Convex Optimization

Stephen Boyd   Lieven Vandenberghe

Revised slides by Stephen Boyd, Lieven Vandenberghe, and Parth Nobel

1. Introduction

Outline

Mathematical optimization

Convex optimization

Optimization problem

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

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

- variations: maximize objective, multiple objectives, ...

Finding good (or best) actions

- x represents some **action**, *e.g.*,
  - trades in a portfolio
  - airplane control surface deflections
  - schedule or assignment
  - resource allocation
- constraints limit actions or impose conditions on outcome
- the smaller the objective $f_0(x)$, the better
  - total cost (or negative profit)
  - deviation from desired or target outcome
  - risk
  - fuel use

Finding good models

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

Worst-case analysis (pessimization)

- variables are actions or parameters out of our control
  (and possibly under the control of an adversary)
- constraints limit the possible values of the parameters
- minimizing $-f_0(x)$ finds **worst possible parameter values**

- if the worst possible value of $f_0(x)$ is tolerable, you're OK
- it's good to know what the worst possible scenario can be

Optimization-based models

- model an entity as taking actions that solve an optimization problem
  - an individual makes choices that maximize expected utility
  - an organism acts to maximize its reproductive success
  - reaction rates in a cell maximize growth
  - currents in a circuit minimize total power

- (except the last) these are **very crude** models
- and yet, they often work very well

Basic use model for mathematical optimization

- instead of saying how to choose (action, model) x
- you articulate what you want (by stating the problem)
- then let an algorithm decide on (action, model) x

Can you solve it?

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

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

Nonlinear optimization

traditional techniques for general nonconvex problems involve compromises

local optimization methods (nonlinear programming)

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

global optimization methods

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

Outline

Mathematical optimization

Convex optimization

Convex optimization

convex optimization problem:

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

- variable $x \in \mathbf{R}^n$
- equality constraints are linear
- $f_0, \ldots, f_m$ are **convex**: for $\theta \in [0, 1]$,

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

i.e., f_i have nonnegative (upward) curvature

When is an optimization problem hard to solve?

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

- the classical view is **wrong**

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

Solving convex optimization problems

- many different algorithms (that run on many platforms)
  - interior-point methods for up to 10000s of variables
  - first-order methods for larger problems
  - do not require initial point, babysitting, or tuning
- can develop and deploy quickly using modeling languages such as CVXPY
- solvers are reliable, so can be embedded
- code generation yields real-time solvers that execute in milliseconds (*e.g.*, on Falcon 9 and Heavy for landing)

Modeling languages for convex optimization

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

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

- gets close to the basic idea: **say what you want, not how to get it**

CVXPY example: non-negative least squares

math:

$$
\begin{array}{ll}
\text{minimize} & \|Ax - b\|_2^2 \\
\text{subject to} & x \geq 0
\end{array}
$$

