrt{2\pi}) \int_{-\infty}^x e^{-t^2/2} dt$ is CDF of $\mathcal{N}(0, 1)$

- robust LP

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

with $\eta \geq 1/2$, is equivalent to the SOCP

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

Geometric programming

monomial function

$$f(x) = cx_1^{a_1}x_2^{a_2}\cdots x_n^{a_n}, \quad \text{dom } f = \mathbf{R}_{++}^n$$

with $c > 0$; exponent α_i can be any real number

posynomial function: sum of monomials

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

geometric program (GP)

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

with f_i posynomial, h_i monomial

Geometric program in convex form

change variables to $y_i = \log x_i$, and take logarithm of cost, constraints

- monomial $f(x) = cx_1^{a_1} \cdots x_n^{a_n}$ transforms to

$$\log f(e^{y_1}, \dots, e^{y_n}) = a^T y + b \quad (b = \log c)$$

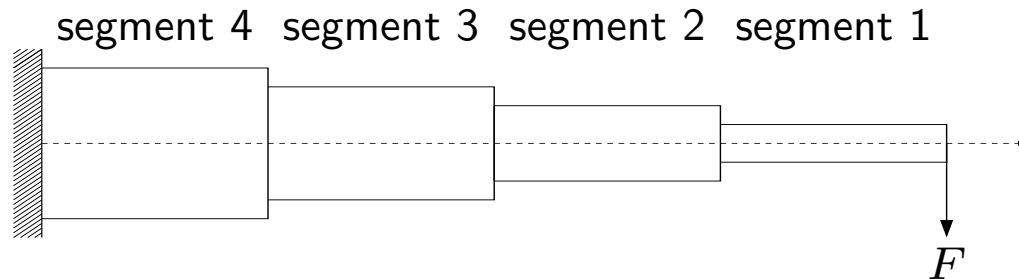
- posynomial $f(x) = \sum_{k=1}^K c_k x_1^{a_{1k}} x_2^{a_{2k}} \cdots x_n^{a_{nk}}$ transforms to

$$\log f(e^{y_1}, \dots, e^{y_n}) = \log \left(\sum_{k=1}^K e^{a_k^T y + b_k} \right) \quad (b_k = \log c_k)$$

- geometric program transforms to convex problem

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

Design of cantilever beam



- N segments with unit lengths, rectangular cross-sections of size $w_i \times h_i$
- given vertical force F applied at the right end

design problem

minimize total weight

subject to upper & lower bounds on w_i, h_i

upper bound & lower bounds on aspect ratios h_i/w_i

upper bound on stress in each segment

upper bound on vertical deflection at the end of the beam

variables: w_i, h_i for $i = 1, \dots, N$

objective and constraint functions

- total weight $w_1h_1 + \cdots + w_Nh_N$ is posynomial
- aspect ratio h_i/w_i and inverse aspect ratio w_i/h_i are monomials
- maximum stress in segment i is given by $6iF/(w_ih_i^2)$, a monomial
- the vertical deflection y_i and slope v_i of central axis at the right end of segment i are defined recursively as

$$v_i = 12(i - 1/2) \frac{F}{Ew_ih_i^3} + v_{i+1}$$
$$y_i = 6(i - 1/3) \frac{F}{Ew_ih_i^3} + v_{i+1} + y_{i+1}$$

for $i = N, N - 1, \dots, 1$, with $v_{N+1} = y_{N+1} = 0$ (E is Young's modulus)

v_i and y_i are posynomial functions of w, h

formulation as a GP

$$\begin{aligned} \text{minimize} \quad & w_1 h_1 + \cdots + w_N h_N \\ \text{subject to} \quad & w_{\max}^{-1} w_i \leq 1, \quad w_{\min} w_i^{-1} \leq 1, \quad i = 1, \dots, N \\ & h_{\max}^{-1} h_i \leq 1, \quad h_{\min} h_i^{-1} \leq 1, \quad i = 1, \dots, N \\ & S_{\max}^{-1} w_i^{-1} h_i \leq 1, \quad S_{\min} w_i h_i^{-1} \leq 1, \quad i = 1, \dots, N \\ & 6iF\sigma_{\max}^{-1} w_i^{-1} h_i^{-2} \leq 1, \quad i = 1, \dots, N \\ & y_{\max}^{-1} y_1 \leq 1 \end{aligned}$$

note

- we write $w_{\min} \leq w_i \leq w_{\max}$ and $h_{\min} \leq h_i \leq h_{\max}$

$$w_{\min}/w_i \leq 1, \quad w_i/w_{\max} \leq 1, \quad h_{\min}/h_i \leq 1, \quad h_i/h_{\max} \leq 1$$

- we write $S_{\min} \leq h_i/w_i \leq S_{\max}$ as

$$S_{\min} w_i / h_i \leq 1, \quad h_i / (w_i S_{\max}) \leq 1$$

Minimizing spectral radius of nonnegative matrix

Perron-Frobenius eigenvalue $\lambda_{\text{pf}}(A)$

- exists for (elementwise) positive $A \in \mathbf{R}^{n \times n}$
- a real, positive eigenvalue of A , equal to spectral radius $\max_i |\lambda_i(A)|$
- determines asymptotic growth (decay) rate of A^k : $A^k \sim \lambda_{\text{pf}}^k$ as $k \rightarrow \infty$
- alternative characterization: $\lambda_{\text{pf}}(A) = \inf\{\lambda \mid Av \preceq \lambda v \text{ for some } v \succ 0\}$

minimizing spectral radius of matrix of posynomials

- minimize $\lambda_{\text{pf}}(A(x))$, where the elements $A(x)_{ij}$ are posynomials of x
- equivalent geometric program:

$$\begin{array}{ll} \text{minimize} & \lambda \\ \text{subject to} & \sum_{j=1}^n A(x)_{ij} v_j / (\lambda v_i) \leq 1, \quad i = 1, \dots, n \end{array}$$

variables λ, v, x

Generalized inequality constraints

convex problem with generalized inequality constraints

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

- $f_0 : \mathbf{R}^n \rightarrow \mathbf{R}$ convex; $f_i : \mathbf{R}^n \rightarrow \mathbf{R}^{k_i}$ K_i -convex w.r.t. proper cone K_i
- same properties as standard convex problem (convex feasible set, local optimum is global, etc.)

conic form problem: special case with affine objective and constraints

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & Fx + g \preceq_K 0 \\ & Ax = b\end{array}$$

extends linear programming ($K = \mathbf{R}_+^m$) to nonpolyhedral cones

Semidefinite program (SDP)

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & x_1 F_1 + x_2 F_2 + \cdots + x_n F_n + G \preceq 0 \\ & Ax = b\end{array}$$

with $F_i, G \in \mathbf{S}^k$

- inequality constraint is called linear matrix inequality (LMI)
- includes problems with multiple LMI constraints: for example,

$$x_1 \hat{F}_1 + \cdots + x_n \hat{F}_n + \hat{G} \preceq 0, \quad x_1 \tilde{F}_1 + \cdots + x_n \tilde{F}_n + \tilde{G} \preceq 0$$

is equivalent to single LMI

$$x_1 \begin{bmatrix} \hat{F}_1 & 0 \\ 0 & \tilde{F}_1 \end{bmatrix} + x_2 \begin{bmatrix} \hat{F}_2 & 0 \\ 0 & \tilde{F}_2 \end{bmatrix} + \cdots + x_n \begin{bmatrix} \hat{F}_n & 0 \\ 0 & \tilde{F}_n \end{bmatrix} + \begin{bmatrix} \hat{G} & 0 \\ 0 & \tilde{G} \end{bmatrix} \preceq 0$$

LP and SOCP as SDP

LP and equivalent SDP

$$\begin{array}{ll} \text{LP:} & \begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \end{array} \end{array} \qquad \begin{array}{ll} \text{SDP:} & \begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \mathbf{diag}(Ax - b) \preceq 0 \end{array} \end{array}$$

(note different interpretation of generalized inequality \preceq)

SOCP and equivalent SDP

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

$$\begin{array}{ll} \text{SDP:} & \begin{array}{ll} \text{minimize} & f^T x \\ \text{subject to} & \begin{bmatrix} (c_i^T x + d_i)I & A_i x + b_i \\ (A_i x + b_i)^T & c_i^T x + d_i \end{bmatrix} \succeq 0, \quad i = 1, \dots, m \end{array} \end{array}$$

Eigenvalue minimization

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

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

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

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 $A_i \in \mathbf{S}^{p \times q}$)

equivalent SDP

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

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

$$\begin{aligned} \|A\|_2 \leq t & \iff A^T A \preceq t^2 I, \quad t \geq 0 \\ & \iff \begin{bmatrix} tI & A \\ A^T & tI \end{bmatrix} \succeq 0 \end{aligned}$$

Vector optimization

general vector optimization problem

$$\begin{array}{ll} \text{minimize (w.r.t. } K) & f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \dots, m \\ & h_i(x) \leq 0, \quad i = 1, \dots, p \end{array}$$

vector objective $f_0 : \mathbf{R}^n \rightarrow \mathbf{R}^q$, minimized w.r.t. proper cone $K \in \mathbf{R}^q$

convex vector optimization problem

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

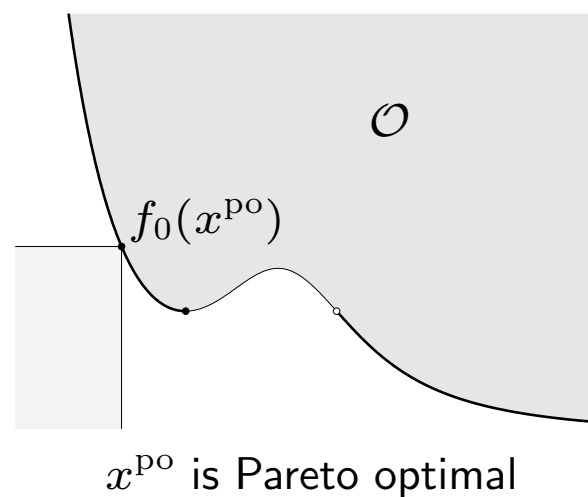
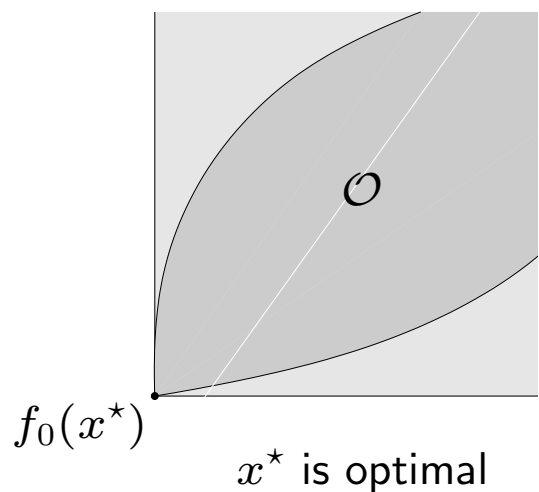
with f_0 K -convex, f_1, \dots, f_m convex

Optimal and Pareto optimal points

set of achievable objective values

$$\mathcal{O} = \{f_0(x) \mid x \text{ feasible}\}$$

- feasible x is **optimal** if $f_0(x)$ is a minimum value of \mathcal{O}
- feasible x is **Pareto optimal** if $f_0(x)$ is a minimal value of \mathcal{O}



Multicriterion optimization

vector optimization problem with $K = \mathbf{R}_+^q$

$$f_0(x) = (F_1(x), \dots, F_q(x))$$

- q different objectives F_i ; roughly speaking we want all F_i 's to be small
- feasible x^* is optimal if

$$y \text{ feasible} \implies f_0(x^*) \preceq f_0(y)$$

if there exists an optimal point, the objectives are noncompeting

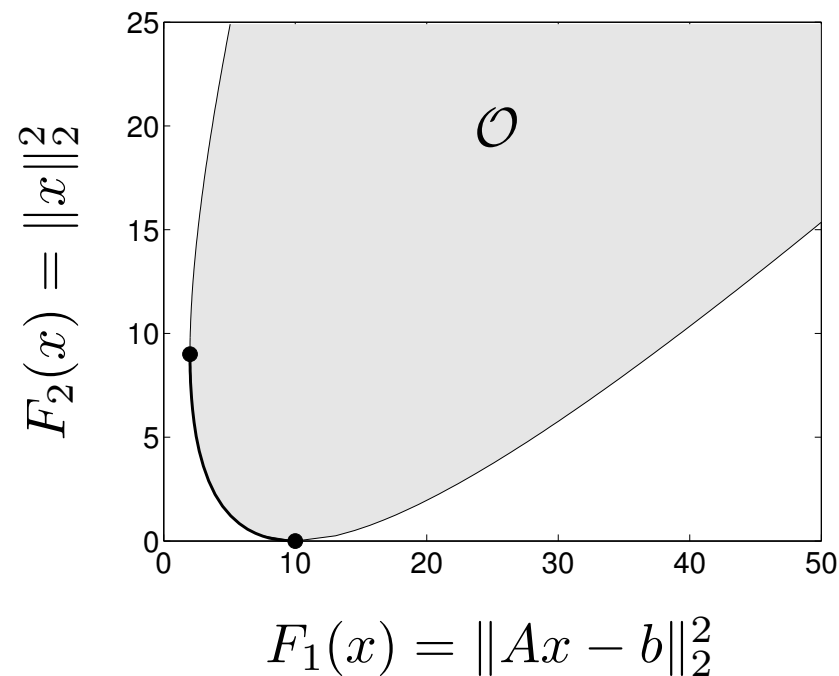
- feasible x^{po} is Pareto optimal if

$$y \text{ feasible, } f_0(y) \preceq f_0(x^{\text{po}}) \implies f_0(x^{\text{po}}) = f_0(y)$$

if there are multiple Pareto optimal values, there is a trade-off between the objectives

Regularized least-squares

$$\text{minimize (w.r.t. } \mathbf{R}_+^2) \quad (\|Ax - b\|_2^2, \|x\|_2^2)$$



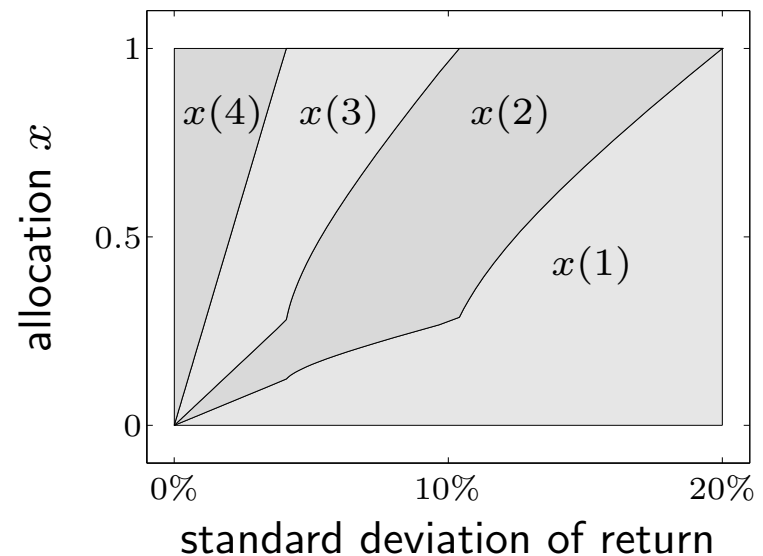
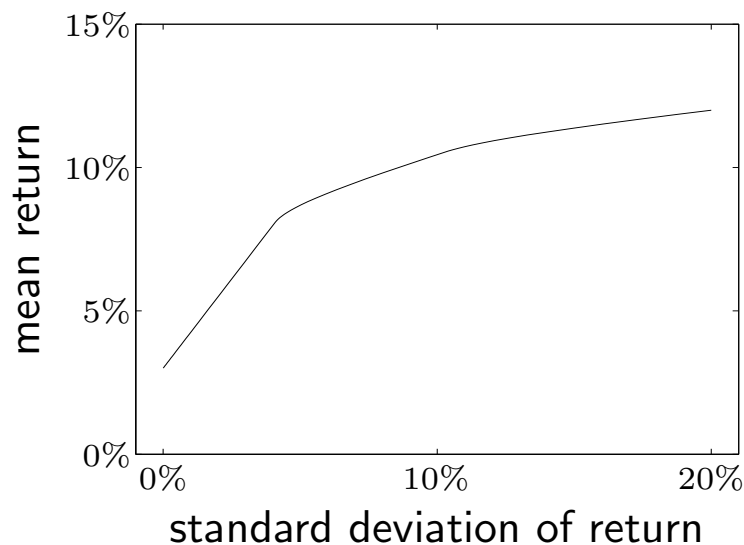
example for $A \in \mathbf{R}^{100 \times 10}$; heavy line is formed by Pareto optimal points

Risk return trade-off in portfolio optimization

$$\begin{array}{ll} \text{minimize (w.r.t. } \mathbf{R}_+^2) & (-\bar{p}^T x, x^T \Sigma x) \\ \text{subject to} & \mathbf{1}^T x = 1, \quad x \succeq 0 \end{array}$$

- $x \in \mathbf{R}^n$ is investment portfolio; x_i is fraction invested in asset i
- $p \in \mathbf{R}^n$ is vector of relative asset price changes; modeled as a random variable with mean \bar{p} , covariance Σ
- $\bar{p}^T x = \mathbf{E} r$ is expected return; $x^T \Sigma x = \mathbf{var} r$ is return variance

example

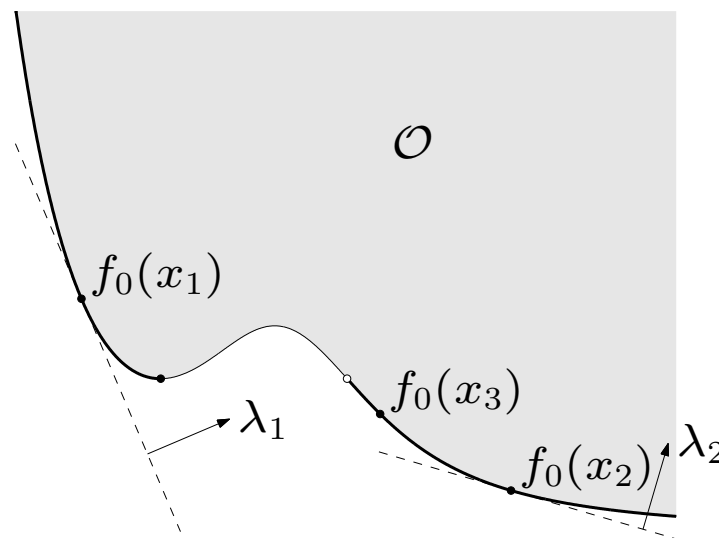


Scalarization

to find Pareto optimal points: choose $\lambda \succ_{K^*} 0$ and solve scalar problem

$$\begin{array}{ll} \text{minimize} & \lambda^T f_0(x) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \dots, m \\ & h_i(x) = 0, \quad i = 1, \dots, p \end{array}$$

if x is optimal for scalar problem,
then it is Pareto-optimal for vector
optimization problem



for convex vector optimization problems, can find (almost) all Pareto
optimal points by varying $\lambda \succ_{K^*} 0$

Scalarization for multicriterion problems

to find Pareto optimal points, minimize positive weighted sum

$$\lambda^T f_0(x) = \lambda_1 F_1(x) + \cdots + \lambda_q F_q(x)$$

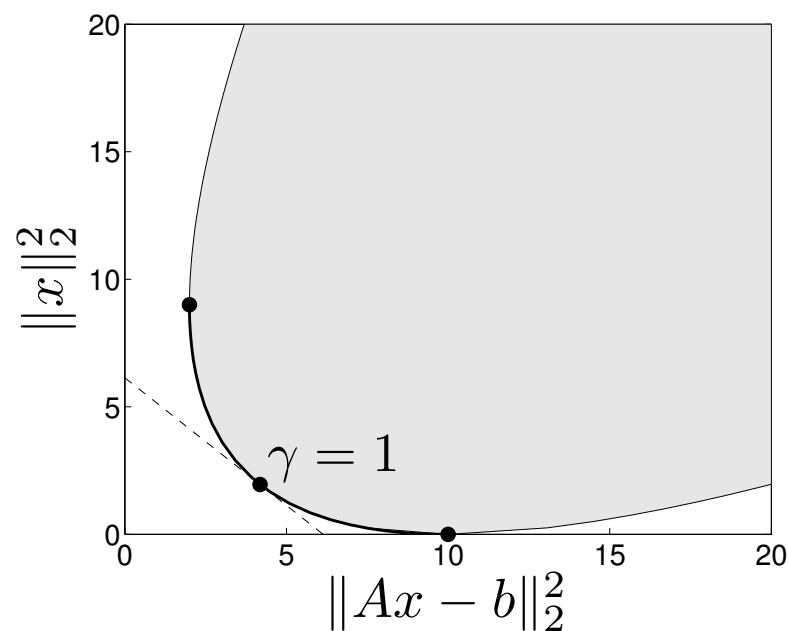
examples

- regularized least-squares problem of page 4–43

take $\lambda = (1, \gamma)$ with $\gamma > 0$

minimize $\|Ax - b\|_2^2 + \gamma \|x\|_2^2$

for fixed γ , a LS problem



- risk-return trade-off of page 4–44

$$\begin{array}{ll}\text{minimize} & -\bar{p}^T x + \gamma x^T \Sigma x \\ \text{subject to} & \mathbf{1}^T x = 1, \quad x \succeq 0\end{array}$$

for fixed $\gamma > 0$, a quadratic program

5. Duality

- Lagrange dual problem
- weak and strong duality
- geometric interpretation
- optimality conditions
- perturbation and sensitivity analysis
- examples
- generalized inequalities

Lagrangian

standard form problem (not necessarily convex)

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

variable $x \in \mathbf{R}^n$, domain \mathcal{D} , optimal value p^*

Lagrangian: $L : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^p \rightarrow \mathbf{R}$, with $\text{dom } L = \mathcal{D} \times \mathbf{R}^m \times \mathbf{R}^p$,

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \nu_i h_i(x)$$

- weighted sum of objective and constraint functions
- λ_i is Lagrange multiplier associated with $f_i(x) \leq 0$
- ν_i is Lagrange multiplier associated with $h_i(x) = 0$

Lagrange dual function

Lagrange dual function: $g : \mathbf{R}^m \times \mathbf{R}^p \rightarrow \mathbf{R}$,

$$\begin{aligned} g(\lambda, \nu) &= \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) \\ &= \inf_{x \in \mathcal{D}} \left(f_0(x) + \sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \nu_i h_i(x) \right) \end{aligned}$$

g is concave, can be $-\infty$ for some λ, ν

lower bound property: if $\lambda \succeq 0$, then $g(\lambda, \nu) \leq p^*$

proof: if \tilde{x} is feasible and $\lambda \succeq 0$, then

$$f_0(\tilde{x}) \geq L(\tilde{x}, \lambda, \nu) \geq \inf_{x \in \mathcal{D}} L(x, \lambda, \nu) = g(\lambda, \nu)$$

minimizing over all feasible \tilde{x} gives $p^* \geq g(\lambda, \nu)$

Least-norm solution of linear equations

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

dual function

- Lagrangian is $L(x, \nu) = x^T x + \nu^T (Ax - b)$
- to minimize L over x , set gradient equal to zero:

$$\nabla_x L(x, \nu) = 2x + A^T \nu = 0 \quad \implies \quad x = -(1/2)A^T \nu$$

- plug in in L to obtain g :

$$g(\nu) = L((-1/2)A^T \nu, \nu) = -\frac{1}{4}\nu^T A A^T \nu - b^T \nu$$

a concave function of ν

lower bound property: $p^* \geq -(1/4)\nu^T A A^T \nu - b^T \nu$ for all ν

Standard form LP

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

dual function

- Lagrangian is

$$\begin{aligned}L(x, \lambda, \nu) &= c^T x + \nu^T (Ax - b) - \lambda^T x \\ &= -b^T \nu + (c + A^T \nu - \lambda)^T x\end{aligned}$$

- L is linear in x , hence

$$g(\lambda, \nu) = \inf_x L(x, \lambda, \nu) = \begin{cases} -b^T \nu & A^T \nu - \lambda + c = 0 \\ -\infty & \text{otherwise} \end{cases}$$

g is linear on affine domain $\{(\lambda, \nu) \mid A^T \nu - \lambda + c = 0\}$, hence concave

lower bound property: $p^* \geq -b^T \nu$ if $A^T \nu + c \succeq 0$

Equality constrained norm minimization

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

dual function

$$g(\nu) = \inf_x (\|x\| - \nu^T Ax + b^T \nu) = \begin{cases} b^T \nu & \|A^T \nu\|_* \leq 1 \\ -\infty & \text{otherwise} \end{cases}$$

where $\|v\|_* = \sup_{\|u\| \leq 1} u^T v$ is dual norm of $\|\cdot\|$

proof: follows from $\inf_x (\|x\| - y^T x) = 0$ if $\|y\|_* \leq 1$, $-\infty$ otherwise

- if $\|y\|_* \leq 1$, then $\|x\| - y^T x \geq 0$ for all x , with equality if $x = 0$
- if $\|y\|_* > 1$, choose $x = tu$ where $\|u\| \leq 1$, $u^T y = \|y\|_* > 1$:

$$\|x\| - y^T x = t(\|u\| - \|y\|_*) \rightarrow -\infty \quad \text{as } t \rightarrow \infty$$

lower bound property: $p^* \geq b^T \nu$ if $\|A^T \nu\|_* \leq 1$

Two-way partitioning

$$\begin{array}{ll}\text{minimize} & x^T W x \\ \text{subject to} & x_i^2 = 1, \quad i = 1, \dots, n\end{array}$$

- a nonconvex problem; feasible set contains 2^n discrete points
- interpretation: partition $\{1, \dots, n\}$ in two sets; W_{ij} is cost of assigning i, j to the same set; $-W_{ij}$ is cost of assigning to different sets

dual function

$$\begin{aligned} g(\nu) &= \inf_x (x^T W x + \sum_i \nu_i (x_i^2 - 1)) = \inf_x x^T (W + \mathbf{diag}(\nu)) x - \mathbf{1}^T \nu \\ &= \begin{cases} -\mathbf{1}^T \nu & W + \mathbf{diag}(\nu) \succeq 0 \\ -\infty & \text{otherwise} \end{cases} \end{aligned}$$

lower bound property: $p^* \geq -\mathbf{1}^T \nu$ if $W + \mathbf{diag}(\nu) \succeq 0$

example: $\nu = -\lambda_{\min}(W)\mathbf{1}$ gives bound $p^* \geq n\lambda_{\min}(W)$

Lagrange dual and conjugate function

$$\begin{array}{ll}\text{minimize} & f_0(x) \\ \text{subject to} & Ax \preceq b, \quad Cx = d\end{array}$$

dual function

$$\begin{aligned}g(\lambda, \nu) &= \inf_{x \in \text{dom } f_0} (f_0(x) + (A^T \lambda + C^T \nu)^T x - b^T \lambda - d^T \nu) \\ &= -f_0^*(-A^T \lambda - C^T \nu) - b^T \lambda - d^T \nu\end{aligned}$$

- recall definition of conjugate $f^*(y) = \sup_{x \in \text{dom } f} (y^T x - f(x))$
- simplifies derivation of dual if conjugate of f_0 is known

example: entropy maximization

$$f_0(x) = \sum_{i=1}^n x_i \log x_i, \quad f_0^*(y) = \sum_{i=1}^n e^{y_i - 1}$$

The dual problem

Lagrange dual problem

$$\begin{array}{ll}\text{maximize} & g(\lambda, \nu) \\ \text{subject to} & \lambda \succeq 0\end{array}$$

- finds best lower bound on p^* , obtained from Lagrange dual function
- a convex optimization problem; optimal value denoted d^*
- λ, ν are dual feasible if $\lambda \succeq 0, (\lambda, \nu) \in \mathbf{dom} g$
- often simplified by making implicit constraint $(\lambda, \nu) \in \mathbf{dom} g$ explicit

example: standard form LP and its dual (page 5–5)

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

$$\begin{array}{ll}\text{maximize} & -b^T \nu \\ \text{subject to} & A^T \nu + c \succeq 0\end{array}$$

Weak and strong duality

weak duality: $d^* \leq p^*$

- always holds (for convex and nonconvex problems)
- can be used to find nontrivial lower bounds for difficult problems
for example, solving the SDP

$$\begin{array}{ll} \text{maximize} & -\mathbf{1}^T \nu \\ \text{subject to} & W + \mathbf{diag}(\nu) \succeq 0 \end{array}$$

gives a lower bound for the two-way partitioning problem on page 5–7

strong duality: $d^* = p^*$

- does not hold in general
- (usually) holds for convex problems
- conditions that guarantee strong duality in convex problems are called **constraint qualifications**

Slater's constraint qualification

strong duality holds for a convex problem

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

if it is strictly feasible, *i.e.*,

$$\exists x \in \mathbf{int} \mathcal{D} : \quad f_i(x) < 0, \quad i = 1, \dots, m, \quad Ax = b$$

- also guarantees that the dual optimum is attained (if $p^* > -\infty$)
- can be sharpened: *e.g.*, can replace $\mathbf{int} \mathcal{D}$ with $\mathbf{relint} \mathcal{D}$ (interior relative to affine hull); linear inequalities do not need to hold with strict inequality, . . .
- there exist many other types of constraint qualifications

Inequality form LP

primal problem

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

dual function

$$g(\lambda) = \inf_x ((c + A^T \lambda)^T x - b^T \lambda) = \begin{cases} -b^T \lambda & A^T \lambda + c = 0 \\ -\infty & \text{otherwise} \end{cases}$$

dual problem

$$\begin{array}{ll}\text{maximize} & -b^T \lambda \\ \text{subject to} & A^T \lambda + c = 0, \quad \lambda \succeq 0\end{array}$$

- from Slater's condition: $p^* = d^*$ if $A\tilde{x} \prec b$ for some \tilde{x}
- in fact, $p^* = d^*$ except when primal and dual are infeasible

Quadratic program

primal problem (assume $P \in \mathbf{S}_{++}^n$)

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

dual function

$$g(\lambda) = \inf_x (x^T P x + \lambda^T (Ax - b)) = -\frac{1}{4} \lambda^T A P^{-1} A^T \lambda - b^T \lambda$$

dual problem

$$\begin{array}{ll}\text{maximize} & -(1/4) \lambda^T A P^{-1} A^T \lambda - b^T \lambda \\ \text{subject to} & \lambda \succeq 0\end{array}$$

- from Slater's condition: $p^* = d^*$ if $A\tilde{x} \prec b$ for some \tilde{x}
- in fact, $p^* = d^*$ always

A nonconvex problem with strong duality

$$\begin{array}{ll}\text{minimize} & x^T A x + 2b^T x \\ \text{subject to} & x^T x \leq 1\end{array}$$

$A \not\succeq 0$, hence nonconvex

dual function: $g(\lambda) = \inf_x (x^T (A + \lambda I)x + 2b^T x - \lambda)$

- unbounded below if $A + \lambda I \not\succeq 0$ or if $A + \lambda I \succeq 0$ and $b \notin \mathcal{R}(A + \lambda I)$
- minimized by $x = -(A + \lambda I)^\dagger b$ otherwise: $g(\lambda) = -b^T (A + \lambda I)^\dagger b - \lambda$

dual problem and equivalent SDP:

$$\begin{array}{ll}\text{maximize} & -b^T (A + \lambda I)^\dagger b - \lambda \\ \text{subject to} & A + \lambda I \succeq 0 \\ & b \in \mathcal{R}(A + \lambda I)\end{array}$$

$$\begin{array}{ll}\text{maximize} & -t - \lambda \\ \text{subject to} & \begin{bmatrix} A + \lambda I & b \\ b^T & t \end{bmatrix} \succeq 0\end{array}$$

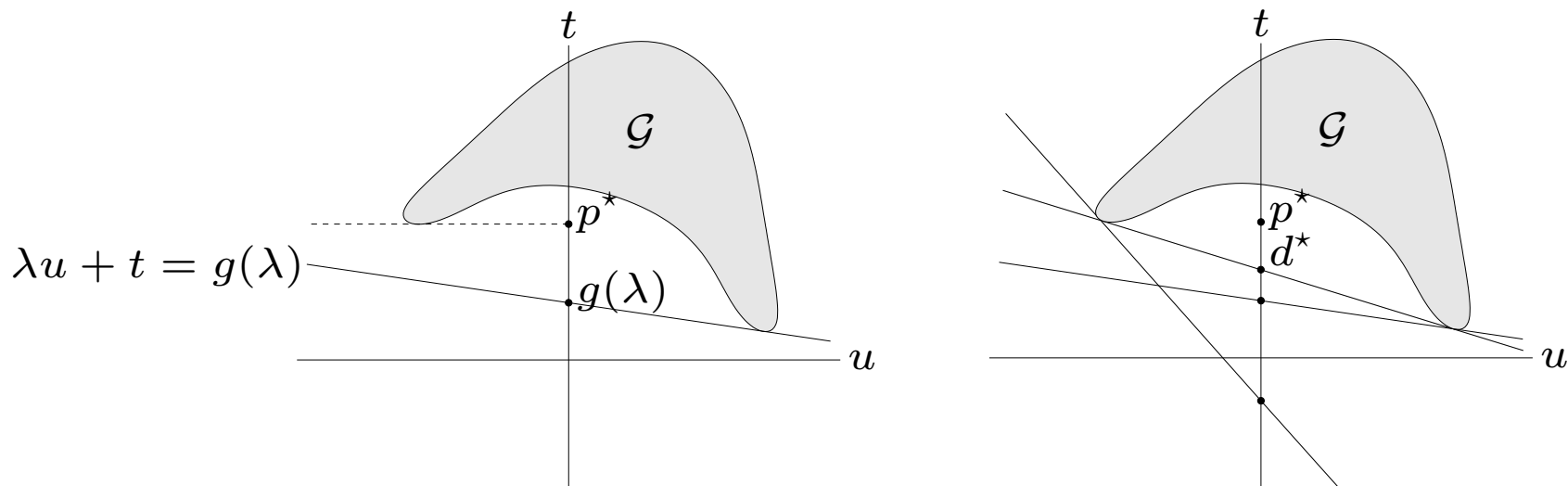
strong duality although primal problem is not convex (not easy to show)

Geometric interpretation

for simplicity, consider problem with one constraint $f_1(x) \leq 0$

interpretation of dual function:

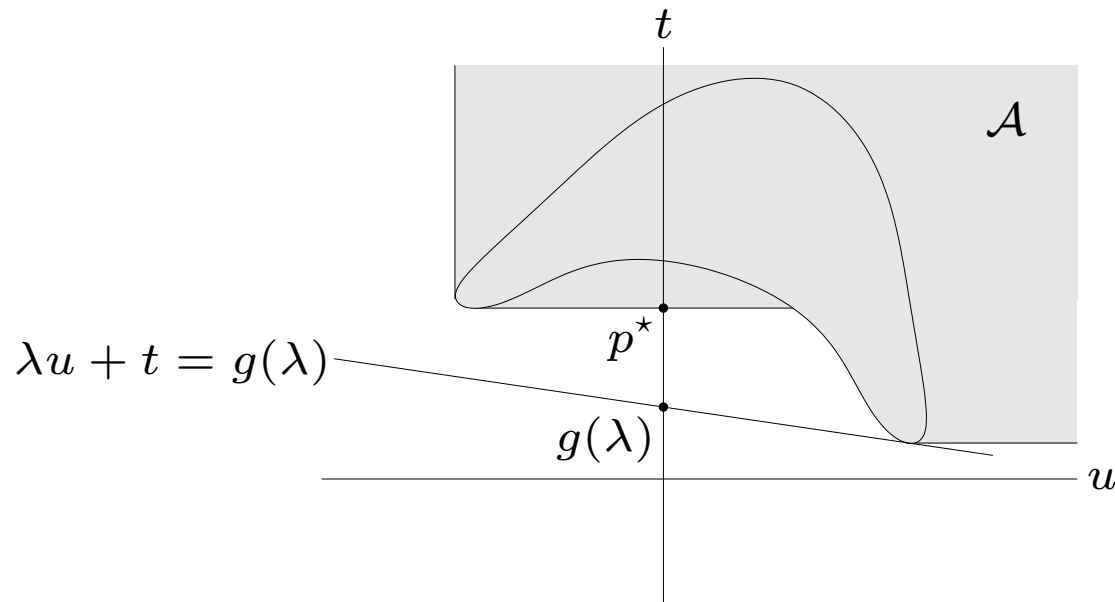
$$g(\lambda) = \inf_{(u,t) \in \mathcal{G}} (t + \lambda u), \quad \text{where } \mathcal{G} = \{(f_1(x), f_0(x)) \mid x \in \mathcal{D}\}$$



- $\lambda u + t = g(\lambda)$ is (non-vertical) supporting hyperplane to \mathcal{G}
- hyperplane intersects t -axis at $t = g(\lambda)$

epigraph variation: same interpretation if \mathcal{G} is replaced with

$$\mathcal{A} = \{(u, t) \mid f_1(x) \leq u, f_0(x) \leq t \text{ for some } x \in \mathcal{D}\}$$



strong duality

- holds if there is a non-vertical supporting hyperplane to \mathcal{A} at $(0, p^*)$
- for convex problem, \mathcal{A} is convex, hence has supp. hyperplane at $(0, p^*)$
- Slater's condition: if there exist $(\tilde{u}, \tilde{t}) \in \mathcal{A}$ with $\tilde{u} < 0$, then supporting hyperplanes at $(0, p^*)$ must be non-vertical

Complementary slackness

assume strong duality holds, x^* is primal optimal, (λ^*, ν^*) is dual optimal

$$\begin{aligned} f_0(x^*) = g(\lambda^*, \nu^*) &= \inf_x \left(f_0(x) + \sum_{i=1}^m \lambda_i^* f_i(x) + \sum_{i=1}^p \nu_i^* h_i(x) \right) \\ &\leq f_0(x^*) + \sum_{i=1}^m \lambda_i^* f_i(x^*) + \sum_{i=1}^p \nu_i^* h_i(x^*) \\ &\leq f_0(x^*) \end{aligned}$$

hence, the two inequalities hold with equality

- x^* minimizes $L(x, \lambda^*, \nu^*)$
- $\lambda_i^* f_i(x^*) = 0$ for $i = 1, \dots, m$ (known as complementary slackness):

$$\lambda_i^* > 0 \implies f_i(x^*) = 0, \quad f_i(x^*) < 0 \implies \lambda_i^* = 0$$

Karush-Kuhn-Tucker (KKT) conditions

the following four conditions are called KKT conditions (for a problem with differentiable f_i, h_i):

1. primal constraints: $f_i(x) \leq 0, i = 1, \dots, m, h_i(x) = 0, i = 1, \dots, p$
2. dual constraints: $\lambda \succeq 0$
3. complementary slackness: $\lambda_i f_i(x) = 0, i = 1, \dots, m$
4. gradient of Lagrangian with respect to x vanishes:

$$\nabla f_0(x) + \sum_{i=1}^m \lambda_i \nabla f_i(x) + \sum_{i=1}^p \nu_i \nabla h_i(x) = 0$$

from page 5–17: if strong duality holds and x, λ, ν are optimal, then they must satisfy the KKT conditions

KKT conditions for convex problem

if \tilde{x} , $\tilde{\lambda}$, $\tilde{\nu}$ satisfy KKT for a convex problem, then they are optimal:

- from complementary slackness: $f_0(\tilde{x}) = L(\tilde{x}, \tilde{\lambda}, \tilde{\nu})$
- from 4th condition (and convexity): $g(\tilde{\lambda}, \tilde{\nu}) = L(\tilde{x}, \tilde{\lambda}, \tilde{\nu})$

hence, $f_0(\tilde{x}) = g(\tilde{\lambda}, \tilde{\nu})$

if **Slater's condition** is satisfied:

x is optimal if and only if there exist λ , ν that satisfy KKT conditions

- recall that Slater implies strong duality, and dual optimum is attained
- generalizes optimality condition $\nabla f_0(x) = 0$ for unconstrained problem

example: water-filling (assume $\alpha_i > 0$)

$$\begin{array}{ll}\text{minimize} & -\sum_{i=1}^n \log(x_i + \alpha_i) \\ \text{subject to} & x \succeq 0, \quad \mathbf{1}^T x = 1\end{array}$$

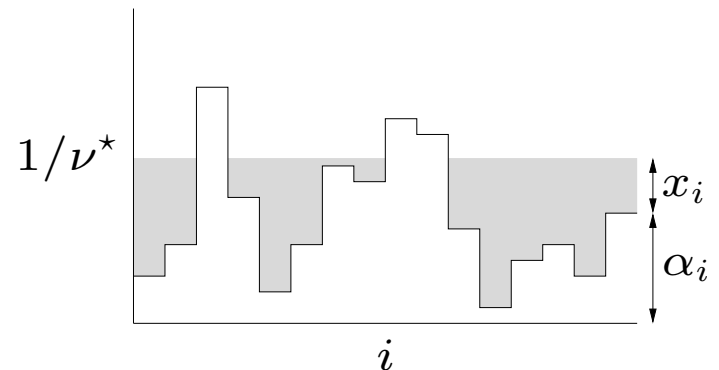
x is optimal iff $x \succeq 0$, $\mathbf{1}^T x = 1$, and there exist $\lambda \in \mathbf{R}^n$, $\nu \in \mathbf{R}$ such that

$$\lambda \succeq 0, \quad \lambda_i x_i = 0, \quad \frac{1}{x_i + \alpha_i} + \lambda_i = \nu$$

- if $\nu < 1/\alpha_i$: $\lambda_i = 0$ and $x_i = 1/\nu - \alpha_i$
- if $\nu \geq 1/\alpha_i$: $\lambda_i = \nu - 1/\alpha_i$ and $x_i = 0$
- determine ν from $\mathbf{1}^T x = \sum_{i=1}^n \max\{0, 1/\nu - \alpha_i\} = 1$

interpretation

- n patches; level of patch i is at height α_i
- flood area with unit amount of water
- resulting level is $1/\nu^*$



Perturbation and sensitivity analysis

(unperturbed) optimization problem and its dual

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

$$\begin{array}{ll}\text{maximize} & g(\lambda, \nu) \\ \text{subject to} & \lambda \succeq 0\end{array}$$

perturbed problem and its dual

$$\begin{array}{ll}\text{min.} & f_0(x) \\ \text{s.t.} & f_i(x) \leq u_i, \quad i = 1, \dots, m \\ & h_i(x) = v_i, \quad i = 1, \dots, p\end{array}$$

$$\begin{array}{ll}\text{max.} & g(\lambda, \nu) - u^T \lambda - v^T \nu \\ \text{s.t.} & \lambda \succeq 0\end{array}$$

- x is primal variable; u, v are parameters
- $p^*(u, v)$ is optimal value as a function of u, v
- we are interested in information about $p^*(u, v)$ that we can obtain from the solution of the unperturbed problem and its dual

global sensitivity result

assume strong duality holds for unperturbed problem, and that λ^*, ν^* are dual optimal for unperturbed problem

apply weak duality to perturbed problem:

$$\begin{aligned} p^*(u, v) &\geq g(\lambda^*, \nu^*) - u^T \lambda^* - v^T \nu^* \\ &= p^*(0, 0) - u^T \lambda^* - v^T \nu^* \end{aligned}$$

sensitivity interpretation

- if λ_i^* large: p^* increases greatly if we tighten constraint i ($u_i < 0$)
- if λ_i^* small: p^* does not decrease much if we loosen constraint i ($u_i > 0$)
- if ν_i^* large and positive: p^* increases greatly if we take $v_i < 0$;
if ν_i^* large and negative: p^* increases greatly if we take $v_i > 0$
- if ν_i^* small and positive: p^* does not decrease much if we take $v_i > 0$;
if ν_i^* small and negative: p^* does not decrease much if we take $v_i < 0$

local sensitivity: if (in addition) $p^*(u, v)$ is differentiable at $(0, 0)$, then

$$\lambda_i^* = -\frac{\partial p^*(0, 0)}{\partial u_i}, \quad \nu_i^* = -\frac{\partial p^*(0, 0)}{\partial v_i}$$