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

CVXPY code:

```
import cvxpy as cp

A, b = ...

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

Brief history of convex optimization

- **theory (convex analysis)**: 1900–1970
- **algorithms**
  - 1947: simplex algorithm for linear programming (Dantzig)
  - 1960s: early interior-point methods (Fiacco & McCormick, Dikin, ...)
  - 1970s: ellipsoid method and other subgradient methods
  - 1980s & 90s: interior-point methods (Karmarkar, Nesterov & Nemirovski)
  - since 2000s: many methods for large-scale convex optimization
- **applications**
  - before 1990: mostly in operations research, a few in engineering
  - since 1990: many applications in engineering (control, signal processing, communications, circuit design, ...)
  - since 2000s: machine learning and statistics, finance

Summary

convex optimization problems

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

2. Convex sets

Outline

Some standard convex sets

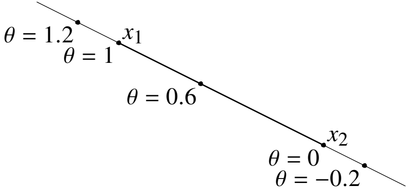
Operations that preserve convexity

Generalized inequalities

Separating and supporting hyperplanes

Affine set

line through x_1, x_2: all points of form $x = \theta x_1 + (1 - \theta)x_2$, with $\theta \in \mathbf{R}$



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)

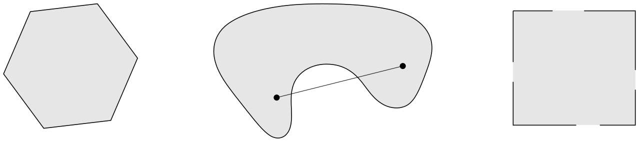
Convex set

line segment between x_1 and x_2: all points of form $x = \theta x_1 + (1-\theta)x_2$, with $0 \le \theta \le 1$

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

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

examples (one convex, two nonconvex sets)



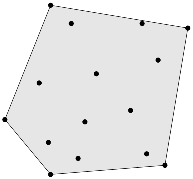
Convex combination and convex hull

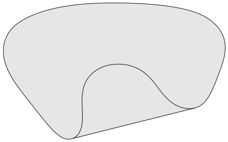
convex combination of $x_1, \ldots, x_k$: any point x of the form

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

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

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



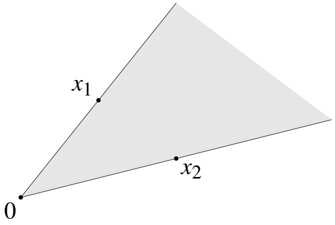


Convex cone

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

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

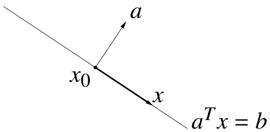
with $\theta_1 \geq 0$, $\theta_2 \geq 0$



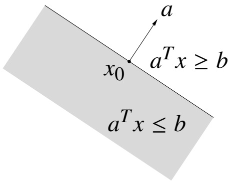
convex cone: set that contains all conic combinations of points in the set

Hyperplanes and halfspaces

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



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



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

Euclidean balls and ellipsoids

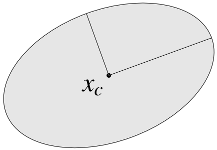
(Euclidean) ball with center x_c and radius r:

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

ellipsoid: set of the form

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

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



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

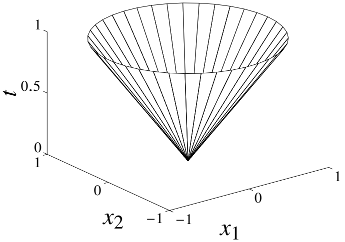
Norm balls and norm cones

- **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:** $\{(x, t) \mid \|x\| \leq t\}$
- norm balls and cones are convex

Euclidean norm cone

$$\{(x, t) \mid \|x\|_2 \leq t\} \subset \mathbf{R}^{n+1}$$

is called **second-order cone**

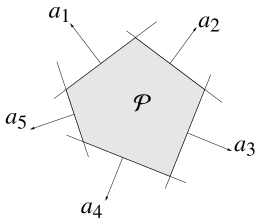


Polyhedra

- **polyhedron** is solution set of finitely many linear inequalities and equalities

$$\{x \mid Ax \preceq b,\ Cx = d\}$$

  ($A \in \mathbf{R}^{m\times n}$, $C \in \mathbf{R}^{p\times n}$, $\preceq$ is componentwise inequality)
- intersection of finite number of halfspaces and hyperplanes
- example with no equality constraints; a_i^T are rows of A



Positive semidefinite cone

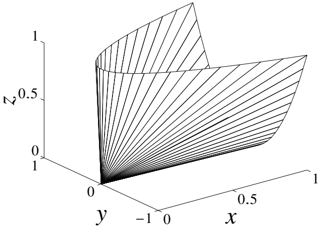
notation:

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

$$X \in \mathbf{S}^n_+ \quad \Longleftrightarrow \quad z^T X z \geq 0 \text{ for all } z$$

- $\mathbf{S}^n_+$ is a convex cone, the **positive semidefinite cone**
- $\mathbf{S}^n_{++} = \{X \in \mathbf{S}^n \mid X \succ 0\}$: positive definite (symmetric) $n \times n$ matrices

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



Outline

Some standard convex sets

Operations that preserve convexity

Generalized inequalities

Separating and supporting hyperplanes

Showing a set is convex

methods for establishing convexity of a set C

1. apply definition: show $x_1, x_2 \in C$, $0 \leq \theta \leq 1 \implies \theta x_1 + (1-\theta)x_2 \in C$
   - recommended only for **very simple** sets

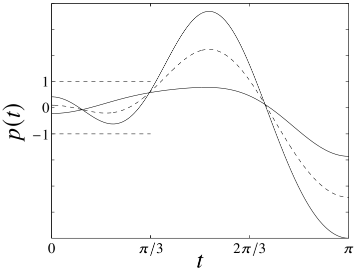
2. use convex functions (next lecture)

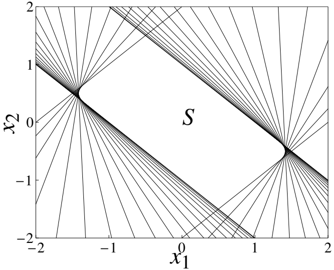
3. show that C is obtained from simple convex sets (hyperplanes, halfspaces, norm balls, ...) by operations that preserve convexity
   - intersection
   - affine mapping
   - perspective mapping
   - linear-fractional mapping

you'll mostly use methods 2 and 3

Intersection

- the intersection of (any number of) convex sets is convex
- **example:**
  - $S = \{x \in \mathbf{R}^m \mid |p(t)| \leq 1 \text{ for } |t| \leq \pi/3\}$, with $p(t) = x_1 \cos t + \cdots + x_m \cos mt$
  - write $S = \bigcap_{|t| \leq \pi/3} \{x \mid |p(t)| \leq 1\}$, *i.e.*, an intersection of (convex) slabs
- picture for $m = 2$:





Affine mappings

- suppose $f : \mathbf{R}^n \to \mathbf{R}^m$ is affine, *i.e.*, $f(x) = Ax + b$ with $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$
- the **image** of a convex set under f is convex

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

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

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

Examples

- scaling, translation: $aS + b = \{ax + b \mid x \in S\}$, $a, b \in \mathbf{R}$
- projection onto some coordinates: $\{x \mid (x, y) \in S\}$
- if $S \subseteq \mathbf{R}^n$ is convex and $c \in \mathbf{R}^n$, $c^T S = \{c^T x \mid x \in S\}$ is an interval
- 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** $P : \mathbf{R}^{n+1} \to \mathbf{R}^n$:

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

- images and inverse images of convex sets under perspective are convex

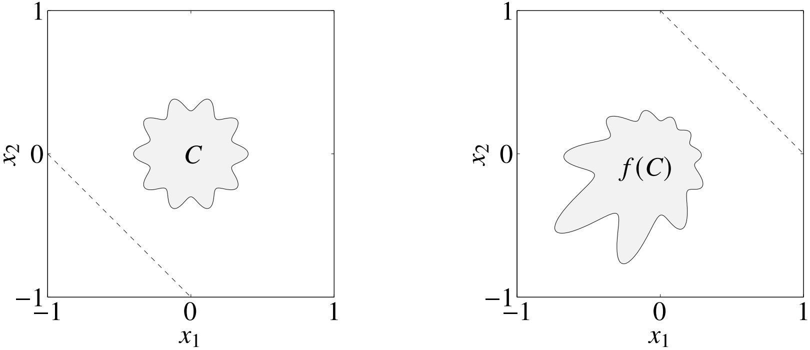
- **linear-fractional function** $f : \mathbf{R}^n \to \mathbf{R}^m$:

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

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

Linear-fractional function example

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



Outline

Some standard convex sets

Operations that preserve convexity

Generalized inequalities

Separating and supporting hyperplanes

Proper cones

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

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

Generalized inequality

- (nonstrict and strict) **generalized inequality** defined by a proper cone K:

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

- **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 ($K = \mathbf{S}^n_+$): $X \preceq_{\mathbf{S}^n_+} Y \iff Y - X$ positive semidefinite

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

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

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

Outline

Some standard convex sets

Operations that preserve convexity

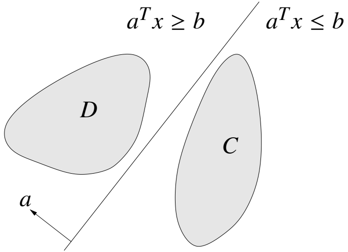
Generalized inequalities

Separating and supporting hyperplanes

Separating hyperplane theorem

- if C and D are nonempty disjoint (*i.e.*, $C \cap D = \emptyset$) convex sets, there exist $a \neq 0$, b s.t.

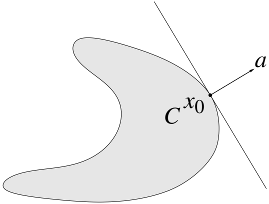
$$a^T x \leq b \text{ for } x \in C, \qquad a^T x \geq b \text{ for } x \in D$$



- 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

- suppose x_0 is a boundary point of set $C \subset \mathbf{R}^n$
- **supporting hyperplane** to C at x_0 has form $\{x \mid a^T x = a^T x_0\}$, 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

3. Convex functions

Outline

Convex functions

Operations that preserve convexity

Constructive convex analysis

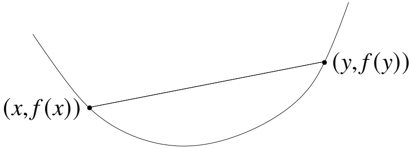
Perspective and conjugate

Quasiconvexity

Definition

- $f : \mathbf{R}^n \to \mathbf{R}$ is convex if $\mathbf{dom}\, f$ is a convex set and for all $x, y \in \mathbf{dom}\, f$, $0 \le \theta \le 1$,

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



- f is concave if $-f$ is convex
- f is strictly convex if $\mathbf{dom}\, f$ is convex and for $x, y \in \mathbf{dom}\, f$, $x \ne y$, $0 < \theta < 1$,

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

Examples on $\mathbf{R}$

convex functions:

- 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 $p \geq 1$
- positive part (relu): $\max\{0, x\}$

concave functions:

- 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 $\mathbf{R}_{++}$
- entropy: $-x \log x$ on $\mathbf{R}_{++}$
- negative part: $\min\{0, x\}$

Examples on $\mathbf{R}^n$

convex functions:

- affine functions: $f(x) = a^T x + b$
- any norm, *e.g.*, the ℓ_p norms
  - $\|x\|_p = (|x_1|^p + \cdots + |x_n|^p)^{1/p}$ for $p \geq 1$
  - $\|x\|_\infty = \max\{|x_1|, \ldots, |x_n|\}$
- sum of squares: $\|x\|_2^2 = x_1^2 + \cdots + x_n^2$
- max function: $\max(x) = \max\{x_1, x_2, \ldots, x_n\}$
- softmax or log-sum-exp function: $\log(\exp x_1 + \cdots + \exp x_n)$

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

- $X \in \mathbf{R}^{m\times n}$ ($m \times n$ matrices) is the variable
- general affine function has form

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

  for some $A \in \mathbf{R}^{m\times n}$, $b \in \mathbf{R}$
- spectral norm (maximum singular value) is convex

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

- log-determinant: for $X \in \mathbf{S}^n_{++}$, $f(X) = \log\det X$ is concave

Extended-value extension

- suppose f is convex on $\mathbf{R}^n$, with domain $\mathbf{dom}\, f$
- its extended-value extension $\tilde{f}$ is function $\tilde{f} : \mathbf{R}^n \to \mathbf{R} \cup \{\infty\}$

$$\tilde{f}(x) = \begin{cases} f(x) & x \in \mathbf{dom}\, f \\ \infty & x \notin \mathbf{dom}\, f \end{cases}$$

- often simplifies notation; for example, the condition

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

  (as an inequality in $\mathbf{R} \cup \{\infty\}$), means the same as the two conditions
  - $\mathbf{dom}\, f$ is convex
  - $x, y \in \mathbf{dom}\, f,\ 0 \leq \theta \leq 1 \quad \Longrightarrow \quad f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta)f(y)$

Restriction of a convex function to a line

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

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

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

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

Example

- $f : \mathbf{S}^n \to \mathbf{R}$ with $f(X) = \log\det X$, $\mathbf{dom}\, f = \mathbf{S}^n_{++}$
- consider line in $\mathbf{S}^n$ given by $X + tV$, $X \in \mathbf{S}^n_{++}$, $V \in \mathbf{S}^n$, $t \in \mathbf{R}$

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

  where λ_i are the eigenvalues of $X^{-1/2}VX^{-1/2}$
- g is concave in t (for any choice of $X \in \mathbf{S}^n_{++}$, $V \in \mathbf{S}^n$); hence f is concave

First-order condition

- f is **differentiable** if $\mathbf{dom}\, f$ is open and the gradient

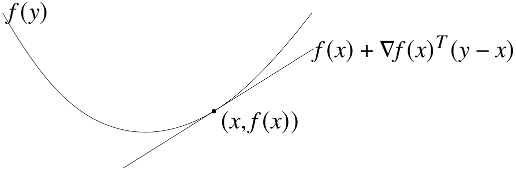
$$\nabla f(x) = \left(\frac{\partial f(x)}{\partial x_1}, \frac{\partial f(x)}{\partial x_2}, \ldots, \frac{\partial f(x)}{\partial x_n} \right) \in \mathbf{R}^n$$

  exists at each $x \in \mathbf{dom}\, f$

- **1st-order condition:** differentiable f with convex domain is convex if and only if

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

- first order Taylor approximation of convex f is a **global underestimator** of f



Second-order conditions

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

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

  exists at each $x \in \mathbf{dom}\, f$

- **2nd-order conditions:** for twice differentiable f with convex domain
  - f is convex if and only if $\nabla^2 f(x) \succeq 0$ for all $x \in \mathbf{dom}\, f$
  - if $\nabla^2 f(x) \succ 0$ for all $x \in \mathbf{dom}\, f$, then f is strictly convex

Examples

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

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

  convex if $P \succeq 0$ (concave if $P \preceq 0$)

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

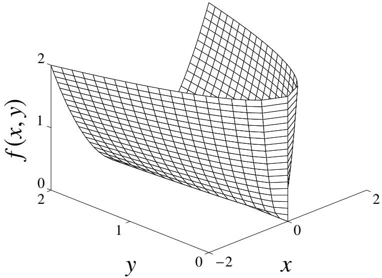
$$\nabla f(x) = 2A^T(Ax - b), \qquad \nabla^2 f(x) = 2A^T A$$

  convex (for any A)

- **quadratic-over-linear:** $f(x, y) = x^2/y$, $y > 0$

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

  convex for $y > 0$



More examples

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

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

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

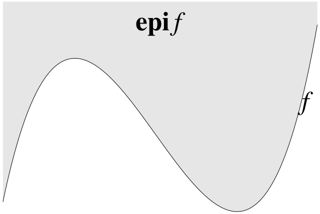
$$v^T \nabla^2 f(x) v = \frac{(\sum_k z_k v_k^2)(\sum_k z_k) - (\sum_k v_k z_k)^2}{(\sum_k z_k)^2} \geq 0$$

  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 above)

Epigraph and sublevel set

- α**-sublevel set** of $f : \mathbf{R}^n \to \mathbf{R}$ is $C_\alpha = \{x \in \mathbf{dom}\, f \mid f(x) \le \alpha\}$
- sublevel sets of convex functions are convex sets (but converse is false)
- **epigraph** of $f : \mathbf{R}^n \to \mathbf{R}$ is $\mathbf{epi}\, f = \{(x, t) \in \mathbf{R}^{n+1} \mid x \in \mathbf{dom}\, f,\ f(x) \le t\}$