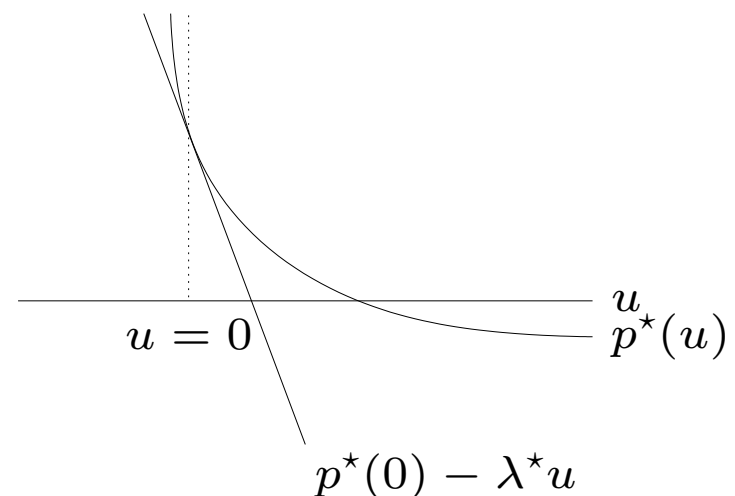
proof (for λ_i^*): from global sensitivity result,

$$\frac{\partial p^*(0, 0)}{\partial u_i} = \lim_{t \searrow 0} \frac{p^*(te_i, 0) - p^*(0, 0)}{t} \geq -\lambda_i^*$$

$$\frac{\partial p^*(0, 0)}{\partial u_i} = \lim_{t \nearrow 0} \frac{p^*(te_i, 0) - p^*(0, 0)}{t} \leq -\lambda_i^*$$

hence, equality

$p^*(u)$ for a problem with one (inequality) constraint:



Duality and problem reformulations

- equivalent formulations of a problem can lead to very different duals
- reformulating the primal problem can be useful when the dual is difficult to derive, or uninteresting

common reformulations

- introduce new variables and equality constraints
- make explicit constraints implicit or vice-versa
- transform objective or constraint functions
e.g., replace $f_0(x)$ by $\phi(f_0(x))$ with ϕ convex, increasing

Introducing new variables and equality constraints

$$\text{minimize } f_0(Ax + b)$$

- dual function is constant: $g = \inf_x L(x) = \inf_x f_0(Ax + b) = p^*$
- we have strong duality, but dual is quite useless

reformulated problem and its dual

$$\begin{array}{ll} \text{minimize} & f_0(y) \\ \text{subject to} & Ax + b - y = 0 \end{array}$$

$$\begin{array}{ll} \text{maximize} & b^T \nu - f_0^*(\nu) \\ \text{subject to} & A^T \nu = 0 \end{array}$$

dual function follows from

$$\begin{aligned} g(\nu) &= \inf_{x,y} (f_0(y) - \nu^T y + \nu^T Ax + b^T \nu) \\ &= \begin{cases} -f_0^*(\nu) + b^T \nu & A^T \nu = 0 \\ -\infty & \text{otherwise} \end{cases} \end{aligned}$$

norm approximation problem: minimize $\|Ax - b\|$

$$\begin{array}{ll}\text{minimize} & \|y\| \\ \text{subject to} & y = Ax - b\end{array}$$

can look up conjugate of $\|\cdot\|$, or derive dual directly

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

(see page 5–4)

dual of norm approximation problem

$$\begin{array}{ll}\text{maximize} & b^T \nu \\ \text{subject to} & A^T \nu = 0, \quad \|\nu\|_* \leq 1\end{array}$$

Implicit constraints

LP with box constraints: primal and dual problem

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & -\mathbf{1} \preceq x \preceq \mathbf{1} \end{array} \qquad \begin{array}{ll} \text{maximize} & -b^T \nu - \mathbf{1}^T \lambda_1 - \mathbf{1}^T \lambda_2 \\ \text{subject to} & c + A^T \nu + \lambda_1 - \lambda_2 = 0 \\ & \lambda_1 \succeq 0, \quad \lambda_2 \succeq 0 \end{array}$$

reformulation with box constraints made implicit

$$\begin{array}{ll} \text{minimize} & f_0(x) = \begin{cases} c^T x & -\mathbf{1} \preceq x \preceq \mathbf{1} \\ \infty & \text{otherwise} \end{cases} \\ \text{subject to} & Ax = b \end{array}$$

dual function

$$\begin{aligned} g(\nu) &= \inf_{-\mathbf{1} \preceq x \preceq \mathbf{1}} (c^T x + \nu^T (Ax - b)) \\ &= -b^T \nu - \|A^T \nu + c\|_1 \end{aligned}$$

dual problem: maximize $-b^T \nu - \|A^T \nu + c\|_1$

Problems with generalized inequalities

$$\begin{array}{ll}\text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \preceq_{K_i} 0, \quad i = 1, \dots, m \\ & h_i(x) = 0, \quad i = 1, \dots, p\end{array}$$

\preceq_{K_i} is generalized inequality on \mathbf{R}^{k_i}

definitions are parallel to scalar case:

- Lagrange multiplier for $f_i(x) \preceq_{K_i} 0$ is vector $\lambda_i \in \mathbf{R}^{k_i}$
- Lagrangian $L : \mathbf{R}^n \times \mathbf{R}^{k_1} \times \dots \times \mathbf{R}^{k_m} \times \mathbf{R}^p \rightarrow \mathbf{R}$, is defined as

$$L(x, \lambda_1, \dots, \lambda_m, \nu) = f_0(x) + \sum_{i=1}^m \lambda_i^T f_i(x) + \sum_{i=1}^p \nu_i h_i(x)$$

- dual function $g : \mathbf{R}^{k_1} \times \dots \times \mathbf{R}^{k_m} \times \mathbf{R}^p \rightarrow \mathbf{R}$, is defined as

$$g(\lambda_1, \dots, \lambda_m, \nu) = \inf_{x \in \mathcal{D}} L(x, \lambda_1, \dots, \lambda_m, \nu)$$

lower bound property: if $\lambda_i \succeq_{K_i^*} 0$, then $g(\lambda_1, \dots, \lambda_m, \nu) \leq p^*$

proof: if \tilde{x} is feasible and $\lambda \succeq_{K_i^*} 0$, then

$$\begin{aligned} f_0(\tilde{x}) &\geq f_0(\tilde{x}) + \sum_{i=1}^m \lambda_i^T f_i(\tilde{x}) + \sum_{i=1}^p \nu_i h_i(\tilde{x}) \\ &\geq \inf_{x \in \mathcal{D}} L(x, \lambda_1, \dots, \lambda_m, \nu) \\ &= g(\lambda_1, \dots, \lambda_m, \nu) \end{aligned}$$

minimizing over all feasible \tilde{x} gives $p^* \geq g(\lambda_1, \dots, \lambda_m, \nu)$

dual problem

$$\begin{aligned} &\text{maximize} && g(\lambda_1, \dots, \lambda_m, \nu) \\ &\text{subject to} && \lambda_i \succeq_{K_i^*} 0, \quad i = 1, \dots, m \end{aligned}$$

- weak duality: $p^* \geq d^*$ always
- strong duality: $p^* = d^*$ for convex problem with constraint qualification (for example, Slater's: primal problem is strictly feasible)

Semidefinite program

primal SDP ($F_i, G \in \mathbf{S}^k$)

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & x_1 F_1 + \cdots + x_n F_n \preceq G\end{array}$$

- Lagrange multiplier is matrix $Z \in \mathbf{S}^k$
- Lagrangian $L(x, Z) = c^T x + \mathbf{tr}(Z(x_1 F_1 + \cdots + x_n F_n - G))$
- dual function

$$g(Z) = \inf_x L(x, Z) = \begin{cases} -\mathbf{tr}(GZ) & \mathbf{tr}(F_i Z) + c_i = 0, \quad i = 1, \dots, n \\ -\infty & \text{otherwise} \end{cases}$$

dual SDP

$$\begin{array}{ll}\text{maximize} & -\mathbf{tr}(GZ) \\ \text{subject to} & Z \succeq 0, \quad \mathbf{tr}(F_i Z) + c_i = 0, \quad i = 1, \dots, n\end{array}$$

$p^* = d^*$ if primal SDP is strictly feasible ($\exists x$ with $x_1 F_1 + \cdots + x_n F_n \prec G$)

6. Approximation and fitting

- norm approximation
- least-norm problems
- regularized approximation
- robust approximation

Norm approximation

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

($A \in \mathbf{R}^{m \times n}$ with $m \geq n$, $\|\cdot\|$ is a norm on \mathbf{R}^m)

interpretations of solution $x^* = \operatorname{argmin}_x \|Ax - b\|$:

- **geometric:** Ax^* is point in $\mathcal{R}(A)$ closest to b
- **estimation:** linear measurement model

$$y = Ax + v$$

y are measurements, x is unknown, v is measurement error

given $y = b$, best guess of x is x^*

- **optimal design:** x are design variables (input), Ax is result (output)
 x^* is design that best approximates desired result b

examples

- least-squares approximation ($\|\cdot\|_2$): solution satisfies normal equations

$$A^T A x = A^T b$$

$$(x^* = (A^T A)^{-1} A^T b \text{ if } \mathbf{rank} A = n)$$

- Chebyshev approximation ($\|\cdot\|_\infty$): can be solved as an LP

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & -t\mathbf{1} \preceq Ax - b \preceq t\mathbf{1} \end{array}$$

- sum of absolute residuals approximation ($\|\cdot\|_1$): can be solved as an LP

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T y \\ \text{subject to} & -y \preceq Ax - b \preceq y \end{array}$$

Penalty function approximation

$$\begin{array}{ll} \text{minimize} & \phi(r_1) + \cdots + \phi(r_m) \\ \text{subject to} & r = Ax - b \end{array}$$

($A \in \mathbf{R}^{m \times n}$, $\phi : \mathbf{R} \rightarrow \mathbf{R}$ is a convex penalty function)

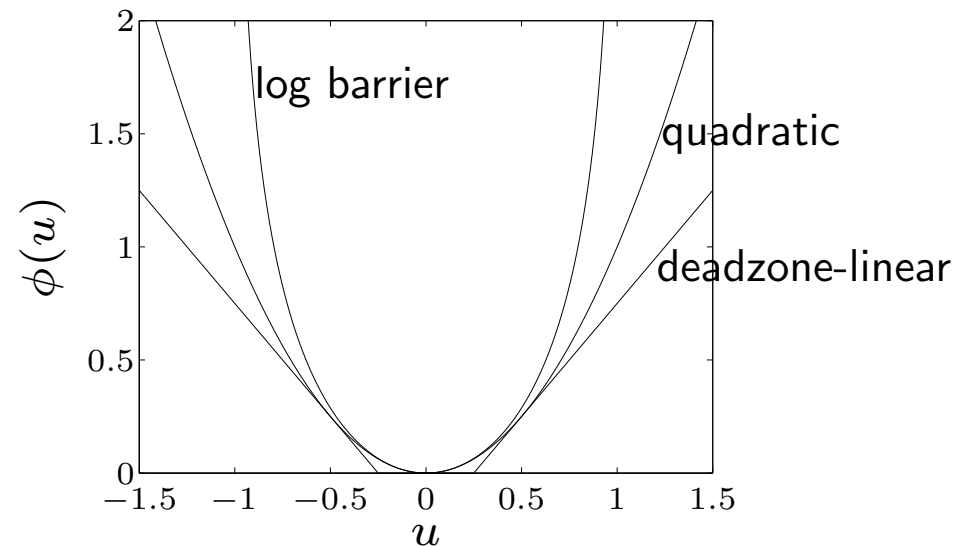
examples

- quadratic: $\phi(u) = u^2$
- deadzone-linear with width a :

$$\phi(u) = \max\{0, |u| - a\}$$

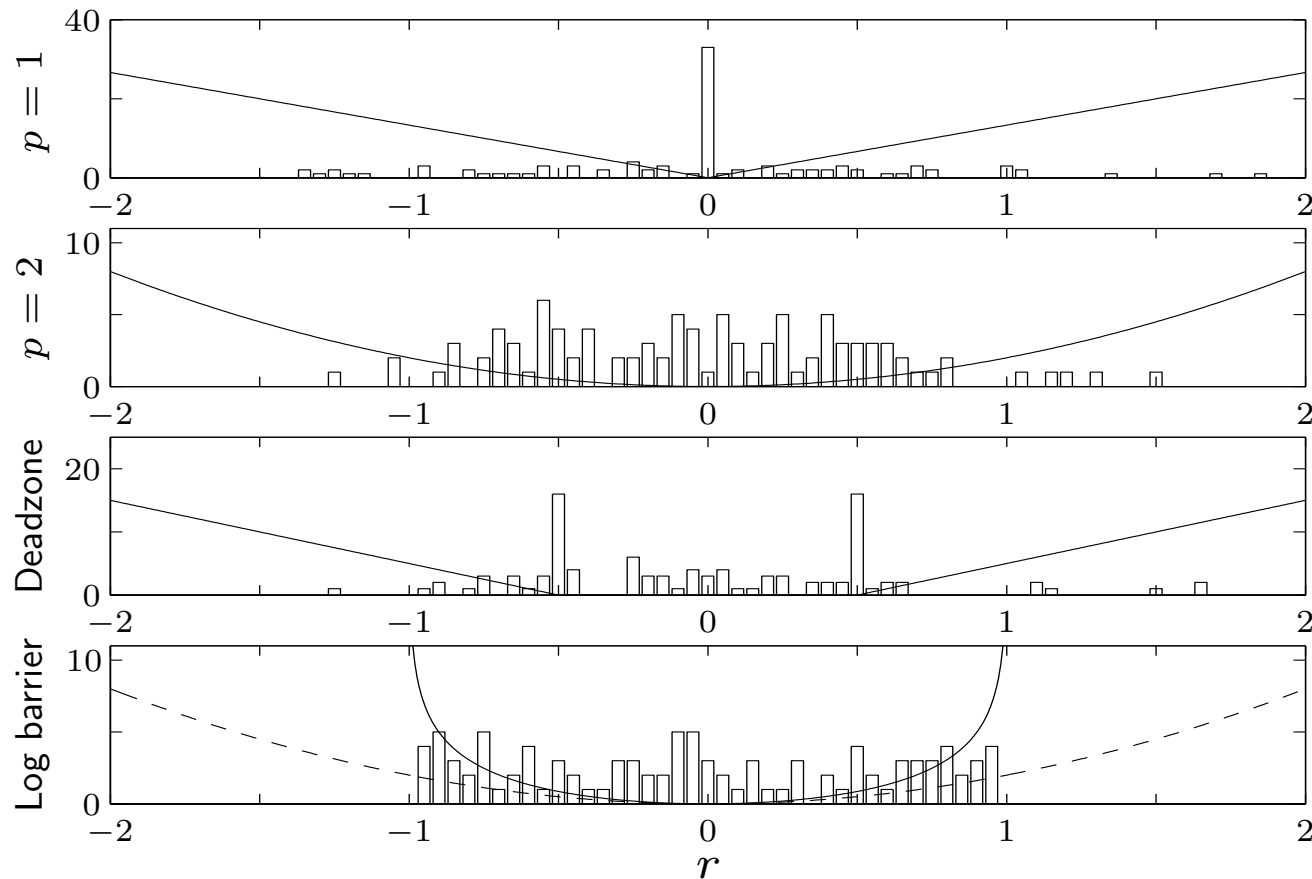
- log-barrier with limit a :

$$\phi(u) = \begin{cases} -a^2 \log(1 - (u/a)^2) & |u| < a \\ \infty & \text{otherwise} \end{cases}$$



example ($m = 100, n = 30$): histogram of residuals for penalties

$$\phi(u) = |u|, \quad \phi(u) = u^2, \quad \phi(u) = \max\{0, |u| - a\}, \quad \phi(u) = -\log(1 - u^2)$$

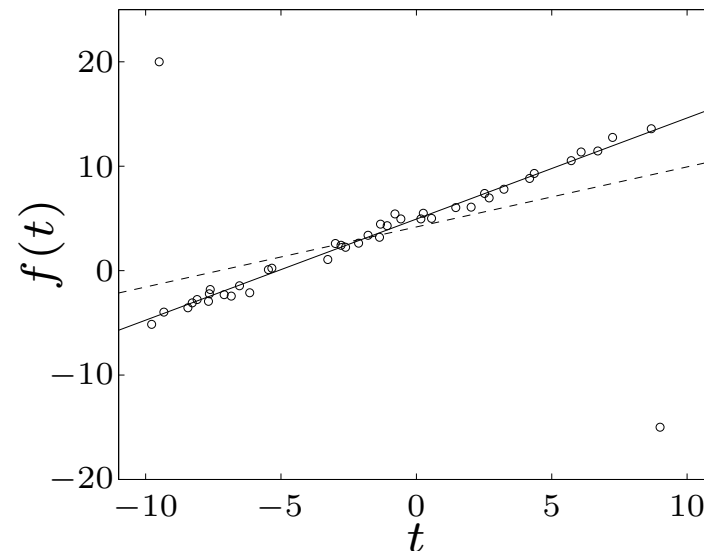
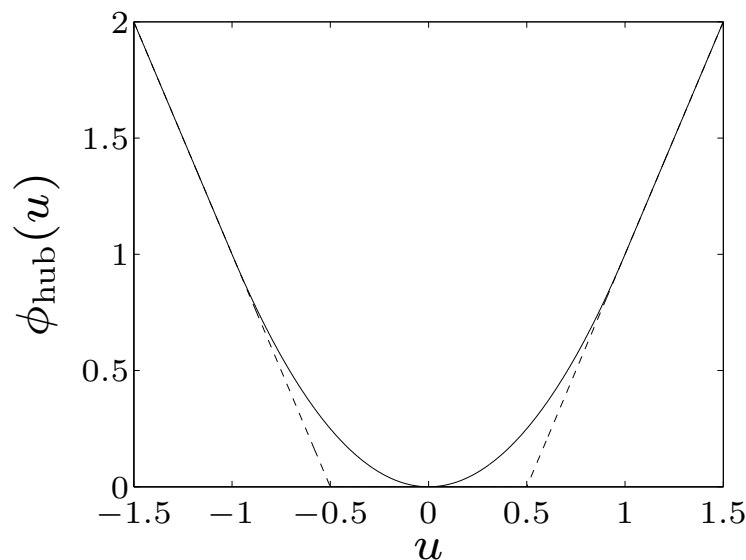


shape of penalty function has large effect on distribution of residuals

Huber penalty function (with parameter M)

$$\phi_{\text{hub}}(u) = \begin{cases} u^2 & |u| \leq M \\ M(2|u| - M) & |u| > M \end{cases}$$

linear growth for large u makes approximation less sensitive to outliers



- left: Huber penalty for $M = 1$
- right: affine function $f(t) = \alpha + \beta t$ fitted to 42 points t_i, y_i (circles) using quadratic (dashed) and Huber (solid) penalty

Least-norm problems

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

($A \in \mathbf{R}^{m \times n}$ with $m \leq n$, $\|\cdot\|$ is a norm on \mathbf{R}^n)

interpretations of solution $x^* = \operatorname{argmin}_{Ax=b} \|x\|$:

- **geometric:** x^* is point in affine set $\{x \mid Ax = b\}$ with minimum distance to 0
- **estimation:** $b = Ax$ are (perfect) measurements of x ; x^* is smallest ('most plausible') estimate consistent with measurements
- **design:** x are design variables (inputs); b are required results (outputs)
 x^* is smallest ('most efficient') design that satisfies requirements

examples

- least-squares solution of linear equations ($\|\cdot\|_2$):
can be solved via optimality conditions

$$2x + A^T \nu = 0, \quad Ax = b$$

- minimum sum of absolute values ($\|\cdot\|_1$): can be solved as an LP

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T y \\ \text{subject to} & -y \preceq x \preceq y, \quad Ax = b \end{array}$$

tends to produce sparse solution x^\star

extension: least-penalty problem

$$\begin{array}{ll} \text{minimize} & \phi(x_1) + \cdots + \phi(x_n) \\ \text{subject to} & Ax = b \end{array}$$

$\phi : \mathbf{R} \rightarrow \mathbf{R}$ is convex penalty function

Regularized approximation

$$\text{minimize (w.r.t. } \mathbf{R}_+^2 \text{)} \quad (\|Ax - b\|, \|x\|)$$

$A \in \mathbf{R}^{m \times n}$, norms on \mathbf{R}^m and \mathbf{R}^n can be different

interpretation: find good approximation $Ax \approx b$ with small x

- **estimation:** linear measurement model $y = Ax + v$, with prior knowledge that $\|x\|$ is small
- **optimal design:** small x is cheaper or more efficient, or the linear model $y = Ax$ is only valid for small x
- **robust approximation:** good approximation $Ax \approx b$ with small x is less sensitive to errors in A than good approximation with large x

Scalarized problem

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

- solution for $\gamma > 0$ traces out optimal trade-off curve
- other common method: minimize $\|Ax - b\|^2 + \delta \|x\|^2$ with $\delta > 0$

Tikhonov regularization

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

can be solved as a least-squares problem

$$\text{minimize} \quad \left\| \begin{bmatrix} A \\ \sqrt{\delta} I \end{bmatrix} x - \begin{bmatrix} b \\ 0 \end{bmatrix} \right\|_2^2$$

$$\text{solution } x^* = (A^T A + \delta I)^{-1} A^T b$$

Optimal input design

linear dynamical system with impulse response h :

$$y(t) = \sum_{\tau=0}^t h(\tau)u(t-\tau), \quad t = 0, 1, \dots, N$$

input design problem: multicriterion problem with 3 objectives

1. tracking error with desired output y_{des} : $J_{\text{track}} = \sum_{t=0}^N (y(t) - y_{\text{des}}(t))^2$
2. input magnitude: $J_{\text{mag}} = \sum_{t=0}^N u(t)^2$
3. input variation: $J_{\text{der}} = \sum_{t=0}^{N-1} (u(t+1) - u(t))^2$

track desired output using a small and slowly varying input signal

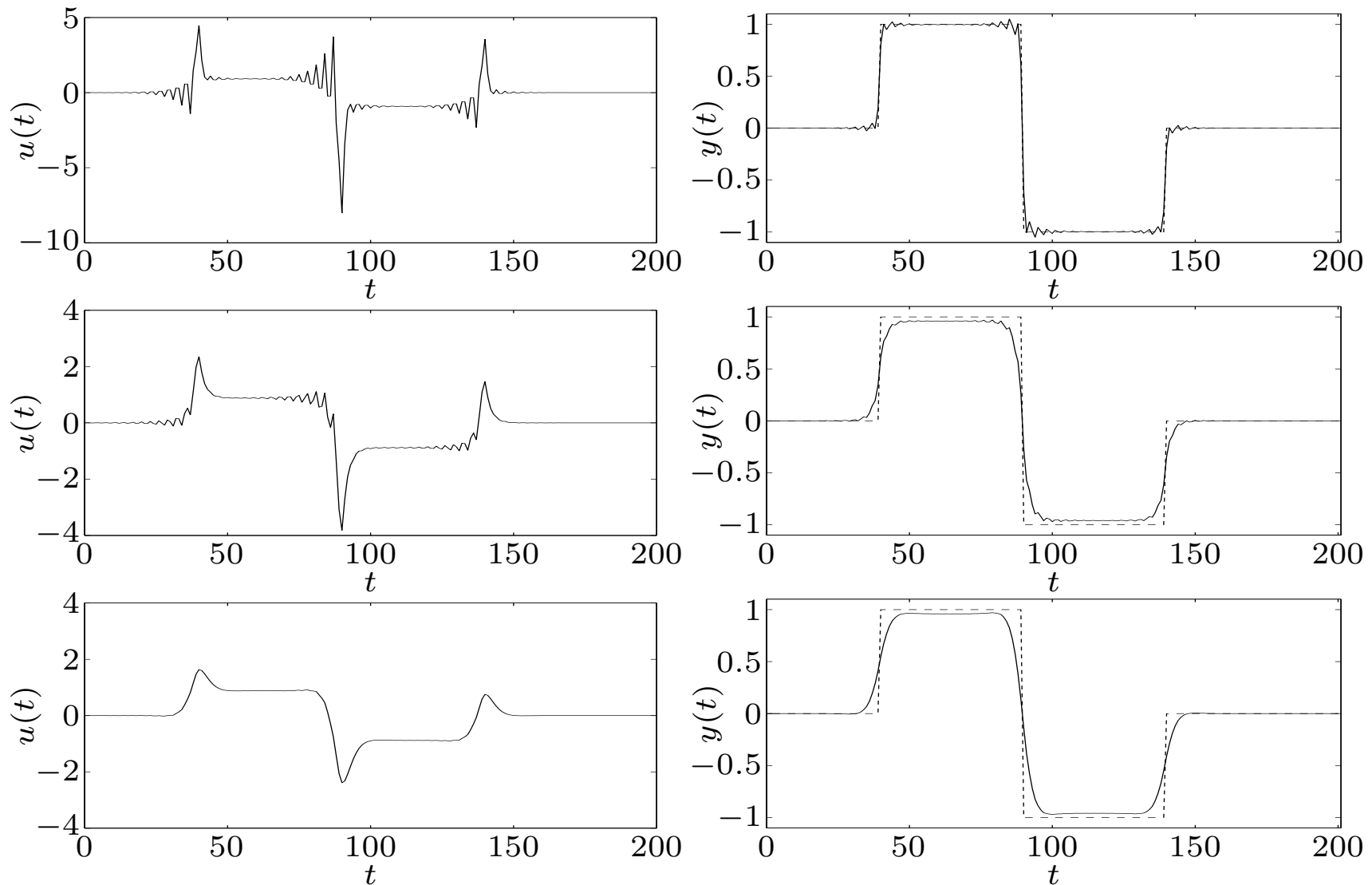
regularized least-squares formulation

$$\text{minimize} \quad J_{\text{track}} + \delta J_{\text{der}} + \eta J_{\text{mag}}$$

for fixed δ, η , a least-squares problem in $u(0), \dots, u(N)$

example: 3 solutions on optimal trade-off curve

(top) $\delta = 0$, small η ; (middle) $\delta = 0$, larger η ; (bottom) large δ



Signal reconstruction

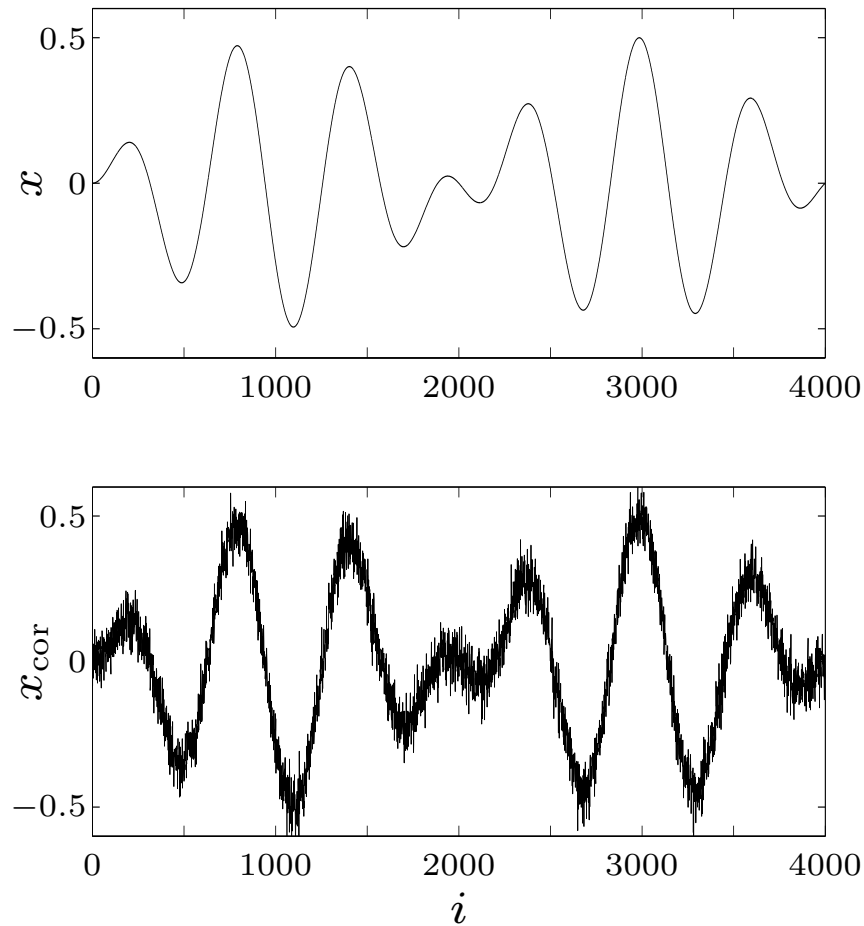
$$\text{minimize (w.r.t. } \mathbf{R}_+^2) \quad (\|\hat{x} - x_{\text{cor}}\|_2, \phi(\hat{x}))$$

- $x \in \mathbf{R}^n$ is unknown signal
- $x_{\text{cor}} = x + v$ is (known) corrupted version of x , with additive noise v
- variable \hat{x} (reconstructed signal) is estimate of x
- $\phi : \mathbf{R}^n \rightarrow \mathbf{R}$ is regularization function or smoothing objective

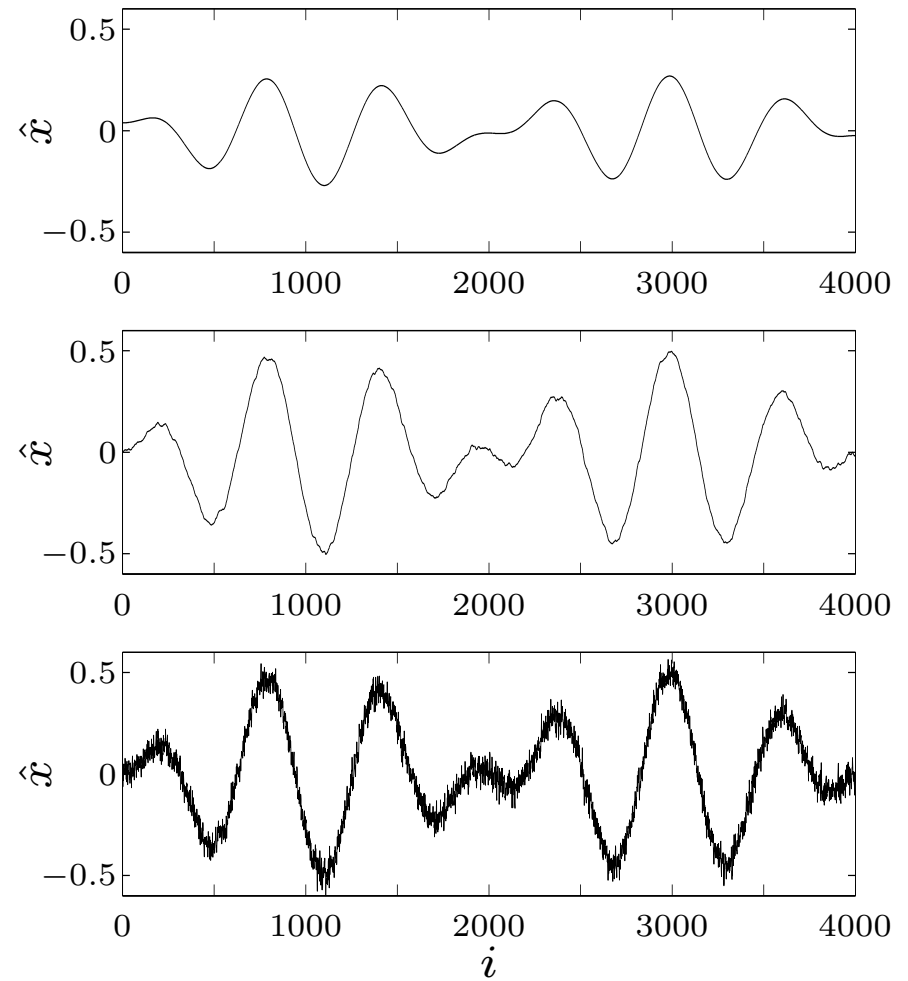
examples: quadratic smoothing, total variation smoothing:

$$\phi_{\text{quad}}(\hat{x}) = \sum_{i=1}^{n-1} (\hat{x}_{i+1} - \hat{x}_i)^2, \quad \phi_{\text{tv}}(\hat{x}) = \sum_{i=1}^{n-1} |\hat{x}_{i+1} - \hat{x}_i|$$

quadratic smoothing example

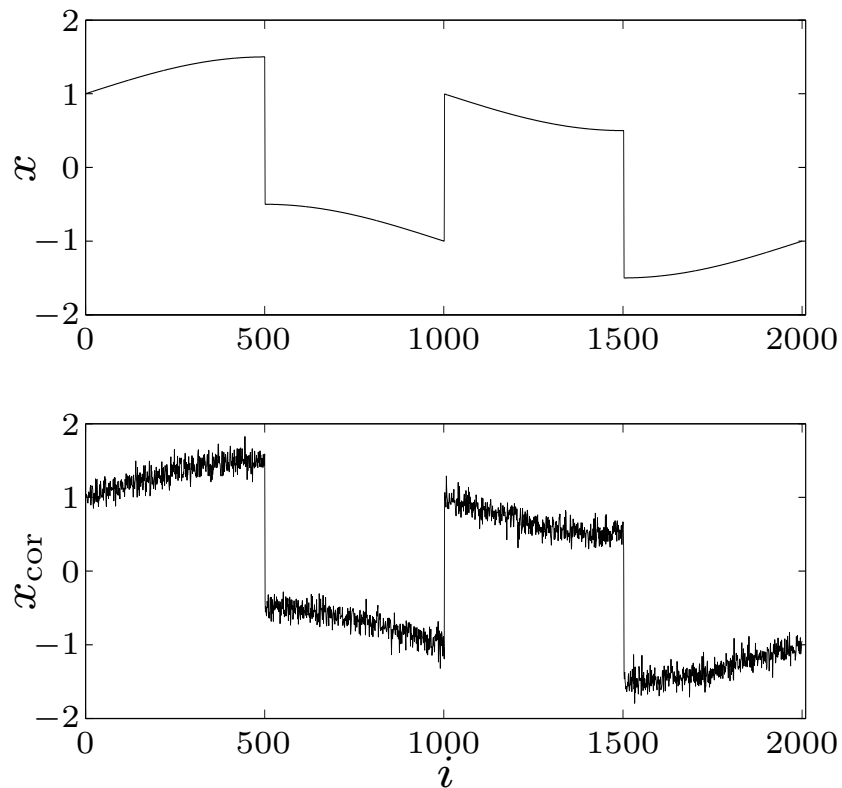


original signal x and noisy
signal x_{cor}

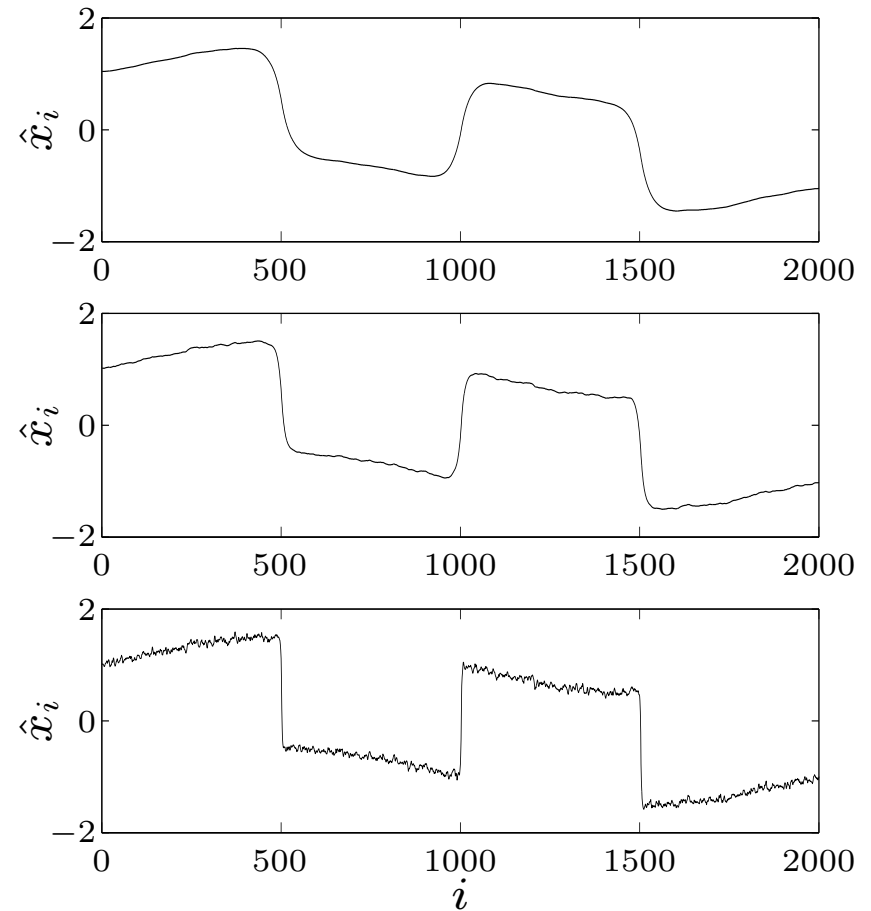


three solutions on trade-off curve
 $\|\hat{x} - x_{\text{cor}}\|_2$ versus $\phi_{\text{quad}}(\hat{x})$

total variation reconstruction example

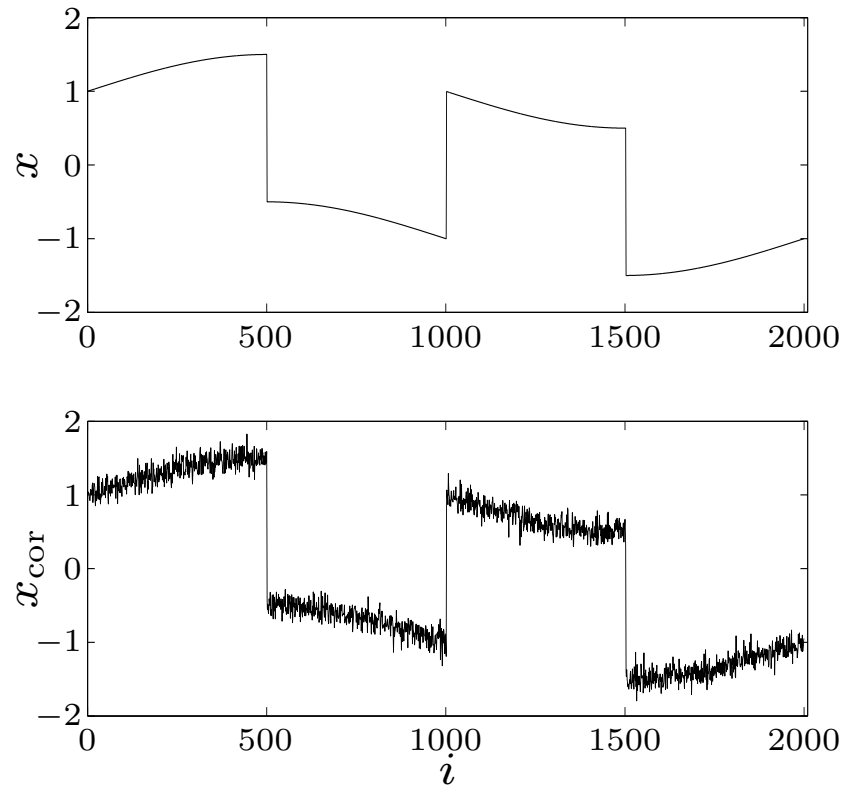


original signal x and noisy
signal x_{cor}

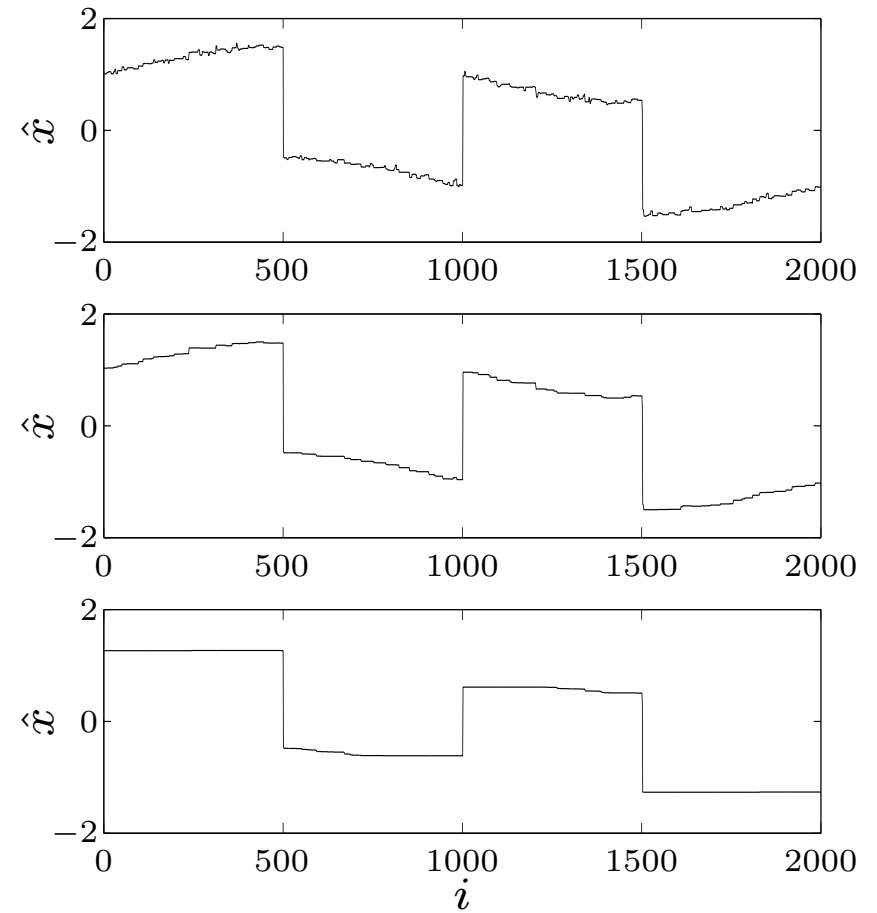


three solutions on trade-off curve
 $\|\hat{x} - x_{\text{cor}}\|_2$ versus $\phi_{\text{quad}}(\hat{x})$

quadratic smoothing smooths out noise **and** sharp transitions in signal



original signal x and noisy
signal x_{cor}



three solutions on trade-off curve
 $\|\hat{x} - x_{\text{cor}}\|_2$ versus $\phi_{\text{tv}}(\hat{x})$

total variation smoothing preserves sharp transitions in signal

Robust approximation

minimize $\|Ax - b\|$ with uncertain A

two approaches:

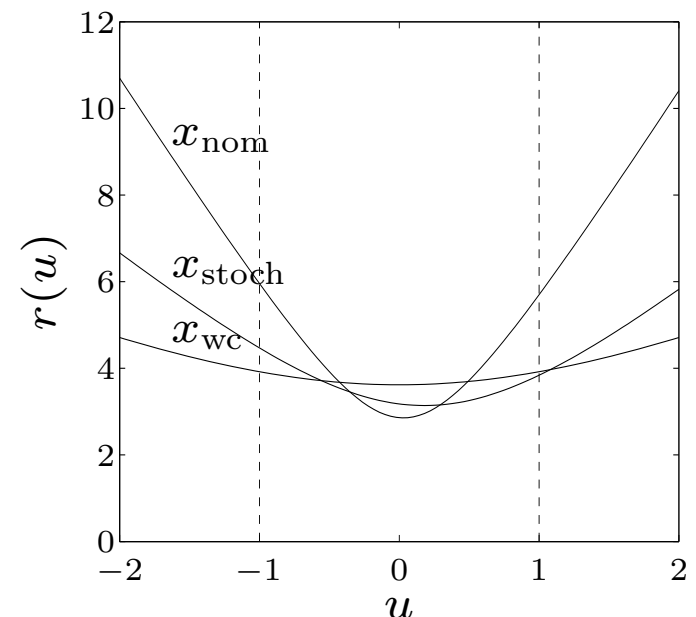
- **stochastic**: assume A is random, minimize $\mathbf{E} \|Ax - b\|$
- **worst-case**: set \mathcal{A} of possible values of A , minimize $\sup_{A \in \mathcal{A}} \|Ax - b\|$

tractable only in special cases (certain norms $\|\cdot\|$, distributions, sets \mathcal{A})

example: $A(u) = A_0 + uA_1$

- x_{nom} minimizes $\|A_0x - b\|_2^2$
- x_{stoch} minimizes $\mathbf{E} \|A(u)x - b\|_2^2$
with u uniform on $[-1, 1]$
- x_{wc} minimizes $\sup_{-1 \leq u \leq 1} \|A(u)x - b\|_2^2$

figure shows $r(u) = \|A(u)x - b\|_2$



stochastic robust LS with $A = \bar{A} + U$, U random, $\mathbf{E} U = 0$, $\mathbf{E} U^T U = P$

$$\text{minimize } \mathbf{E} \|(\bar{A} + U)x - b\|_2^2$$

- explicit expression for objective:

$$\begin{aligned} \mathbf{E} \|Ax - b\|_2^2 &= \mathbf{E} \|\bar{A}x - b + Ux\|_2^2 \\ &= \|\bar{A}x - b\|_2^2 + \mathbf{E} x^T U^T U x \\ &= \|\bar{A}x - b\|_2^2 + x^T P x \end{aligned}$$

- hence, robust LS problem is equivalent to LS problem

$$\text{minimize } \|\bar{A}x - b\|_2^2 + \|P^{1/2}x\|_2^2$$

- for $P = \delta I$, get Tikhonov regularized problem

$$\text{minimize } \|\bar{A}x - b\|_2^2 + \delta \|x\|_2^2$$

worst-case robust LS with $\mathcal{A} = \{\bar{A} + u_1 A_1 + \cdots + u_p A_p \mid \|u\|_2 \leq 1\}$

$$\text{minimize} \quad \sup_{A \in \mathcal{A}} \|Ax - b\|_2^2 = \sup_{\|u\|_2 \leq 1} \|P(x)u + q(x)\|_2^2$$

where $P(x) = \begin{bmatrix} A_1 x & A_2 x & \cdots & A_p x \end{bmatrix}$, $q(x) = \bar{A}x - b$

- from page 5–14, strong duality holds between the following problems

$$\begin{array}{ll} \text{maximize} & \|Pu + q\|_2^2 \\ \text{subject to} & \|u\|_2^2 \leq 1 \end{array} \qquad \begin{array}{ll} \text{minimize} & t + \lambda \\ \text{subject to} & \begin{bmatrix} I & P & q \\ P^T & \lambda I & 0 \\ q^T & 0 & t \end{bmatrix} \succeq 0 \end{array}$$

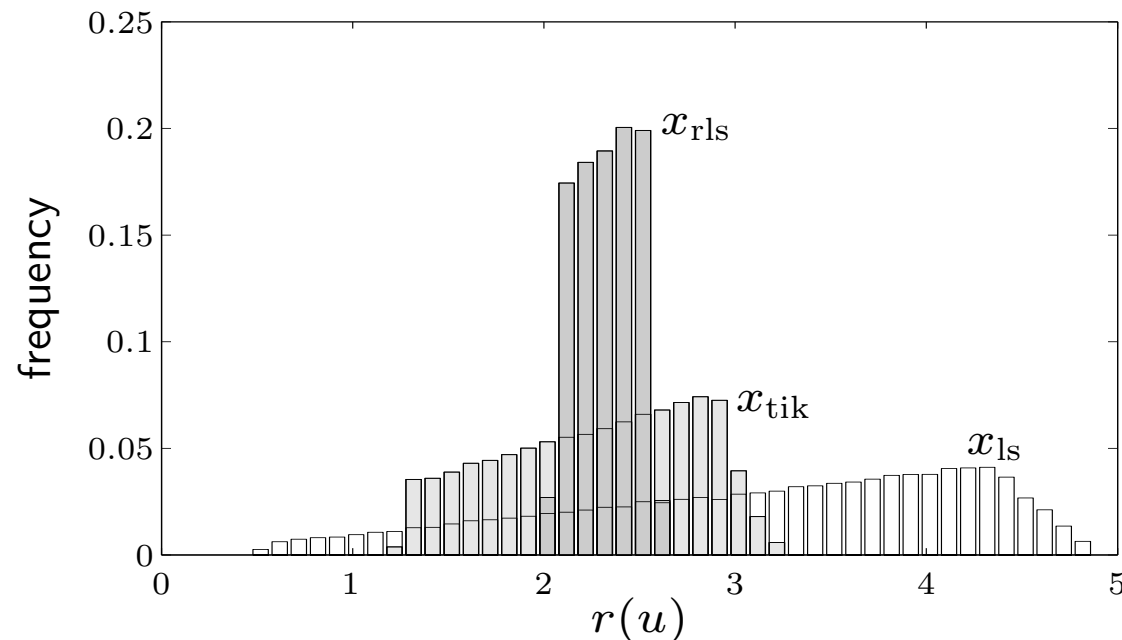
- hence, robust LS problem is equivalent to SDP

$$\begin{array}{ll} \text{minimize} & t + \lambda \\ \text{subject to} & \begin{bmatrix} I & P(x) & q(x) \\ P(x)^T & \lambda I & 0 \\ q(x)^T & 0 & t \end{bmatrix} \succeq 0 \end{array}$$

example: histogram of residuals

$$r(u) = \|(A_0 + u_1 A_1 + u_2 A_2)x - b\|_2$$

with u uniformly distributed on unit disk, for three values of x



- x_{ls} minimizes $\|A_0 x - b\|_2$
- x_{tik} minimizes $\|A_0 x - b\|_2^2 + \|x\|_2^2$ (Tikhonov solution)
- x_{wc} minimizes $\sup_{\|u\|_2 \leq 1} \|A_0 x - b\|_2^2 + \|x\|_2^2$

7. Statistical estimation

- maximum likelihood estimation
- optimal detector design
- experiment design

Parametric distribution estimation

- distribution estimation problem: estimate probability density $p(y)$ of a random variable from observed values
- parametric distribution estimation: choose from a family of densities $p_x(y)$, indexed by a parameter x

maximum likelihood estimation

$$\text{maximize (over } x) \quad \log p_x(y)$$

- y is observed value
- $l(x) = \log p_x(y)$ is called log-likelihood function
- can add constraints $x \in C$ explicitly, or define $p_x(y) = 0$ for $x \notin C$
- a convex optimization problem if $\log p_x(y)$ is concave in x for fixed y

Linear measurements with IID noise

linear measurement model

$$y_i = a_i^T x + v_i, \quad i = 1, \dots, m$$

- $x \in \mathbf{R}^n$ is vector of unknown parameters
- v_i is IID measurement noise, with density $p(z)$
- y_i is measurement: $y \in \mathbf{R}^m$ has density $p_x(y) = \prod_{i=1}^m p(y_i - a_i^T x)$

maximum likelihood estimate: any solution x of

$$\text{maximize } l(x) = \sum_{i=1}^m \log p(y_i - a_i^T x)$$

(y is observed value)

examples

- Gaussian noise $\mathcal{N}(0, \sigma^2)$: $p(z) = (2\pi\sigma^2)^{-1/2} e^{-z^2/(2\sigma^2)}$,

$$l(x) = -\frac{m}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^m (a_i^T x - y_i)^2$$

ML estimate is LS solution

- Laplacian noise: $p(z) = (1/(2a)) e^{-|z|/a}$,

$$l(x) = -m \log(2a) - \frac{1}{a} \sum_{i=1}^m |a_i^T x - y_i|$$

ML estimate is ℓ_1 -norm solution

- uniform noise on $[-a, a]$:

$$l(x) = \begin{cases} -m \log(2a) & |a_i^T x - y_i| \leq a, \quad i = 1, \dots, m \\ -\infty & \text{otherwise} \end{cases}$$

ML estimate is any x with $|a_i^T x - y_i| \leq a$

Logistic regression

random variable $y \in \{0, 1\}$ with distribution

$$p = \mathbf{prob}(y = 1) = \frac{\exp(a^T u + b)}{1 + \exp(a^T u + b)}$$

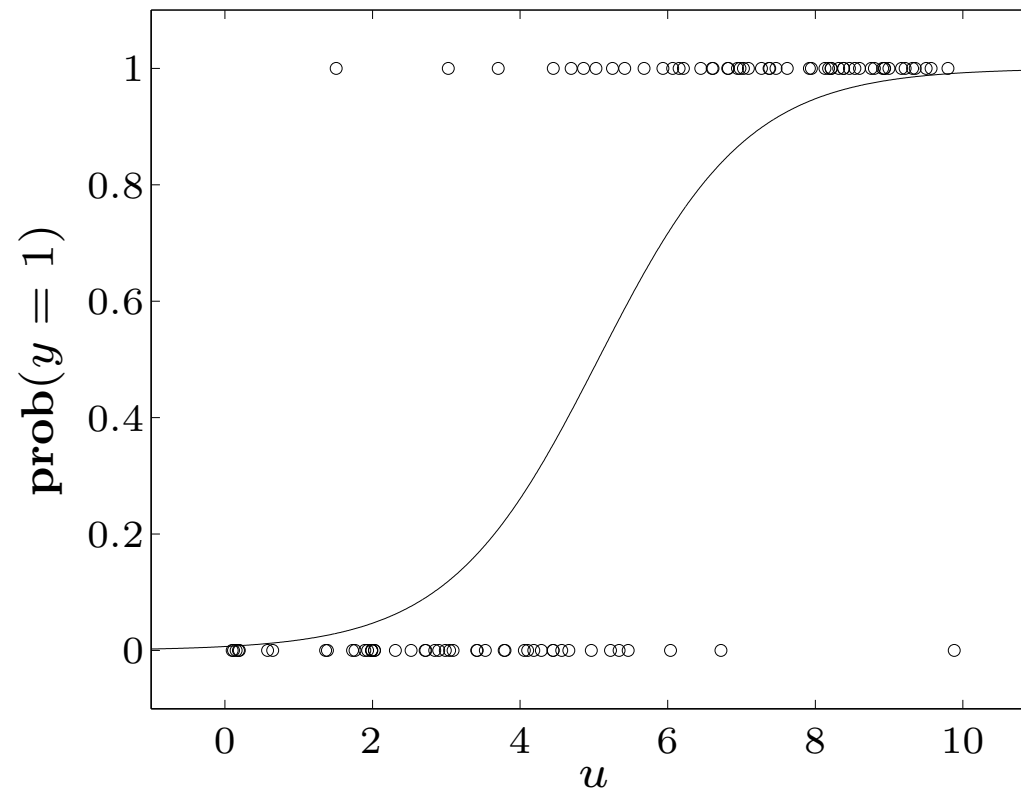
- a, b are parameters; $u \in \mathbf{R}^n$ are (observable) explanatory variables
- estimation problem: estimate a, b from m observations (u_i, y_i)

log-likelihood function (for $y_1 = \dots = y_k = 1, y_{k+1} = \dots = y_m = 0$):

$$\begin{aligned} l(a, b) &= \log \left(\prod_{i=1}^k \frac{\exp(a^T u_i + b)}{1 + \exp(a^T u_i + b)} \prod_{i=k+1}^m \frac{1}{1 + \exp(a^T u_i + b)} \right) \\ &= \sum_{i=1}^k (a^T u_i + b) - \sum_{i=1}^m \log(1 + \exp(a^T u_i + b)) \end{aligned}$$

concave in a, b

example ($n = 1$, $m = 50$ measurements)



- circles show 50 points (u_i, y_i)
- solid curve is ML estimate of $p = \exp(au + b) / (1 + \exp(au + b))$

(Binary) hypothesis testing

detection (hypothesis testing) problem

given observation of a random variable $X \in \{1, \dots, n\}$, choose between:

- hypothesis 1: X was generated by distribution $p = (p_1, \dots, p_n)$
- hypothesis 2: X was generated by distribution $q = (q_1, \dots, q_n)$

randomized detector

- a nonnegative matrix $T \in \mathbf{R}^{2 \times n}$, with $\mathbf{1}^T T = \mathbf{1}$
- if we observe $X = k$, we choose hypothesis 1 with probability t_{1k} , hypothesis 2 with probability t_{2k}
- if all elements of T are 0 or 1, it is called a deterministic detector

detection probability matrix:

$$D = \begin{bmatrix} T_p & T_q \end{bmatrix} = \begin{bmatrix} 1 - P_{\text{fp}} & P_{\text{fn}} \\ P_{\text{fp}} & 1 - P_{\text{fn}} \end{bmatrix}$$

- P_{fp} is probability of selecting hypothesis 2 if X is generated by distribution 1 (false positive)
- P_{fn} is probability of selecting hypothesis 1 if X is generated by distribution 2 (false negative)

multicriterion formulation of detector design

$$\begin{array}{ll} \text{minimize (w.r.t. } \mathbf{R}_+^2) & (P_{\text{fp}}, P_{\text{fn}}) = ((Tp)_2, (Tq)_1) \\ \text{subject to} & t_{1k} + t_{2k} = 1, \quad k = 1, \dots, n \\ & t_{ik} \geq 0, \quad i = 1, 2, \quad k = 1, \dots, n \end{array}$$

variable $T \in \mathbf{R}^{2 \times n}$

scalarization (with weight $\lambda > 0$)

$$\begin{array}{ll}\text{minimize} & (Tp)_2 + \lambda(Tq)_1 \\ \text{subject to} & t_{1k} + t_{2k} = 1, \quad t_{ik} \geq 0, \quad i = 1, 2, \quad k = 1, \dots, n\end{array}$$

an LP with a simple analytical solution

$$(t_{1k}, t_{2k}) = \begin{cases} (1, 0) & p_k \geq \lambda q_k \\ (0, 1) & p_k < \lambda q_k \end{cases}$$

- a deterministic detector, given by a likelihood ratio test
- if $p_k = \lambda q_k$ for some k , any value $0 \leq t_{1k} \leq 1$, $t_{1k} = 1 - t_{2k}$ is optimal (*i.e.*, Pareto-optimal detectors include non-deterministic detectors)

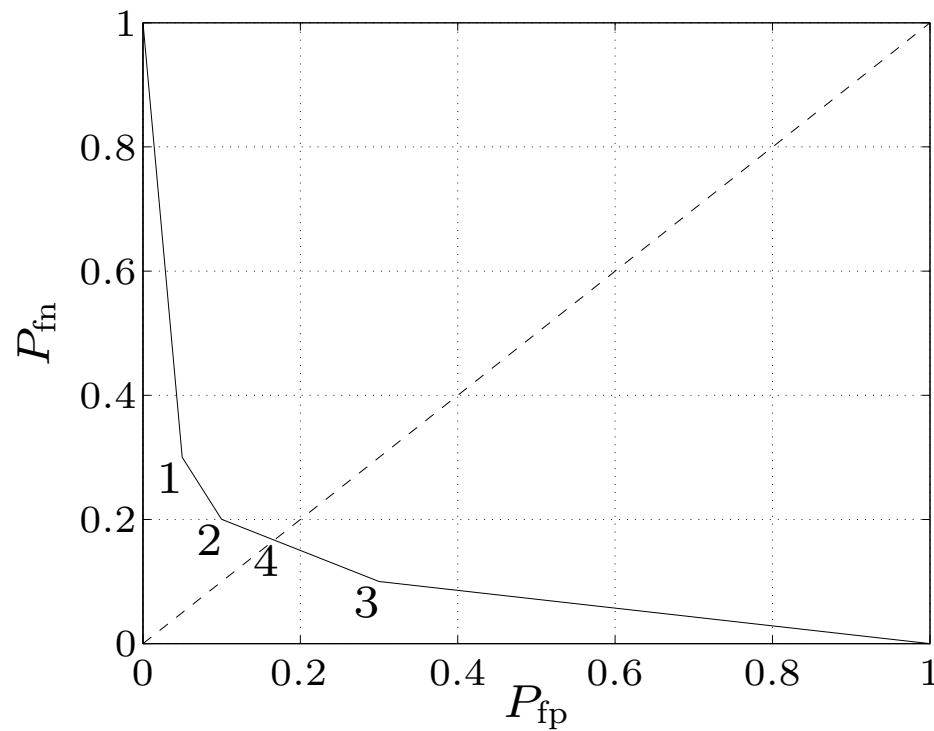
minimax detector

$$\begin{array}{ll}\text{minimize} & \max\{P_{\text{fp}}, P_{\text{fn}}\} = \max\{(Tp)_2, (Tq)_1\} \\ \text{subject to} & t_{1k} + t_{2k} = 1, \quad t_{ik} \geq 0, \quad i = 1, 2, \quad k = 1, \dots, n\end{array}$$

an LP; solution is usually not deterministic

example

$$P = \begin{bmatrix} 0.70 & 0.10 \\ 0.20 & 0.10 \\ 0.05 & 0.70 \\ 0.05 & 0.10 \end{bmatrix}$$



solutions 1, 2, 3 (and endpoints) are deterministic; 4 is minimax detector

Experiment design

m linear measurements $y_i = a_i^T x + w_i$, $i = 1, \dots, m$ of unknown $x \in \mathbf{R}^n$

- measurement errors w_i are IID $\mathcal{N}(0, 1)$
- ML (least-squares) estimate is

$$\hat{x} = \left(\sum_{i=1}^m a_i a_i^T \right)^{-1} \sum_{i=1}^m y_i a_i$$

- error $e = \hat{x} - x$ has zero mean and covariance

$$E = \mathbf{E} e e^T = \left(\sum_{i=1}^m a_i a_i^T \right)^{-1}$$

confidence ellipsoids are given by $\{x \mid (x - \hat{x})^T E^{-1} (x - \hat{x}) \leq \beta\}$

experiment design: choose $a_i \in \{v_1, \dots, v_p\}$ (a set of possible test vectors) to make E 'small'

vector optimization formulation

$$\begin{array}{ll} \text{minimize (w.r.t. } \mathbf{S}_+^n) & E = \left(\sum_{k=1}^p m_k v_k v_k^T \right)^{-1} \\ \text{subject to} & m_k \geq 0, \quad m_1 + \cdots + m_p = m \\ & m_k \in \mathbf{Z} \end{array}$$

- variables are m_k (\neq vectors a_i equal to v_k)
- difficult in general, due to integer constraint

relaxed experiment design

assume $m \gg p$, use $\lambda_k = m_k/m$ as (continuous) real variable

$$\begin{array}{ll} \text{minimize (w.r.t. } \mathbf{S}_+^n) & E = (1/m) \left(\sum_{k=1}^p \lambda_k v_k v_k^T \right)^{-1} \\ \text{subject to} & \lambda \succeq 0, \quad \mathbf{1}^T \lambda = 1 \end{array}$$

- common scalarizations: minimize $\log \det E$, $\text{tr } E$, $\lambda_{\max}(E)$, \dots
- can add other convex constraints, *e.g.*, bound experiment cost $c^T \lambda \leq B$

***D*-optimal design**

$$\begin{array}{ll}\text{minimize} & \log \det \left(\sum_{k=1}^p \lambda_k v_k v_k^T \right)^{-1} \\ \text{subject to} & \lambda \succeq 0, \quad \mathbf{1}^T \lambda = 1\end{array}$$

interpretation: minimizes volume of confidence ellipsoids

dual problem

$$\begin{array}{ll}\text{maximize} & \log \det W + n \log n \\ \text{subject to} & v_k^T W v_k \leq 1, \quad k = 1, \dots, p\end{array}$$

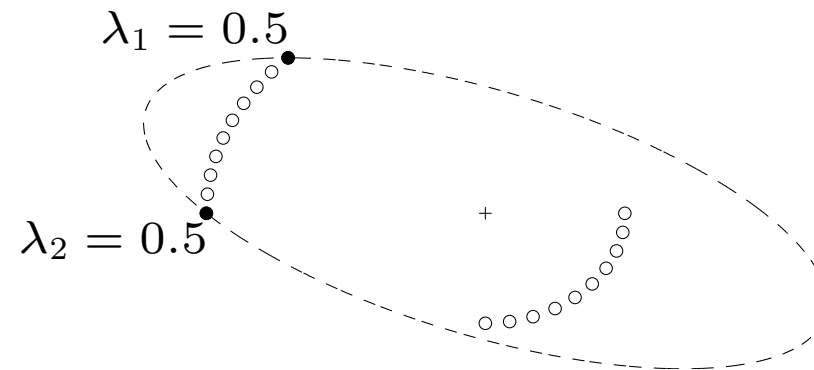
interpretation: $\{x \mid x^T W x \leq 1\}$ is minimum volume ellipsoid centered at origin, that includes all test vectors v_k

complementary slackness: for λ , W primal and dual optimal

$$\lambda_k (1 - v_k^T W v_k) = 0, \quad k = 1, \dots, p$$

optimal experiment uses vectors v_k on boundary of ellipsoid defined by W

example ($p = 20$)



design uses two vectors, on boundary of ellipse defined by optimal W

derivation of dual of page 7–13

first reformulate primal problem with new variable X :

$$\begin{array}{ll}\text{minimize} & \log \det X^{-1} \\ \text{subject to} & X = \sum_{k=1}^p \lambda_k v_k v_k^T, \quad \lambda \succeq 0, \quad \mathbf{1}^T \lambda = 1\end{array}$$

$$L(X, \lambda, Z, z, \nu) = \log \det X^{-1} + \text{tr} \left(Z \left(X - \sum_{k=1}^p \lambda_k v_k v_k^T \right) \right) - z^T \lambda + \nu (\mathbf{1}^T \lambda - 1)$$

- minimize over X by setting gradient to zero: $-X^{-1} + Z = 0$
- minimum over λ_k is $-\infty$ unless $-v_k^T Z v_k - z_k + \nu = 0$

dual problem

$$\begin{array}{ll}\text{maximize} & n + \log \det Z - \nu \\ \text{subject to} & v_k^T Z v_k \leq \nu, \quad k = 1, \dots, p\end{array}$$

change variable $W = Z/\nu$, and optimize over ν to get dual of page 7–13

8. Geometric problems

- extremal volume ellipsoids
- centering
- classification
- placement and facility location

Minimum volume ellipsoid around a set

Löwner-John ellipsoid of a set C : minimum volume ellipsoid \mathcal{E} s.t. $C \subseteq \mathcal{E}$

- parametrize \mathcal{E} as $\mathcal{E} = \{v \mid \|Av + b\|_2 \leq 1\}$; w.l.o.g. assume $A \in \mathbf{S}_{++}^n$
- $\text{vol } \mathcal{E}$ is proportional to $\det A^{-1}$; to compute minimum volume ellipsoid,

$$\begin{array}{ll} \text{minimize (over } A, b) & \log \det A^{-1} \\ \text{subject to} & \sup_{v \in C} \|Av + b\|_2 \leq 1 \end{array}$$

convex, but evaluating the constraint can be hard (for general C)

finite set $C = \{x_1, \dots, x_m\}$:

$$\begin{array}{ll} \text{minimize (over } A, b) & \log \det A^{-1} \\ \text{subject to} & \|Ax_i + b\|_2 \leq 1, \quad i = 1, \dots, m \end{array}$$

also gives Löwner-John ellipsoid for polyhedron $\text{conv}\{x_1, \dots, x_m\}$

Maximum volume inscribed ellipsoid

maximum volume ellipsoid \mathcal{E} inside a convex set $C \subseteq \mathbf{R}^n$

- parametrize \mathcal{E} as $\mathcal{E} = \{Bu + d \mid \|u\|_2 \leq 1\}$; w.l.o.g. assume $B \in \mathbf{S}_{++}^n$
- $\text{vol } \mathcal{E}$ is proportional to $\det B$; can compute \mathcal{E} by solving

$$\begin{array}{ll}\text{maximize} & \log \det B \\ \text{subject to} & \sup_{\|u\|_2 \leq 1} I_C(Bu + d) \leq 0\end{array}$$

(where $I_C(x) = 0$ for $x \in C$ and $I_C(x) = \infty$ for $x \notin C$)

convex, but evaluating the constraint can be hard (for general C)

polyhedron $\{x \mid a_i^T x \leq b_i, i = 1, \dots, m\}$:

$$\begin{array}{ll}\text{maximize} & \log \det B \\ \text{subject to} & \|Ba_i\|_2 + a_i^T d \leq b_i, \quad i = 1, \dots, m\end{array}$$

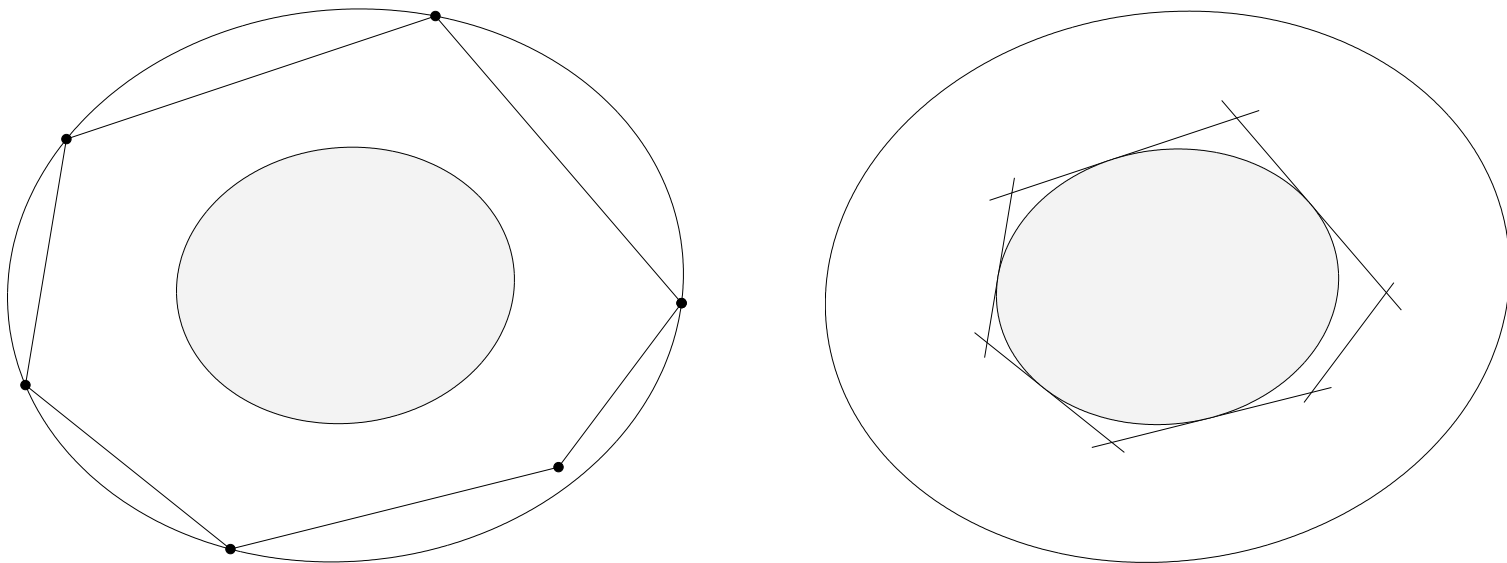
(constraint follows from $\sup_{\|u\|_2 \leq 1} a_i^T (Bu + d) = \|Ba_i\|_2 + a_i^T d$)

Efficiency of ellipsoidal approximations

$C \subseteq \mathbf{R}^n$ convex, bounded, with nonempty interior

- Löwner-John ellipsoid, shrunk by a factor n , lies inside C
- maximum volume inscribed ellipsoid, expanded by a factor n , covers C

example (for two polyhedra in \mathbf{R}^2)

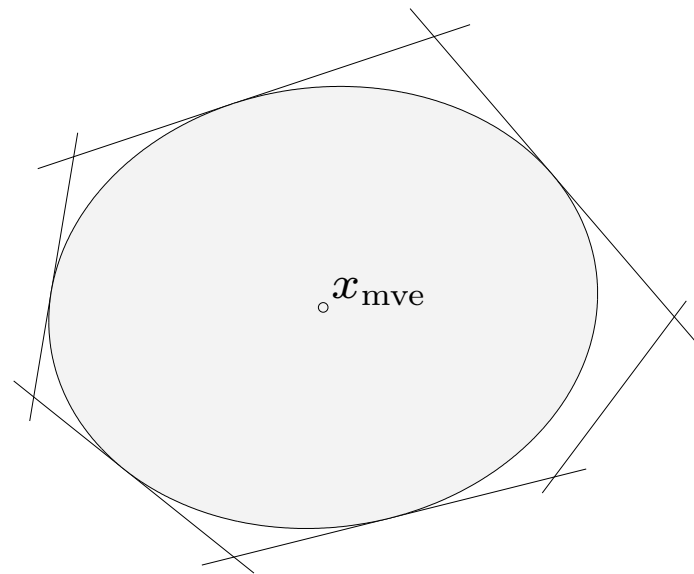
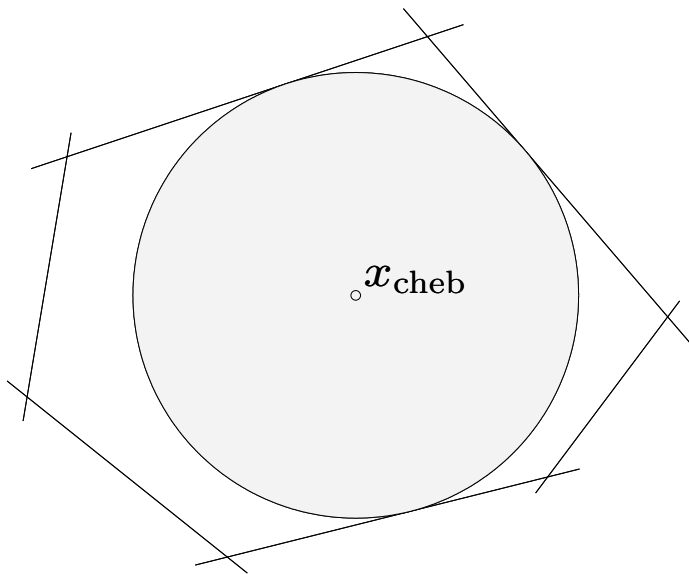


factor n can be improved to \sqrt{n} if C is symmetric

Centering

some possible definitions of 'center' of a convex set C :

- center of largest inscribed ball ('Chebyshev center')
for polyhedron, can be computed via linear programming (page 4–19)
- center of maximum volume inscribed ellipsoid (page 8–3)



MVE center is invariant under affine coordinate transformations

Analytic center of a set inequalities

the analytic center of set of convex inequalities and linear equations

$$f_i(x) \leq 0, \quad i = 1, \dots, m, \quad Fx = g$$

is defined as the optimal point of

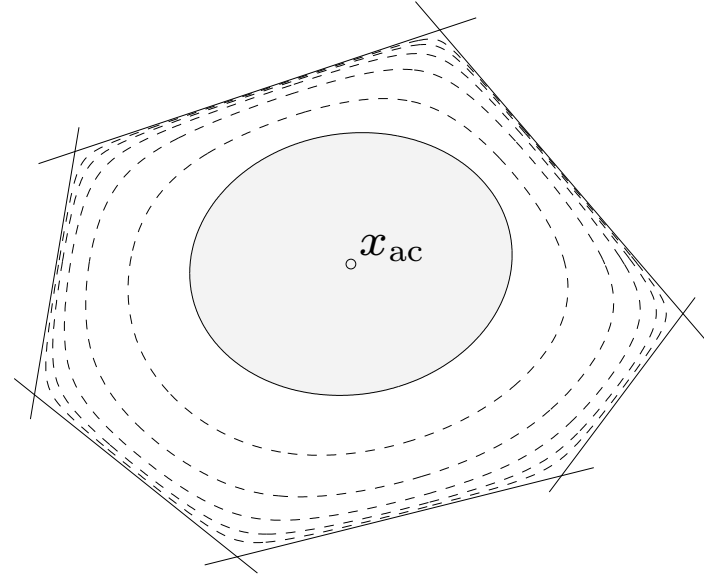
$$\begin{array}{ll} \text{minimize} & -\sum_{i=1}^m \log(-f_i(x)) \\ \text{subject to} & Fx = g \end{array}$$

- more easily computed than MVE or Chebyshev center (see later)
- not just a property of the feasible set: two sets of inequalities can describe the same set, but have different analytic centers

analytic center of linear inequalities $a_i^T x \leq b_i, i = 1, \dots, m$

x_{ac} is minimizer of

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



inner and outer ellipsoids from analytic center:

$$\mathcal{E}_{\text{inner}} \subseteq \{x \mid a_i^T x \leq b_i, i = 1, \dots, m\} \subseteq \mathcal{E}_{\text{outer}}$$

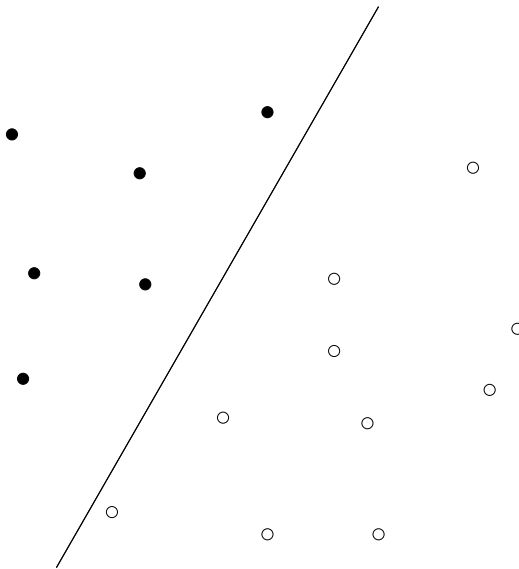
where

$$\begin{aligned} \mathcal{E}_{\text{inner}} &= \{x \mid (x - x_{ac})^T \nabla^2 \phi(x_{ac})(x - x_{ac}) \leq 1\} \\ \mathcal{E}_{\text{outer}} &= \{x \mid (x - x_{ac})^T \nabla^2 \phi(x_{ac})(x - x_{ac}) \leq m(m-1)\} \end{aligned}$$

Linear discrimination

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

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



homogeneous in a , b , hence equivalent to

$$a^T x_i + b_i \geq 1, \quad i = 1, \dots, N, \quad a^T y_i + b_i \leq -1, \quad i = 1, \dots, M$$

a set of linear inequalities in a , b

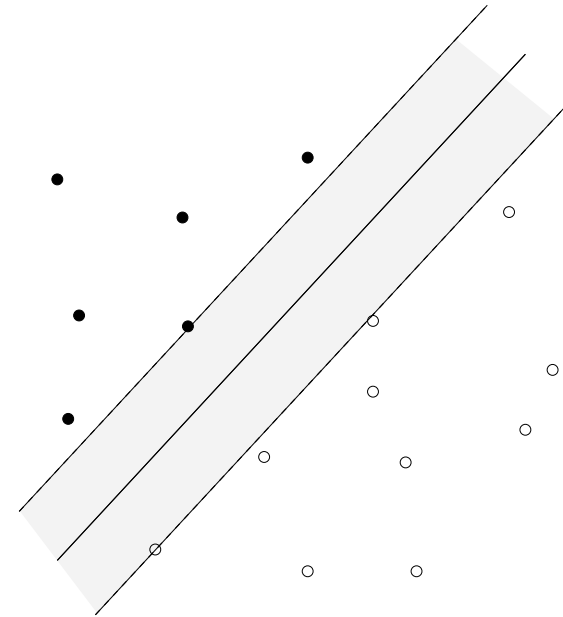
Robust linear discrimination

(Euclidean) distance between hyperplanes

$$\mathcal{H}_1 = \{z \mid a^T z + b = 1\}$$

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

is $\text{dist}(\mathcal{H}_1, \mathcal{H}_2) = 2/\|a\|_2$



to separate two sets of points by maximum margin,

$$\begin{aligned} & \text{minimize} && (1/2)\|a\|_2 \\ & \text{subject to} && a^T x_i + b \geq 1, \quad i = 1, \dots, N \\ & && a^T y_i + b \leq -1, \quad i = 1, \dots, M \end{aligned} \tag{1}$$

(after squaring objective) a QP in a, b

Lagrange dual of maximum margin separation problem (1)

$$\begin{aligned} & \text{maximize} && \mathbf{1}^T \lambda + \mathbf{1}^T \mu \\ & \text{subject to} && 2 \left\| \sum_{i=1}^N \lambda_i x_i - \sum_{i=1}^M \mu_i y_i \right\|_2 \leq 1 \\ & && \mathbf{1}^T \lambda = \mathbf{1}^T \mu, \quad \lambda \succeq 0, \quad \mu \succeq 0 \end{aligned} \tag{2}$$

from duality, optimal value is inverse of maximum margin of separation

interpretation

- change variables to $\theta_i = \lambda_i / \mathbf{1}^T \lambda$, $\gamma_i = \mu_i / \mathbf{1}^T \mu$, $t = 1 / (\mathbf{1}^T \lambda + \mathbf{1}^T \mu)$
- invert objective to minimize $1 / (\mathbf{1}^T \lambda + \mathbf{1}^T \mu) = t$

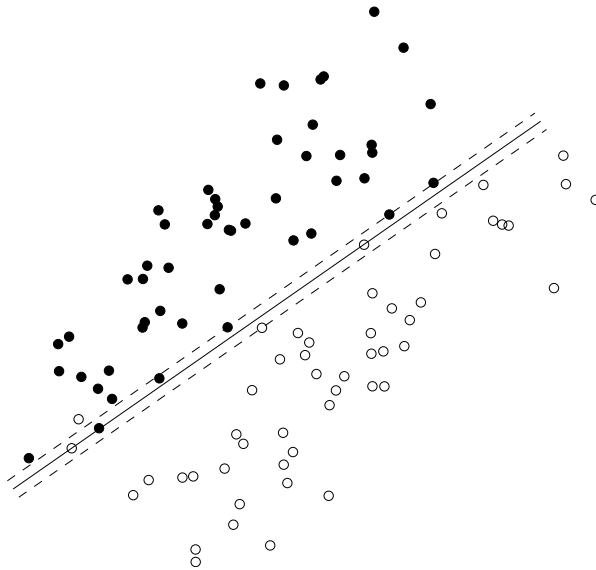
$$\begin{aligned} & \text{minimize} && t \\ & \text{subject to} && \left\| \sum_{i=1}^N \theta_i x_i - \sum_{i=1}^M \gamma_i y_i \right\|_2 \leq t \\ & && \theta \succeq 0, \quad \mathbf{1}^T \theta = 1, \quad \gamma \succeq 0, \quad \mathbf{1}^T \gamma = 1 \end{aligned}$$

optimal value is distance between convex hulls

Approximate linear separation of non-separable sets

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

- an LP in a, b, u, v
- at optimum, $u_i = \max\{0, 1 - a^T x_i - b\}$, $v_i = \max\{0, 1 + a^T y_i + b\}$
- can be interpreted as a heuristic for minimizing #misclassified points

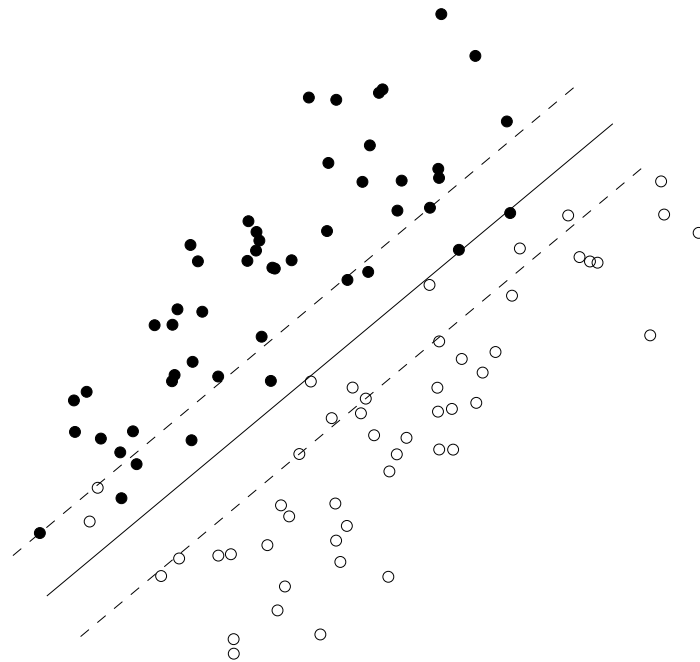


Support vector classifier

$$\begin{array}{ll}\text{minimize} & \|a\|_2 + \gamma(\mathbf{1}^T u + \mathbf{1}^T v) \\ \text{subject to} & a^T x_i + b \geq 1 - u_i, \quad i = 1, \dots, N \\ & a^T y_i + b \leq -1 + v_i, \quad i = 1, \dots, M \\ & u \succeq 0, \quad v \succeq 0\end{array}$$

produces point on trade-off curve between inverse of margin $2/\|a\|_2$ and classification error, measured by total slack $\mathbf{1}^T u + \mathbf{1}^T v$

same example as previous page,
with $\gamma = 0.1$:



Nonlinear discrimination

separate two sets of points by a nonlinear function:

$$f(x_i) > 0, \quad i = 1, \dots, N, \quad f(y_i) < 0, \quad i = 1, \dots, M$$

- choose a linearly parametrized family of functions

$$f(z) = \theta^T F(z)$$

$F = (F_1, \dots, F_k) : \mathbf{R}^n \rightarrow \mathbf{R}^k$ are basis functions

- solve a set of linear inequalities in θ :