- 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 $x, y \in \mathbf{dom} f$, $0 \le \theta \le 1$,

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

- **extension:** if f is convex and z is a random variable on $\mathbf{dom} f$,

$$f(\mathbf{E}\, z) \le \mathbf{E}\, f(z)$$

- basic inequality is special case with discrete distribution

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

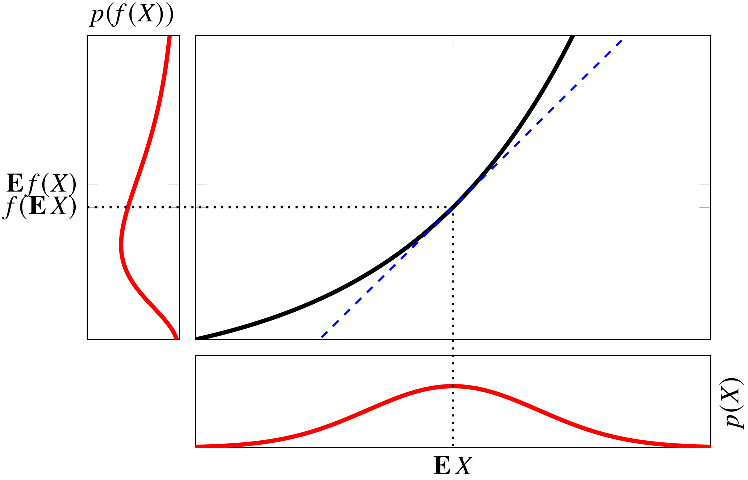
Example: log-normal random variable

- suppose $X \sim \mathcal{N}(\mu, \sigma^2)$
- with $f(u) = \exp u$, $Y = f(X)$ is log-normal
- we have $\mathbf{E} f(X) = \exp(\mu + \sigma^2/2)$
- Jensen's inequality is

$$f(\mathbf{E}\, X) = \exp \mu \leq \mathbf{E} f(X) = \exp(\mu + \sigma^2/2)$$

  which indeed holds since $\exp \sigma^2/2 > 1$

Example: log-normal random variable



Outline

Convex functions

Operations that preserve convexity

Constructive convex analysis

Perspective and conjugate

Quasiconvexity

Showing a function is convex

methods for establishing convexity of a function f

1. verify definition (often simplified by restricting to a line)
2. for twice differentiable functions, show $\nabla^2 f(x) \succeq 0$
   - recommended only for **very simple** functions
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

you'll mostly use methods 2 and 3

Nonnegative scaling, sum, and integral

- **nonnegative multiple:** αf is convex if f is convex, $\alpha \geq 0$
- **sum:** $f_1 + f_2$ convex if f_1, f_2 convex
- **infinite sum:** if $f_1, f_2, \ldots$ are convex functions, infinite sum $\sum_{i=1}^{\infty} f_i$ is convex
- **integral:** if $f(x, \alpha)$ is convex in x for each $\alpha \in \mathcal{A}$, then $\displaystyle\int_{\alpha \in \mathcal{A}} f(x, \alpha)\, d\alpha$ is convex

- there are analogous rules for concave functions

Composition with affine function

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

examples

- log barrier for linear inequalities

$$f(x) = -\sum_{i=1}^{m} \log(b_i - a_i^T x), \qquad \mathbf{dom}\, f = \{x \mid a_i^T x < b_i, i = 1, \ldots, m\}$$

- norm approximation error: $f(x) = \|Ax - b\|$ (any norm)

Pointwise maximum

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

examples

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

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

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

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

Pointwise supremum

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

- distance to farthest point in a set C: $f(x) = \sup_{y \in C} \|x - y\|$
- maximum eigenvalue of symmetric matrix: for $X \in \mathbf{S}^n$, $\lambda_{\max}(X) = \sup_{\|y\|_2 = 1} y^T X y$ is convex
- support function of a set C: $S_C(x) = \sup_{y \in C} y^T x$ is convex

Partial minimization

- the function $g(x) = \inf_{y \in C} f(x, y)$ is called the **partial minimization** of f (w.r.t. y)
- if $f(x, y)$ is convex in (x, y) and C is a convex set, then partial minimization g is convex

examples

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

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

  minimizing over y gives $g(x) = \inf_y f(x, y) = x^T (A - BC^{-1} B^T) x$
  g is convex, hence Schur complement $A - BC^{-1} B^T \succeq 0$

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

Composition with scalar functions

- composition of $g : \mathbf{R}^n \to \mathbf{R}$ and $h : \mathbf{R} \to \mathbf{R}$ is $f(x) = h(g(x))$ (written as $f = h \circ g$)
- composition f is convex if
  - g convex, h convex, $\tilde{h}$ nondecreasing
  - or g concave, h convex, $\tilde{h}$ nonincreasing

  (monotonicity must hold for extended-value extension $\tilde{h}$)
- proof (for $n = 1$, differentiable g, h)

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

examples

- $f(x) = \exp g(x)$ is convex if g is convex
- $f(x) = 1/g(x)$ is convex if g is concave and positive

General composition rule

- composition of $g : \mathbf{R}^n \to \mathbf{R}^k$ and $h : \mathbf{R}^k \to \mathbf{R}$ is $f(x) = h(g(x)) = h(g_1(x), g_2(x), \ldots, g_k(x))$
- f is convex if h is convex and for each i one of the following holds
  - g_i convex, $\tilde{h}$ nondecreasing in its ith argument
  - g_i concave, $\tilde{h}$ nonincreasing in its ith argument
  - g_i affine

- you will use this composition rule **constantly** throughout this course
- you need to commit this rule to memory

Examples

- $\log \sum_{i=1}^{m} \exp g_i(x)$ is convex if g_i are convex
- $f(x) = p(x)^2/q(x)$ is convex if
  - p is nonnegative and convex
  - q is positive and concave

- composition rule subsumes others, *e.g.*,
  - αf is convex if f is, and $\alpha \geq 0$
  - sum of convex (concave) functions is convex (concave)
  - max of convex functions is convex
  - min of concave functions is concave

Outline

Convex functions

Operations that preserve convexity

Constructive convex analysis

Perspective and conjugate

Quasiconvexity

Constructive convexity verification

- start with function f given as **expression**
- build parse tree for expression
  - leaves are variables or constants
  - nodes are functions of child expressions
- use composition rule to tag subexpressions as convex, concave, affine, or none
- if root node is labeled convex (concave), then f is convex (concave)
- extension: tag sign of each expression, and use sign-dependent monotonicity

- this is sufficient to show f is convex (concave), but not necessary
- this method for checking convexity (concavity) is readily automated

Example

the function

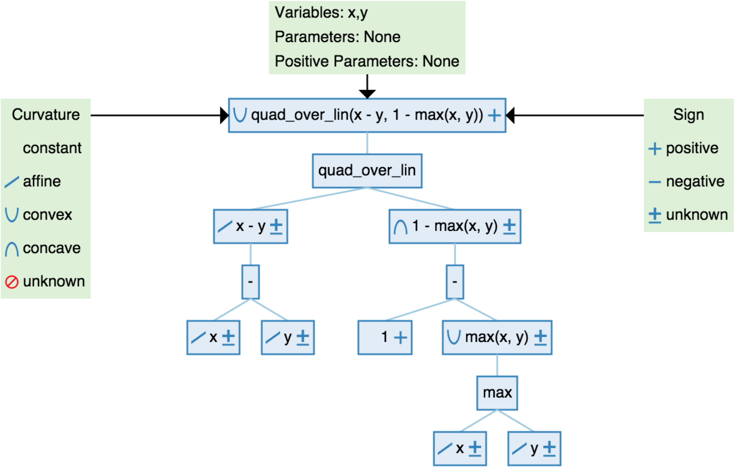
$$f(x,y) = \frac{(x-y)^2}{1-\max(x,y)}, \qquad x < 1, \quad y < 1$$

is convex

constructive analysis:

- (leaves) x, y, and 1 are affine
- $\max(x,y)$ is convex; $x-y$ is affine
- $1-\max(x,y)$ is concave
- function u^2/v is convex, monotone decreasing in v for $v > 0$
- f is composition of u^2/v with $u = x-y$, $v = 1-\max(x,y)$, hence convex

Example (from dcp.stanford.edu)



Disciplined convex programming

in **disciplined convex programming** (DCP) users construct convex and concave functions as expressions using constructive convex analysis

- expressions formed from
  - **variables**,
  - **constants**,
  - and **atomic functions** from a library
- atomic functions have known convexity, monotonicity, and sign properties
- all subexpressions match general composition rule
- a valid DCP function is
  - convex-by-construction
  - 'syntactically' convex (can be checked 'locally')
- convexity depends only on attributes of atomic functions, not their meanings
  - *e.g.*, could swap $\sqrt{\cdot}$ and $\sqrt[4]{\cdot}$, or $\exp \cdot$ and $(\cdot)_+$, since their attributes match

CVXPY example

$$\frac{(x-y)^2}{1-\max(x,y)}, \qquad x<1, \quad y<1$$