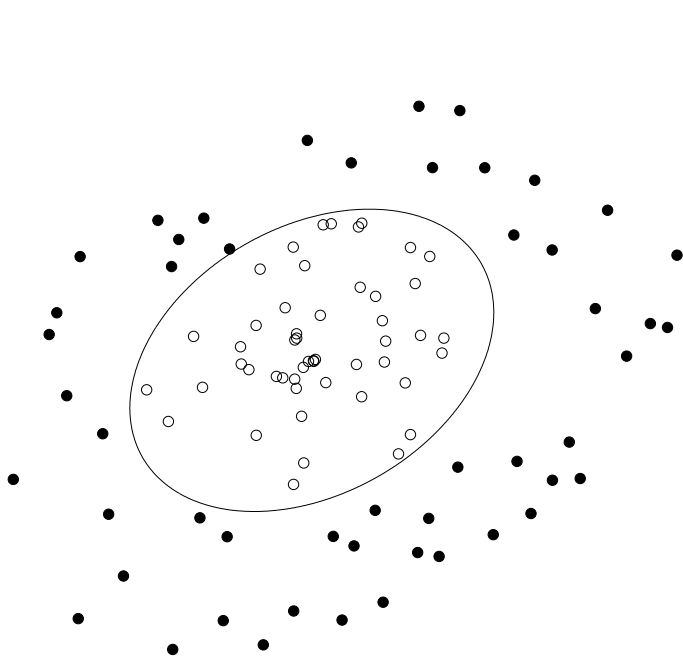
$$\theta^T F(x_i) \geq 1, \quad i = 1, \dots, N, \quad \theta^T F(y_i) \leq -1, \quad i = 1, \dots, M$$

quadratic discrimination: $f(z) = z^T P z + q^T z + r$

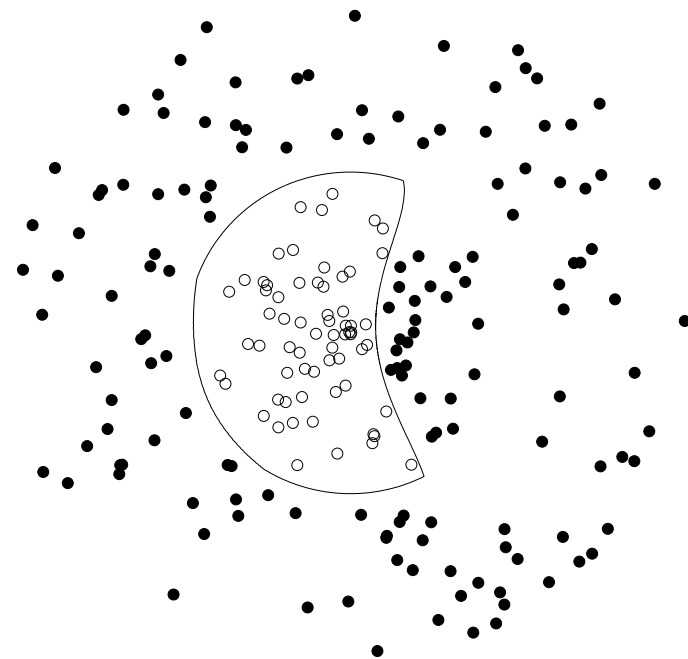
$$x_i^T P x_i + q^T x_i + r \geq 1, \quad y_i^T P y_i + q^T y_i + r \leq -1$$

can add additional constraints (*e.g.*, $P \preceq -I$ to separate by an ellipsoid)

polynomial discrimination: $F(z)$ are all monomials up to a given degree



separation by ellipsoid



separation by 4th degree polynomial

Placement and facility location

- N points with coordinates $x_i \in \mathbf{R}^2$ (or \mathbf{R}^3)
- some positions x_i are given; the other x_i 's are variables
- for each pair of points, a cost function $f_{ij}(x_i, x_j)$

placement problem

$$\text{minimize } \sum_{i \neq j} f_{ij}(x_i, x_j)$$

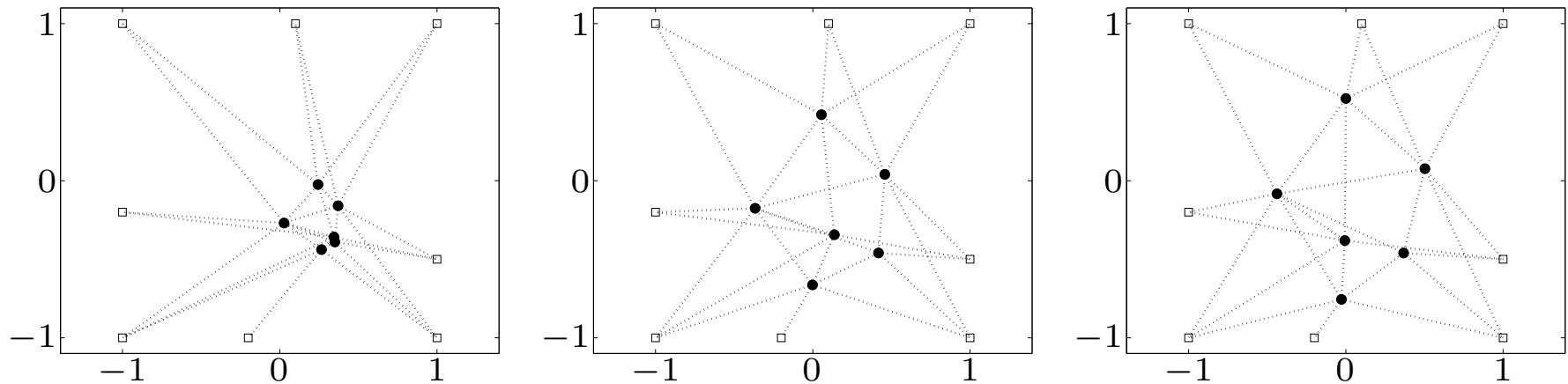
variables are positions of free points

interpretations

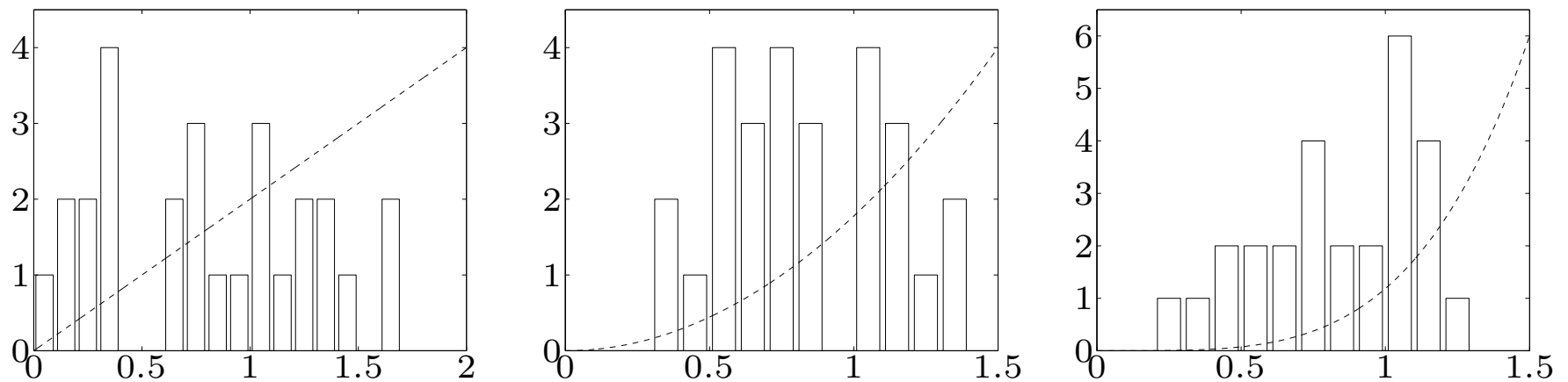
- points represent plants or warehouses; f_{ij} is transportation cost between facilities i and j
- points represent cells on an IC; f_{ij} represents wirelength

example: minimize $\sum_{(i,j) \in \mathcal{A}} h(\|x_i - x_j\|_2)$, with 6 free points, 27 links

optimal placement for $h(z) = z$, $h(z) = z^2$, $h(z) = z^4$



histograms of connection lengths $\|x_i - x_j\|_2$



9. Numerical linear algebra background

- matrix structure and algorithm complexity
- solving linear equations with factored matrices
- LU, Cholesky, LDL^T factorization
- block elimination and the matrix inversion lemma
- solving underdetermined equations

Matrix structure and algorithm complexity

cost (execution time) of solving $Ax = b$ with $A \in \mathbf{R}^{n \times n}$

- for general methods, grows as n^3
- less if A is structured (banded, sparse, Toeplitz, . . .)

flop counts

- flop (floating-point operation): one addition, subtraction, multiplication, or division of two floating-point numbers
- to estimate complexity of an algorithm: express number of flops as a (polynomial) function of the problem dimensions, and simplify by keeping only the leading terms
- not an accurate predictor of computation time on modern computers
- useful as a rough estimate of complexity

vector-vector operations ($x, y \in \mathbf{R}^n$)

- inner product $x^T y$: $2n - 1$ flops (or $2n$ if n is large)
- sum $x + y$, scalar multiplication αx : n flops

matrix-vector product $y = Ax$ with $A \in \mathbf{R}^{m \times n}$

- $m(2n - 1)$ flops (or $2mn$ if n large)
- $2N$ if A is sparse with N nonzero elements
- $2p(n + m)$ if A is given as $A = UV^T$, $U \in \mathbf{R}^{m \times p}$, $V \in \mathbf{R}^{n \times p}$

matrix-matrix product $C = AB$ with $A \in \mathbf{R}^{m \times n}$, $B \in \mathbf{R}^{n \times p}$

- $mp(2n - 1)$ flops (or $2mnp$ if n large)
- less if A and/or B are sparse
- $(1/2)m(m + 1)(2n - 1) \approx m^2 n$ if $m = p$ and C symmetric

Linear equations that are easy to solve

diagonal matrices ($a_{ij} = 0$ if $i \neq j$): n flops

$$x = A^{-1}b = (b_1/a_{11}, \dots, b_n/a_{nn})$$

lower triangular ($a_{ij} = 0$ if $j > i$): n^2 flops

$$x_1 := b_1/a_{11}$$

$$x_2 := (b_2 - a_{21}x_1)/a_{22}$$

$$x_3 := (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33}$$

$$\vdots$$

$$x_n := (b_n - a_{n1}x_1 - a_{n2}x_2 - \dots - a_{n,n-1}x_{n-1})/a_{nn}$$

called forward substitution

upper triangular ($a_{ij} = 0$ if $j < i$): n^2 flops via backward substitution

orthogonal matrices: $A^{-1} = A^T$

- $2n^2$ flops to compute $x = A^T b$ for general A
- less with structure, *e.g.*, if $A = I - 2uu^T$ with $\|u\|_2 = 1$, we can compute $x = A^T b = b - 2(u^T b)u$ in $4n$ flops

permutation matrices:

$$a_{ij} = \begin{cases} 1 & j = \pi_i \\ 0 & \text{otherwise} \end{cases}$$

where $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ is a permutation of $(1, 2, \dots, n)$

- interpretation: $Ax = (x_{\pi_1}, \dots, x_{\pi_n})$
- satisfies $A^{-1} = A^T$, hence cost of solving $Ax = b$ is 0 flops

example:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad A^{-1} = A^T = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

The factor-solve method for solving $Ax = b$

- factor A as a product of simple matrices (usually 2 or 3):

$$A = A_1 A_2 \cdots A_k$$

(A_i diagonal, upper or lower triangular, etc)

- compute $x = A^{-1}b = A_k^{-1} \cdots A_2^{-1} A_1^{-1}b$ by solving k ‘easy’ equations

$$A_1 x_1 = b, \quad A_2 x_2 = x_1, \quad \dots, \quad A_k x_k = x_{k-1}$$

cost of factorization step usually dominates cost of solve step

equations with multiple righthand sides

$$Ax_1 = b_1, \quad Ax_2 = b_2, \quad \dots, \quad Ax_m = b_m$$

cost: one factorization plus m solves

LU factorization

every nonsingular matrix A can be factored as

$$A = PLU$$

with P a permutation matrix, L lower triangular, U upper triangular

cost: $(2/3)n^3$ flops

Solving linear equations by LU factorization.

given a set of linear equations $Ax = b$, with A nonsingular.

1. *LU factorization.* Factor A as $A = PLU$ $((2/3)n^3$ flops).
2. *Permutation.* Solve $Pz_1 = b$ (0 flops).
3. *Forward substitution.* Solve $Lz_2 = z_1$ (n^2 flops).
4. *Backward substitution.* Solve $Ux = z_2$ (n^2 flops).

cost: $(2/3)n^3 + 2n^2 \approx (2/3)n^3$ for large n

sparse LU factorization

$$A = P_1 L U P_2$$

- adding permutation matrix P_2 offers possibility of sparser L , U (hence, cheaper factor and solve steps)
- P_1 and P_2 chosen (heuristically) to yield sparse L , U
- choice of P_1 and P_2 depends on sparsity pattern and values of A
- cost is usually much less than $(2/3)n^3$; exact value depends in a complicated way on n , number of zeros in A , sparsity pattern

Cholesky factorization

every positive definite A can be factored as

$$A = LL^T$$

with L lower triangular

cost: $(1/3)n^3$ flops

Solving linear equations by Cholesky factorization.

given a set of linear equations $Ax = b$, with $A \in \mathbf{S}_{++}^n$.

1. *Cholesky factorization.* Factor A as $A = LL^T$ ($(1/3)n^3$ flops).
2. *Forward substitution.* Solve $Lz_1 = b$ (n^2 flops).
3. *Backward substitution.* Solve $L^T x = z_1$ (n^2 flops).

cost: $(1/3)n^3 + 2n^2 \approx (1/3)n^3$ for large n

sparse Cholesky factorization

$$A = PLL^T P^T$$

- adding permutation matrix P offers possibility of sparser L
- P chosen (heuristically) to yield sparse L
- choice of P only depends on sparsity pattern of A (unlike sparse LU)
- cost is usually much less than $(1/3)n^3$; exact value depends in a complicated way on n , number of zeros in A , sparsity pattern

LDL^T factorization

every nonsingular symmetric matrix A can be factored as

$$A = PLDL^T P^T$$

with P a permutation matrix, L lower triangular, D block diagonal with 1×1 or 2×2 diagonal blocks

cost: $(1/3)n^3$

- cost of solving symmetric sets of linear equations by LDL^T factorization:
 $(1/3)n^3 + 2n^2 \approx (1/3)n^3$ for large n
- for sparse A , can choose P to yield sparse L ; cost $\ll (1/3)n^3$

Equations with structured sub-blocks

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \quad (1)$$

- variables $x_1 \in \mathbf{R}^{n_1}$, $x_2 \in \mathbf{R}^{n_2}$; blocks $A_{ij} \in \mathbf{R}^{n_i \times n_j}$
- if A_{11} is nonsingular, can eliminate x_1 : $x_1 = A_{11}^{-1}(b_1 - A_{12}x_2)$; to compute x_2 , solve

$$(A_{22} - A_{21}A_{11}^{-1}A_{12})x_2 = b_2 - A_{21}A_{11}^{-1}b_1$$

Solving linear equations by block elimination.

given a nonsingular set of linear equations (1), with A_{11} nonsingular.

1. Form $A_{11}^{-1}A_{12}$ and $A_{11}^{-1}b_1$.
 2. Form $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$ and $\tilde{b} = b_2 - A_{21}A_{11}^{-1}b_1$.
 3. Determine x_2 by solving $Sx_2 = \tilde{b}$.
 4. Determine x_1 by solving $A_{11}x_1 = b_1 - A_{12}x_2$.
-

dominant terms in flop count

- step 1: $f + n_2 s$ (f is cost of factoring A_{11} ; s is cost of solve step)
- step 2: $2n_2^2 n_1$ (cost dominated by product of A_{21} and $A_{11}^{-1} A_{12}$)
- step 3: $(2/3)n_2^3$

total: $f + n_2 s + 2n_2^2 n_1 + (2/3)n_2^3$

examples

- general A_{11} ($f = (2/3)n_1^3$, $s = 2n_1^2$): no gain over standard method

$$\text{\#flops} = (2/3)n_1^3 + 2n_1^2 n_2 + 2n_2^2 n_1 + (2/3)n_2^3 = (2/3)(n_1 + n_2)^3$$

- block elimination is useful for structured A_{11} ($f \ll n_1^3$)

for example, diagonal ($f = 0$, $s = n_1$): $\text{\#flops} \approx 2n_2^2 n_1 + (2/3)n_2^3$

Structured matrix plus low rank term

$$(A + BC)x = b$$

- $A \in \mathbf{R}^{n \times n}$, $B \in \mathbf{R}^{n \times p}$, $C \in \mathbf{R}^{p \times n}$
- assume A has structure ($Ax = b$ easy to solve)

first write as

$$\begin{bmatrix} A & B \\ C & -I \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

now apply block elimination: solve

$$(I + CA^{-1}B)y = CA^{-1}b,$$

then solve $Ax = b - By$

this proves the **matrix inversion lemma**: if A and $A + BC$ nonsingular,

$$(A + BC)^{-1} = A^{-1} - A^{-1}B(I + CA^{-1}B)^{-1}CA^{-1}$$

example: A diagonal, B, C dense

- method 1: form $D = A + BC$, then solve $Dx = b$

cost: $(2/3)n^3 + 2pn^2$

- method 2 (via matrix inversion lemma): solve

$$(I + CA^{-1}B)y = A^{-1}b, \quad (2)$$

then compute $x = A^{-1}b - A^{-1}By$

total cost is dominated by (2): $2p^2n + (2/3)p^3$ (*i.e.*, linear in n)

Underdetermined linear equations

if $A \in \mathbf{R}^{p \times n}$ with $p < n$, $\text{rank } A = p$,

$$\{x \mid Ax = b\} = \{Fz + \hat{x} \mid z \in \mathbf{R}^{n-p}\}$$

- \hat{x} is (any) particular solution
- columns of $F \in \mathbf{R}^{n \times (n-p)}$ span nullspace of A
- there exist several numerical methods for computing F
(QR factorization, rectangular LU factorization, . . .)

10. Unconstrained minimization

- terminology and assumptions
- gradient descent method
- steepest descent method
- Newton's method
- self-concordant functions
- implementation

Unconstrained minimization

$$\text{minimize } f(x)$$

- f convex, twice continuously differentiable (hence $\text{dom } f$ open)
- we assume optimal value $p^* = \inf_x f(x)$ is attained (and finite)

unconstrained minimization methods

- produce sequence of points $x^{(k)} \in \text{dom } f$, $k = 0, 1, \dots$ with

$$f(x^{(k)}) \rightarrow p^*$$

- can be interpreted as iterative methods for solving optimality condition

$$\nabla f(x^*) = 0$$

Initial point and sublevel set

algorithms in this chapter require a starting point $x^{(0)}$ such that

- $x^{(0)} \in \mathbf{dom} f$
- sublevel set $S = \{x \mid f(x) \leq f(x^{(0)})\}$ is closed

2nd condition is hard to verify, except when *all* sublevel sets are closed:

- equivalent to condition that $\mathbf{epi} f$ is closed
- true if $\mathbf{dom} f = \mathbf{R}^n$
- true if $f(x) \rightarrow \infty$ as $x \rightarrow \mathbf{bd} \mathbf{dom} f$

examples of differentiable functions with closed sublevel sets:

$$f(x) = \log\left(\sum_{i=1}^m \exp(a_i^T x + b_i)\right), \quad f(x) = -\sum_{i=1}^m \log(b_i - a_i^T x)$$

Strong convexity and implications

f is strongly convex on S if there exists an $m > 0$ such that

$$\nabla^2 f(x) \succeq mI \quad \text{for all } x \in S$$

implications

- for $x, y \in S$,

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2} \|x - y\|_2^2$$

hence, S is bounded

- $p^\star > -\infty$, and for $x \in S$,

$$f(x) - p^\star \leq \frac{1}{2m} \|\nabla f(x)\|_2^2$$

useful as stopping criterion (if you know m)

Descent methods

$$x^{(k+1)} = x^{(k)} + t^{(k)} \Delta x^{(k)} \quad \text{with } f(x^{(k+1)}) < f(x^{(k)})$$

- other notations: $x^+ = x + t\Delta x$, $x := x + t\Delta x$
- Δx is the *step*, or *search direction*; t is the *step size*, or *step length*
- from convexity, $f(x^+) < f(x)$ implies $\nabla f(x)^T \Delta x < 0$
(*i.e.*, Δx is a *descent direction*)

General descent method.

given a starting point $x \in \text{dom } f$.

repeat

1. Determine a descent direction Δx .
2. *Line search.* Choose a step size $t > 0$.
3. *Update.* $x := x + t\Delta x$.

until stopping criterion is satisfied.

Line search types

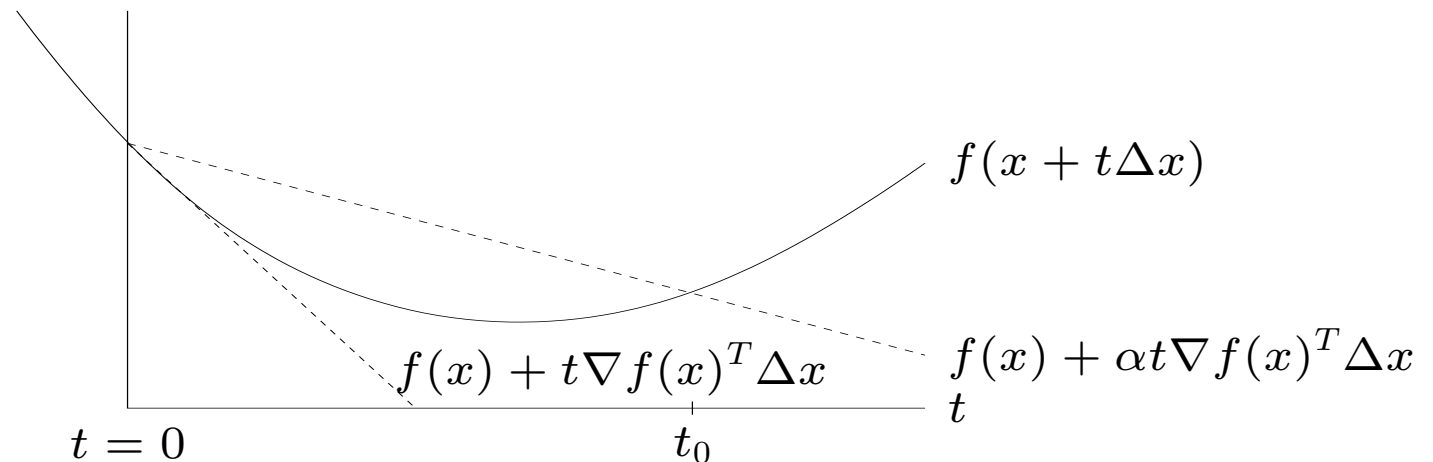
exact line search: $t = \operatorname{argmin}_{t>0} f(x + t\Delta x)$

backtracking line search (with parameters $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$)

- starting at $t = 1$, repeat $t := \beta t$ until

$$f(x + t\Delta x) < f(x) + \alpha t \nabla f(x)^T \Delta x$$

- graphical interpretation: backtrack until $t \leq t_0$



Gradient descent method

general descent method with $\Delta x = -\nabla f(x)$

given a starting point $x \in \text{dom } f$.

repeat

1. $\Delta x := -\nabla f(x)$.
2. *Line search*. Choose step size t via exact or backtracking line search.
3. *Update*. $x := x + t\Delta x$.

until stopping criterion is satisfied.

- stopping criterion usually of the form $\|\nabla f(x)\|_2 \leq \epsilon$
- convergence result: for strongly convex f ,

$$f(x^{(k)}) - p^* \leq c^k (f(x^{(0)}) - p^*)$$

$c \in (0, 1)$ depends on m , $x^{(0)}$, line search type

- very simple, but often very slow; rarely used in practice

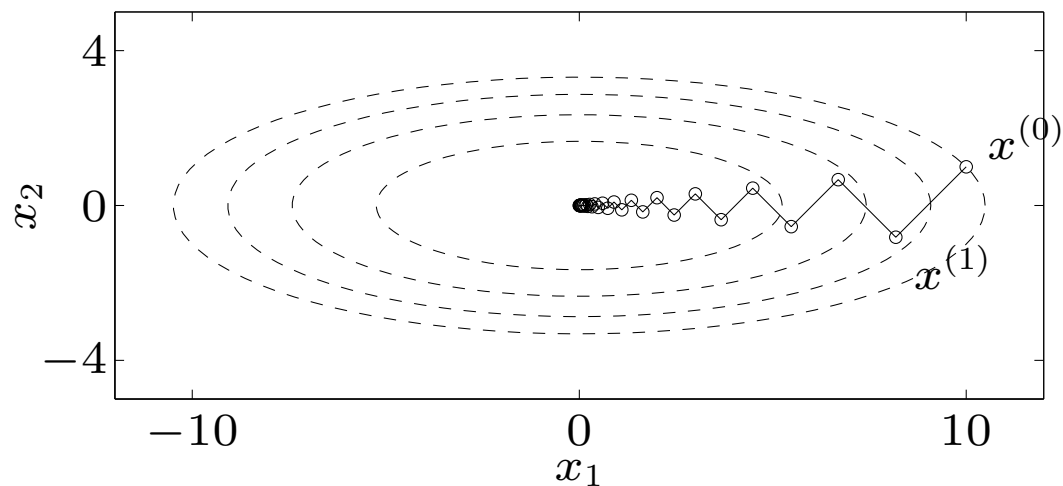
quadratic problem in \mathbf{R}^2

$$f(x) = (1/2)(x_1^2 + \gamma x_2^2) \quad (\gamma > 0)$$

with exact line search, starting at $x^{(0)} = (\gamma, 1)$:

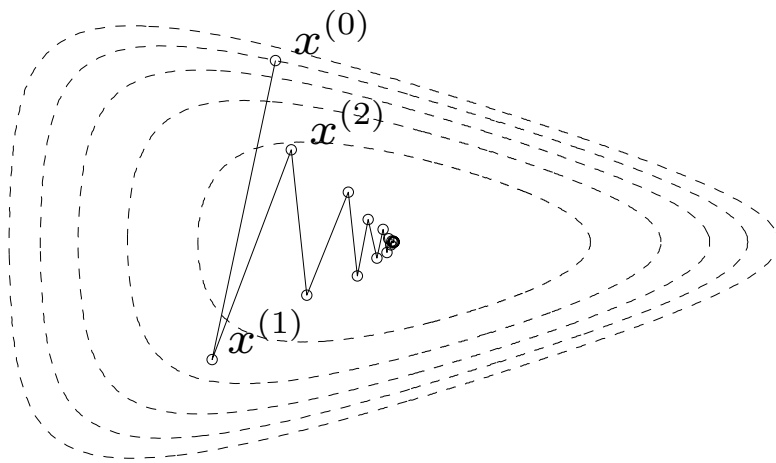
$$x_1^{(k)} = \gamma \left(\frac{\gamma - 1}{\gamma + 1} \right)^k, \quad x_2^{(k)} = \left(-\frac{\gamma - 1}{\gamma + 1} \right)^k$$

- very slow if $\gamma \gg 1$ or $\gamma \ll 1$
- example for $\gamma = 10$:

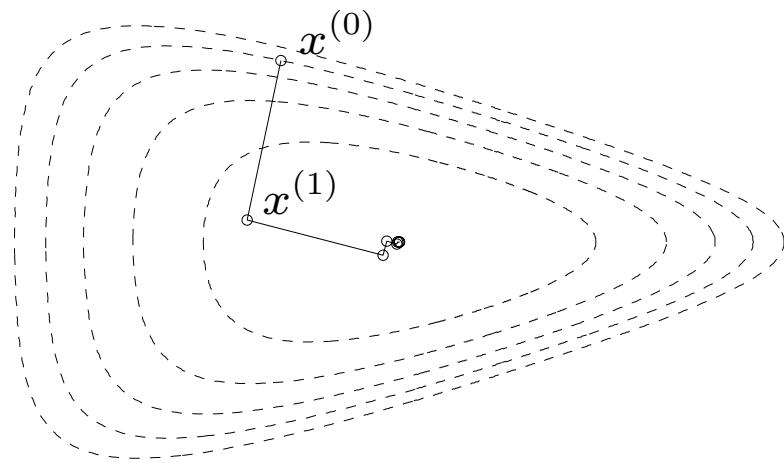


nonquadratic example

$$f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$$



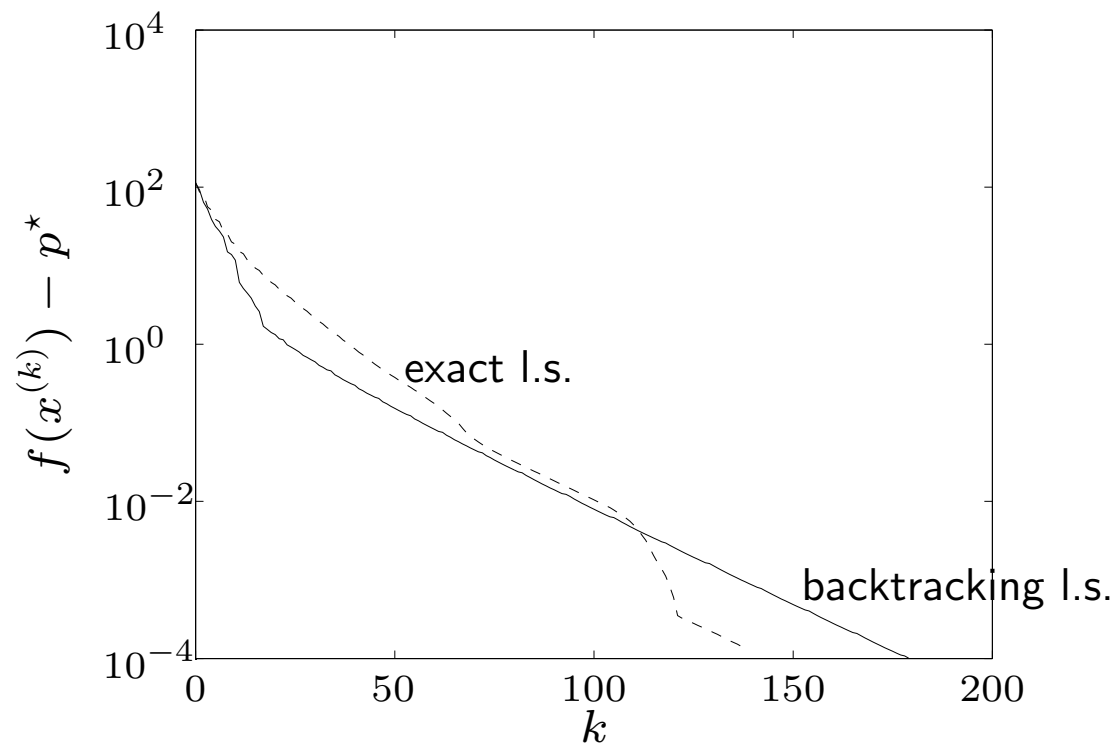
backtracking line search



exact line search

a problem in \mathbf{R}^{100}

$$f(x) = c^T x - \sum_{i=1}^{500} \log(b_i - a_i^T x)$$



‘linear’ convergence, *i.e.*, a straight line on a semilog plot

Steepest descent method

normalized steepest descent direction (at x , for norm $\|\cdot\|$):

$$\Delta x_{\text{nsd}} = \operatorname{argmin}\{\nabla f(x)^T v \mid \|v\| = 1\}$$

interpretation: for small v , $f(x + v) \approx f(x) + \nabla f(x)^T v$;
direction Δx_{nsd} is unit-norm step with most negative directional derivative

(unnormalized) steepest descent direction

$$\Delta x_{\text{sd}} = \|\nabla f(x)\|_* \Delta x_{\text{nsd}}$$

satisfies $\nabla f(x)^T \Delta x_{\text{sd}} = -\|\nabla f(x)\|_*^2$

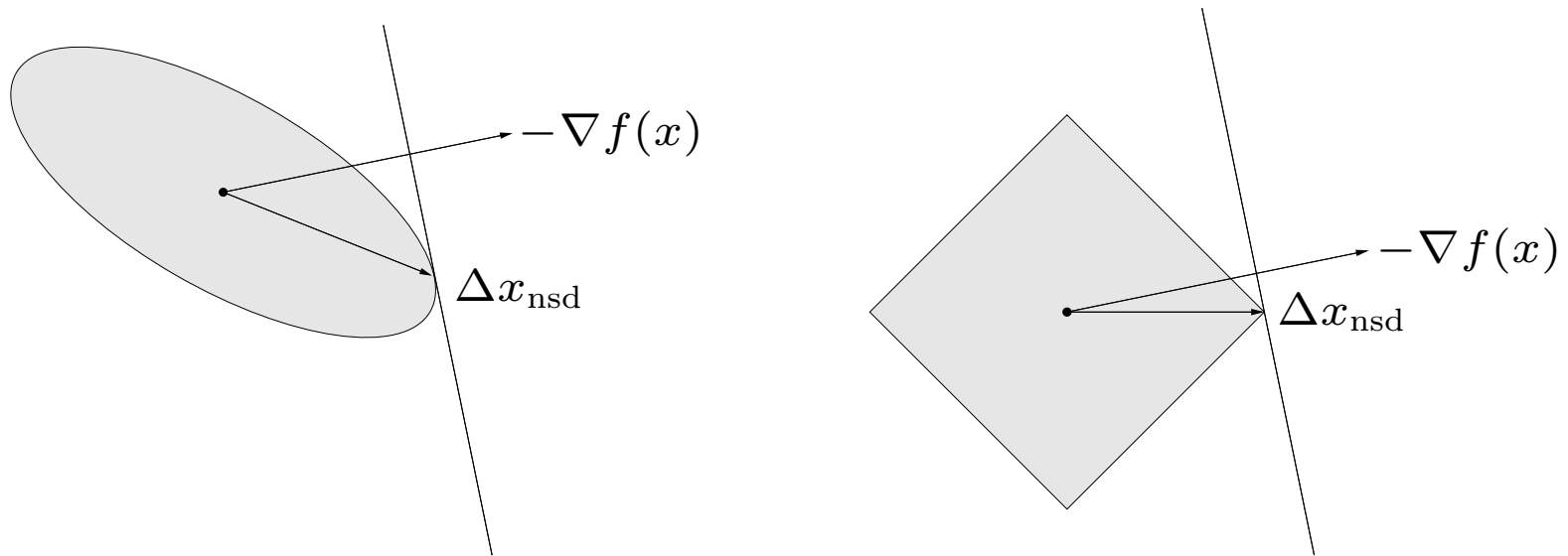
steepest descent method

- general descent method with $\Delta x = \Delta x_{\text{sd}}$
- convergence properties similar to gradient descent

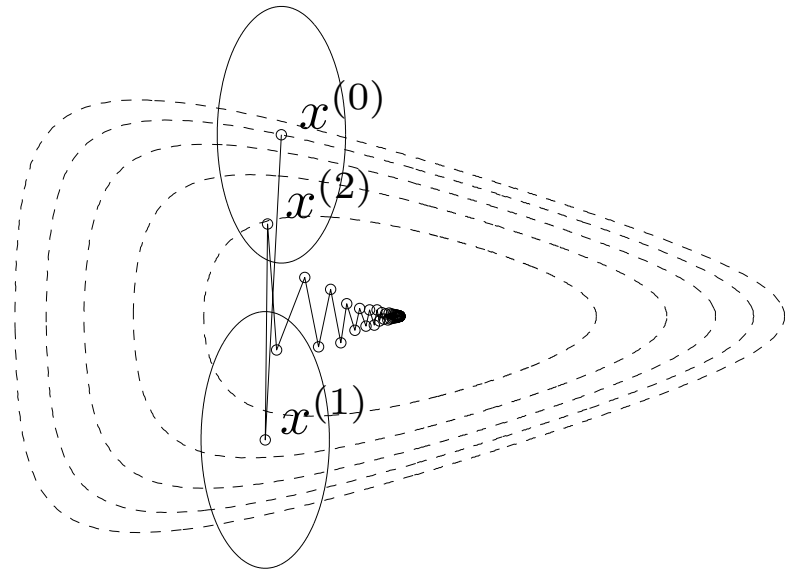
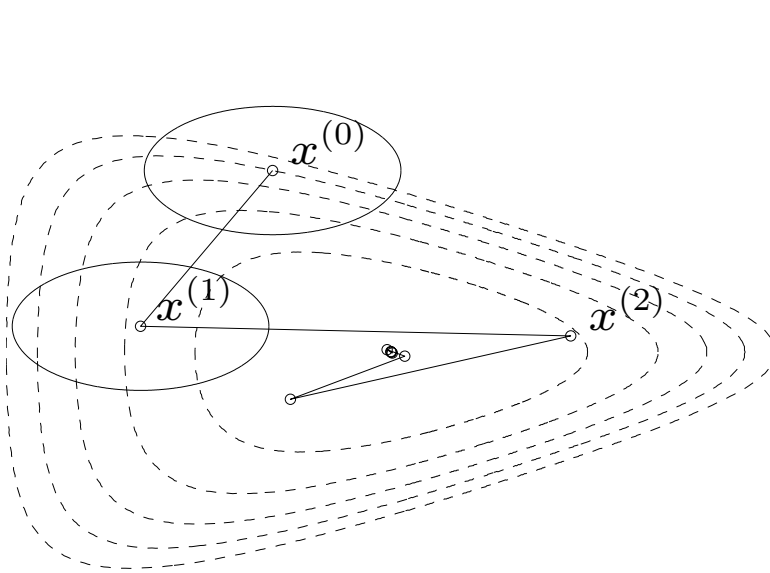
examples

- Euclidean norm: $\Delta x_{\text{sd}} = -\nabla f(x)$
- quadratic norm $\|x\|_P = (x^T P x)^{1/2}$ ($P \in \mathbf{S}_{++}^n$): $\Delta x_{\text{sd}} = -P^{-1} \nabla f(x)$
- ℓ_1 -norm: $\Delta x_{\text{sd}} = -(\partial f(x)/\partial x_i)e_i$, where $|\partial f(x)/\partial x_i| = \|\nabla f(x)\|_\infty$

unit balls and normalized steepest descent directions for a quadratic norm and the ℓ_1 -norm:



choice of norm for steepest descent



- steepest descent with backtracking line search for two quadratic norms
- ellipses show $\{x \mid \|x - x^{(k)}\|_P = 1\}$
- equivalent interpretation of steepest descent with quadratic norm $\|\cdot\|_P$: gradient descent after change of variables $\bar{x} = P^{1/2}x$

shows choice of P has strong effect on speed of convergence

Newton step

$$\Delta x_{\text{nt}} = -\nabla^2 f(x)^{-1} \nabla f(x)$$

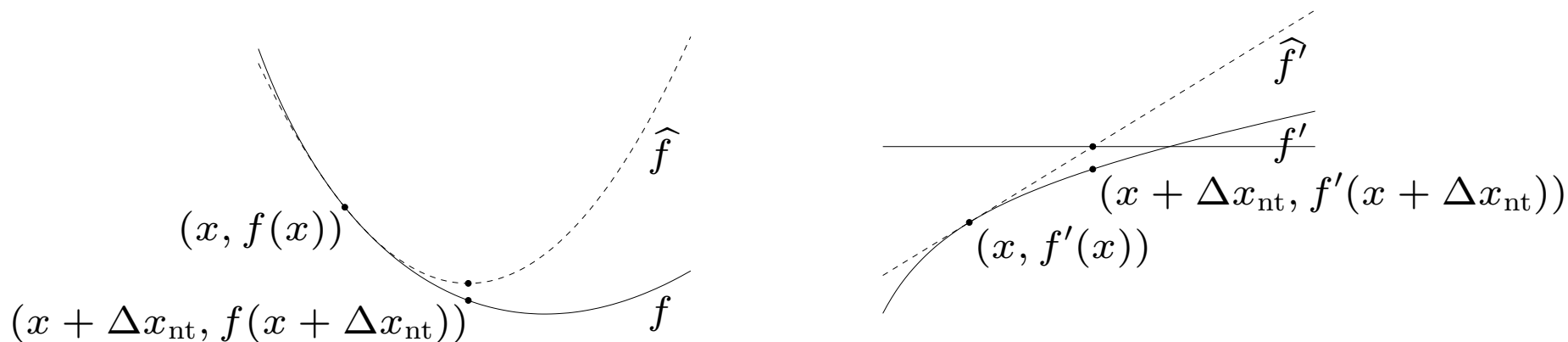
interpretations

- $x + \Delta x_{\text{nt}}$ minimizes second order approximation

$$\hat{f}(x + v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v$$

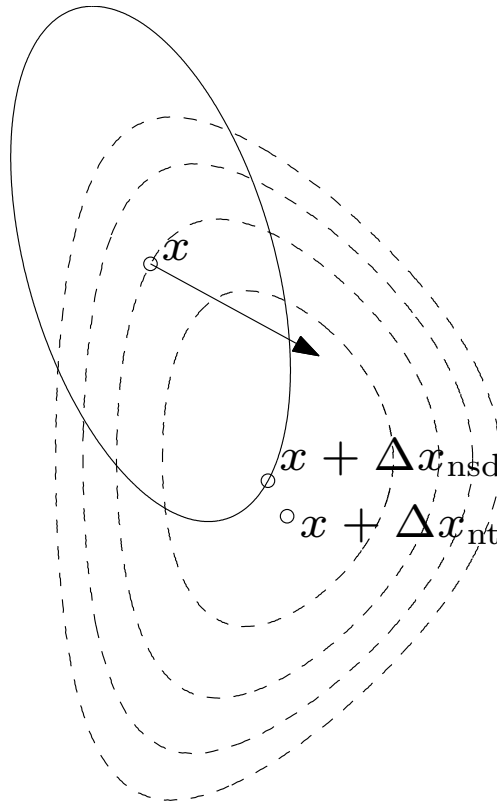
- $x + \Delta x_{\text{nt}}$ solves linearized optimality condition

$$\nabla f(x + v) \approx \nabla \hat{f}(x + v) = \nabla f(x) + \nabla^2 f(x) v = 0$$



- Δx_{nt} is steepest descent direction at x in local Hessian norm

$$\|u\|_{\nabla^2 f(x)} = (u^T \nabla^2 f(x) u)^{1/2}$$



dashed lines are contour lines of f ; ellipse is $\{x + v \mid v^T \nabla^2 f(x) v = 1\}$

arrow shows $-\nabla f(x)$

Newton decrement

$$\lambda(x) = \left(\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x) \right)^{1/2}$$

a measure of the proximity of x to x^\star

properties

- gives an estimate of $f(x) - p^\star$, using quadratic approximation \hat{f} :

$$f(x) - \inf_y \hat{f}(y) = \frac{1}{2} \lambda(x)^2$$

- equal to the norm of the Newton step in the quadratic Hessian norm

$$\lambda(x) = \left(\Delta x_{\text{nt}}^T \nabla^2 f(x) \Delta x_{\text{nt}} \right)^{1/2}$$

- directional derivative in the Newton direction: $\nabla f(x)^T \Delta x_{\text{nt}} = -\lambda(x)^2$
- affine invariant (unlike $\|\nabla f(x)\|_2$)

Newton's method

given a starting point $x \in \text{dom } f$, tolerance $\epsilon > 0$.

repeat

1. *Compute the Newton step and decrement.*

$$\Delta x_{\text{nt}} := -\nabla^2 f(x)^{-1} \nabla f(x); \quad \lambda^2 := \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x).$$

2. *Stopping criterion. quit* if $\lambda^2/2 \leq \epsilon$.

3. *Line search.* Choose step size t by backtracking line search.

4. *Update.* $x := x + t\Delta x_{\text{nt}}$.

affine invariant, *i.e.*, independent of linear changes of coordinates:

Newton iterates for $\tilde{f}(y) = f(Ty)$ with starting point $y^{(0)} = T^{-1}x^{(0)}$ are

$$y^{(k)} = T^{-1}x^{(k)}$$

Classical convergence analysis

assumptions

- f strongly convex on S with constant m
- $\nabla^2 f$ is Lipschitz continuous on S , with constant $L > 0$:

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq L\|x - y\|_2$$

(L measures how well f can be approximated by a quadratic function)

outline: there exist constants $\eta \in (0, m^2/L)$, $\gamma > 0$ such that

- if $\|\nabla f(x)\|_2 \geq \eta$, then $f(x^{(k+1)}) - f(x^{(k)}) \leq -\gamma$
- if $\|\nabla f(x)\|_2 < \eta$, then

$$\frac{L}{2m^2} \|\nabla f(x^{(k+1)})\|_2 \leq \left(\frac{L}{2m^2} \|\nabla f(x^{(k)})\|_2 \right)^2$$

damped Newton phase ($\|\nabla f(x)\|_2 \geq \eta$)

- most iterations require backtracking steps
- function value decreases by at least γ
- if $p^* > -\infty$, this phase ends after at most $(f(x^{(0)}) - p^*)/\gamma$ iterations

quadratically convergent phase ($\|\nabla f(x)\|_2 < \eta$)

- all iterations use step size $t = 1$
- $\|\nabla f(x)\|_2$ converges to zero quadratically: if $\|\nabla f(x^{(k)})\|_2 < \eta$, then

$$\frac{L}{2m^2} \|\nabla f(x^l)\|_2 \leq \left(\frac{L}{2m^2} \|\nabla f(x^k)\|_2 \right)^{2^{l-k}} \leq \left(\frac{1}{2} \right)^{2^{l-k}}, \quad l \geq k$$

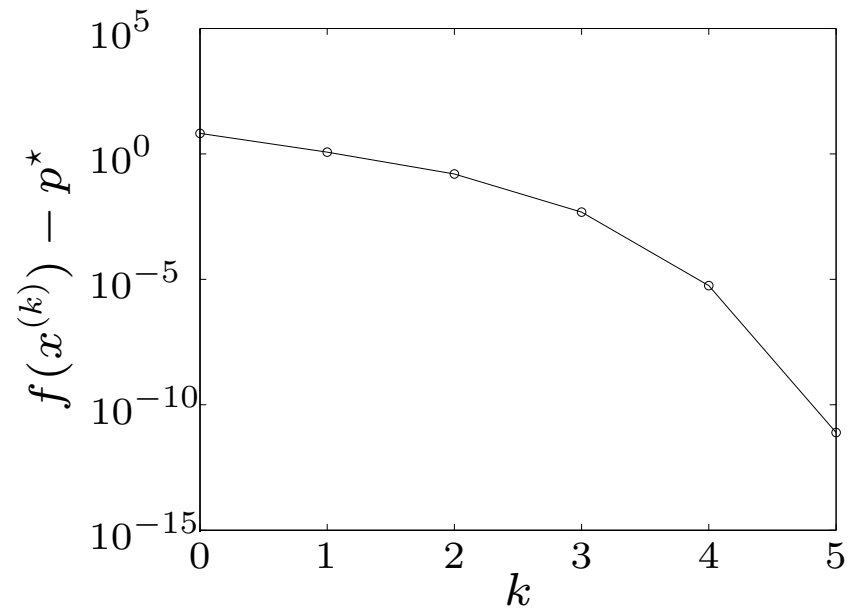
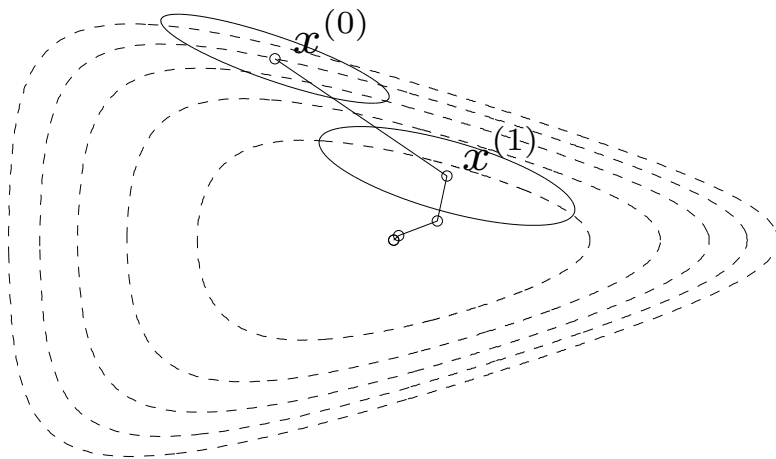
conclusion: number of iterations until $f(x) - p^* \leq \epsilon$ is bounded above by

$$\frac{f(x^{(0)}) - p^*}{\gamma} + \log_2 \log_2(\epsilon_0/\epsilon)$$

- γ, ϵ_0 are constants that depend on $m, L, x^{(0)}$
- second term is small (of the order of 6) and almost constant for practical purposes
- in practice, constants m, L (hence γ, ϵ_0) are usually unknown
- provides qualitative insight in convergence properties (*i.e.*, explains two algorithm phases)

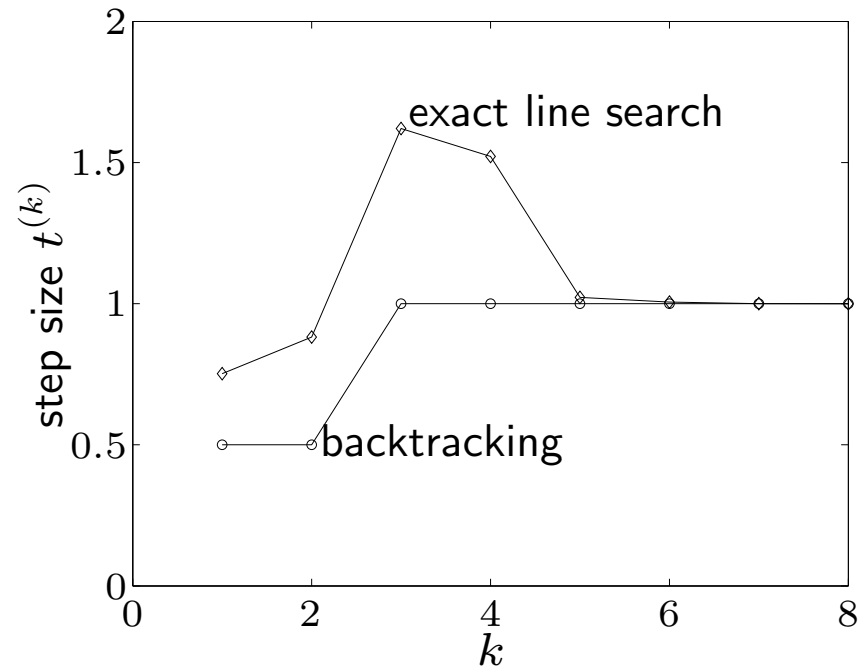
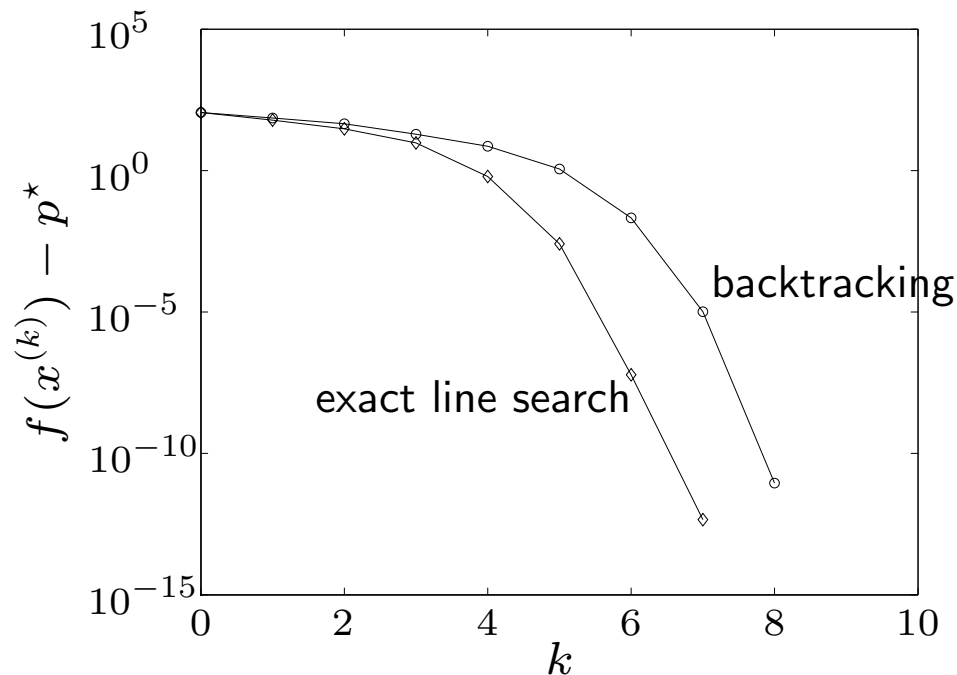
Examples

example in \mathbf{R}^2 (page 10–9)



- backtracking parameters $\alpha = 0.1$, $\beta = 0.7$
- converges in only 5 steps
- quadratic local convergence

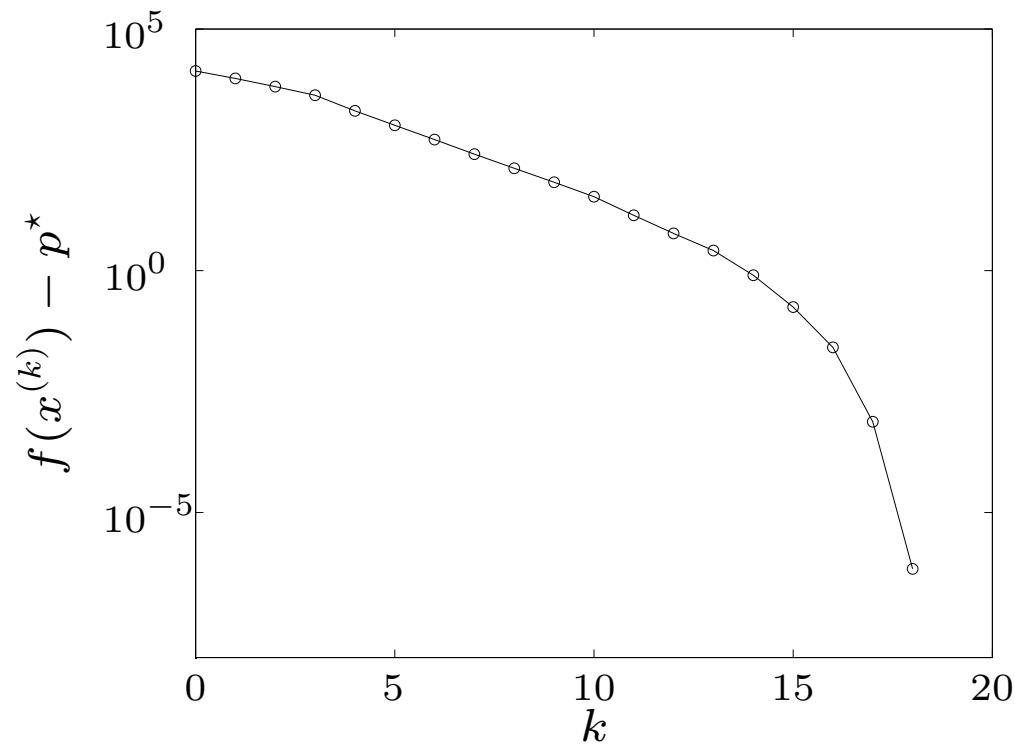
example in \mathbf{R}^{100} (page 10–10)



- backtracking parameters $\alpha = 0.01$, $\beta = 0.5$
- backtracking line search almost as fast as exact l.s. (and much simpler)
- clearly shows two phases in algorithm

example in \mathbf{R}^{10000} (with sparse a_i)

$$f(x) = - \sum_{i=1}^{10000} \log(1 - x_i^2) - \sum_{i=1}^{100000} \log(b_i - a_i^T x)$$



- backtracking parameters $\alpha = 0.01$, $\beta = 0.5$.
- performance similar as for small examples

Self-concordance

shortcomings of classical convergence analysis

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

convergence analysis via self-concordance (Nesterov and Nemirovski)

- does not depend on any unknown constants
- gives affine-invariant bound
- applies to special class of convex functions ('self-concordant' functions)
- developed to analyze polynomial-time interior-point methods for convex optimization

Self-concordant functions

definition

- $f : \mathbf{R} \rightarrow \mathbf{R}$ is self-concordant if $|f'''(x)| \leq 2f''(x)^{3/2}$ for all $x \in \text{dom } f$
- $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is self-concordant if $g(t) = f(x + tv)$ is self-concordant for all $x \in \text{dom } f$, $v \in \mathbf{R}^n$

examples on \mathbf{R}

- linear and quadratic functions
- negative logarithm $f(x) = -\log x$
- negative entropy plus negative logarithm: $f(x) = x \log x - \log x$

affine invariance: if $f : \mathbf{R} \rightarrow \mathbf{R}$ is s.c., then $\tilde{f}(y) = f(ay + b)$ is s.c.:

$$\tilde{f}'''(y) = a^3 f'''(ay + b), \quad \tilde{f}''(y) = a^2 f''(ay + b)$$

Self-concordant calculus

properties

- preserved under positive scaling $\alpha \geq 1$, and sum
- preserved under composition with affine function
- if g is convex with $\text{dom } g = \mathbf{R}_{++}$ and $|g'''(x)| \leq 3g''(x)/x$ then

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

is self-concordant

examples: properties can be used to show that the following are s.c.

- $f(x) = -\sum_{i=1}^m \log(b_i - a_i^T x)$ on $\{x \mid a_i^T x < b_i, i = 1, \dots, m\}$
- $f(X) = -\log \det X$ on \mathbf{S}_{++}^n
- $f(x) = -\log(y^2 - x^T x)$ on $\{(x, y) \mid \|x\|_2 < y\}$

Convergence analysis for self-concordant functions

summary: there exist constants $\eta \in (0, 1/4]$, $\gamma > 0$ such that

- if $\lambda(x) > \eta$, then

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

- if $\lambda(x) \leq \eta$, then

$$2\lambda(x^{(k+1)}) \leq \left(2\lambda(x^{(k)})\right)^2$$

(η and γ only depend on backtracking parameters α, β)

complexity bound: number of Newton iterations bounded by

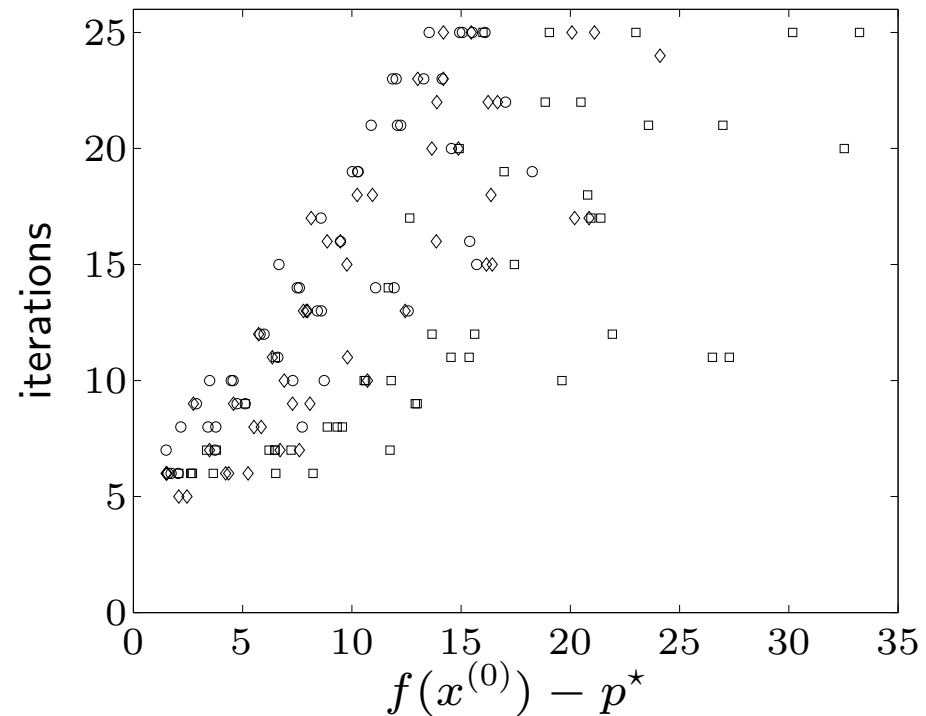
$$\frac{f(x^{(0)}) - p^*}{\gamma} + \log_2 \log_2(1/\epsilon)$$

for $\alpha = 0.1$, $\beta = 0.8$, $\epsilon = 10^{-10}$, bound evaluates to $375(f(x^{(0)}) - p^*) + 6$

numerical example: 150 randomly generated instances of

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

- : $m = 100, n = 50$
□: $m = 1000, n = 500$
◇: $m = 1000, n = 50$



- number of iterations much smaller than $375(f(x^{(0)}) - p^*) + 6$
- bound of the form $c(f(x^{(0)}) - p^*) + 6$ with smaller c (empirically) valid

Implementation

main effort in each iteration: evaluate derivatives and solve Newton system

$$H\Delta x = g$$

where $H = \nabla^2 f(x)$, $g = -\nabla f(x)$

via Cholesky factorization

$$H = LL^T, \quad \Delta x_{\text{nt}} = L^{-T}L^{-1}g, \quad \lambda(x) = \|L^{-1}g\|_2$$

- cost $(1/3)n^3$ flops for unstructured system
- cost $\ll (1/3)n^3$ if H sparse, banded

example of dense Newton system with structure

$$f(x) = \sum_{i=1}^n \psi_i(x_i) + \psi_0(Ax + b), \quad H = D + A^T H_0 A$$

- assume $A \in \mathbf{R}^{p \times n}$, dense, with $p \ll n$
- D diagonal with diagonal elements $\psi_i''(x_i)$; $H_0 = \nabla^2 \psi_0(Ax + b)$

method 1: form H , solve via dense Cholesky factorization: (cost $(1/3)n^3$)

method 2 (page 9–15): factor $H_0 = L_0 L_0^T$; write Newton system as

$$D\Delta x + A^T L_0 w = -g, \quad L_0^T A\Delta x - w = 0$$

eliminate Δx from first equation; compute w and Δx from

$$(I + L_0^T A D^{-1} A^T L_0)w = -L_0^T A D^{-1} g, \quad D\Delta x = -g - A^T L_0 w$$

cost: $2p^2n$ (dominated by computation of $L_0^T A D^{-1} A^T L_0$)

11. Equality constrained minimization

- equality constrained minimization
- eliminating equality constraints
- Newton's method with equality constraints
- infeasible start Newton method
- implementation

Equality constrained minimization

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

- f convex, twice continuously differentiable
- $A \in \mathbf{R}^{p \times n}$ with $\text{rank } A = p$
- we assume p^* is finite and attained

optimality conditions: x^* is optimal iff there exists a ν^* such that

$$\nabla f(x^*) + A^T \nu^* = 0, \quad Ax^* = b$$

equality constrained quadratic minimization (with $P \in \mathbf{S}_+^n$)

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P x + q^T x + r \\ \text{subject to} & Ax = b \end{array}$$

optimality condition:

$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \nu^* \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}$$

- coefficient matrix is called KKT matrix
- KKT matrix is nonsingular if and only if

$$Ax = 0, \quad x \neq 0 \quad \implies \quad x^T P x > 0$$

- equivalent condition for nonsingularity: $P + A^T A \succ 0$

Eliminating equality constraints

represent solution of $\{x \mid Ax = b\}$ as

$$\{x \mid Ax = b\} = \{Fz + \hat{x} \mid z \in \mathbf{R}^{n-p}\}$$

- \hat{x} is (any) particular solution
- range of $F \in \mathbf{R}^{n \times (n-p)}$ is nullspace of A ($\text{rank } F = n - p$ and $AF = 0$)

reduced or eliminated problem

$$\text{minimize } f(Fz + \hat{x})$$

- an unconstrained problem with variable $z \in \mathbf{R}^{n-p}$
- from solution z^* , obtain x^* and ν^* as

$$x^* = Fz^* + \hat{x}, \quad \nu^* = -(AA^T)^{-1}A\nabla f(x^*)$$

example: optimal allocation with resource constraint

$$\begin{array}{ll}\text{minimize} & f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n) \\ \text{subject to} & x_1 + x_2 + \cdots + x_n = b\end{array}$$

eliminate $x_n = b - x_1 - \cdots - x_{n-1}$, *i.e.*, choose

$$\hat{x} = be_n, \quad F = \begin{bmatrix} I \\ -\mathbf{1}^T \end{bmatrix} \in \mathbf{R}^{n \times (n-1)}$$

reduced problem:

$$\text{minimize} \quad f_1(x_1) + \cdots + f_{n-1}(x_{n-1}) + f_n(b - x_1 - \cdots - x_{n-1})$$

(variables x_1, \dots, x_{n-1})

Newton step

Newton step of f at feasible x is given by (1st block) of solution of

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{\text{nt}} \\ w \end{bmatrix} = \begin{bmatrix} -\nabla f(x) \\ 0 \end{bmatrix}$$

interpretations

- Δx_{nt} solves second order approximation (with variable v)

$$\begin{array}{ll} \text{minimize} & \hat{f}(x + v) = f(x) + \nabla f(x)^T v + (1/2)v^T \nabla^2 f(x)v \\ \text{subject to} & A(x + v) = b \end{array}$$

- equations follow from linearizing optimality conditions

$$\nabla f(x + \Delta x_{\text{nt}}) + A^T w = 0, \quad A(x + \Delta x_{\text{nt}}) = b$$

Newton decrement

$$\lambda(x) = (\Delta x_{\text{nt}}^T \nabla^2 f(x) \Delta x_{\text{nt}})^{1/2} = (-\nabla f(x)^T \Delta x_{\text{nt}})^{1/2}$$

properties

- gives an estimate of $f(x) - p^*$ using quadratic approximation \hat{f} :

$$f(x) - \inf_{Ay=b} \hat{f}(y) = \frac{1}{2} \lambda(x)^2$$

- directional derivative in Newton direction:

$$\left. \frac{d}{dt} f(x + t \Delta x_{\text{nt}}) \right|_{t=0} = -\lambda(x)^2$$

- in general, $\lambda(x) \neq (\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x))^{1/2}$

Newton's method with equality constraints

given starting point $x \in \text{dom } f$ with $Ax = b$, tolerance $\epsilon > 0$.

repeat

1. Compute the Newton step and decrement $\Delta x_{\text{nt}}, \lambda(x)$.
 2. *Stopping criterion.* **quit** if $\lambda^2/2 \leq \epsilon$.
 3. *Line search.* Choose step size t by backtracking line search.
 4. *Update.* $x := x + t\Delta x_{\text{nt}}$.
-

- a feasible descent method: $x^{(k)}$ feasible and $f(x^{(k+1)}) < f(x^{(k)})$
- affine invariant

Newton's method and elimination

Newton's method for reduced problem

$$\text{minimize } \tilde{f}(z) = f(Fz + \hat{x})$$

- variables $z \in \mathbf{R}^{n-p}$
- \hat{x} satisfies $A\hat{x} = b$; **rank** $F = n - p$ and $AF = 0$
- Newton's method for \tilde{f} , started at $z^{(0)}$, generates iterates $z^{(k)}$

Newton's method with equality constraints

when started at $x^{(0)} = Fz^{(0)} + \hat{x}$, iterates are

$$x^{(k+1)} = Fz^{(k)} + \hat{x}$$

hence, don't need separate convergence analysis

Newton step at infeasible points

2nd interpretation of page 11–6 extends to infeasible x (*i.e.*, $Ax \neq b$)

linearizing optimality conditions at infeasible x (with $x \in \text{dom } f$) gives

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{\text{nt}} \\ w \end{bmatrix} = - \begin{bmatrix} \nabla f(x) \\ Ax - b \end{bmatrix} \quad (1)$$

primal-dual interpretation

- write optimality condition as $r(y) = 0$, where

$$y = (x, \nu), \quad r(y) = (\nabla f(x) + A^T \nu, Ax - b)$$

- linearizing $r(y) = 0$ gives $r(y + \Delta y) \approx r(y) + Dr(y)\Delta y = 0$:

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{\text{nt}} \\ \Delta \nu_{\text{nt}} \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + A^T \nu \\ Ax - b \end{bmatrix}$$

same as (1) with $w = \nu + \Delta \nu_{\text{nt}}$

Infeasible start Newton method

given starting point $x \in \text{dom } f$, ν , tolerance $\epsilon > 0$, $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$.

repeat

1. Compute primal and dual Newton steps Δx_{nt} , $\Delta \nu_{\text{nt}}$.

2. *Backtracking line search* on $\|r\|_2$.

$t := 1$.

while $\|r(x + t\Delta x_{\text{nt}}, \nu + t\Delta \nu_{\text{nt}})\|_2 > (1 - \alpha t)\|r(x, \nu)\|_2$, $t := \beta t$.

3. *Update*. $x := x + t\Delta x_{\text{nt}}$, $\nu := \nu + t\Delta \nu_{\text{nt}}$.

until $Ax = b$ and $\|r(x, \nu)\|_2 \leq \epsilon$.

- not a descent method: $f(x^{(k+1)}) > f(x^{(k)})$ is possible
- directional derivative of $\|r(y)\|_2^2$ in direction $\Delta y = (\Delta x_{\text{nt}}, \Delta \nu_{\text{nt}})$ is

$$\left. \frac{d}{dt} \|r(y + \Delta y)\|_2^2 \right|_{t=0} = -\|r(y)\|_2^2$$

Solving KKT systems

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = - \begin{bmatrix} g \\ h \end{bmatrix}$$

solution methods

- LDL^T factorization
- elimination (if H nonsingular)

$$AH^{-1}A^Tw = h - AH^{-1}g, \quad Hv = -(g + A^Tw)$$

- elimination with singular H : write as

$$\begin{bmatrix} H + A^TQA & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = - \begin{bmatrix} g + A^TQh \\ h \end{bmatrix}$$

with $Q \succeq 0$ for which $H + A^TQA \succ 0$, and apply elimination

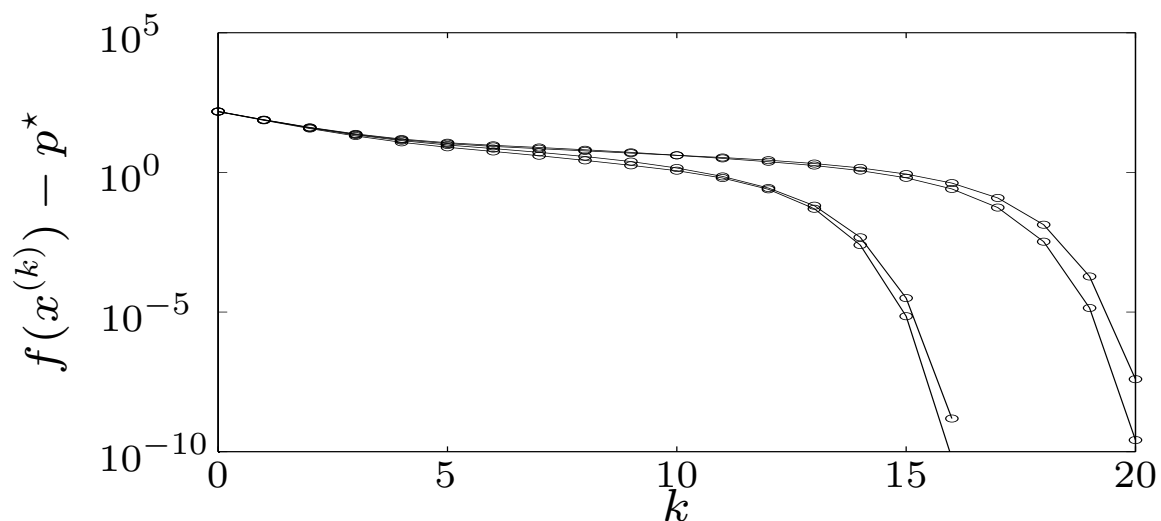
Equality constrained analytic centering

primal problem: minimize $-\sum_{i=1}^n \log x_i$ subject to $Ax = b$

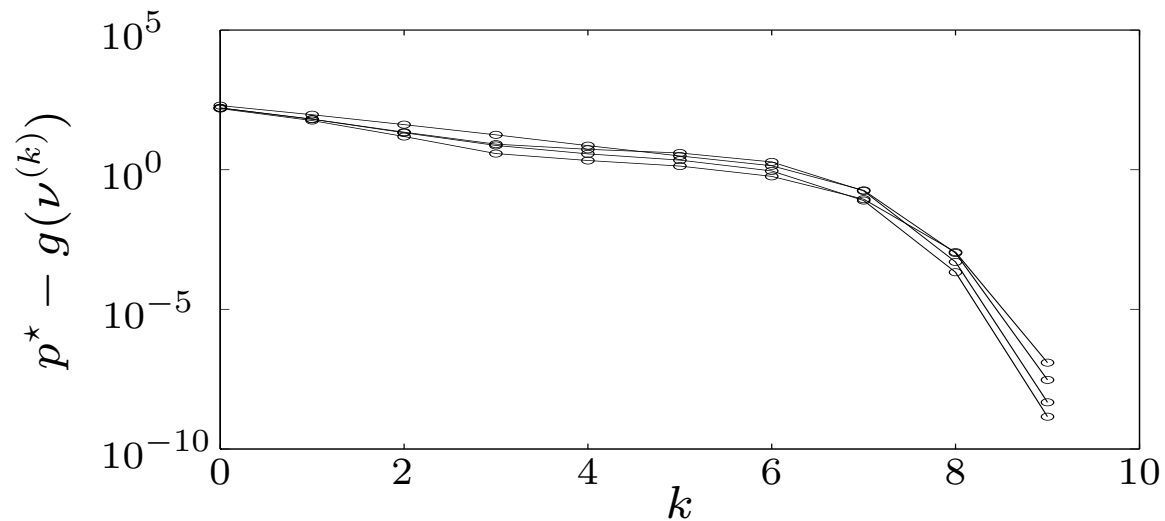
dual problem: maximize $-b^T \nu + \sum_{i=1}^n \log(A^T \nu)_i + n$

three methods for an example with $A \in \mathbf{R}^{100 \times 500}$, different starting points

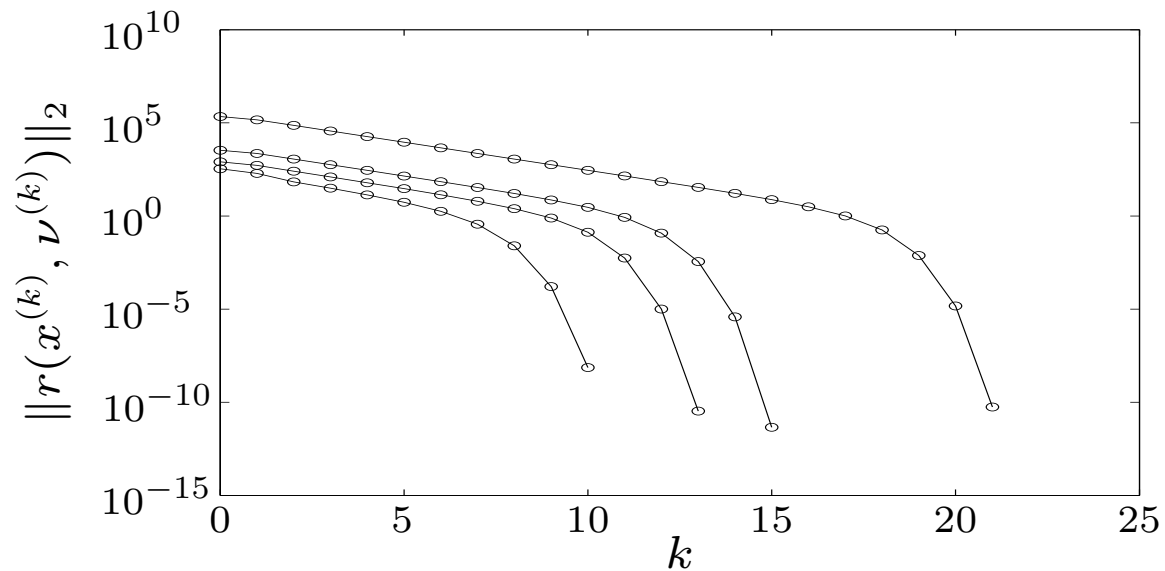
1. Newton method with equality constraints (requires $x^{(0)} \succ 0$, $Ax^{(0)} = b$)



2. Newton method applied to dual problem (requires $A^T \nu^{(0)} \succ 0$)



3. infeasible start Newton method (requires $x^{(0)} \succ 0$)



complexity per iteration of three methods is identical

1. use block elimination to solve KKT system

$$\begin{bmatrix} \mathbf{diag}(x)^{-2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ w \end{bmatrix} = \begin{bmatrix} \mathbf{diag}(x)^{-1} \mathbf{1} \\ 0 \end{bmatrix}$$

reduces to solving $A \mathbf{diag}(x)^2 A^T w = b$

2. solve Newton system $A \mathbf{diag}(A^T \nu)^{-2} A^T \Delta \nu = -b + A \mathbf{diag}(A^T \nu)^{-1} \mathbf{1}$

3. use block elimination to solve KKT system

$$\begin{bmatrix} \mathbf{diag}(x)^{-2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta \nu \end{bmatrix} = \begin{bmatrix} \mathbf{diag}(x)^{-1} \mathbf{1} \\ Ax - b \end{bmatrix}$$

reduces to solving $A \mathbf{diag}(x)^2 A^T w = 2Ax - b$

conclusion: in each case, solve $ADA^T w = h$ with D positive diagonal

Network flow optimization

$$\begin{array}{ll}\text{minimize} & \sum_{i=1}^n \phi_i(x_i) \\ \text{subject to} & Ax = b\end{array}$$

- directed graph with n arcs, $p + 1$ nodes
- x_i : flow through arc i ; ϕ_i : cost flow function for arc i (with $\phi_i''(x) > 0$)
- node-incidence matrix $\tilde{A} \in \mathbf{R}^{(p+1) \times n}$ defined as

$$\tilde{A}_{ij} = \begin{cases} 1 & \text{arc } j \text{ leaves node } i \\ -1 & \text{arc } j \text{ enters node } i \\ 0 & \text{otherwise} \end{cases}$$

- reduced node-incidence matrix $A \in \mathbf{R}^{p \times n}$ is \tilde{A} with last row removed
- $b \in \mathbf{R}^p$ is (reduced) source vector
- **rank** $A = p$ if graph is connected

KKT system

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = - \begin{bmatrix} g \\ h \end{bmatrix}$$

- $H = \mathbf{diag}(\phi_1''(x_1), \dots, \phi_n''(x_n))$, positive diagonal
- solve via elimination:

$$AH^{-1}A^T w = h - AH^{-1}g, \quad Hv = -(g + A^T w)$$

sparsity pattern of coefficient matrix is given by graph connectivity

$$\begin{aligned} (AH^{-1}A^T)_{ij} \neq 0 &\iff (AA^T)_{ij} \neq 0 \\ &\iff \text{nodes } i \text{ and } j \text{ are connected by an arc} \end{aligned}$$

Analytic center of linear matrix inequality

$$\begin{array}{ll}\text{minimize} & -\log \det X \\ \text{subject to} & \text{tr}(A_i X) = b_i, \quad i = 1, \dots, p\end{array}$$

variable $X \in \mathbf{S}^n$

optimality conditions

$$X^* \succ 0, \quad -(X^*)^{-1} + \sum_{j=1}^p \nu_j^* A_j = 0, \quad \text{tr}(A_i X^*) = b_i, \quad i = 1, \dots, p$$

Newton equation at feasible X :

$$X^{-1} \Delta X X^{-1} + \sum_{j=1}^p w_j A_j = X^{-1}, \quad \text{tr}(A_i \Delta X) = 0, \quad i = 1, \dots, p$$

- follows from linear approximation $(X + \Delta X)^{-1} \approx X^{-1} - X^{-1} \Delta X X^{-1}$
- $n(n+1)/2 + p$ variables $\Delta X, w$

solution by block elimination

- eliminate ΔX from first equation: $\Delta X = X - \sum_{j=1}^p w_j X A_j X$
- substitute ΔX in second equation

$$\sum_{j=1}^p \text{tr}(A_i X A_j X) w_j = b_i, \quad i = 1, \dots, p \quad (2)$$

a dense positive definite set of linear equations with variable $w \in \mathbf{R}^p$

flop count (dominant terms) using Cholesky factorization $X = LL^T$:

- form p products $L^T A_j L$: $(3/2)pn^3$
- form $p(p+1)/2$ inner products $\text{tr}((L^T A_i L)(L^T A_j L))$: $(1/2)p^2 n^2$
- solve (2) via Cholesky factorization: $(1/3)p^3$

12. Interior-point methods

- inequality constrained minimization
- logarithmic barrier function and central path
- barrier method
- feasibility and phase I methods
- complexity analysis via self-concordance
- generalized inequalities

Inequality constrained minimization

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

- f_i convex, twice continuously differentiable
- $A \in \mathbf{R}^{p \times n}$ with $\text{rank } A = p$
- we assume p^* is finite and attained
- we assume problem is strictly feasible: there exists \tilde{x} with

$$\tilde{x} \in \text{dom } f_0, \quad f_i(\tilde{x}) < 0, \quad i = 1, \dots, m, \quad A\tilde{x} = b$$

hence, strong duality holds and dual optimum is attained

Examples

- LP, QP, QCQP, GP
- entropy maximization with linear inequality constraints

$$\begin{array}{ll}\text{minimize} & \sum_{i=1}^n x_i \log x_i \\ \text{subject to} & Fx \preceq g \\ & Ax = b\end{array}$$

with $\text{dom } f_0 = \mathbf{R}_{++}^n$

- differentiability may require reformulating the problem, *e.g.*, piecewise-linear minimization or ℓ_∞ -norm approximation via LP
- SDPs and SOCPs are better handled as problems with generalized inequalities (see later)

Logarithmic barrier

reformulation of (1) via indicator function:

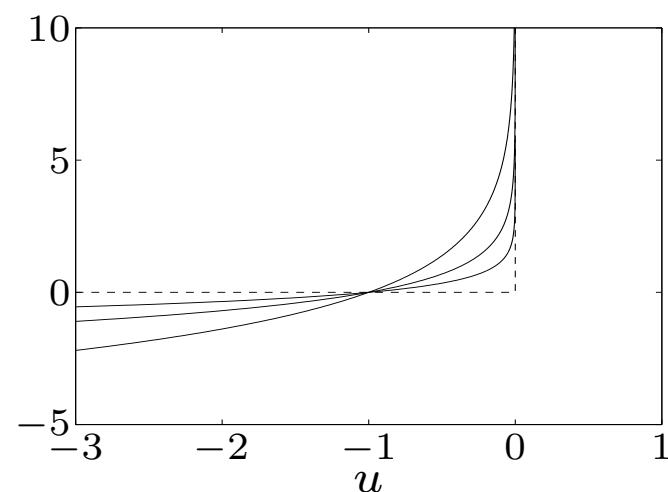
$$\begin{array}{ll} \text{minimize} & f_0(x) + \sum_{i=1}^m I_-(f_i(x)) \\ \text{subject to} & Ax = b \end{array}$$

where $I_-(u) = 0$ if $u \leq 0$, $I_-(u) = \infty$ otherwise (indicator function of \mathbf{R}_-)

approximation via logarithmic barrier

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

- an equality constrained problem
- for $t > 0$, $-(1/t) \log(-u)$ is a smooth approximation of I_-
- approximation improves as $t \rightarrow \infty$



logarithmic barrier function

$$\phi(x) = -\sum_{i=1}^m \log(-f_i(x)), \quad \mathbf{dom} \phi = \{x \mid f_1(x) < 0, \dots, f_m(x) < 0\}$$

- convex (follows from composition rules)
- twice continuously differentiable, with derivatives

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

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

Central path

- for $t > 0$, define $x^*(t)$ as the solution of

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

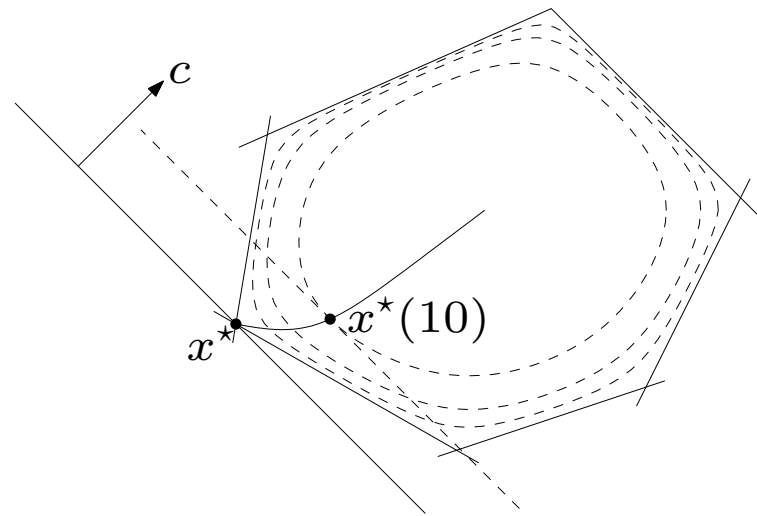
(for now, assume $x^*(t)$ exists and is unique for each $t > 0$)

- central path is $\{x^*(t) \mid t > 0\}$

example: central path for an LP

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

hyperplane $c^T x = c^T x^*(t)$ is tangent to level curve of ϕ through $x^*(t)$



Dual points on central path

$x = x^*(t)$ if there exists a w such that

$$t\nabla f_0(x) + \sum_{i=1}^m \frac{1}{-f_i(x)} \nabla f_i(x) + A^T w = 0, \quad Ax = b$$

- therefore, $x^*(t)$ minimizes the Lagrangian

$$L(x, \lambda^*(t), \nu^*(t)) = f_0(x) + \sum_{i=1}^m \lambda_i^*(t) f_i(x) + \nu^*(t)^T (Ax - b)$$

where we define $\lambda_i^*(t) = 1/(-tf_i(x^*(t)))$ and $\nu^*(t) = w/t$

- this confirms the intuitive idea that $f_0(x^*(t)) \rightarrow p^*$ if $t \rightarrow \infty$:

$$\begin{aligned} p^* &\geq g(\lambda^*(t), \nu^*(t)) \\ &= L(x^*(t), \lambda^*(t), \nu^*(t)) \\ &= f_0(x^*(t)) - m/t \end{aligned}$$

Interpretation via KKT conditions

$x = x^*(t)$, $\lambda = \lambda^*(t)$, $\nu = \nu^*(t)$ satisfy

1. primal constraints: $f_i(x) \leq 0$, $i = 1, \dots, m$, $Ax = b$
2. dual constraints: $\lambda \succeq 0$
3. approximate complementary slackness: $-\lambda_i f_i(x) = 1/t$, $i = 1, \dots, m$
4. gradient of Lagrangian with respect to x vanishes:

$$\nabla f_0(x) + \sum_{i=1}^m \lambda_i \nabla f_i(x) + A^T \nu = 0$$

difference with KKT is that condition 3 replaces $\lambda_i f_i(x) = 0$

Force field interpretation

centering problem (for problem with no equality constraints)

$$\text{minimize } tf_0(x) - \sum_{i=1}^m \log(-f_i(x))$$

force field interpretation

- $tf_0(x)$ is potential of force field $F_0(x) = -t\nabla f_0(x)$
- $-\log(-f_i(x))$ is potential of force field $F_i(x) = (1/f_i(x))\nabla f_i(x)$

the forces balance at $x^*(t)$:

$$F_0(x^*(t)) + \sum_{i=1}^m F_i(x^*(t)) = 0$$

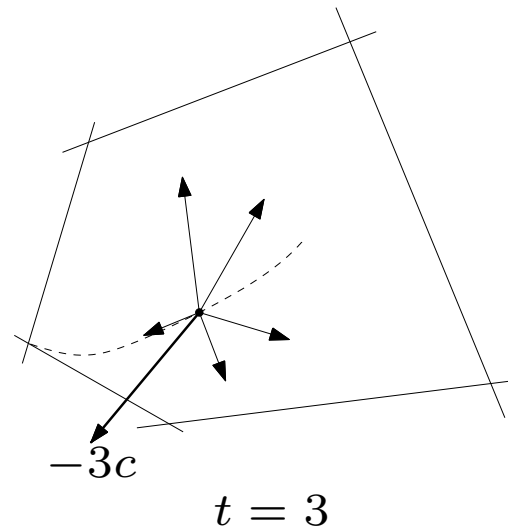
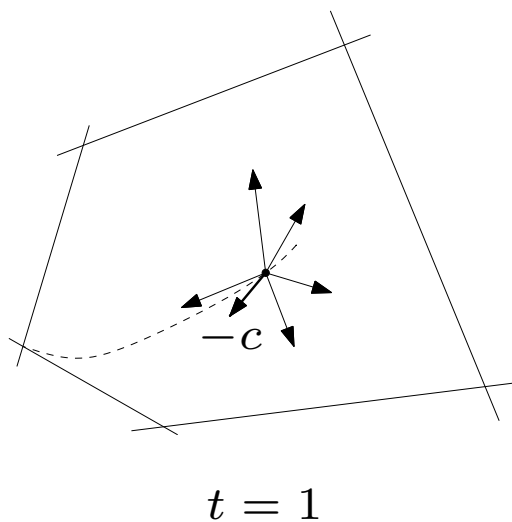
example

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

- objective force field is constant: $F_0(x) = -tc$
- constraint force field decays as inverse distance to constraint hyperplane:

$$F_i(x) = \frac{-a_i}{b_i - a_i^T x}, \quad \|F_i(x)\|_2 = \frac{1}{\mathbf{dist}(x, \mathcal{H}_i)}$$

where $\mathcal{H}_i = \{x \mid a_i^T x = b_i\}$



Barrier method

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

repeat

1. *Centering step.* Compute $x^*(t)$ by minimizing $tf_0 + \phi$, subject to $Ax = b$.
 2. *Update.* $x := x^*(t)$.
 3. *Stopping criterion.* **quit** if $m/t < \epsilon$.
 4. *Increase t .* $t := \mu t$.
-

- terminates with $f_0(x) - p^* \leq \epsilon$ (stopping criterion follows from $f_0(x^*(t)) - p^* \leq m/t$)
- centering usually done using Newton's method, starting at current x
- choice of μ involves a trade-off: large μ means fewer outer iterations, more inner (Newton) iterations; typical values: $\mu = 10$ – 20
- several heuristics for choice of $t^{(0)}$