```
import cvxpy as cp
x = cp.Variable()
y = cp.Variable()
expr = cp.quad_over_lin(x - y, 1 - cp.maximum(x, y))
expr.curvature # Convex
expr.sign # Positive
expr.is_dcp() # True
```

(atom `quad_over_lin(u,v)` includes domain constraint `v>0`)

DCP is only sufficient

- consider convex function $f(x) = \sqrt{1 + x^2}$
- expression `f1 = cp.sqrt(1+cp.square(x))` is **not** DCP
- expression `f2 = cp.norm2([1,x])` **is** DCP
- CVXPY will not recognize `f1` as convex, even though it represents a convex function

Outline

Convex functions

Operations that preserve convexity

Constructive convex analysis

Perspective and conjugate

Quasiconvexity

Perspective

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

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

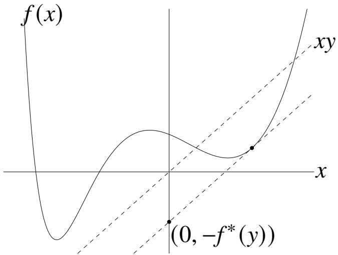
- g is convex if f is convex

examples

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

Conjugate function

- the **conjugate** of a function f is $f^*(y) = \sup_{x \in \mathbf{dom} f} (y^T x - f(x))$



- f^* is convex (even if f is not)
- will be useful in chapter 5

Examples

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

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

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

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

Outline

Convex functions

Operations that preserve convexity

Constructive convex analysis

Perspective and conjugate

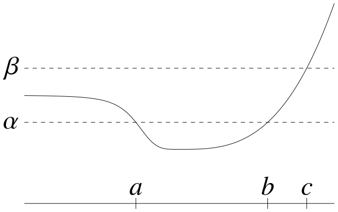
Quasiconvexity

Quasiconvex functions

- $f : \mathbf{R}^n \to \mathbf{R}$ 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 $\mathbf{R}$
- $\operatorname{ceil}(x) = \inf\{z \in \mathbf{Z} \mid z \geq x\}$ 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

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

  is quasilinear

Example: Internal rate of return

- cash flow $x = (x_0, \ldots, x_n)$; x_i is payment in period i (to us if $x_i > 0$)
- we assume $x_0 < 0$ (*i.e.*, an initial investment) and $x_0 + x_1 + \cdots + x_n > 0$
- **net present value** (NPV) of cash flow x, for interest rate r, is $\mathrm{PV}(x, r) = \sum_{i=0}^{n} (1+r)^{-i} x_i$
- **internal rate of return** (IRR) is smallest interest rate for which $\mathrm{PV}(x, r) = 0$:

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

- IRR is quasiconcave: superlevel set is intersection of open halfspaces

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

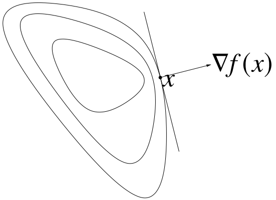
Properties of quasiconvex functions

- **modified Jensen inequality:** for quasiconvex f

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

- **first-order condition:** differentiable f with convex domain is quasiconvex if and only if

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



- **sum** of quasiconvex functions is not necessarily quasiconvex

4. Convex optimization problems

Outline

Optimization problems

Some standard convex problems

Transforming problems

Disciplined convex programming

Geometric programming

Quasiconvex optimization

Multicriterion optimization

Optimization problem in standard form

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

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

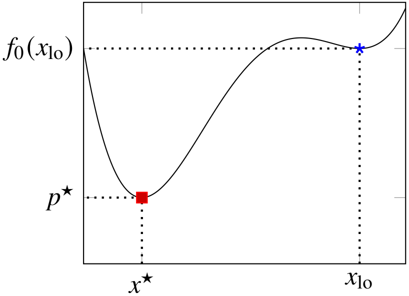
Feasible and optimal points

- $x \in \mathbf{R}^n$ is **feasible** if $x \in \mathbf{dom}\, f_0$ and it satisfies the constraints
- **optimal value** is $p^\star = \inf\{f_0(x) \mid f_i(x) \leq 0,\ i = 1, \ldots, m,\ h_i(x) = 0,\ i = 1, \ldots, p\}$
- $p^\star = \infty$ if problem is infeasible
- $p^\star = -\infty$ if problem is **unbounded below**
- a feasible x is **optimal** if $f_0(x) = p^\star$
- X_{opt} is the set of optimal points

Locally optimal points

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

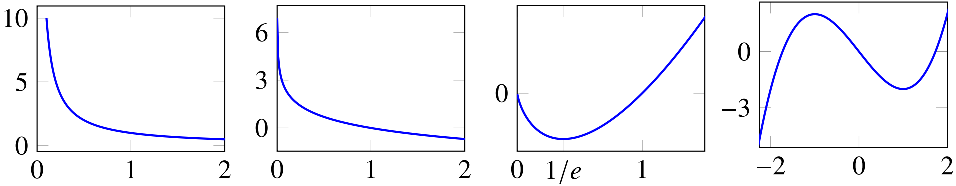
$$
\begin{array}{ll}
\text{minimize (over } z) & f_0(z) \\
\text{subject to} & f_i(z) \leq 0, \quad i = 1, \ldots, m, \quad h_i(z) = 0, \quad i = 1, \ldots, p \\
& \|z - x\|_2 \leq R
\end{array}
$$



Examples

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

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



$f_0(x) = 1/x$ $\quad$ $f_0(x) = -\log x$ $\quad$ $f_0(x) = x \log x$ $\quad$ $f_0(x) = x^3 - 3x$

Implicit and explicit constraints

standard form optimization problem has **implicit constraint**

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

- 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:

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

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

Feasibility problem

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

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

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

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

Standard form convex optimization problem

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

- objective and inequality constraints $f_0, f_1, \ldots, f_m$ are convex
- equality constraints are affine, often written as $Ax = b$
- feasible and optimal sets of a convex optimization problem are convex

- problem is **quasiconvex** if f_0 is quasiconvex, $f_1, \ldots, f_m$ are convex, $h_1, \ldots, h_p$ are affine

Example

- standard form problem

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

- f_0 is convex; feasible set $\{(x_1, x_2) \mid x_1 = -x_2 \leq 0\}$ is convex
- not a convex problem (by our definition) since 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, but there exists a feasible y with $f_0(y) < f_0(x)$
- x locally optimal means there is an $R > 0$ such that

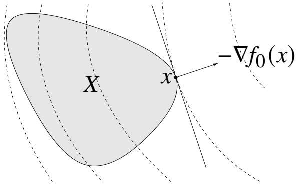
$$z \text{ feasible}, \quad \|z - x\|_2 \le R \quad \Longrightarrow \quad f_0(z) \ge f_0(x)$$

- 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 $f_0(z) \le \theta f_0(y) + (1 - \theta)f_0(x) < f_0(x)$, which contradicts our assumption that x is locally optimal

Optimality criterion for differentiable f_0

- x is optimal for a convex problem 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

Examples

- **unconstrained problem**: x minimizes $f_0(x)$ if and only if $\nabla f_0(x) = 0$
- **equality constrained problem**: x minimizes $f_0(x)$ subject to $Ax = b$ if and only if there exists a ν such that
$$Ax = b, \qquad \nabla f_0(x) + A^T \nu = 0$$
- **minimization over nonnegative orthant**: x minimizes $f_0(x)$ over $\mathbf{R}^n_+$ if and only if
$$x \succeq 0, \qquad \begin{cases} \nabla f_0(x)_i \geq 0 & x_i = 0 \\ \nabla f_0(x)_i = 0 & x_i > 0 \end{cases}$$

Outline

Optimization problems

Some standard convex problems

Transforming problems

Disciplined convex programming

Geometric programming

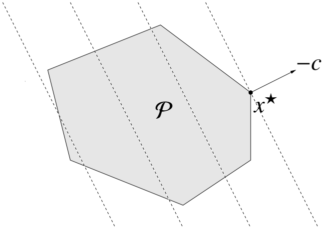
Quasiconvex optimization

Multicriterion optimization

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



Example: Diet problem

- choose nonnegative quantities $x_1, \ldots, x_n$ of n foods
- one unit of food j costs c_j and contains amount A_{ij} of nutrient i
- healthy diet requires nutrient i in quantity at least b_i
- to find cheapest healthy diet, solve

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

- express in standard LP form as

$$\begin{array}{ll} \text{minimize} & c^T x \\ \text{subject to} & \begin{bmatrix} -A \\ -I \end{bmatrix} x \preceq \begin{bmatrix} -b \\ 0 \end{bmatrix} \end{array}$$

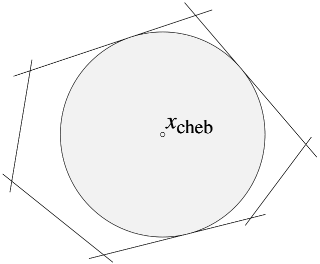
Example: Piecewise-linear minimization

- minimize convex piecewise-linear function $f_0(x) = \max_{i=1,\dots,m}(a_i^T x + b_i)$, $x \in \mathbf{R}^n$
- equivalent to LP

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

  with variables $x \in \mathbf{R}^n$, $t \in \mathbf{R}$
- constraints describe $\mathbf{epi}\, f_0$

Example: Chebyshev center of a polyhedron



Chebyshev center of $\mathcal{P} = \{x \mid a_i^T x \le b_i,\ i = 1, \ldots, m\}$ is center of largest inscribed ball $\mathcal{B} = \{x_c + u \mid \|u\|_2 \le r\}$

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

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

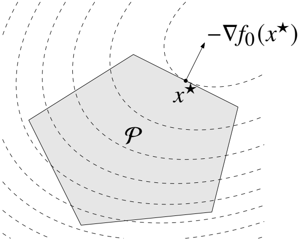
- hence, x_c, r can be determined by solving LP with variables x_c, r

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

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



Example: Least squares

- **least squares** problem: minimize $\|Ax - b\|_2^2$
- analytical solution $x^\star = A^\dagger b$ ($A^\dagger$ is pseudo-inverse)
- can add linear constraints, *e.g.*,
  - $x \geq 0$ (**nonnegative least squares**)
  - $x_1 \leq x_2 \leq \cdots \leq x_n$ (**isotonic regression**)

Example: Linear program with random cost

- LP with random cost c, with mean $\bar{c}$ and covariance Σ