Convergence analysis

number of outer (centering) iterations: exactly

$$\left\lceil \frac{\log(m/(\epsilon t^{(0)}))}{\log \mu} \right\rceil$$

plus the initial centering step (to compute $x^*(t^{(0)})$)

centering problem

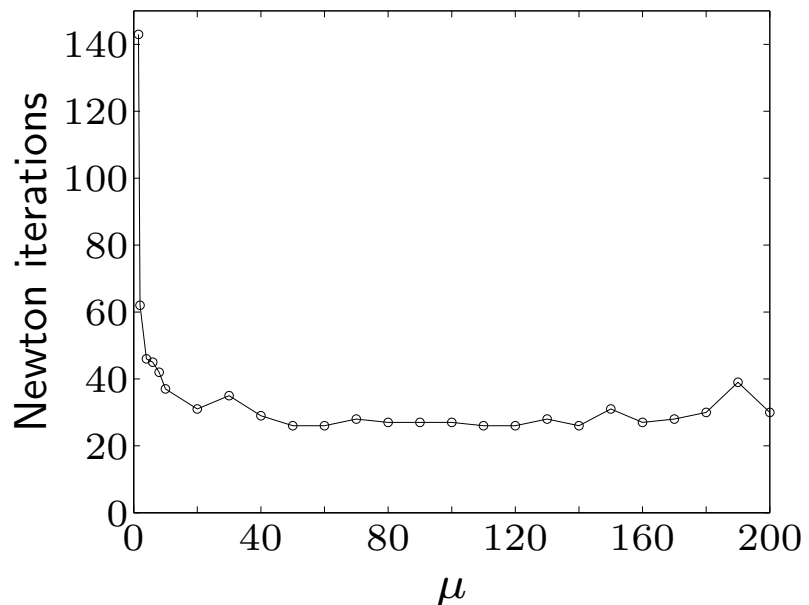
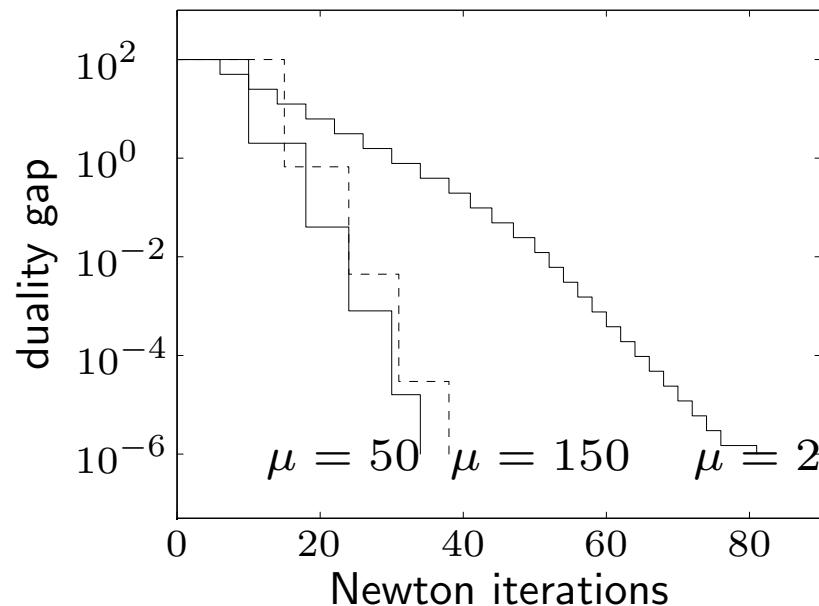
$$\text{minimize } tf_0(x) + \phi(x)$$

see convergence analysis of Newton's method

- $tf_0 + \phi$ must have closed sublevel sets for $t \geq t^{(0)}$
- classical analysis requires strong convexity, Lipschitz condition
- analysis via self-concordance requires self-concordance of $tf_0 + \phi$

Examples

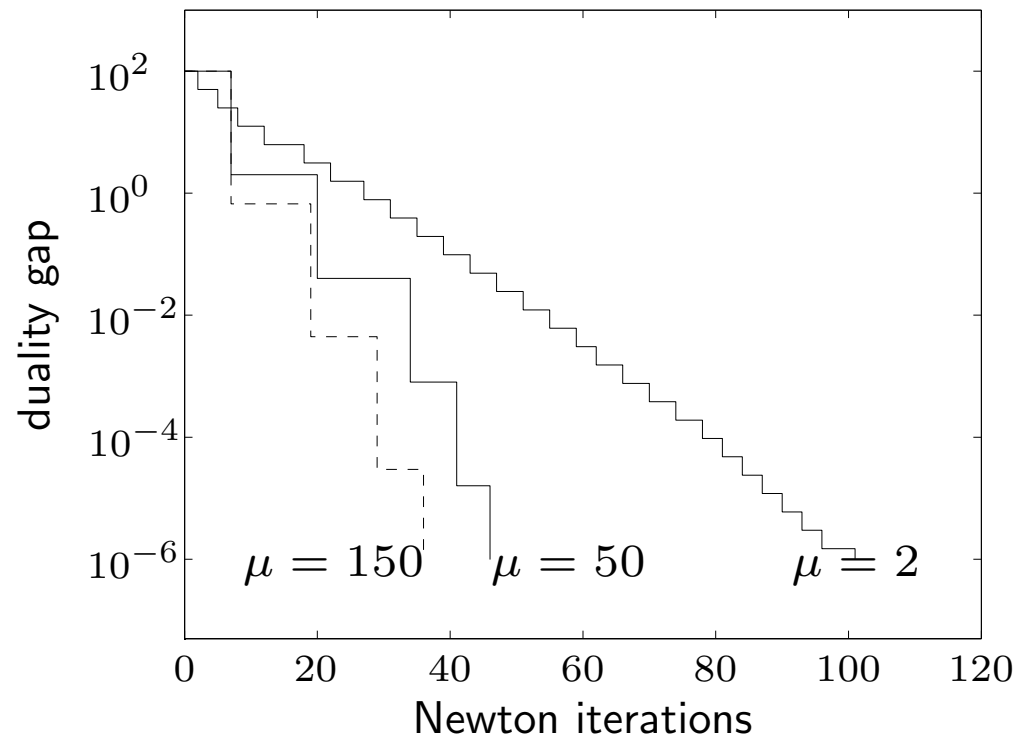
inequality form LP ($m = 100$ inequalities, $n = 50$ variables)



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

geometric program ($m = 100$ inequalities and $n = 50$ variables)

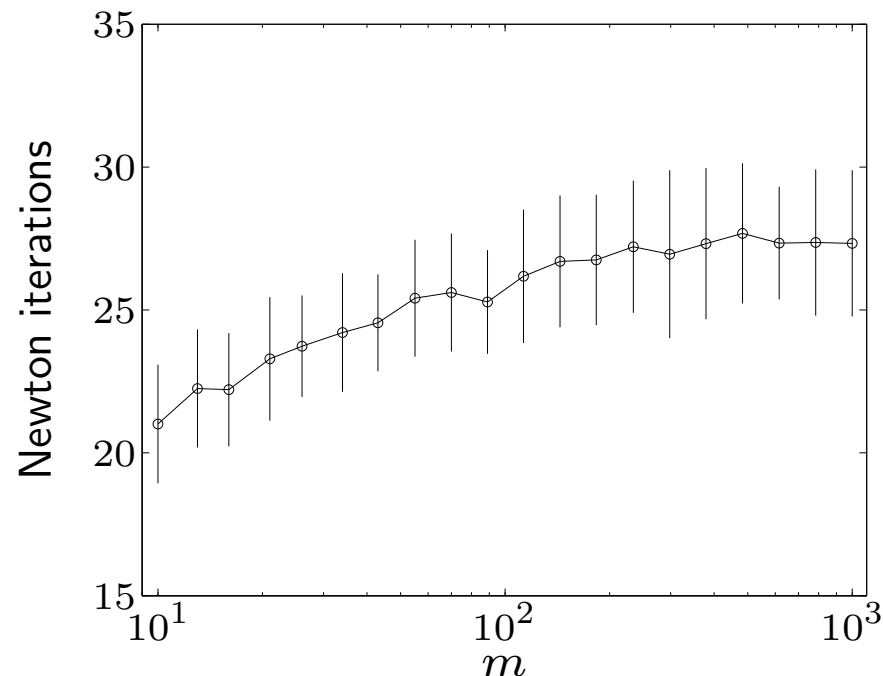
$$\begin{array}{ll} \text{minimize} & \log \left(\sum_{k=1}^5 \exp(a_{0k}^T x + b_{0k}) \right) \\ \text{subject to} & \log \left(\sum_{k=1}^5 \exp(a_{ik}^T x + b_{ik}) \right) \leq 0, \quad i = 1, \dots, m \end{array}$$



family of standard LPs ($A \in \mathbf{R}^{m \times 2m}$)

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

$m = 10, \dots, 1000$; for each m , solve 100 randomly generated instances



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

Feasibility and phase I methods

feasibility problem: find x such that

$$f_i(x) \leq 0, \quad i = 1, \dots, m, \quad Ax = b \quad (2)$$

phase I: computes strictly feasible starting point for barrier method

basic phase I method

$$\begin{array}{ll} \text{minimize (over } x, s) & s \\ \text{subject to} & f_i(x) \leq s, \quad i = 1, \dots, m \\ & Ax = b \end{array} \quad (3)$$

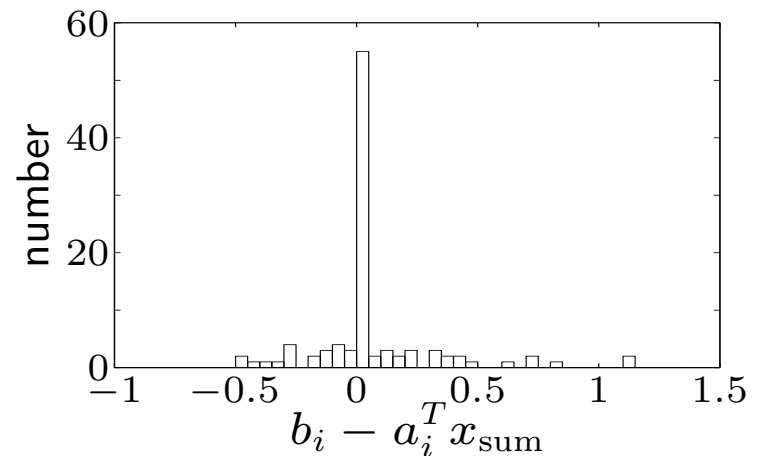
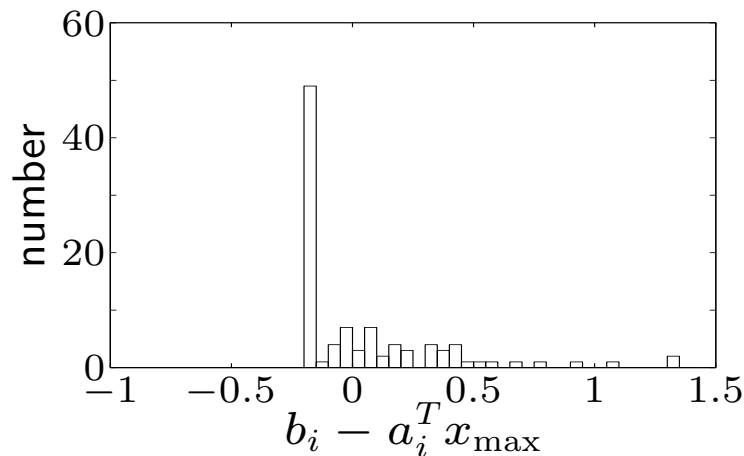
- if x, s feasible, with $s < 0$, then x is strictly feasible for (2)
- if optimal value \bar{p}^* of (3) is positive, then problem (2) is infeasible
- if $\bar{p}^* = 0$ and attained, then problem (2) is feasible (but not strictly);
if $\bar{p}^* = 0$ and not attained, then problem (2) is infeasible

sum of infeasibilities phase I method

$$\begin{array}{ll}\text{minimize} & \mathbf{1}^T s \\ \text{subject to} & s \succeq 0, \quad f_i(x) \leq s_i, \quad i = 1, \dots, m \\ & Ax = b\end{array}$$

for infeasible problems, produces a solution that satisfies many more inequalities than basic phase I method

example (infeasible set of 100 linear inequalities in 50 variables)

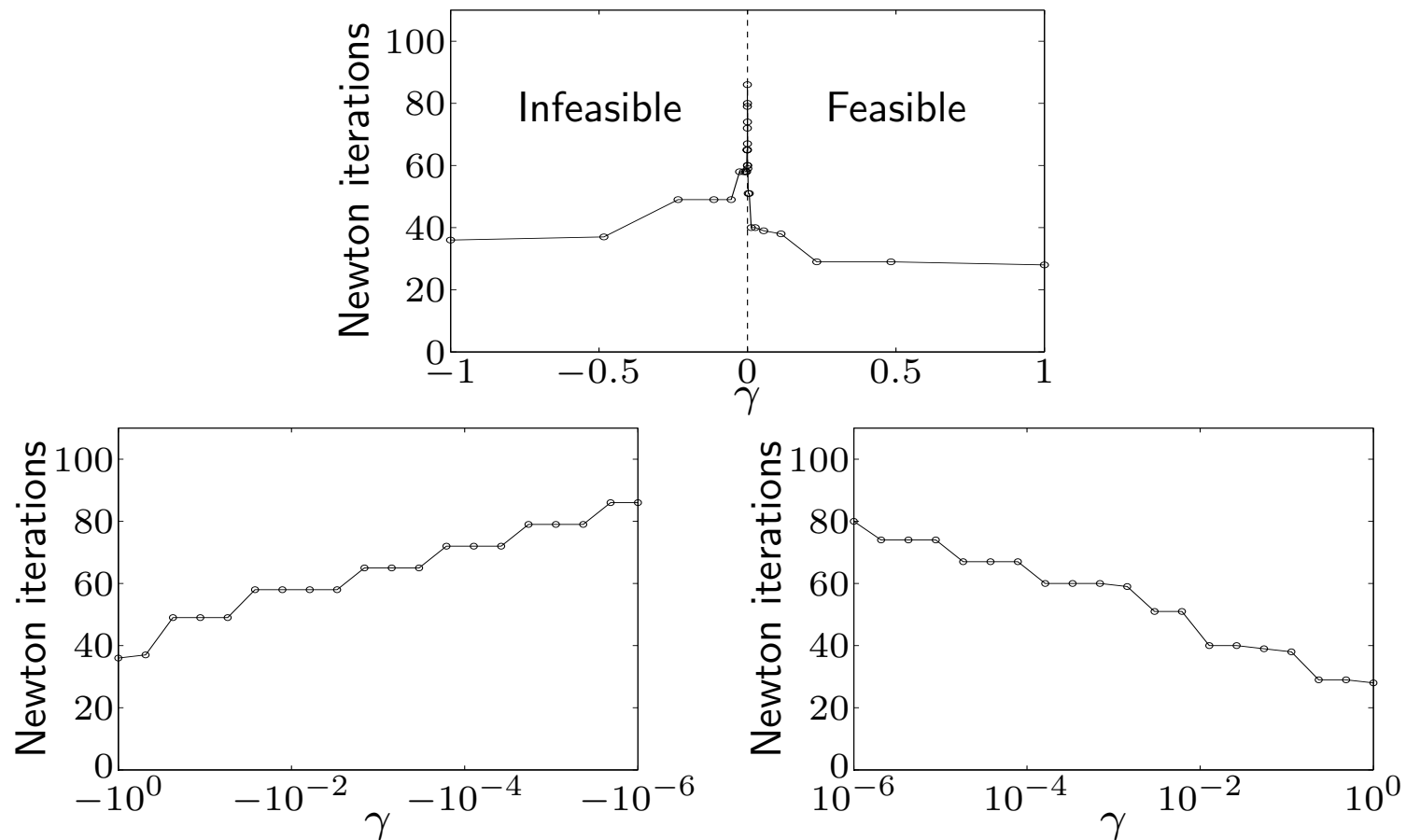


left: basic phase I solution; satisfies 39 inequalities

right: sum of infeasibilities phase I solution; satisfies 79 solutions

example: family of linear inequalities $Ax \preceq b + \gamma \Delta b$

- data chosen to be strictly feasible for $\gamma > 0$, infeasible for $\gamma \leq 0$
- use basic phase I, terminate when $s < 0$ or dual objective is positive



number of iterations roughly proportional to $\log(1/|\gamma|)$

Complexity analysis via self-concordance

same assumptions as on page 12–2, plus:

- sublevel sets (of f_0 , on the feasible set) are bounded
- $tf_0 + \phi$ is self-concordant with closed sublevel sets

second condition

- holds for LP, QP, QCQP
- may require reformulating the problem, *e.g.*,

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n x_i \log x_i \\ \text{subject to} & Fx \preceq g \end{array} \quad \longrightarrow \quad \begin{array}{ll} \text{minimize} & \sum_{i=1}^n x_i \log x_i \\ \text{subject to} & Fx \preceq g, \quad x \succeq 0 \end{array}$$

- needed for complexity analysis; barrier method works even when self-concordance assumption does not apply

Newton iterations per centering step: from self-concordance theory

$$\# \text{Newton iterations} \leq \frac{\mu t f_0(x) + \phi(x) - \mu t f_0(x^+) - \phi(x^+)}{\gamma} + c$$

- bound on effort of computing $x^+ = x^*(\mu t)$ starting at $x = x^*(t)$
- γ, c are constants (depend only on Newton algorithm parameters)
- from duality (with $\lambda = \lambda^*(t), \nu = \nu^*(t)$):

$$\begin{aligned} & \mu t f_0(x) + \phi(x) - \mu t f_0(x^+) - \phi(x^+) \\ &= \mu t f_0(x) - \mu t f_0(x^+) + \sum_{i=1}^m \log(-\mu t \lambda_i f_i(x^+)) - m \log \mu \\ &\leq \mu t f_0(x) - \mu t f_0(x^+) - \mu t \sum_{i=1}^m \lambda_i f_i(x^+) - m - m \log \mu \\ &\leq \mu t f_0(x) - \mu t g(\lambda, \nu) - m - m \log \mu \\ &= m(\mu - 1 - \log \mu) \end{aligned}$$

total number of Newton iterations (excluding first centering step)

$$\# \text{Newton iterations} \leq N = \left\lceil \frac{\log(m/(t^{(0)}\epsilon))}{\log \mu} \right\rceil \left(\frac{m(\mu - 1 - \log \mu)}{\gamma} + c \right)$$

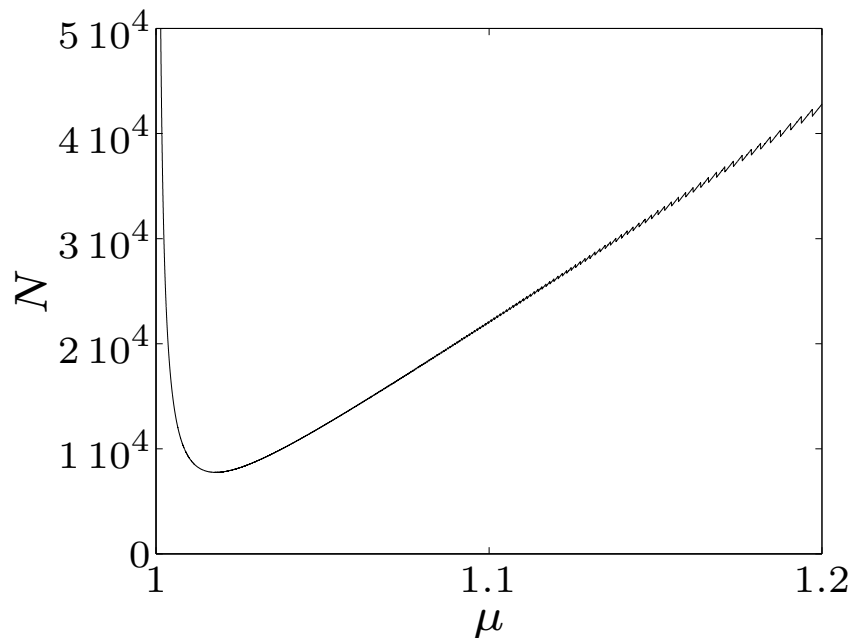


figure shows N for typical values of γ , c ,

$$m = 100, \quad \frac{m}{t^{(0)}\epsilon} = 10^5$$

- confirms trade-off in choice of μ
- in practice, #iterations is in the tens; not very sensitive for $\mu \geq 10$

polynomial-time complexity of barrier method

- for $\mu = 1 + 1/\sqrt{m}$:

$$N = O \left(\sqrt{m} \log \left(\frac{m/t^{(0)}}{\epsilon} \right) \right)$$

- number of Newton iterations for fixed gap reduction is $O(\sqrt{m})$
- multiply with cost of one Newton iteration (a polynomial function of problem dimensions), to get bound on number of flops

this choice of μ optimizes worst-case complexity; in practice we choose μ fixed ($\mu = 10, \dots, 20$)

Generalized inequalities

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

- f_0 convex, $f_i : \mathbf{R}^n \rightarrow \mathbf{R}^{k_i}$, $i = 1, \dots, m$, convex with respect to proper cones $K_i \in \mathbf{R}^{k_i}$
- f_i twice continuously differentiable
- $A \in \mathbf{R}^{p \times n}$ with $\text{rank } A = p$
- we assume p^* is finite and attained
- we assume problem is strictly feasible; hence strong duality holds and dual optimum is attained

examples of greatest interest: SOCP, SDP

Generalized logarithm for proper cone

$\psi : \mathbf{R}^q \rightarrow \mathbf{R}$ is generalized logarithm for proper cone $K \subseteq \mathbf{R}^q$ if:

- $\text{dom } \psi = \text{int } K$ and $\nabla^2 \psi(y) \prec 0$ for $y \succ_K 0$
- $\psi(sy) = \psi(y) + \theta \log s$ for $y \succ_K 0$, $s > 0$ (θ is the degree of ψ)

examples

- nonnegative orthant $K = \mathbf{R}_+^n$: $\psi(y) = \sum_{i=1}^n \log y_i$, with degree $\theta = n$
- positive semidefinite cone $K = \mathbf{S}_+^n$:

$$\psi(Y) = \log \det Y \quad (\theta = n)$$

- second-order cone $K = \{y \in \mathbf{R}^{n+1} \mid (y_1^2 + \cdots + y_n^2)^{1/2} \leq y_{n+1}\}$:

$$\psi(y) = \log(y_{n+1}^2 - y_1^2 - \cdots - y_n^2) \quad (\theta = 2)$$

properties (without proof): for $y \succ_K 0$,

$$\nabla\psi(y) \succeq_{K^*} 0, \quad y^T \nabla\psi(y) = \theta$$

- nonnegative orthant \mathbf{R}_+^n : $\psi(y) = \sum_{i=1}^n \log y_i$

$$\nabla\psi(y) = (1/y_1, \dots, 1/y_n), \quad y^T \nabla\psi(y) = n$$

- positive semidefinite cone \mathbf{S}_+^n : $\psi(Y) = \log \det Y$

$$\nabla\psi(Y) = Y^{-1}, \quad \text{tr}(Y \nabla\psi(Y)) = n$$

- second-order cone $K = \{y \in \mathbf{R}^{n+1} \mid (y_1^2 + \dots + y_n^2)^{1/2} \leq y_{n+1}\}$:

$$\psi(y) = \frac{2}{y_{n+1}^2 - y_1^2 - \dots - y_n^2} \begin{bmatrix} -y_1 \\ \vdots \\ -y_n \\ y_{n+1} \end{bmatrix}, \quad y^T \nabla\psi(y) = 2$$

Logarithmic barrier and central path

logarithmic barrier for $f_1(x) \preceq_{K_1} 0, \dots, f_m(x) \preceq_{K_m} 0$:

$$\phi(x) = - \sum_{i=1}^m \psi_i(-f_i(x)), \quad \text{dom } \phi = \{x \mid f_i(x) \prec_{K_i} 0, \ i = 1, \dots, m\}$$

- ψ_i is generalized logarithm for K_i , with degree θ_i
- ϕ is convex, twice continuously differentiable

central path: $\{x^*(t) \mid t > 0\}$ where $x^*(t)$ solves

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

Dual points on central path

$x = x^*(t)$ if there exists $w \in \mathbf{R}^p$,

$$t \nabla f_0(x) + \sum_{i=1}^m Df_i(x)^T \nabla \psi_i(-f_i(x)) + A^T w = 0$$

$(Df_i(x) \in \mathbf{R}^{k_i \times n}$ is derivative matrix of f_i)

- therefore, $x^*(t)$ minimizes Lagrangian $L(x, \lambda^*(t), \nu^*(t))$, where

$$\lambda_i^*(t) = \frac{1}{t} \nabla \psi_i(-f_i(x^*(t))), \quad \nu^*(t) = \frac{w}{t}$$

- from properties of ψ_i : $\lambda_i^*(t) \succ_{K_i^*} 0$, with duality gap

$$f_0(x^*(t)) - g(\lambda^*(t), \nu^*(t)) = (1/t) \sum_{i=1}^m \theta_i$$

example: semidefinite programming (with $F_i \in \mathbf{S}^p$)

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & F(x) = \sum_{i=1}^n x_i F_i + G \preceq 0\end{array}$$

- logarithmic barrier: $\phi(x) = \log \det(-F(x)^{-1})$
- central path: $x^*(t)$ minimizes $tc^T x - \log \det(-F(x))$; hence

$$tc_i - \text{tr}(F_i F(x^*(t))^{-1}) = 0, \quad i = 1, \dots, n$$

- dual point on central path: $Z^*(t) = -(1/t)F(x^*(t))^{-1}$ is feasible for

$$\begin{array}{ll}\text{maximize} & \text{tr}(GZ) \\ \text{subject to} & \text{tr}(F_i Z) + c_i = 0, \quad i = 1, \dots, n \\ & Z \succeq 0\end{array}$$

- duality gap on central path: $c^T x^*(t) - \text{tr}(GZ^*(t)) = p/t$

Barrier method

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

repeat

1. *Centering step.* Compute $x^*(t)$ by minimizing $tf_0 + \phi$, subject to $Ax = b$.
 2. *Update.* $x := x^*(t)$.
 3. *Stopping criterion.* **quit** if $(\sum_i \theta_i)/t < \epsilon$.
 4. *Increase t .* $t := \mu t$.
-

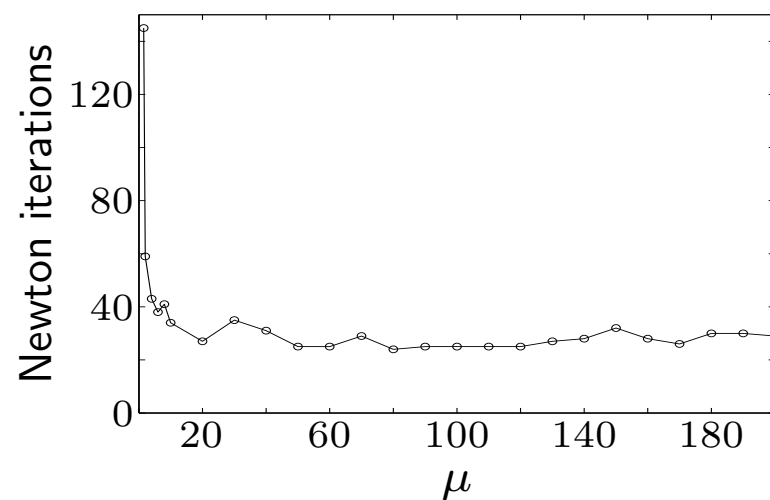
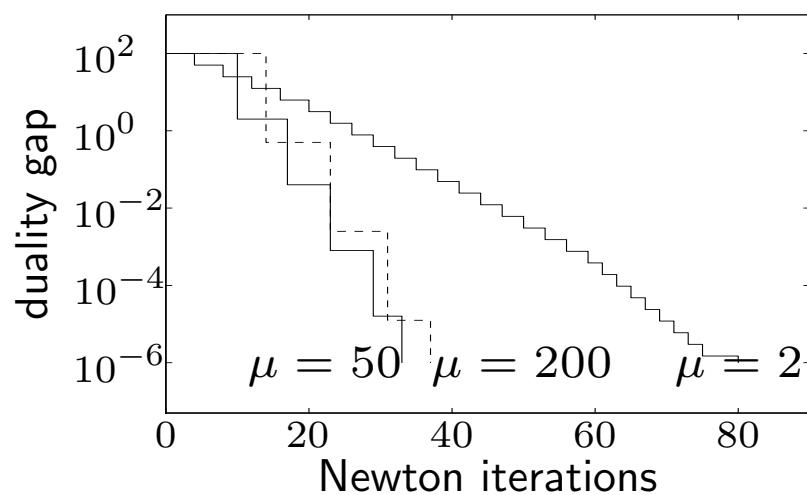
- only difference is duality gap m/t on central path is replaced by $\sum_i \theta_i/t$
- number of outer iterations:

$$\left\lceil \frac{\log((\sum_i \theta_i)/(\epsilon t^{(0)}))}{\log \mu} \right\rceil$$

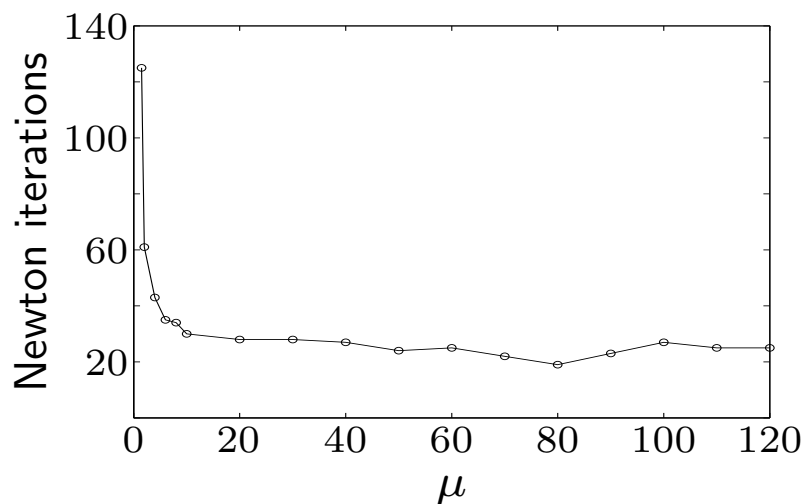
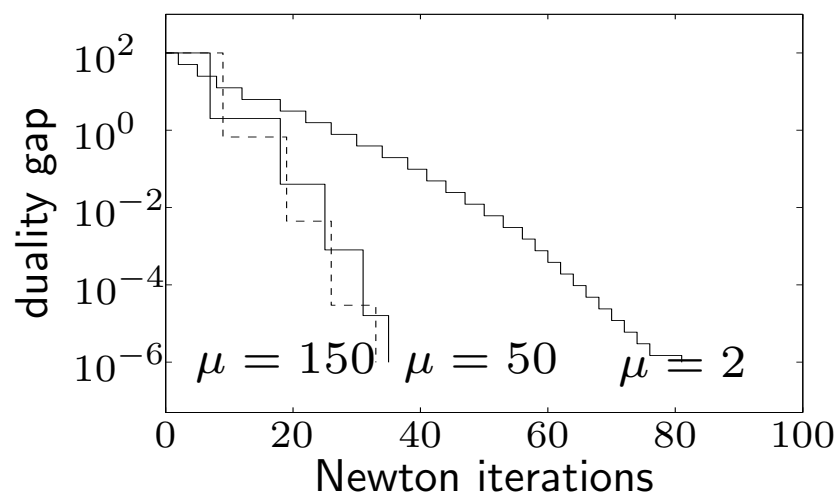
- complexity analysis via self-concordance applies to SDP, SOCP

Examples

second-order cone program (50 variables, 50 SOC constraints in \mathbf{R}^6)



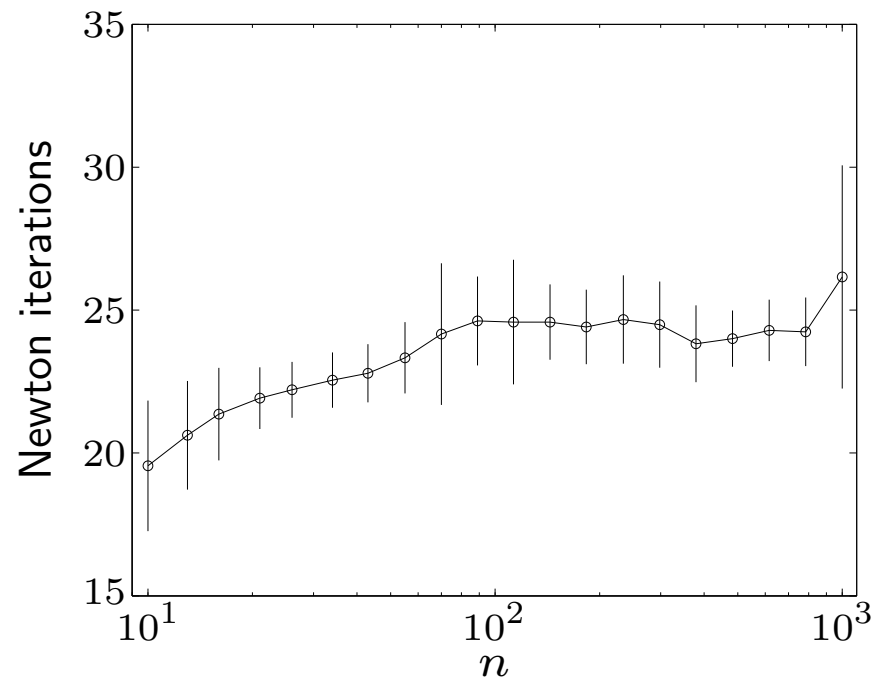
semidefinite program (100 variables, LMI constraint in \mathbf{S}^{100})



family of SDPs ($A \in \mathbf{S}^n$, $x \in \mathbf{R}^n$)

$$\begin{array}{ll}\text{minimize} & \mathbf{1}^T x \\ \text{subject to} & A + \mathbf{diag}(x) \succeq 0\end{array}$$

$n = 10, \dots, 1000$, for each n solve 100 randomly generated instances



Primal-dual interior-point methods

more efficient than barrier method when high accuracy is needed

- update primal and dual variables at each iteration; no distinction between inner and outer iterations
- often exhibit superlinear asymptotic convergence
- search directions can be interpreted as Newton directions for modified KKT conditions
- can start at infeasible points
- cost per iteration same as barrier method

13. Conclusions

- main ideas of the course
- importance of modeling in optimization

Modeling

mathematical optimization

- problems in engineering design, data analysis and statistics, economics, management, . . . , can often be expressed as mathematical optimization problems
- techniques exist to take into account multiple objectives or uncertainty in the data

tractability

- roughly speaking, tractability in optimization requires convexity
- algorithms for nonconvex optimization find local (suboptimal) solutions, or are very expensive
- surprisingly many applications can be formulated as convex problems

Theoretical consequences of convexity

local optima are global

extensive duality theory

- systematic way of deriving lower bounds on optimal value
- necessary and sufficient optimality conditions
- certificates of infeasibility
- sensitivity analysis

Practical consequences of convexity

convex problems can be solved globally and efficiently

- algorithms with a polynomial worst-case complexity
- basic algorithms (*e.g.*, Newton, barrier method, . . .) are easy to implement and work well for small and medium size problems (larger problems if structure is exploited)
- more and more high-quality implementations of advanced algorithms and modeling tools are becoming available

Convex optimization examples

force/moment generation with thrusters

minimum-time optimal control

optimal transmitter power allocation

phased-array antenna beamforming

optimal receiver location

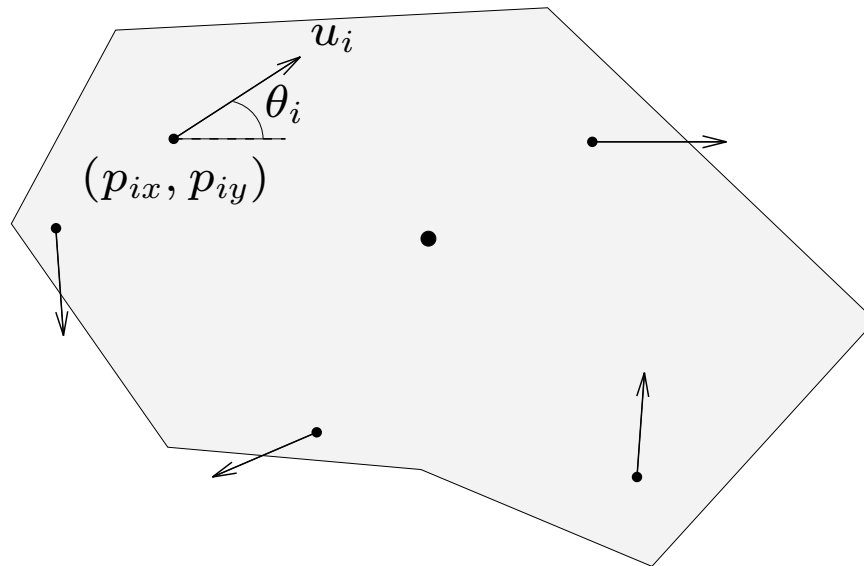
power allocation in FDM system

optimizing structural dynamics

Force/moment generation with thrusters

rigid body with center of mass at origin $p = 0 \in \mathbf{R}^2$

n forces with magnitude u_i , acting at $p_i = (p_{ix}, p_{iy})$, in direction θ_i



resulting horizontal force: $F_x = \sum_{i=1}^n u_i \cos \theta_i$

resulting vertical force: $F_y = \sum_{i=1}^n u_i \sin \theta_i$

resulting torque: $T = \sum_{i=1}^n (p_{iy} u_i \cos \theta_i - p_{ix} u_i \sin \theta_i)$

force limits: $0 \leq u_i \leq 1$ (thrusters)

fuel usage: $u_1 + \dots + u_n$

problem: find thruster forces u_i that yield given desired forces and torques $F_x^{\text{des}}, F_y^{\text{des}}, T^{\text{des}}$, and minimize fuel usage (if feasible)

can be expressed as LP:

$$\begin{aligned} & \text{minimize} && \mathbf{1}^T u \\ & \text{subject to} && Fu = f^{\text{des}} \\ & && 0 \leq u_i \leq 1, \quad i = 1, \dots, n \end{aligned}$$

where

$$F = \begin{bmatrix} \cos \theta_1 & & \cos \theta_n \\ \sin \theta_1 & & \sin \theta_n \\ p_{1y} \cos \theta_1 & p_{1x} \sin \theta_1 & p_{ny} \cos \theta_n & p_{nx} \sin \theta_n \end{bmatrix},$$

$$f^{\text{des}} = (F_x^{\text{des}}, F_y^{\text{des}}, T^{\text{des}}), \quad \mathbf{1} = (1, 1, 1)$$

Extensions of thruster problem

opposing thruster pairs:

$$\begin{array}{ll}\text{minimize} & \|u\|_1 = \sum_{i=1}^n |u_i| \\ \text{subject to} & Fu = f^{\text{des}} \\ & |u_i| \leq 1, \quad i = 1, \dots, n\end{array}$$

can express as LP

more accurate fuel use model:

$$\begin{array}{ll}\text{minimize} & \sum_{i=1}^n \phi_i(u_i) \\ \text{subject to} & Fu = f^{\text{des}} \\ & 0 \leq u_i \leq 1, \quad i = 1, \dots, n\end{array}$$

ϕ_i are piecewise linear increasing convex functions
can express as LP

minimize maximum force/moment error:

$$\begin{array}{ll}\text{minimize} & \|Fu - f^{\text{des}}\|_{\infty} \\ \text{subject to} & 0 \leq u_i \leq 1, \quad i = 1, \dots, n\end{array}$$

can express as LP

minimize number of thrusters used:

$$\begin{array}{ll}\text{minimize} & \# \text{ thrusters on} \\ \text{subject to} & Fu = f^{\text{des}} \\ & 0 \leq u_i \leq 1, \quad i = 1, \dots, n\end{array}$$

can't express as LP

(but we could check feasibility of each of the 2^n subsets of thrusters)

Minimum-time optimal control

linear dynamical system:

$$x(t+1) = Ax(t) + Bu(t), \quad t = 0, 1, \dots, K, \quad x(0) = x_0$$

inputs limited to range $[-1, 1]$:

$$\|u(t)\|_\infty \leq 1, \quad t = 0, 1, \dots, K$$

settling time:

$$f(u(0), \dots, u(K)) = \min \{T \mid x(t) = 0 \text{ for } T \leq t \leq K+1\}$$

settling time f is quasiconvex function of $(u(0), \dots, u(K))$:

$$f(u(0), u(1), \dots, u(K)) \leq T$$

if and only if for all $t = T, \dots, K + 1$

$$x(t) = A^t x_0 + A^{t-1} B u(0) + \dots + B u(t-1) = 0$$

i.e., sublevel sets are affine

minimum-time optimal control problem:

$$\begin{aligned} & \text{minimize} && f(u(0), u(1), \dots, u(K)) \\ & \text{subject to} && \|u(t)\|_\infty \leq 1, \quad t = 0, \dots, K \end{aligned}$$

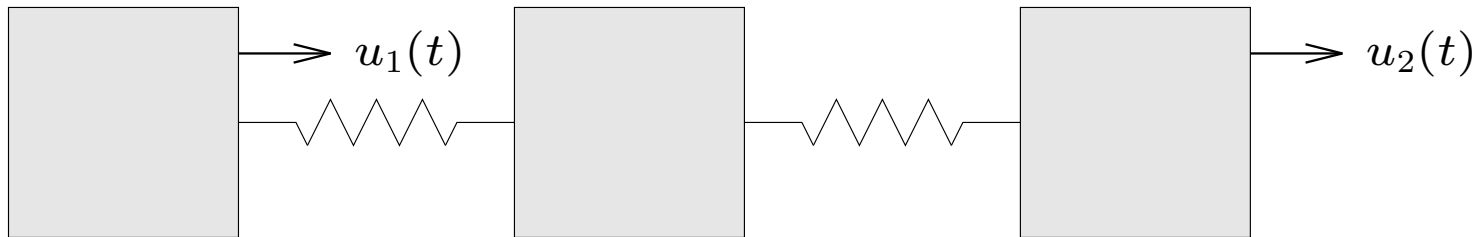
with variables $u(0), \dots, u(K)$

a quasiconvex problem; can be solved via bisection with LPs

Minimum-time control example

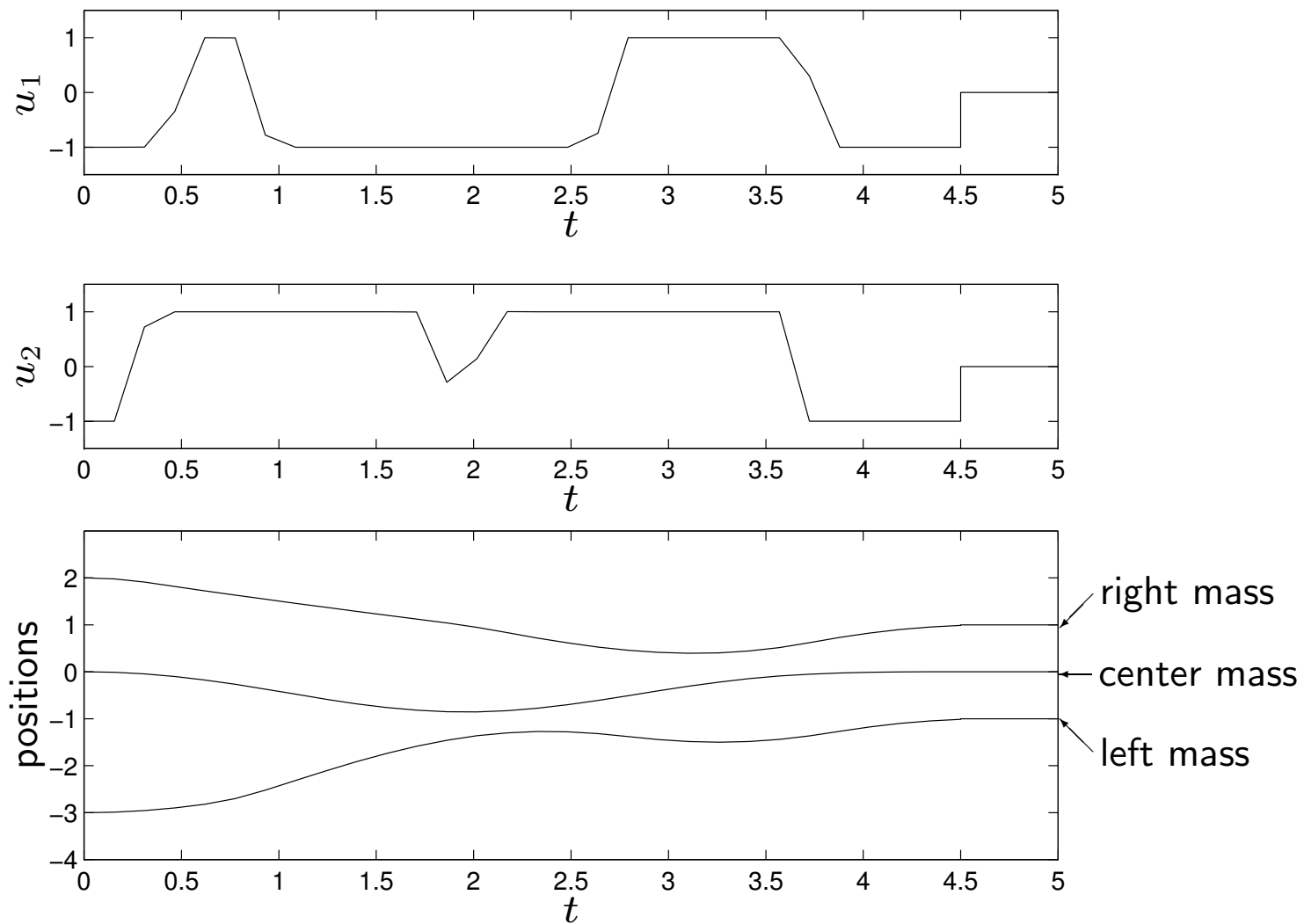
three unit masses, connected by two unit springs with equilibrium length one

$u(t) \in \mathbf{R}^2$ is force on left & right masses over time interval $(0.15t, 0.15(t+1)]$



problem: pick $u(0), \dots, u(K)$ to bring masses to positions $(-1, 0, 1)$ (at rest), as quickly as possible, from initial condition $(-3, 0, 2)$ (at rest)

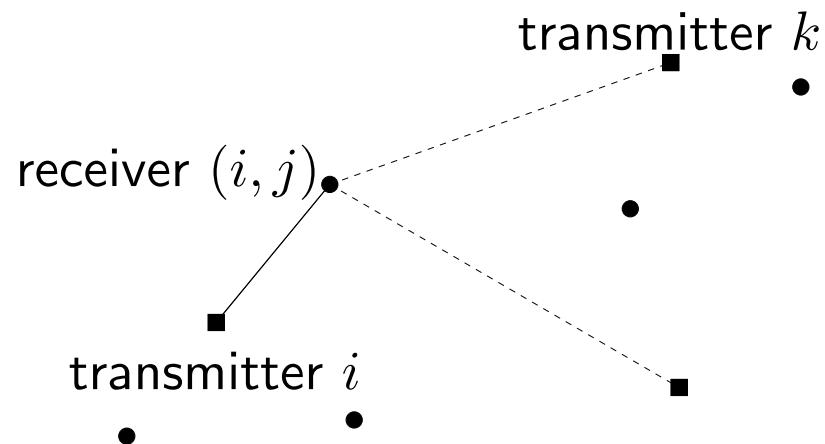
optimal solution:



Optimal transmitter power allocation

m transmitters, mn receivers all at same frequency

transmitter i wants to transmit to n receivers labeled (i, j) , $j = 1, \dots, n$



A_{ijk} is path gain from transmitter k to receiver (i, j)

N_{ij} is (self) noise power of receiver (i, j)

variables: transmitter powers p_k , $k = 1, \dots, m$

at receiver (i, j) :

signal power:

$$S_{ij} = A_{iji}p_i$$

noise plus interference power:

$$I_{ij} = \sum_{k \neq i} A_{ijk}p_k + N_{ij}$$

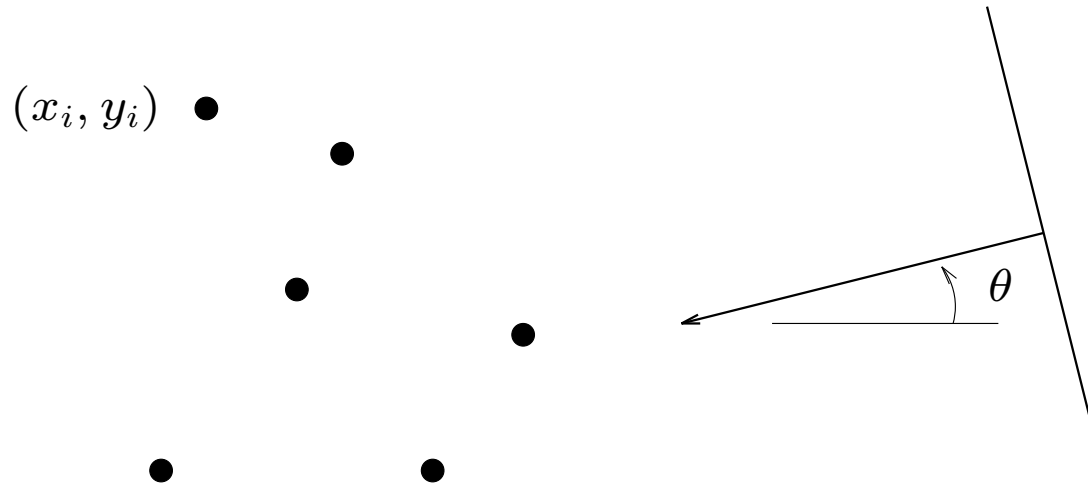
signal to interference/noise ratio (SINR): S_{ij}/I_{ij}

problem: choose p_i to maximize smallest SINR:

$$\begin{array}{ll} \text{maximize} & \min_{i,j} \frac{A_{iji}p_i}{\sum_{k \neq i} A_{ijk}p_k + N_{ij}} \\ \text{subject to} & 0 \leq p_i \leq p_{\max} \end{array}$$

. . . a (generalized) linear fractional program

Phased-array antenna beamforming



omnidirectional antenna elements at positions $(x_1, y_1), \dots, (x_n, y_n)$

unit plane wave incident from angle θ induces in i th element a signal $e^{j(x_i \cos \theta + y_i \sin \theta - \omega t)}$

($j = \sqrt{-1}$, frequency ω , wavelength λ)

demodulate to get output $e^{j(x_i \cos \theta + y_i \sin \theta)} \in \mathbf{C}$

linearly combine with complex weights w_i :

$$y(\theta) = \sum_{i=1}^n w_i e^{j(x_i \cos \theta + y_i \sin \theta)}$$

$y(\theta)$ is (complex) *antenna array gain pattern*

$|y(\theta)|$ gives sensitivity of array as function of incident angle

depends on design variables **Re** w , **Im** w

(called *antenna array weights* or *shading coefficients*)

design problem: choose w to achieve desired gain pattern

Sidelobe level minimization

make $|y(\theta)|$ small for $|\theta - \theta_{\text{tar}}| >$

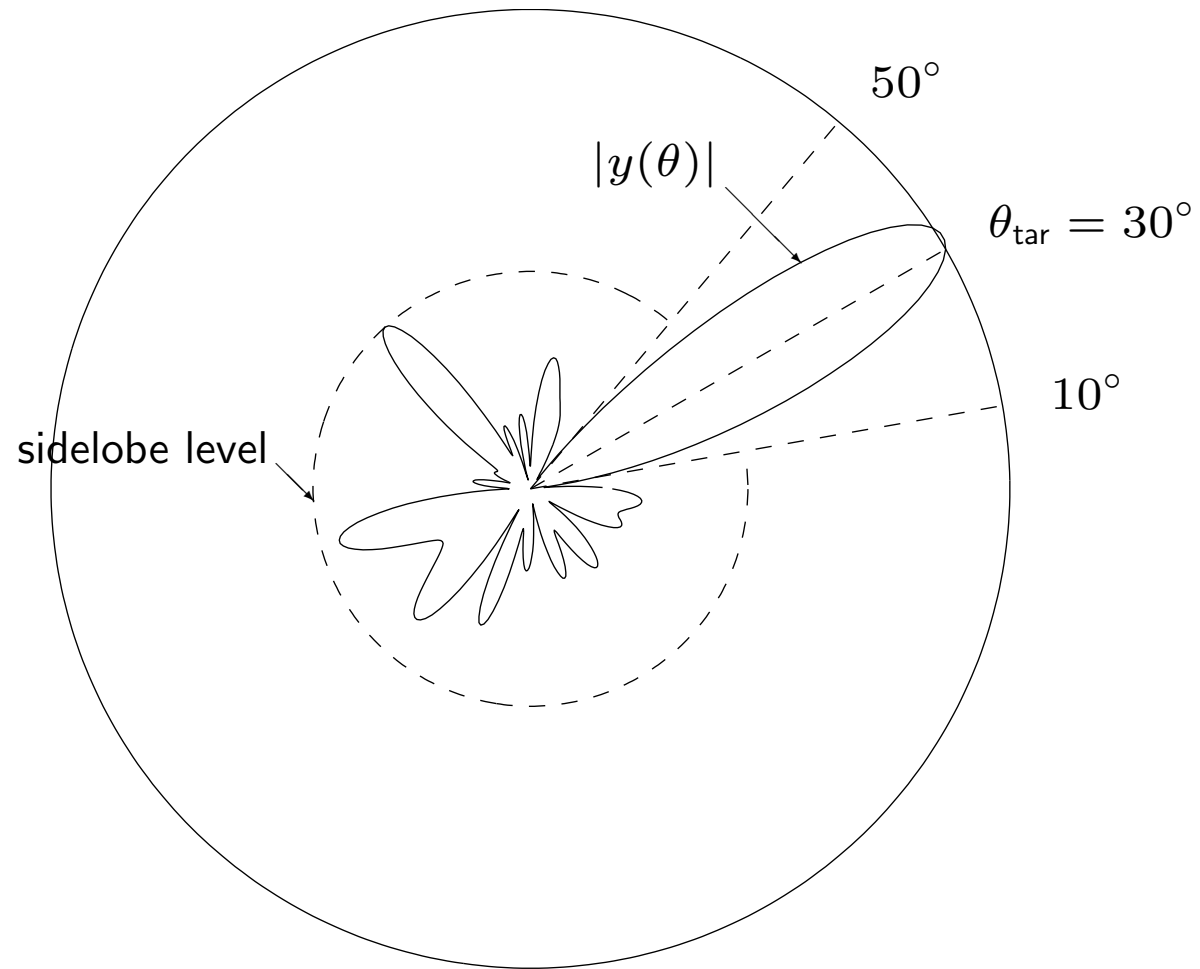
(θ_{tar} : target direction; 2β : beamwidth)

via least-squares (discretize angles)

$$\begin{array}{ll} \text{minimize} & \sum_i |y(\theta_i)|^2 \\ \text{subject to} & y(\theta_{\text{tar}}) = 1 \end{array}$$

(sum is over angles outside beam)

least-squares problem with two (real) linear equality constraints



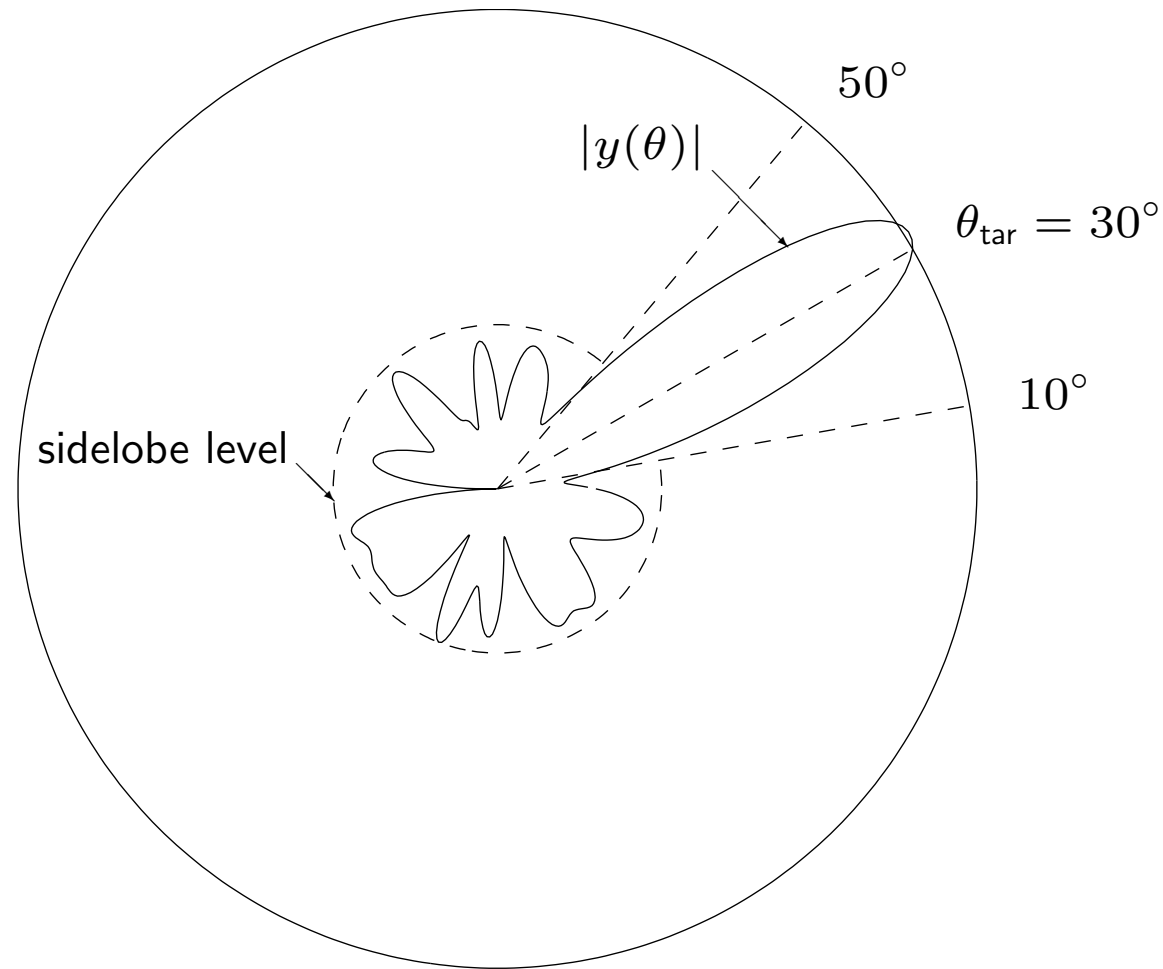
minimize sidelobe level (discretize angles)

$$\begin{array}{ll}\text{minimize} & \max_i |y(\theta_i)| \\ \text{subject to} & y(\theta_{\text{tar}}) = 1\end{array}$$

(max over angles outside beam)

can be cast as SOCP

$$\begin{array}{ll}\text{minimize} & t \\ \text{subject to} & |y(\theta_i)| \leq t \\ & y(\theta_{\text{tar}}) = 1\end{array}$$



Extensions

convex (& quasiconvex) extensions:

$$y(\theta_0) = 0 \text{ (null in direction } \theta_0)$$

w is real (amplitude only shading)

$$|w_i| \leq 1 \text{ (attenuation only shading)}$$

minimize $\sum_{i=1}^n |w_i|^2$ (thermal noise power in y)

minimize beamwidth given a maximum sidelobe level

nonconvex extension:

maximize number of zero weights

Optimal receiver location

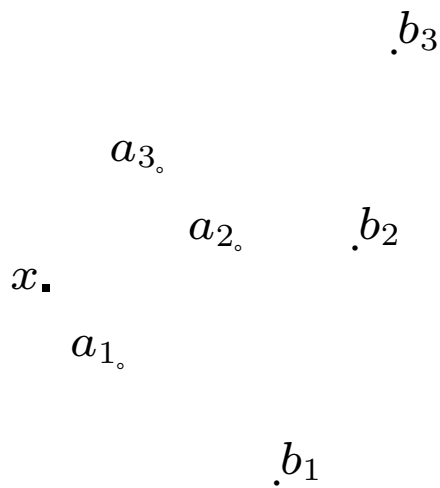
N transmitter frequencies $1, \dots, N$

transmitters at locations $a_i, b_i \in \mathbf{R}^2$ use frequency i

transmitters at a_1, a_2, \dots, a_N are the wanted ones

transmitters at b_1, b_2, \dots, b_N are interfering

receiver at position $x \in \mathbf{R}^2$



(signal) receiver power from a_i : $\|x - a_i\|$ (2.1)

(interfering) receiver power from b_i : $\|x - b_i\|$ (2.1)

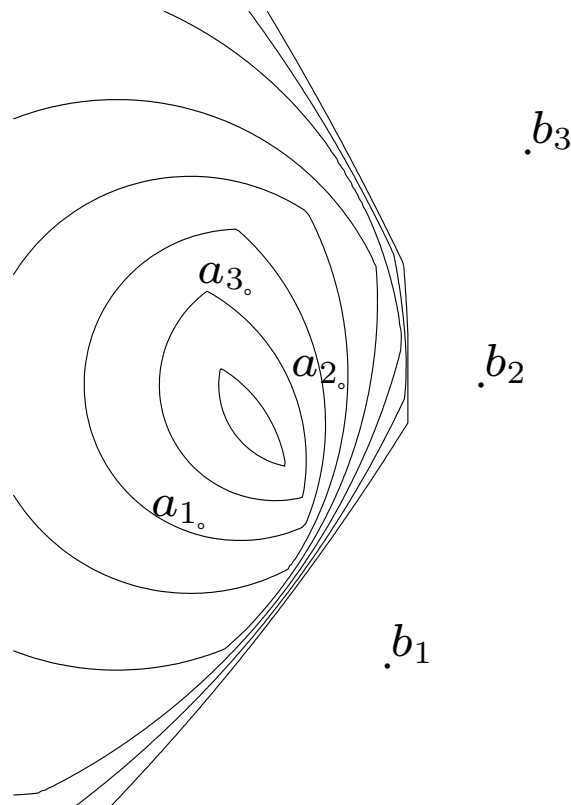
worst signal to interference ratio, over all frequencies, is

$$S/I = \min_i \frac{\|x - a_i\|}{\|x - b_i\|}$$

what receiver location x maximizes S/I ?

S/I is quasiconcave on $\{x \mid S/I \leq 1\}$, *i.e.*, on

$$\{x \mid \|x - a_i\| \leq \|x - b_i\|, i = 1, \dots, N\}$$



can use bisection; every iteration is a convex quadratic feasibility problem

Power allocation in FDM system

frequency division multiplex (FDM) system



signal u_i modulates carrier frequency f_i with power p_i

channel is slightly nonlinear

powers affect signal power, interference power at each y_i

problem: choose powers to maximize minimum SINR
(signal to noise & interference ratio)

demodulated signal power in y_i proportional to p_i

noise power in y_i is $\frac{2}{i}$

interference power in y_i is sum of crosstalk & intermodulation products from nonlinearity

crosstalk power c_i is linear in powers:

$$c = Cp, \quad C_{ij} = 0$$

C is often tridiagonal, *i.e.*, have crosstalk from adjacent channels only

intermodulation power: k th order IM products have frequencies

$$\pm f_{i_1} \pm f_{i_2} \pm \dots \pm f_{i_k}$$

with power proportional to $p_{i_1} p_{i_2} \dots p_{i_k}$

e.g., for frequencies 1, 2, 3:

frequency	IM product	pwr. prop. to
2	$1 + 1$	p_1^2
3	$1 + 2$	$p_1 p_2$
1	$2 - 1$	$p_2 p_1$
1	$3 - 2$	$p_3 p_2$
2	$3 - 1$	$p_3 p_1$
3	$1 + 1 + 1$	p_1^3
1	$1 + 1 - 1$	p_1^3
2	$2 + 1 - 1$	$p_2 p_1^2$
\vdots	\vdots	\vdots

total IM power at f_i is (complicated) polynomial of p_1, \dots, p_n , with nonnegative coefficients

inverse SINR at frequency i

$$\frac{\text{noise} + \text{crosstalk} + \text{IM power}}{\text{signal power}}$$

is posynomial function of p_1, \dots, p_n

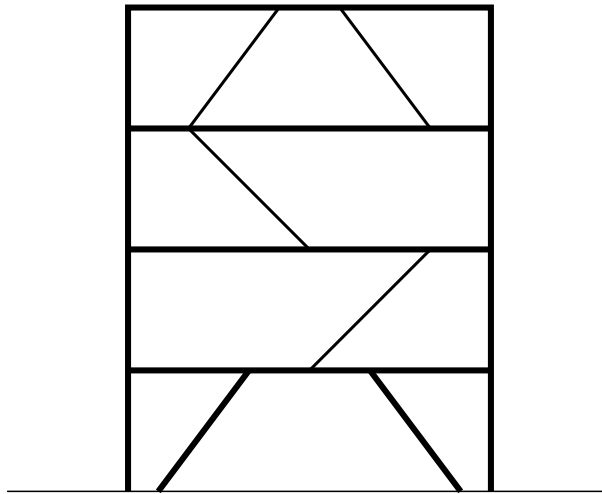
hence, problem such as

$$\begin{array}{ll} \text{maximize} & \min_i \text{SINR}_i \\ \text{subject to} & 0 < p_i \leq P_{\max} \end{array}$$

is geometric program

Optimizing structural dynamics

linear elastic structure



dynamics (ignoring damping): $M\ddot{d} + Kd = 0$

$d(t) \in \mathbf{R}^k$: vector of displacements

$M = M^T$ M is mass matrix; $K = K^T$ K is stiffness matrix

Fundamental frequency

solutions have form

$$d_i(t) = \sum_{j=1}^k a_{ij} \cos(\omega_j t + \phi_j)$$

where $0 < \omega_1 < \omega_2 < \dots < \omega_k$ are the modal frequencies, *i.e.*, positive solutions of $\det(\omega^2 M - K) = 0$

fundamental frequency:

$$\omega_1 = \sqrt{\lambda_{\min}(M^{-1}K)} = \sqrt{\lambda_{\min}(M^{-1/2}KM^{-1/2})}$$

- structure behaves like mass at frequencies below ω_1
- gives stiffness measure (the larger ω_1 , the stiffer the structure)

$\omega_1^2 = \lambda_{\min}(M^{-1}K) \iff \lambda_{\min}(M^{-1}K) \geq 0$ so ω_1 is quasiconcave function of M, K

design variables: x_i , cross-sectional area of structural member i
(geometry of structure fixed)

$$M(x) = M_0 + \sum_i x_i M_i, \quad K(x) = K_0 + \sum_i x_i K_i$$

structure weight $w = w_0 + \sum_i x_i w_i$

problem: minimize weight s.t. $\omega_1 \leq \Omega$, limits on cross-sectional areas

as SDP:

$$\begin{array}{ll} \text{minimize} & w_0 + \sum_i x_i w_i \\ \text{subject to} & \Omega^2 M(x) - K(x) \preceq 0 \\ & l_i \leq x_i \leq u_i \end{array}$$

Filter design

FIR filters

Chebyshev design

linear phase filter design

equalizer design

filter magnitude specifications

FIR Filters

Finite impulse response (FIR) Filter:

$$y(t) = \sum_{i=0}^{n-1} h_i u(t-i), \quad t \in \mathbf{Z}$$

(sequence) $u : \mathbf{Z} \rightarrow \mathbf{R}$ is *input signal*

(sequence) $y : \mathbf{Z} \rightarrow \mathbf{R}$ is *output signal*

h_i are called *filter coefficients*

n is *filter order or length*

Iter frequency response: $H : \mathbf{R} \rightarrow \mathbf{C}$

$$\begin{aligned} H(\omega) &= h_0 + h_1 e^{-j\omega} + \dots + h_{n-1} e^{-j(n-1)\omega} \\ &= \sum_{t=0}^{n-1} h_t \cos t\omega + j \sum_{t=0}^{n-1} h_t \sin t\omega \end{aligned}$$

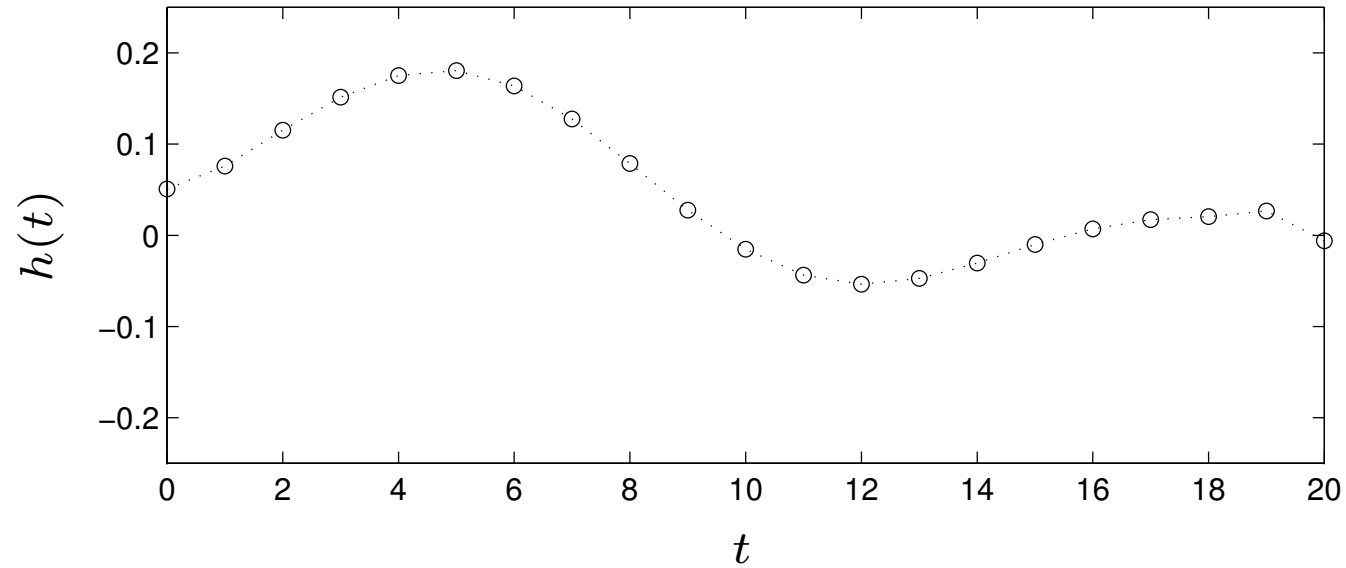
j means $\sqrt{-1}$ here (EE tradition)

H is periodic and conjugate symmetric, so only need to know/specify for $0 \leq \omega$

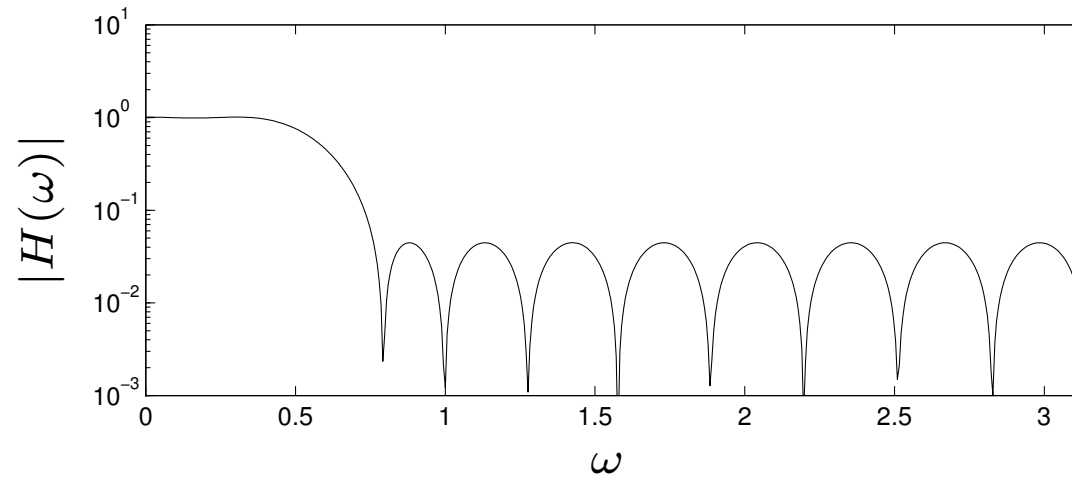
FIR Iiter design problem: choose h so it and H satisfy/optimize specs

example: (lowpass) FIR filter, order $n = 21$

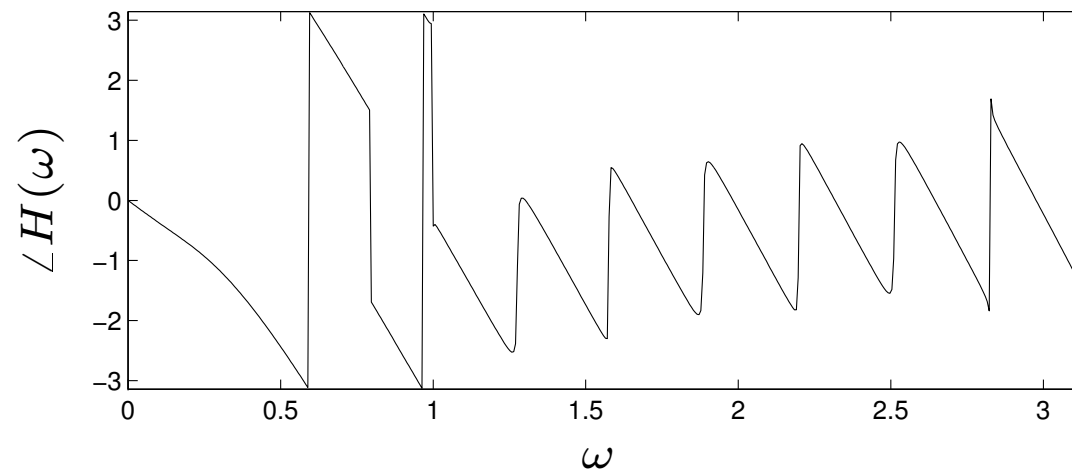
impulse response h :



frequency response magnitude (*i.e.*, $|H(\omega)|$):



frequency response phase (*i.e.*, $\angle H(\omega)$):



Chebyshev design

$$\text{minimize } \max_{\omega \in [0, \pi]} |H(\omega) - H_{\text{des}}(\omega)|$$

h is optimization variable

$H_{\text{des}} : \mathbf{R} \rightarrow \mathbf{C}$ is (given) **desired transfer function**

convex problem

can add constraints, *e.g.*, $|h_i| \leq 1$

sample (discretize) frequency:

$$\text{minimize } \max_{k=1, \dots, m} |H(\omega_k) - H_{\text{des}}(\omega_k)|$$

sample points $0 = \omega_1 < \dots < \omega_m = \pi$ are fixed (*e.g.*, $\omega_k = k\pi/m$)

$m \geq n$ (common rule-of-thumb: $m = 15n$)

yields approximation (relaxation) of problem above

Chebyshev design via SOCP:

$$\begin{array}{ll} \text{minimize} & t \\ \text{subject to} & A^{(k)}h \preceq b^{(k)} \quad t, \quad k = 1, \dots, m \end{array}$$

where

$$A^{(k)} = \begin{bmatrix} 1 & \cos \omega_k \\ 0 & \sin \omega_k \end{bmatrix} \quad \begin{bmatrix} \cos(n-1)\omega_k \\ \sin(n-1)\omega_k \end{bmatrix}$$

$$b^{(k)} = \begin{bmatrix} \Re H_{\text{des}}(\omega_k) \\ \Im H_{\text{des}}(\omega_k) \end{bmatrix}$$

$$h = \begin{bmatrix} h_0 \\ h_{n-1} \end{bmatrix}$$

Linear phase filters

suppose

$n = 2N + 1$ is odd

impulse response is symmetric about midpoint:

$$h_t = h_{n-1-t}, \quad t = 0, \dots, n-1$$

then

$$\begin{aligned} H(\omega) &= h_0 + h_1 e^{j\omega} + \dots + h_{n-1} e^{j(n-1)\omega} \\ &= e^{jN\omega} (2h_0 \cos N\omega + 2h_1 \cos(N-1)\omega + \dots + h_N) \\ &\triangleq e^{jN\omega} \tilde{H}(\omega) \end{aligned}$$

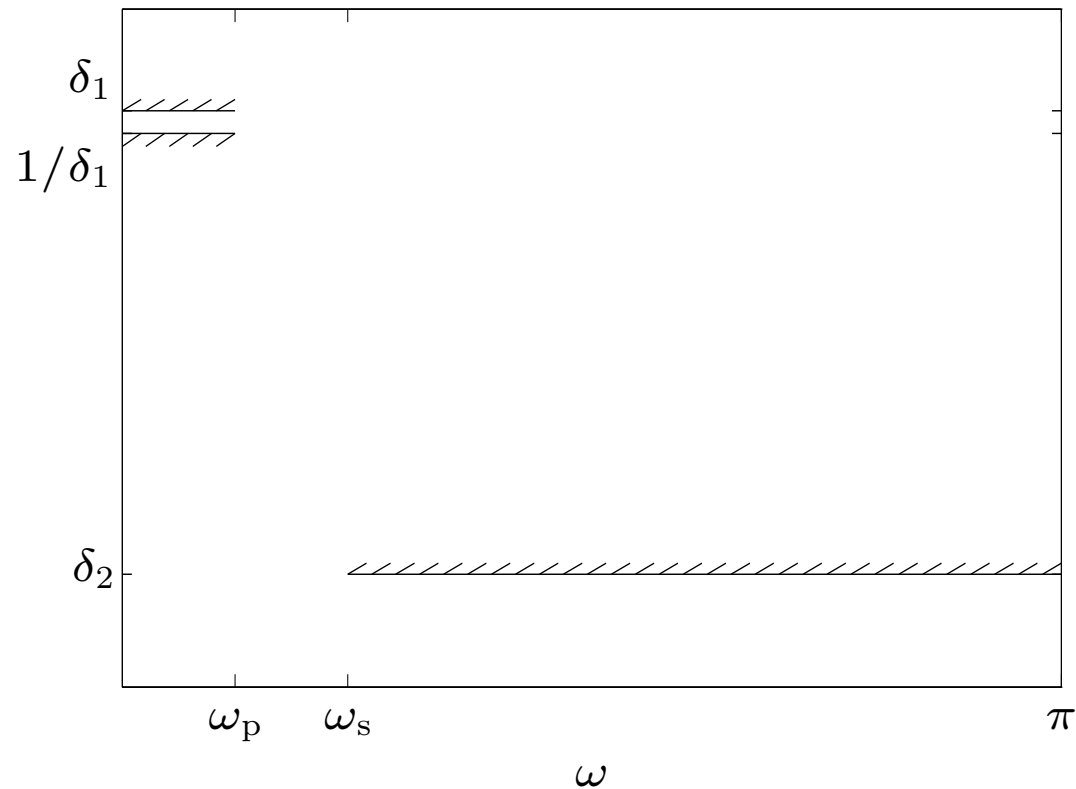
term $e^{-jN\omega}$ represents N -sample delay

$\tilde{H}(\omega)$ is **real**

$$|H(\omega)| = |\tilde{H}(\omega)|$$

called **linear phase filter** ($\angle H(\omega)$ is linear except for jumps of $\pm \pi$)

Lowpass filter specifications



idea:

pass frequencies in *passband* $[0, \omega_p]$

block frequencies in *stopband* $[\omega_s, \pi]$

specifications:

maximum *passband ripple* ($\pm 20 \log_{10} \epsilon_1$ in dB):

$$1/\epsilon_1 \geq |H(\omega)| \geq 1, \quad 0 \leq \omega \leq \omega_p$$

minimum *stopband attenuation* ($-20 \log_{10} \epsilon_2$ in dB):

$$|H(\omega)| \leq \epsilon_2, \quad \omega_s \leq \omega$$

Linear phase lowpass filter design

sample frequency

can assume wlog $\tilde{H}(0) > 0$, so ripple spec is

$$1/\delta_1 \leq \tilde{H}(\omega_k) \leq 1$$

design for **maximum stopband attenuation**:

$$\begin{aligned} &\text{minimize} && \delta_2 \\ &\text{subject to} && 1/\delta_1 \leq \tilde{H}(\omega_k) \leq 1, \quad 0 \leq \omega_k \leq \omega_p \\ & && \tilde{H}(\omega_k) \leq \delta_2, \quad \omega_s \leq \omega_k \end{aligned}$$

passband ripple δ_1 is given

an LP in variables h, δ_2

known (and used) since 1960's

can add other constraints, *e.g.*, $|h_i|$

variations and extensions:

× δ_2 , minimize δ_1 (convex, but not LP)

× δ_1 and δ_2 , minimize ω_s (quasiconvex)

× δ_1 and δ_2 , minimize order n (quasiconvex)

example

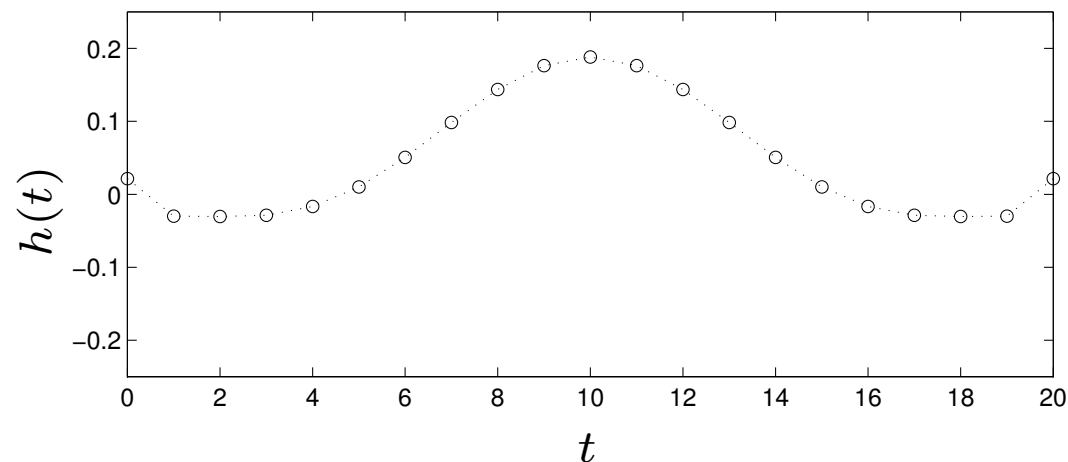
linear phase filter, $n = 21$

passband $[0, 0.12]$; stopband $[0.24, \infty)$

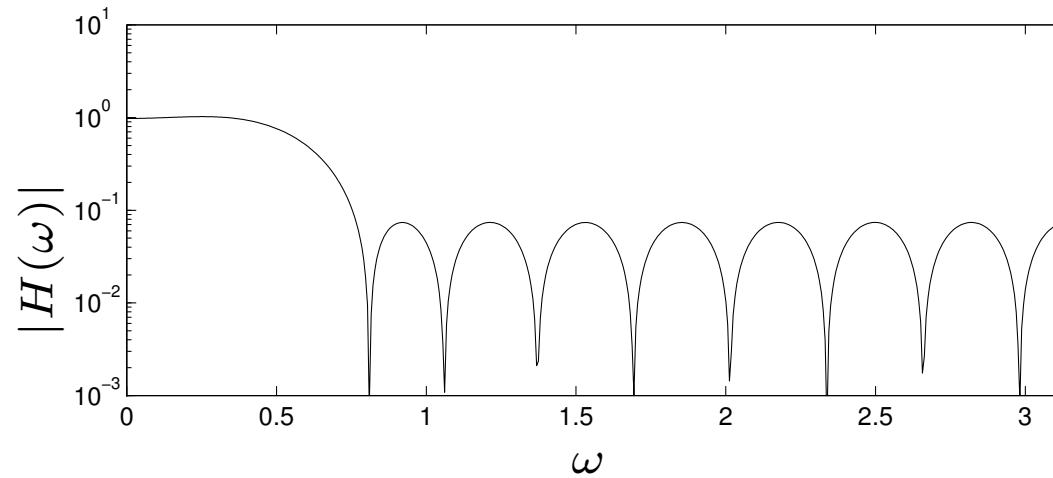
max ripple $\delta_1 = 1.012$ (± 0.1 dB)

design for maximum stopband attenuation

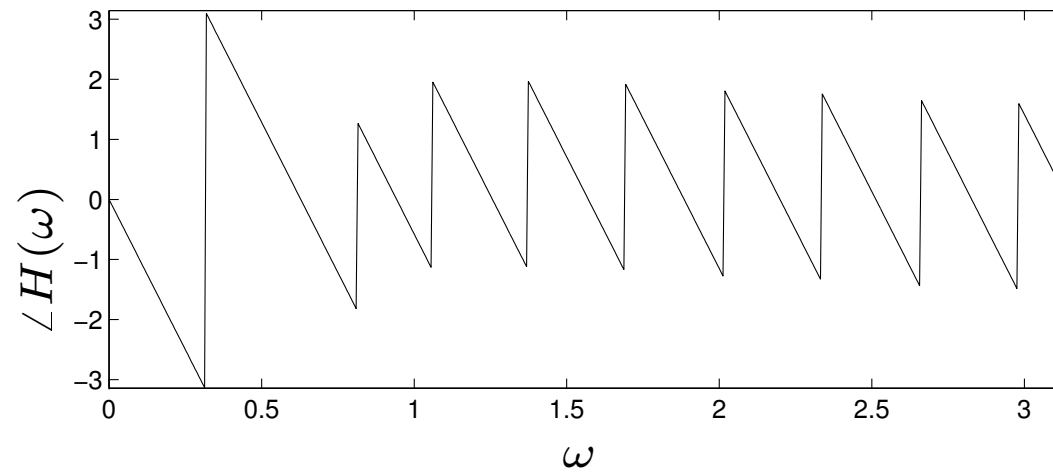
impulse response h :



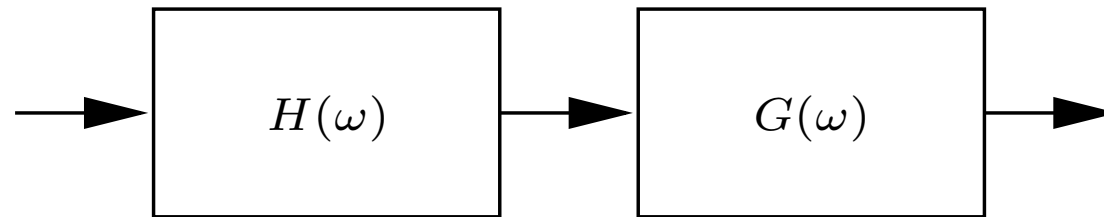
frequency response magnitude (*i.e.*, $|H(\omega)|$):



frequency response phase (*i.e.*, $\angle H(\omega)$):



Equalizer design



equalization: given

G (unequalized frequency response)

G_{des} (desired frequency response)

design (FIR equalizer) H so that $\tilde{G} \triangleq GH = G_{\text{des}}$

common choice: $G_{\text{des}}(\omega) = e^{jD\omega}$ (delay)

i.e., equalization is deconvolution (up to delay)

can add constraints on H , *e.g.*, limits on $|h_i|$ or $\max_{\omega} |H(\omega)|$

Chebyshev equalizer design:

$$\text{minimize } \max_{\omega \in [0, \pi]} \tilde{G}(\omega) - G_{\text{des}}(\omega)$$

convex; SOCP after sampling frequency

time-domain equalization: optimize impulse response \tilde{g} of equalized system

e.g., with $G_{\text{des}}(\omega) = e^{-jD\omega}$,

$$g_{\text{des}}(t) = \begin{cases} 1 & t = D \\ 0 & t \neq D \end{cases}$$

sample design:

$$\begin{array}{ll} \text{minimize} & \max_{t \neq D} |\tilde{g}(t)| \\ \text{subject to} & \tilde{g}(D) = 1 \end{array}$$

an LP

can use $\sum_{t \neq D} \tilde{g}(t)^2$ or $\sum_{t \neq D} |\tilde{g}(t)|$

extensions:

can impose (convex) constraints

can mix time- and frequency-domain specifications

can equalize multiple systems, *i.e.*, choose H so

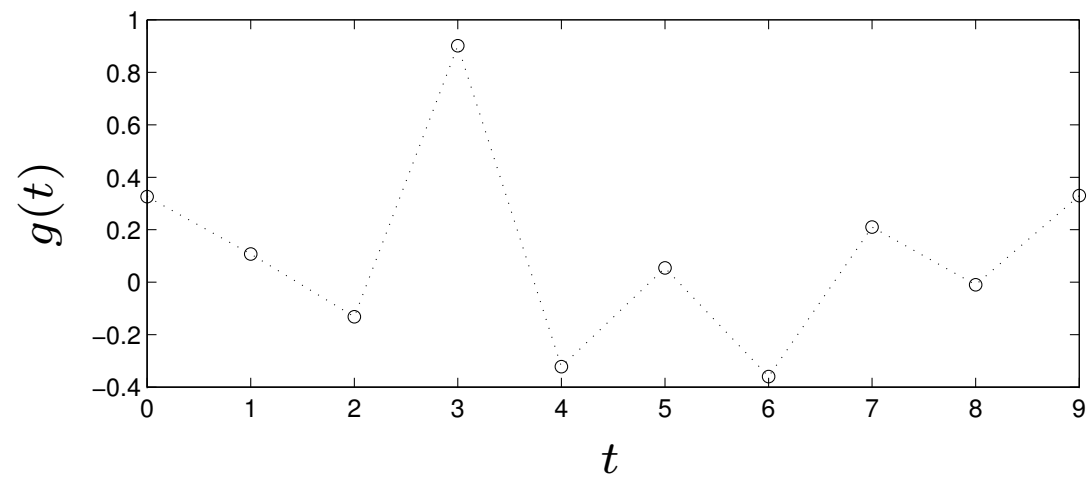
$$G^{(k)}H = G_{\text{des}}, \quad k = 1, \dots, K$$