- hence, LP objective $c^T x$ is random variable with mean $\bar{c}^T x$ and variance $x^T \Sigma x$
- **risk-averse** problem:

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

- $\gamma > 0$ is **risk aversion parameter**; controls the trade-off between expected cost and variance (risk)
- express as QP

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

Quadratically constrained quadratic program (QCQP)

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

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

Second-order cone programming

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

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

- 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

Example: Robust linear programming

suppose constraint vectors a_i are uncertain in the LP

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

two common approaches to handling uncertainty

- **deterministic worst-case**: constraints must hold for all $a_i \in \mathcal{E}_i$ (uncertainty ellipsoids)

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

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

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

Deterministic worst-case approach

- uncertainty ellipsoids are $\mathcal{E}_i = \{\bar{a}_i + P_i u \mid \|u\|_2 \leq 1\}$, $(\bar{a}_i \in \mathbf{R}^n,\ P_i \in \mathbf{R}^{n\times n})$
- center of $\mathcal{E}_i$ is $\bar{a}_i$; semi-axes determined by singular values/vectors of P_i
- robust LP

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

- equivalent to SOCP

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

(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

- assume $a_i \sim \mathcal{N}(\bar{a}_i, \Sigma_i)$
- $a_i^T x \sim \mathcal{N}(\bar{a}_i^T x, x^T \Sigma_i x)$, so

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

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

- $\mathbf{prob}(a_i^T x \le b_i) \ge \eta$ can be expressed as $\bar{a}_i^T x + \Phi^{-1}(\eta) \|\Sigma_i^{1/2} x\|_2 \le b_i$
- for $\eta \ge 1/2$, robust LP equivalent to SOCP

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

Conic form problem

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

- constraint $Fx + g \preceq_K 0$ involves a generalized inequality with respect to a proper cone K
- linear programming is a conic form problem with $K = \mathbf{R}^m_+$
- as with standard convex problem
  - feasible and optimal sets are convex
  - any local optimum is global

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,

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

is equivalent to single LMI

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

Example: Matrix norm minimization

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

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

equivalent SDP

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

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

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

Outline

Optimization problems

Some standard convex problems

Transforming problems

Disciplined convex programming

Geometric programming

Quasiconvex optimization

Multicriterion optimization

Change of variables

- $\phi : \mathbf{R}^n \to \mathbf{R}^n$ is one-to-one with $\phi(\mathbf{dom}\,\phi) \supseteq \mathcal{D}$
- consider (possibly non-convex) problem

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

- change variables to z with $x = \phi(z)$
- can solve equivalent problem

$$\begin{array}{ll} \text{minimize} & \tilde{f}_0(z) \\ \text{subject to} & \tilde{f}_i(z) \leq 0, \quad i = 1, \ldots, m \\ & \tilde{h}_i(z) = 0, \quad i = 1, \ldots, p \end{array}$$

  where $\tilde{f}_i(z) = f_i(\phi(z))$ and $\tilde{h}_i(z) = h_i(\phi(z))$
- recover original optimal point as $x^\star = \phi(z^\star)$

Example

- **non-convex** problem

$$\begin{array}{ll} \text{minimize} & x_1/x_2 + x_3/x_1 \\ \text{subject to} & x_2/x_3 + x_1 \leq 1 \end{array}$$

  with implicit constraint $x \succ 0$

- change variables using $x = \phi(z) = \exp z$ to get

$$\begin{array}{ll} \text{minimize} & \exp(z_1 - z_2) + \exp(z_3 - z_1) \\ \text{subject to} & \exp(z_2 - z_3) + \exp(z_1) \leq 1 \end{array}$$

  which is **convex**

Transformation of objective and constraint functions

suppose

- ϕ_0 is monotone increasing
- $\psi_i(u) \leq 0$ if and only if $u \leq 0$, $i = 1, \ldots, m$
- $\varphi_i(u) = 0$ if and only if $u = 0$, $i = 1, \ldots, p$

standard form optimization problem is equivalent to

$$\begin{array}{ll} \text{minimize} & \phi_0(f_0(x)) \\ \text{subject to} & \psi_i(f_i(x)) \leq 0, \quad i = 1, \ldots, m \\ & \varphi_i(h_i(x)) = 0, \quad i = 1, \ldots, p \end{array}$$

example: minimizing $\|Ax - b\|$ is equivalent to minimizing $\|Ax - b\|^2$

Converting maximization to minimization

- suppose ϕ_0 is monotone decreasing
- the maximization problem

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

  is equivalent to the minimization problem

$$\begin{array}{ll} \text{minimize} & \phi_0(f_0(x)) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \dots, m \\ & h_i(x) = 0, \quad i = 1, \dots, p \end{array}$$

- **examples**:
  - $\phi_0(u) = -u$ transforms maximizing a concave function to minimizing a convex function
  - $\phi_0(u) = 1/u$ transforms maximizing a concave positive function to minimizing a convex function

Eliminating equality constraints

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

is equivalent to

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

where F and x_0 are such that $Ax = b \iff x = Fz + x_0$ for some z

Introducing equality constraints

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

is equivalent to

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

Introducing slack variables for linear inequalities

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

is equivalent to

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

Epigraph form

standard form convex problem is equivalent to

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

Minimizing over some variables

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

is equivalent to

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

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

LP and SOCP as SDP

LP and equivalent SDP

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

(note different interpretation of generalized inequalities $\preceq$ in LP and SDP)

SOCP and equivalent SDP

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

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

Convex relaxation

- start with **nonconvex problem**: minimize $h(x)$ subject to $x \in C$
- find convex function $\hat{h}$ with $\hat{h}(x) \leq h(x)$ for all $x \in \mathbf{dom}\, h$ (*i.e.*, a pointwise lower bound on h)
- find set $\hat{C} \supseteq C$ (*e.g.*, $\hat{C} = \mathbf{conv}\, C$) described by linear equalities and convex inequalities

$$\hat{C} = \{x \mid f_i(x) \leq 0,\ i = 1, \ldots, m,\ f_m(x) \leq 0,\ Ax = b\}$$

- convex problem

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

  is a **convex relaxation** of the original problem
- optimal value of relaxation is lower bound on optimal value of original problem

Example: Boolean LP

- **mixed integer linear program (MILP)**:

$$\begin{array}{ll} \text{minimize} & c^T(x,z) \\ \text{subject to} & F(x,z) \preceq g, \quad A(x,z) = b, \quad z \in \{0,1\}^q \end{array}$$

  with variables $x \in \mathbf{R}^n$, $z \in \mathbf{R}^q$
- z_i are called **Boolean variables**
- this problem is in general hard to solve

- **LP relaxation**: replace $z \in \{0,1\}^q$ with $z \in [0,1]^q$
- optimal value of relaxation LP is lower bound on MILP
- can use as heuristic for approximately solving MILP, *e.g.*, **relax and round**

Outline

Optimization problems

Some standard convex problems

Transforming problems

Disciplined convex programming

Geometric programming

Quasiconvex optimization

Multicriterion optimization

Disciplined convex program

- specify objective as
  - minimize {scalar convex expression}, or
  - maximize {scalar concave expression}
- specify constraints as
  - {convex expression} <= {concave expression} or
  - {concave expression} >= {convex expression} or
  - {affine expression} == {affine expression}
- curvature of expressions are DCP certified, *i.e.*, follow composition rule
- DCP-compliant problems can be automatically transformed to standard forms, then solved

CVXPY example

math:

$$
\begin{array}{ll}
\text{minimize} & \|x\|_1 \\
\text{subject to} & Ax = b \\
& \|x\|_\infty \leq 1
\end{array}
$$

- x is the variable
- A, b are given

CVXPY code:

```
import cvxpy as cp

A, b = ...

x = cp.Variable(n)
obj = cp.norm(x, 1)
constr = [
  A @ x == b,
  cp.norm(x, 'inf') <= 1,
]
prob = cp.Problem(cp.Minimize(obj), constr)
prob.solve()
```

How CVXPY works

- starts with your optimization problem $\mathcal{P}_1$
- finds a sequence of equivalent problems $\mathcal{P}_2, \dots, \mathcal{P}_N$
- final problem $\mathcal{P}_N$ matches a standard form (*e.g.*, LP, QP, SOCP, or SDP)
- calls a specialized solver on $\mathcal{P}_N$
- retrieves solution of original problem by reversing the transformations

$$\underset{\substack{\uparrow \\ \text{your problem}}}{\mathcal{P}_1} \iff \mathcal{P}_2 \iff \cdots \iff \mathcal{P}_{N-1} \iff \underset{\substack{\uparrow \\ \text{standard problem}}}{\mathcal{P}_N}$$

Outline

Optimization problems

Some standard convex problems

Transforming problems

Disciplined convex programming

Geometric programming

Quasiconvex optimization

Multicriterion optimization

Geometric programming

- **monomial function**:

$$f(x) = c x_1^{a_1} x_2^{a_2} \cdots x_n^{a_n}, \qquad \mathbf{dom}\, f = \mathbf{R}^n_{++}$$

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

- **posynomial function**: sum of monomials

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

- **geometric program (GP)**

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

  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 $f(x) = c x_1^{a_1} \cdots x_n^{a_n}$ transforms to

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

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

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

- geometric program transforms to convex problem

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

Examples: Frobenius norm diagonal scaling

- we seek diagonal matrix $D = \mathbf{diag}(d)$, $d \succ 0$, to minimize $\|DMD^{-1}\|_F^2$
- express as

$$\|DMD^{-1}\|_F^2 = \sum_{i,j=1}^{n} \left(DMD^{-1}\right)_{ij}^2 = \sum_{i,j=1}^{n} M_{ij}^2 d_i^2 / d_j^2$$

- a posynomial in d (with exponents 0, 2, and -2)
- in convex form, with $y = \log d$,

$$\log \|DMD^{-1}\|_F^2 = \log \left(\sum_{i,j=1}^{n} \exp\left(2(y_i - y_j + \log |M_{ij}|)\right) \right)$$

Outline

Optimization problems

Some standard convex problems

Transforming problems

Disciplined convex programming

Geometric programming

Quasiconvex optimization

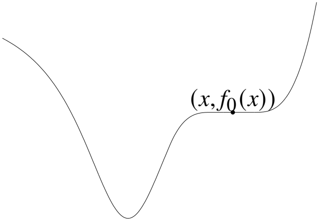
Multicriterion optimization

Quasiconvex optimization

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

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

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



Linear-fractional program

- **linear-fractional program**

$$\begin{array}{ll} \text{minimize} & (c^T x + d)/(e^T x + f) \\ \text{subject to} & Gx \preceq h, \quad Ax = b \end{array}$$

  with variable x and implicit constraint $e^T x + f > 0$

- equivalent to the LP (with variables y, z)

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

- recover $x^\star = y^\star / z^\star$

Von Neumann model of a growing economy

- $x, x^+ \in \mathbf{R}^n_{++}$: activity levels of n economic sectors, in current and next period
- $(Ax)_i$: amount of good i produced in current period
- $(Bx^+)_i$: amount of good i consumed in next period
- $Bx^+ \preceq Ax$: goods consumed next period no more than produced this period
- x_i^+/x_i: growth rate of sector i
- allocate activity to maximize growth rate of slowest growing sector

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

- a quasiconvex problem with variables x, x^+

Convex representation of sublevel sets

- 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:

- $f_0(x) = p(x)/q(x)$, with p convex and nonnegative, q concave and positive
- 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 $\phi_t(x) \leq 0$

Bisection method for quasiconvex optimization

- for fixed t, consider convex feasiblity problem

$$\phi_t(x) \leq 0, \qquad f_i(x) \leq 0, \quad i = 1, \ldots, m, \qquad Ax = b \tag{1}$$

  if feasible, we can conclude that $t \geq p^\star$; if infeasible, $t \leq p^\star$

- bisection method:

---

given $l \leq p^\star$, $u \geq p^\star$, 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

Outline

Optimization problems

Some standard convex problems

Transforming problems

Disciplined convex programming

Geometric programming

Quasiconvex optimization

Multicriterion optimization

Multicriterion optimization

- **multicriterion** or **multi-objective** problem:

$$\begin{array}{ll} \text{minimize} & f_0(x) = (F_1(x), \ldots, F_q(x)) \\ \text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m, \quad Ax = b \end{array}$$

- objective is the **vector** $f_0(x) \in \mathbf{R}^q$
- q different objectives $F_1, \ldots, F_q$; roughly speaking we want all F_i's to be small
- feasible $x^\star$ is **optimal** if y feasible $\implies f_0(x^\star) \preceq f_0(y)$
- this means that $x^\star$ simultaneously minimizes each F_i; the objectives are **noncompeting**
- not surprisingly, this doesn't happen very often

Pareto optimality

- feasible x **dominates** another feasible $\tilde{x}$ if $f_0(x) \preceq f_0(\tilde{x})$ and for at least one i, $F_i(x) < F_i(\tilde{x})$
- *i.e.*, x meets $\tilde{x}$ on all objectives, and beats it on at least one

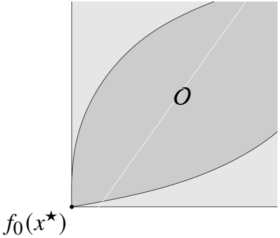
- feasible x^{po} is **Pareto optimal** if it is not dominated by any feasible point
- can be expressed as: y feasible, $f_0(y) \preceq f_0(x^{\text{po}}) \implies f_0(x^{\text{po}}) = f_0(y)$

- there are typically many Pareto optimal points
- for $q = 2$, set of Pareto optimal objective values is the **optimal trade-off curve**
- for $q = 3$, set of Pareto optimal objective values is the **optimal trade-off surface**

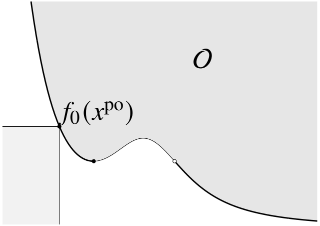
Optimal and Pareto optimal points

set of achievable objective values $\mathcal{O} = \{f_0(x) \mid x \text{ feasible}\}$

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



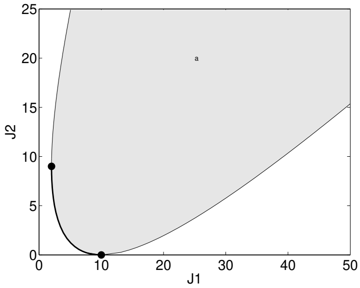
$x^\star$ is optimal



x^{po} is Pareto optimal

Regularized least-squares

- minimize $(\|Ax - b\|_2^2, \|x\|_2^2)$ (first objective is loss; second is regularization)
- example with $A \in \mathbf{R}^{100\times 10}$; heavy line shows Pareto optimal points

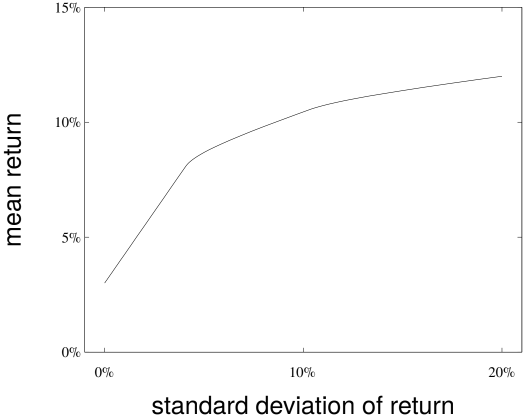


Risk return trade-off in portfolio optimization

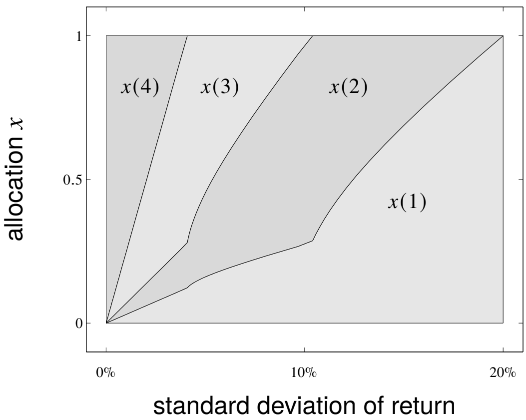
- variable $x \in \mathbf{R}^n$ is investment portfolio, with x_i fraction invested in asset i
- $\bar{p} \in \mathbf{R}^n$ is mean, Σ is covariance of asset returns
- portfolio return has mean $\bar{p}^T x$, variance $x^T \Sigma x$

- minimize $(-\bar{p}^T x, x^T \Sigma x)$, subject to $\mathbf{1}^T x = 1$, $x \succeq 0$