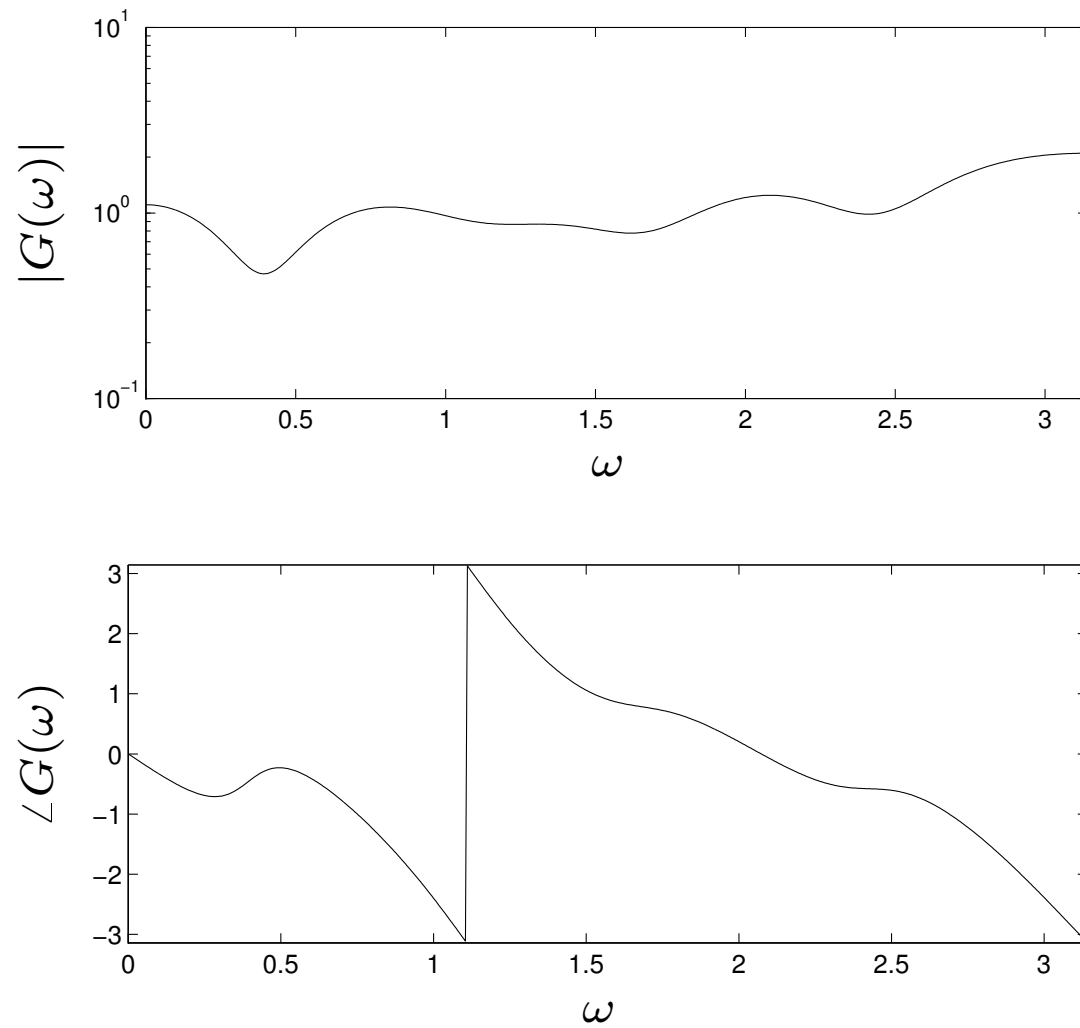
can equalize multi-input multi-output systems
(*i.e.*, G and H are matrices)

extends to multidimensional systems, *e.g.*, image processing

Equalizer design example

unequalized system G is 10th order FIR:



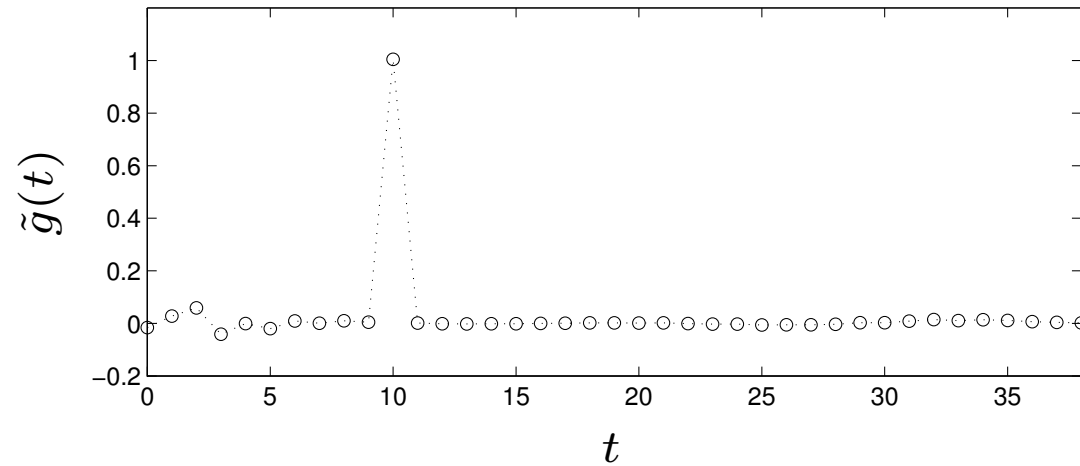


design 30th order FIR equalizer with $\tilde{G}(\omega) = e^{-j10\omega}$

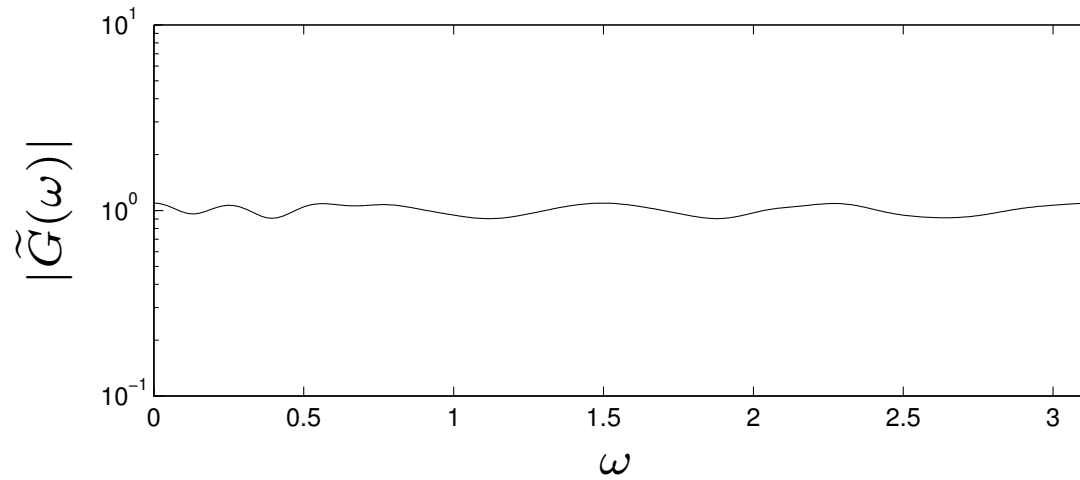
Chebyshev equalizer design:

$$\text{minimize } \max_{\omega} \tilde{G}(\omega) \quad e^{j10\omega}$$

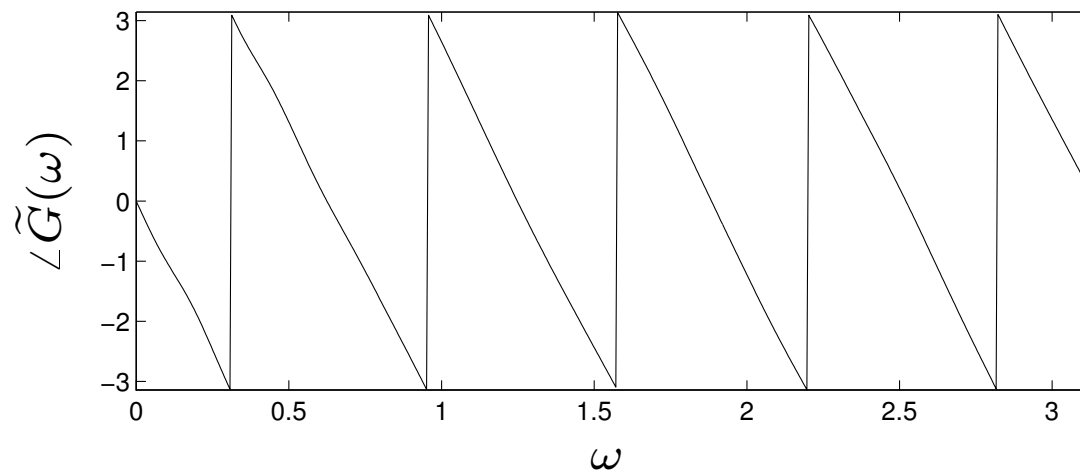
equalized system impulse response \tilde{g}



equalized frequency response magnitude $|\tilde{G}|$



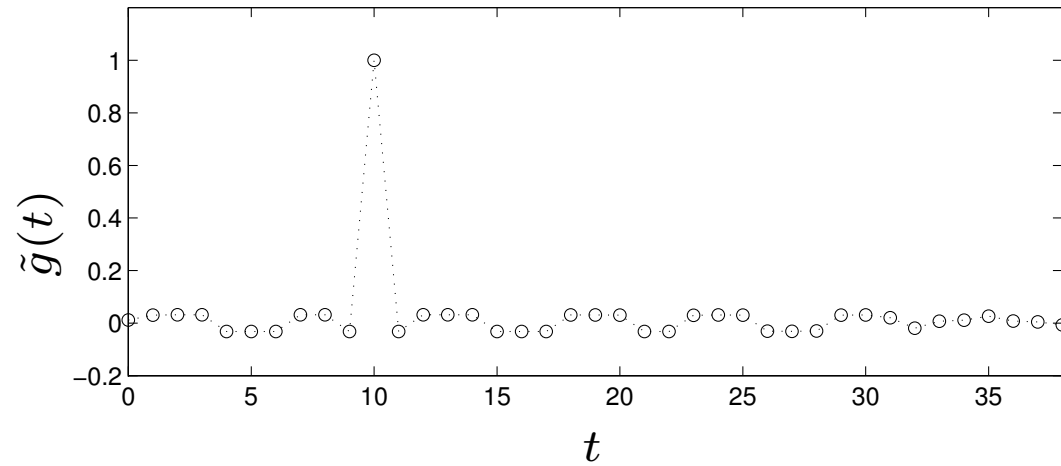
equalized frequency response phase $\angle \tilde{G}$



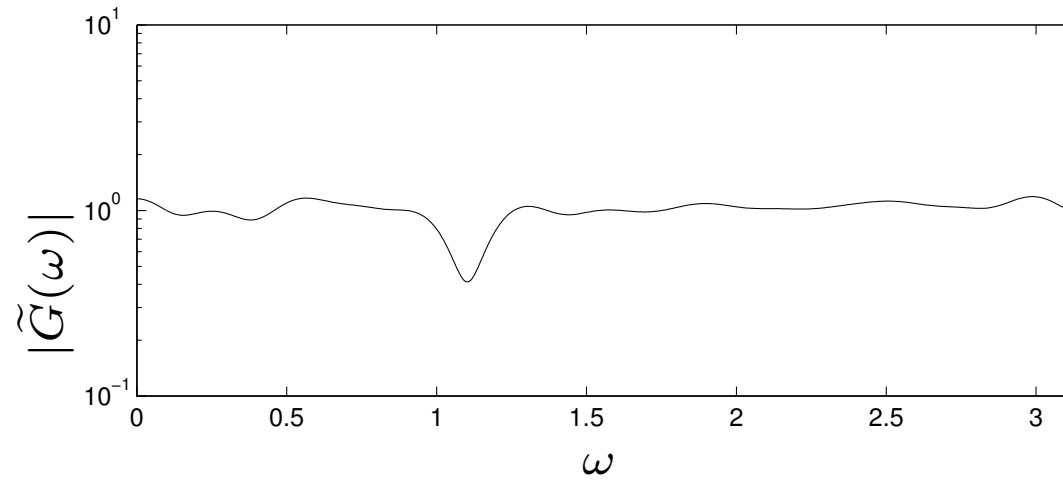
time-domain equalizer design:

$$\text{minimize } \max_{t \neq 10} |\tilde{g}(t)|$$

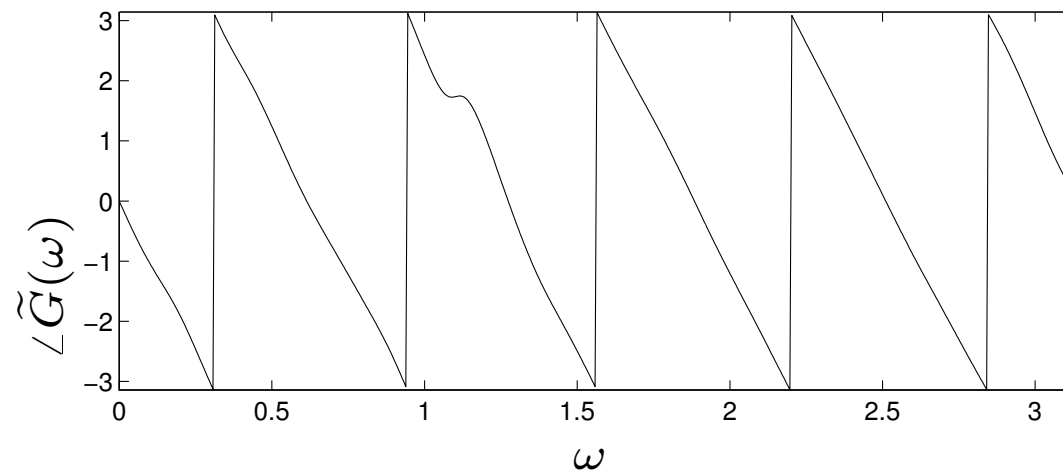
equalized system impulse response \tilde{g}



equalized frequency response magnitude $|\tilde{G}|$



equalized frequency response phase $\angle \tilde{G}$



Filter magnitude specifications

transfer function *magnitude spec* has form

$$L(\omega) \leq |H(\omega)| \leq U(\omega), \quad \omega \in [0, \pi]$$

where $L, U : \mathbf{R} \rightarrow \mathbf{R}_+$ are given

lower bound is **not** convex in filter coefficients h

arises in many applications, *e.g.*, audio, spectrum shaping

can change variables to solve via convex optimization

Autocorrelation coefficients

autocorrelation coefficients associated with impulse response $h = (h_0, \dots, h_{n-1}) \in \mathbf{R}^n$ are

$$r_t = \sum h_k h_{k+t}$$

(we take $h_k = 0$ for $k < 0$ or $k \geq n$)

$$r_t = r_{-t}; \quad r_t = 0 \text{ for } |t| \geq n$$

hence suffices to specify $r = (r_0, \dots, r_{n-1}) \in \mathbf{R}^n$

Fourier transform of autocorrelation coefficients is

$$R(\omega) = \sum e^{-j\omega t} r_t = r_0 + \sum_{t=1}^{n-1} 2r_t \cos \omega t = |H(\omega)|^2$$

always have $R(\omega) \geq 0$ for all ω

can express magnitude specification as

$$L(\omega)^2 \leq R(\omega) \leq U(\omega)^2, \quad \omega \in [0, \pi]$$

... **convex** in r

Spectral factorization

question: when is $r \in \mathbf{R}^n$ the autocorrelation coefficients of some $h \in \mathbf{R}^n$?

answer: (*spectral factorization theorem*) if and only if $R(\omega) \geq 0$ for all ω

spectral factorization condition is convex in r

many algorithms for spectral factorization, *i.e.*, finding an h s.t.

$$R(\omega) = |H(\omega)|^2$$

magnitude design via autocorrelation coefficients:

use r as variable (instead of h)

add spec. fact. condition $R(\omega) \geq 0$ for all ω

optimize over r

use spectral factorization to recover h

log-Chebyshev magnitude design

choose h to minimize

$$\max_{\omega} |20 \log_{10} |H(\omega)| - 20 \log_{10} D(\omega)|$$

D is desired transfer function magnitude

($D(\omega) > 0$ for all ω)

nd minimax logarithmic (dB) t

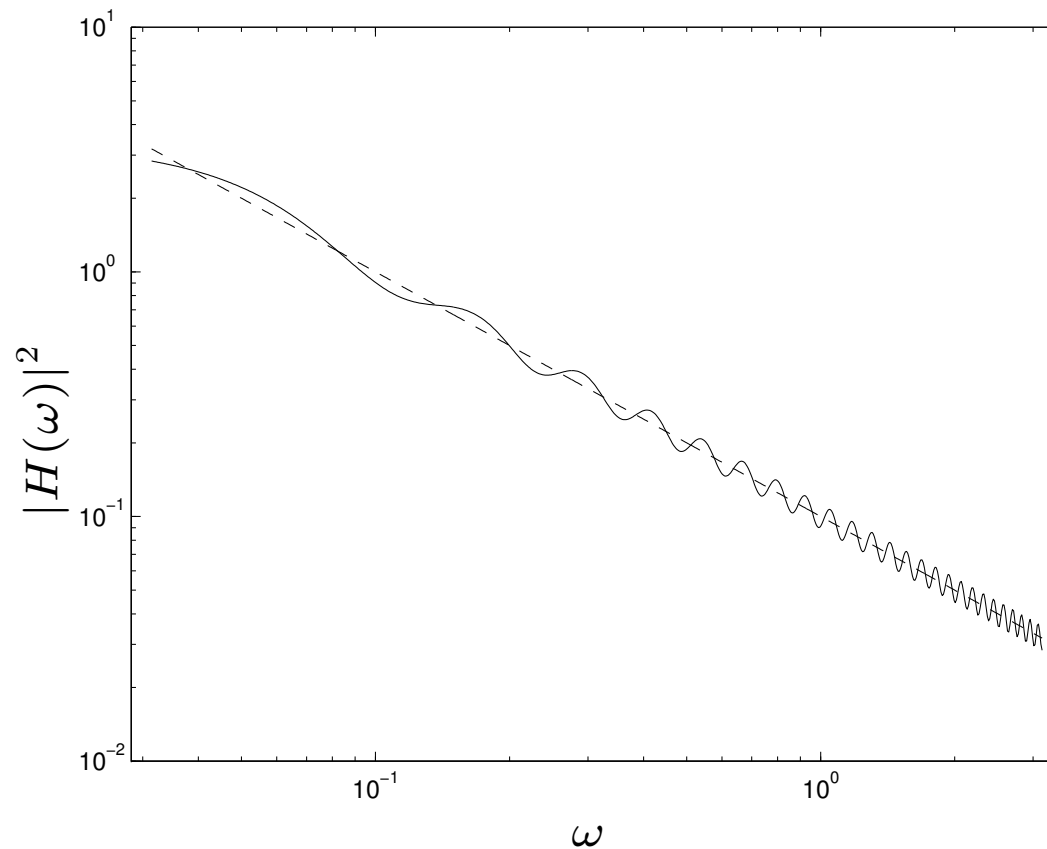
reformulate as

$$\begin{aligned} &\text{minimize} \quad t \\ &\text{subject to} \quad D(\omega)^2/t - R(\omega) - tD(\omega)^2, \quad 0 \leq \omega \leq \pi \end{aligned}$$

convex in variables r, t

constraint includes spectral factorization condition

example: $1/f$ (pink noise) lter (*i.e.*, $D(\omega) = 1/\sqrt{\omega}$), $n = 50$,
log-Chebyshev design over 0.01 ω



optimal τ : $\pm 0.5\text{dB}$

Disciplined convex programming and the `cvx` modeling framework

Michael C. Grant
Information Systems Laboratory
Stanford University

April 27, 2006

Agenda

Today's topics:

- **disciplined convex programming**, a methodology for practical convex optimization; and
- **the `cvx` modeling system**: a MATLAB-based software package which supports this methodology

Where you are now

What you have seen thus far:

- Fundamental definitions of convex sets and convex/concave functions
- An extensive *convexity calculus*: a collection of operations, combinations and transformations that are known to preserve convexity
- An introduction to convex optimization problems, or *convex programs*
- Several specific classes of convex programs: LPs, SDPs, GPs, SOCPs... hopefully you can identify a problem from one of these classes when you see one

You have not yet seen:

- How convex optimization problems are **solved**

(well, that and a few other things)

Solvers

A *solver* is an engine for solving a particular type of mathematical problem, such as a convex program

Solvers typically handle only a certain *class* of problems, such as LPs, or SDPs, or GPs

They also require that problems be expressed in a *standard form*

Most problems do not immediately present themselves in standard form, so they must be *transformed* into standard form

Solver example: MATLAB's `linprog`

A program for solving linear programs:

```
x = linprog( c, A, b, A_eq, B_eq, l, u )
```

Problems must be expressed in the following standard form:

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & Ax \preceq b \\ & A_{\text{eq}} x = b_{\text{eq}} \\ & \ell \preceq x \preceq u\end{array}$$

As standard forms go, this one is quite flexible: in fact, the first step `linprog` often does is to convert this problem into a more restricted standard form!

Conversion to standard form: common tricks

Representing free variables as the difference of nonnegative variables:

$$x \text{ free} \implies x_+ - x_-, \quad x_+ \geq 0, \quad x_- \geq 0$$

Eliminating inequality constraints using *slack variables*:

$$a^T x \leq b \implies a^T x + s = b, \quad s \geq 0$$

Splitting equality constraints into inequalities:

$$a^T x = b \implies a^T x \leq b, \quad a^T x \geq b$$

Solver example: SeDuMi

A program for solving LPs, SOCPs, SDPs, and related problems:

```
x = sedumi( A, b, c, K )
```

Solves problems of the form:

$$\begin{array}{ll}\text{minimize} & c^T x \\ \text{subject to} & Ax = b \\ & x \in \mathcal{K} \triangleq \mathcal{K}_1 \times \mathcal{K}_2 \times \cdots \times \mathcal{K}_L\end{array}$$

where each set $\mathcal{K}_i \subseteq \mathbf{R}^{n_i}$, $i = 1, 2, \dots, L$ is chosen from a very short list of cones (see next slide...)

The Matlab variable K gives the number, types, and dimensions of the cones \mathcal{K}_i

Cones supported by SeDuMi

- free variables: \mathbf{R}^{n_i}
- a nonnegative orthant: $\mathbf{R}_+^{n_i}$ (for linear inequalities)
- a real or complex second-order cone:

$$\mathbf{Q}^n \triangleq \{ (x, y) \in \mathbf{R}^n \times \mathbf{R} \mid \|x\|_2 \leq y \}$$

$$\mathbf{Q}_c^n \triangleq \{ (x, y) \in \mathbf{C}^n \times \mathbf{R} \mid \|x\|_2 \leq y \}$$

- a real or complex semidefinite cone:

$$\mathbf{S}_+^n \triangleq \{ X \in \mathbf{R}^{n \times n} \mid X = X^T, X \succeq 0 \}$$

$$\mathbf{H}_+^n \triangleq \{ X \in \mathbf{C}^{n \times n} \mid X = X^H, X \succeq 0 \}$$

The cones must be arranged in this order: *i.e.*, the free variables first, then the nonnegative orthants, then the second-order cones, then the semidefinite cones

Example: Norm approximation

Consider the problem

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

An optimal value x^* minimizes the residuals

$$r_k \triangleq a_k^T x - b_k \quad k = 1, 2, \dots, m$$

according to the measure defined by the norm $\|\cdot\|$

Obviously, the value of x^* depends significantly upon the choice of that norm...

...and so does the process of conversion to standard form

Euclidean (ℓ_2) norm

$$\text{minimize} \quad \|Ax - b\|_2 = \left(\sum_{k=1}^m (a_k^T x - b_k)^2 \right)^{1/2}$$

No need to use `linprog` or `SeDuMi` here: this is a *least squares* problem, with an analytic solution

$$x^* = (A^T A)^{-1} A^T b$$

In MATLAB or Octave, a single command computes the solution:

```
>> x = A \ b;
```


Chebyshev (ℓ_∞) norm

$$\text{minimize } f(Ax - b) \triangleq \|Ax - b\|_\infty = \max_{1 \leq k \leq m} |a_k^T x - b|$$

This can be expressed as a linear program:

$$\begin{array}{ll} \text{minimize} & q \\ \text{subject to} & -q\mathbf{1} \leq Ax - b \leq +q\mathbf{1} \end{array} \quad \Longrightarrow \quad \begin{array}{ll} \text{minimize} & \begin{bmatrix} \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ q \end{bmatrix} \\ \text{subject to} & \begin{bmatrix} A & -\mathbf{1} \\ -A & -\mathbf{1} \end{bmatrix} \begin{bmatrix} x \\ q \end{bmatrix} \leq \begin{bmatrix} b \\ -b \end{bmatrix} \end{array}$$

The `linprog` call:

```
xq = linprog( [zeros(n,1);1], ...  
              [A,-ones(m,1);-A,-ones(m,1)], [b;-b] );  
x = xq(1:n);
```

Manhattan (ℓ_1) norm

$$\text{minimize} \quad f(Ax - b) \triangleq \|Ax - b\|_1 = \sum_{k=1}^m |a_k^T x - b|$$

LP formulation:

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T w \\ \text{subject to} & -w \leq Ax - b \leq w \end{array} \quad \Longrightarrow \quad \begin{array}{ll} \text{minimize} & \begin{bmatrix} \mathbf{0} & \mathbf{1} \end{bmatrix}^T \begin{bmatrix} x \\ w \end{bmatrix} \\ \text{subject to} & \begin{bmatrix} A & -I \\ -A & -I \end{bmatrix} \begin{bmatrix} x \\ w \end{bmatrix} \leq \begin{bmatrix} b \\ -b \end{bmatrix} \end{array}$$

The `linprog` call:

```
xw = linprog( [zeros(n,1);ones(m,1)], ...  
              [A,-eye(m,1);-A,-eye(m,1)], [b;-b] );  
x = xw(1:n);
```

Largest- L norm

Let $|w|_{[k]}$ represent the k -th largest element (in magnitude) of $w \in \mathbf{R}^m$:

$$|w|_{[1]} \geq |w|_{[2]} \geq \cdots \geq |w|_{[m]}$$

Then define the *largest- L norm* as

$$\|w\|_{[L]} \triangleq |w|_{[1]} + |w|_{[2]} + \cdots + |w|_{[L]} \triangleq \sum_{k=1}^L \sigma_k(\mathbf{diag}(w))$$

for any integer $1 \leq L \leq m$. Special cases include

$$\|w\|_{[1]} \equiv \|w\|_{\infty}, \quad \|w\|_{[m]} \triangleq \|w\|_1$$

but novel results are produced for all $1 < L < m$

Largest- L norm (continued)

This norm can be also represented in a linear program:

$$\begin{array}{ll}\text{minimize} & \mathbf{1}^T v + Lq \\ \text{subject to} & -v - \mathbf{1}q \leq Ax - b \leq +v + \mathbf{1}q \\ & v \geq 0\end{array}$$

The `linprog` call:

```
xvq = linprog( [zeros(n,1);ones(m,1);L], ...  
               [A,-eye(m),ones(m,1);-A,-eye(m),-ones(m,1)], ...  
               [], [], [-Inf*ones(n,1);zeros(m,1);-Inf] );  
x = xvq(1:n);
```

This is not much more complex than the ℓ_1 and ℓ_∞ cases—but of course, you have to know this trick (and few do)

Constrained Euclidean (ℓ_2) norm

Add some constraints:

$$\begin{array}{ll}\text{minimize} & \|Ax - b\|_2 \\ \text{subject to} & Cx = d \\ & \ell \leq x \leq u\end{array}$$

This is not a least-squares problem—but it is an SOCP, and SeDuMi can handle it, once it is converted to standard form:

$$\begin{array}{ll}\text{minimize} & z \\ \text{subject to} & Ax - b = y \\ & Cx = d \\ & x - s_\ell = \ell \\ & x + s_u = u \\ & s_\ell, s_u \geq 0 \\ & \|y\|_2 \leq z\end{array} \quad \Longrightarrow \quad \begin{array}{ll}\text{minimize} & \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \end{bmatrix}^T \bar{x} \\ \text{subject to} & \begin{bmatrix} A & & & -I \\ C & & & \\ I & -I & & \\ I & & I & \end{bmatrix} \bar{x} = \begin{bmatrix} b \\ d \\ \ell \\ u \end{bmatrix} \\ & \bar{x} \in \mathbf{R}^n \times \mathbf{R}_+^n \times \mathbf{R}_+^n \times \mathbf{Q}^m\end{array}$$

$s_\ell, s_u \in \mathbf{R}_+^n$ are *slack variables*, which are used quite often to convert inequalities to equations

Constrained Euclidean (ℓ_2) norm

The SeDuMi call:

```
AA = [ A,      zeros(m,n), zeros(m,n), -eye(m),      0 ;  
      C,      zeros(p,n), zeros(p,n), zeros(p,n), 0 ;  
      eye(n), -eye(n),      zeros(n,n), zeros(n,n), 0 ;  
      eye(n), zeros(n,n), eye(n),      zeros(n,n), 0 ];  
bb = [ b ; d ; 1 ; u ];  
cc = [ zeros( 3 * n + m, 1 ) ; 1 ];  
K.f = n; K.l = 2 * n; K.q = m + 1;  
xsyz = sedumi( AA, bb, cc, K );  
x = xsyz( 1 : n );
```

Hopefully it is getting clear: this can be cumbersome

Modeling frameworks

A *modeling framework* simplifies the use of a numerical technology by shielding the user from the underlying mathematical details. Frameworks for LP, QP, NLP, and SDP are directly responsible for increased popularity of these problem classes

Most common modeling frameworks provide a language that allows problems to be specified in a natural syntax, and automatically call an underlying numerical solver

Examples:

- AMPL, GAMS, AIMMS, LINDO, *etc.*: LP, QP, NLP
- SDPSOL, LMITOOL, LMILAB: SDP
- YALMIP: LP, SOCP, SDP

`cvx` is designed to support convex optimization; or more specifically, *disciplined convex programming*

Disciplined convex programming

People don't simply write down optimization problems and hope that they are convex; instead, they draw from a “mental library” of functions and sets with known convexity properties, and combine them in ways that convex analysis guarantees will produce convex results...

...i.e., using the calculus rules discussed in previous lectures

Disciplined convex programming formalizes this methodology

Two key elements:

- An expandable *atom library*: a collection of functions and sets, or *atoms*, with known properties of convexity, monotonicity, and range.
- A ruleset which governs how those atoms can be used and combined to form a valid problem

Compliant problems are called *disciplined convex programs*, or DCPs

The DCP ruleset

A subset of the calculus rules you have already learned:

- objective functions: convex for minimization, concave for maximization
- constraints:
 - equality and inequality constraints
 - each must be separately convex
- expressions:
 - addition, subtraction, scalar multiplication
 - composition with affine functions
 - nonlinear compositions (with appropriate monotonicity conditions). including elementwise maximums, *etc.*

Not included (for now): perspective transformations, conjugates, pointwise supremum, quasiconvexity

Is the ruleset too limited?

These rules constitute a set of *sufficient* conditions for convexity. For example,

```
sqrt( sum( square( x ) ) )
```

is convex, but does not obey the ruleset, because it is the composition of a concave nondecreasing function and a convex function

In this case, the workaround is simple: use `norm(x)` instead

In some cases, it may be necessary to add new functions (more later)

A practical hurdle? Perhaps, but it does mirror the mental process

Disciplined convex programming: benefits

- Verification of convexity is replaced with enforcement of the ruleset, which can be performed reliably and quickly
- Conversion to solvable form is fully automated
- Because the atom library is extensible, generality is not compromised
- New atoms can be created by experts and shared with novices

Demonstration

Implementing atoms

Obviously, the usefulness of disciplined convex programming depends upon the size of the atom library available to us... or at least, on the ability to expand that library as needed

The manner in which atoms are implemented depends heavily on the capabilities of the underlying solver

Currently, `cvx` uses (SeDuMi), whose standard form permits four constraint types:

- Equality constraints
- Inequality constraints
- Second-order cone constraints
- Linear matrix inequalities

How do you implement a variety of functions under these limited conditions?

Graph implementations

A fundamental principle of convex analysis: the close relationship between convex functions and convex sets via the *epigraph*

The *epigraph* of a function $f : \mathbf{R}^n \rightarrow \mathbf{R}$ is

$$\text{epi } f \triangleq \{ (x, y) \in \mathbf{R}^n \times \mathbf{R} \mid f(x) \leq y \}$$

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

For concave functions, the corresponding construction is the *hypograph*

$$\text{hypo } g \triangleq \{ (x, y) \in \mathbf{R}^n \times \mathbf{R} \mid g(x) \geq y \}$$

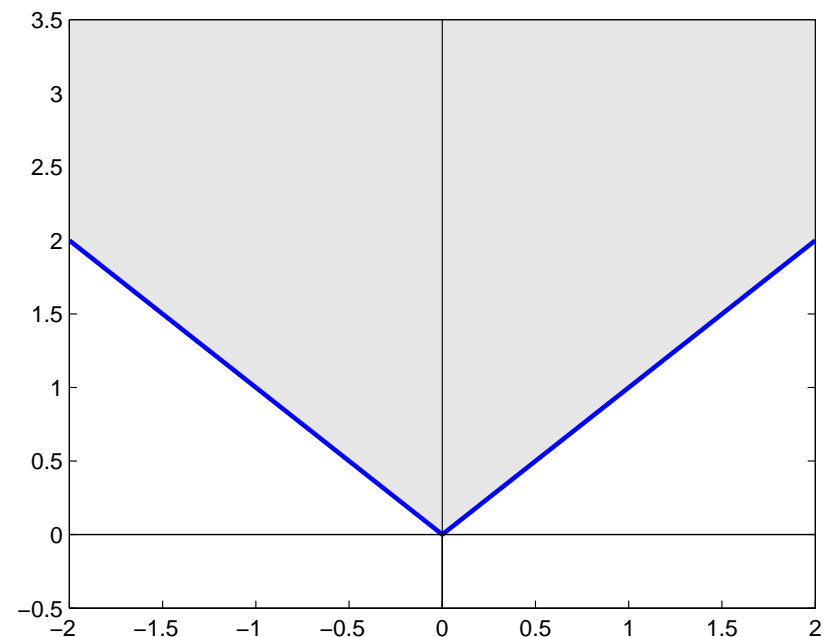
A function is concave **if and only if** its hypograph is a convex set

Graph implementations (continued)

The absolute value function (blue line) and its epigraph (grey region):

$$f(x) \triangleq |x|$$

$$\begin{aligned}\text{epi } f &\triangleq \{ (x, y) \mid |x| \leq y \} \\ &= \{ (x, y) \mid x \leq y, -x \leq y \}\end{aligned}$$



Key observation: for the purposes of convex optimization, the epigraph/hypograph can be used to provide a *complete* description of a function

Graph implementations are, in effect, rules for “rewriting” a function in terms of other atoms by describing its epigraph or hypograph

Example: the two-element maximum

$$f(x, y) \triangleq \max\{x, y\} = \inf \{ z \mid x \leq z, y \leq z \}$$

```
function cvx_optval = max( x, y )  
cvx_begin  
    variable z  
    minimize( z )  
    subject to  
        x <= z;  
        y <= z;  
cvx_end
```


Example: ℓ_1 norm

$$f(x) \triangleq \|x\|_1 = \inf \{ \mathbf{1}^T v \mid -v \leq x \leq v \}$$

```
function cvx_optval = norm_1( x )
n = length( x );
cvx_begin
    variable v(n)
    minimize( sum(v) )
    subject to
        +x <= v;
        -x <= v;
cvx_end
```

This implementation effectively encapsulates the very transformation employed in the norm approximation example

Example: ℓ_2 norm

$$f(x) \triangleq \|x\|_2 \implies \text{epi } f \equiv \mathcal{Q}^n$$

Obviously, the second-order (Lorentz) cone can represent the 2-norm:

```
function cvx_optval = norm_2( x )  
n = length( x );  
cvx_begin  
    variable y  
    { x, y } == lorentz( n );  
cvx_end
```

(In the actual implementation, the complex case is handled too)

Example: Simple squaring

SeDuMi does not even support the basic convex function $f(x) = x^2$ —or does it?

$$x^2 \leq y \implies \begin{bmatrix} y & x \\ x & 1 \end{bmatrix} \succeq 0$$

So $f(x)$ can be represented using a semidefinite program:

```
function cvx_optval = square( x )  
cvx_begin sdp  
    variable y  
    minimize( y )  
    [ y, x ; x, 1 ] >= 0  
cvx_end
```

Of course, if SeDuMi were replaced with a different solver, this implementation might not be necessary

Example: Square root

Graph implementations can freely refer to other graph implementations

For example, the square root function can be implemented using square:

$$f(x) \triangleq \sqrt{x} \geq y \implies y^2 \leq x$$

```
function cvx_optval = sqrt( x )  
cvx_begin  
    variable y  
    maximize( y )  
    subject to  
        square( y ) <= x  
cvx_end
```

Example: eigenvalue functions

If X is symmetric:

$$\lambda_{\min}(X) \geq y \iff X - yI \succeq 0$$

$$\lambda_{\max}(X) \leq y \iff yI - X \succeq 0$$

```
function cvx_optval = lambda_min( X )
n = size(X,1);
cvx_begin
    variable y
    minimize( y )
    X - y * eye( n ) == semidefinite(n);
cvx_end
```

Example: matrix fractional function

$$f : \mathbf{R}^n \times \mathbf{S}^n \rightarrow \mathbf{R}, \quad f(x, Y) \triangleq \begin{cases} x^T Y^{-1} x & Y \succ 0 \\ +\infty & \text{otherwise} \end{cases}$$

Convex in both x and Y , and can be represented using an LMI:

$$f(x, Y) \leq z \quad \Longleftrightarrow \quad \begin{bmatrix} Y & x \\ x^T & z \end{bmatrix} \succeq 0$$

```
function cvx_optval = matrix_frac( x, Y )
n = length(x);
cvx_begin
    variable z
    minimize( z )
    subject to
        [(Y+Y')/2 x; x' z] == semidefinite(n+1);
cvx_end
```

Incomplete specifications

`cvx` actually supports a more general concept we call *incomplete specifications*, which is

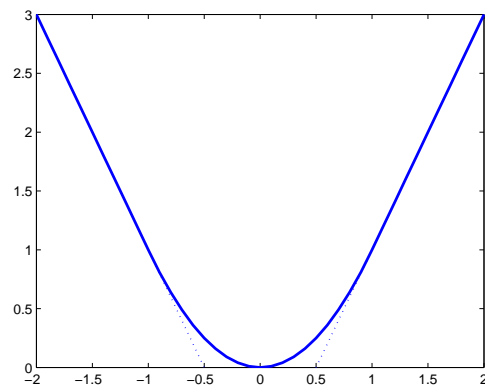
$$\begin{array}{ccc} f(x) = \inf_y \{ g(x, y) \mid (x, y) \in S \} & \text{or} & f(x) = \sup_y \{ g(x, y) \mid (x, y) \in S \} \\ (f, g \text{ convex}) & & (f, g \text{ concave}) \end{array}$$

where S is a convex set

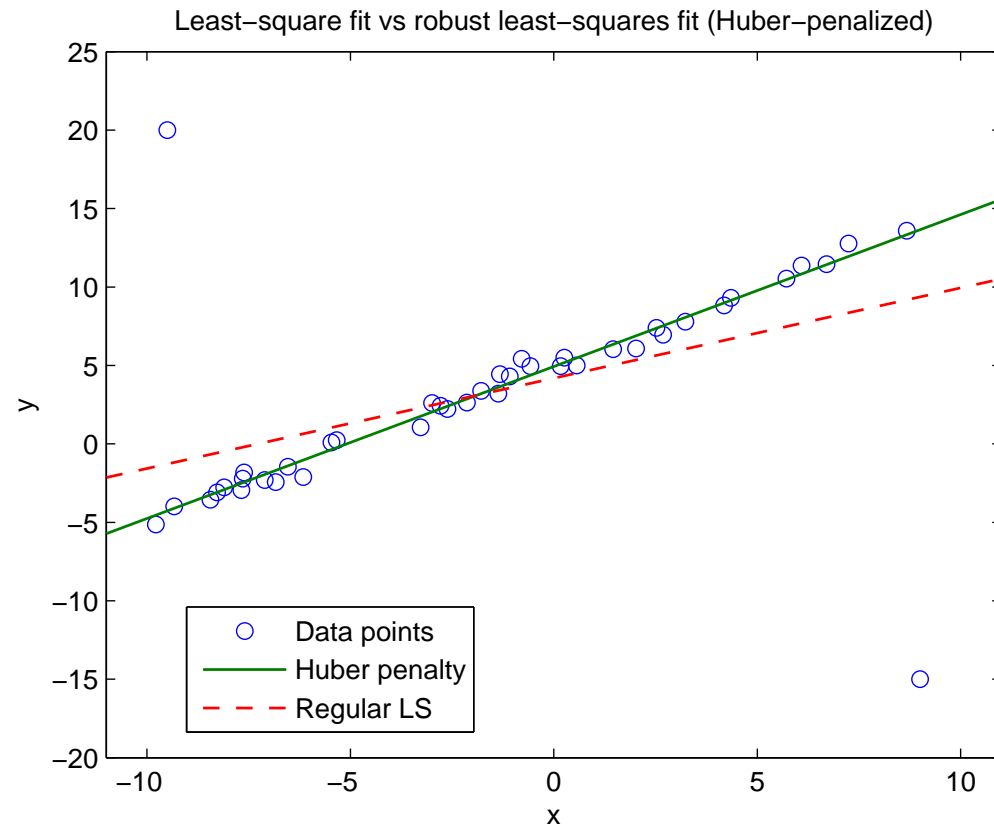
Graph implementations are just special cases of this more general framework

Example: Huber penalty

$$f(x) \triangleq \begin{cases} x^2 & |x| \leq 1 \\ 2|x| - 1 & |x| \geq 1 \end{cases}$$



Useful for robust regression—to suppress outliers



Example: Huber penalty

$$f(x) \triangleq \begin{cases} x^2 & |x| \leq 1 \\ 2|x| - 1 & |x| \geq 1 \end{cases}$$
$$= \inf \{ w^2 + 2v \mid |x| \leq w + v, w \leq 1, v \geq 0 \}$$

```
function cvx_optval = huber( x )
cvx_begin
    variables v w
    minimize( square( w ) + 2 * v )
    subject to
        abs( x ) <= w + v
        w <= 1
        v >= 0
cvx_end
```

Example: distance from an ellipsoid

Let $f_{\mathcal{E}}(x)$ be the distance from a point x to an ellipsoid \mathcal{E} , where \mathcal{E} is described by a pair $(P, y_c) \in \mathbf{R}^{n \times n} \times \mathbf{R}^n$:

$$f(x, y) \triangleq \inf \{ \|x - y\| \mid y \in \mathcal{E} \}$$
$$\mathcal{E} \triangleq \{ y \in \mathbf{R}^n \mid \|P(y - y_c)\|_2 \leq 1 \}$$

Then $g(x, y) \triangleq \|x - y\|_2$, $S \triangleq \mathbf{R}^n \times \mathcal{E}$:

```
function cvx_optval = dist_ellipse( x, P, yc )
n = length( x );
cvx_begin
    variable y(n)
    minimize( norm( x - y, 2 ) )
    subject to
        norm( P * ( y - yc ), 2 ) <= 1;
cvx_end
```