- Pareto optimal portfolios trace out optimal risk-return curve

Example



15%
10%
5%
0%
0%
10%
20%
mean return
standard deviation of return



1
0.5
0
0%
10%
20%
$x(4)$
$x(3)$
$x(2)$
$x(1)$
allocation x
standard deviation of return

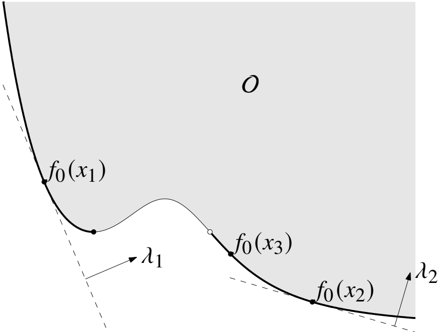
Scalarization

- **scalarization** combines the multiple objectives into one (scalar) objective
- a standard method for finding Pareto optimal points
- choose $\lambda \succ 0$ and solve scalar problem

$$
\begin{array}{ll}
\text{minimize} & \lambda^T f_0(x) = \lambda_1 F_1(x) + \cdots + \lambda_q F_q(x) \\
\text{subject to} & f_i(x) \leq 0, \quad i = 1, \ldots, m, \quad h_i(x) = 0, \quad i = 1, \ldots, p
\end{array}
$$

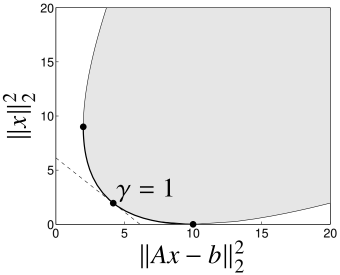
- λ_i are relative weights on the objectives
- if x is optimal for scalar problem, then it is Pareto-optimal for multicriterion problem
- for convex problems, can find (almost) all Pareto optimal points by varying $\lambda \succ 0$

Example



Example: Regularized least-squares

- regularized least-squares problem: minimize $(\|Ax - b\|_2^2, \|x\|_2^2)$
- take $\lambda = (1, \gamma)$ with $\gamma > 0$, and minimize $\|Ax - b\|_2^2 + \gamma\|x\|_2^2$



Example: Risk-return trade-off

- risk-return trade-off: minimize $(-\bar{p}^T x, x^T \Sigma x)$ subject to $\mathbf{1}^T x = 1$, $x \succeq 0$
- with $\lambda = (1, \gamma)$ we obtain scalarized problem

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

- objective is negative **risk-adjusted return**, $\bar{p}^T x - \gamma x^T \Sigma x$
- γ is called the **risk-aversion parameter**

5. Duality

Outline

Lagrangian and dual function

Lagrange dual problem

KKT conditions

Sensitivity analysis

Problem reformulations

Theorems of alternatives

Lagrangian

- **standard form problem** (not necessarily convex)

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

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

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

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

  - 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:** $g : \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}$,

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

- g is concave, can be $-\infty$ for some λ, ν
- **lower bound property:** if $\lambda \succeq 0$, then $g(\lambda, \nu) \leq p^\star$
- 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 $p^\star \geq g(\lambda, \nu)$

Least-norm solution of linear equations

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

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

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

- plug x into L to obtain

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

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

Standard form LP

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

- Lagrangian is

$$L(x, \lambda, \nu) = c^T x + \nu^T (Ax - b) - \lambda^T x = -b^T \nu + (c + A^T \nu - \lambda)^T x$$

- L is affine in x, so

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

- g is linear on affine domain $\{(\lambda, \nu) \mid A^T \nu - \lambda + c = 0\}$, hence concave
- lower bound property: $p^\star \geq -b^T \nu$ 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 is

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

  where $\|v\|_* = \sup_{\|u\| \leq 1} u^T v$ is dual norm of $\| \cdot \|$
- lower bound property: $p^\star \geq b^T \nu$ 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 encoded as $x_i = 1$ and $x_i = -1$
- W_{ij} is cost of assigning i, j to the same set; $-W_{ij}$ is cost of assigning to different sets
- dual function is

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

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

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

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

  where $f^*(y) = \sup_{x \in \mathbf{dom}\, f}(y^T x - f(x))$ is conjugate of f_0
- simplifies derivation of dual if conjugate of f_0 is known
- **example: entropy maximization**

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

Outline

Lagrangian and dual function

Lagrange dual problem

KKT conditions

Sensitivity analysis

Problem reformulations

Theorems of alternatives

The Lagrange 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 $p^\star$, obtained from Lagrange dual function
- a convex optimization problem, even if original **primal** problem is not
- dual optimal value denoted $d^\star$
- 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

(see slide 5.5)

- primal standard form LP:

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

- dual problem is

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

  with $g(\lambda, \nu) = -b^T \nu$ if $A^T \nu - \lambda + c = 0$, $-\infty$ otherwise

- make implicit constraint explicit, and eliminate λ to obtain (transformed) dual problem

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

Weak and strong duality

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

- always holds (for convex and nonconvex problems)
- can be used to find nontrivial lower bounds for difficult problems, *e.g.*, 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: $d^\star = p^\star$

- 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

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

if it is **strictly feasible**, *i.e.*, there is an $x \in \textbf{int}\, \mathcal{D}$ with $f_i(x) < 0$, $i = 1, \ldots, m$, $Ax = b$

- also guarantees that the dual optimum is attained (if $p^\star > -\infty$)
- can be sharpened: *e.g.*,
  - can replace $\textbf{int}\, \mathcal{D}$ with $\textbf{relint}\, \mathcal{D}$ (interior relative to affine hull)
  - affine inequalities do not need to hold with strict inequality
- there are 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

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

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 the sharpened Slater's condition: $p^\star = d^\star$ if the primal problem is feasible
- in fact, $p^\star = d^\star$ except when primal and dual are both 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

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

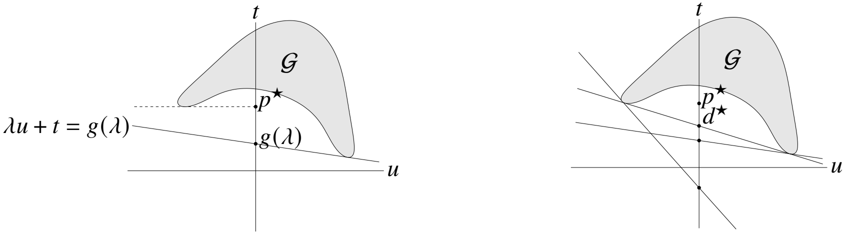
dual problem

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

- from the sharpened Slater's condition: $p^\star = d^\star$ if the primal problem is feasible
- in fact, $p^\star = d^\star$ always

Geometric interpretation

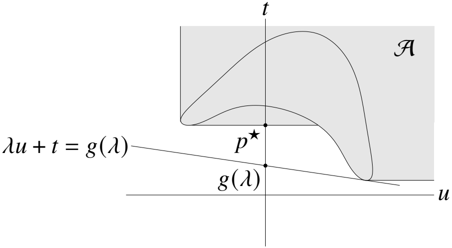
- for simplicity, consider problem with one constraint $f_1(x) \leq 0$
- $\mathcal{G} = \{(f_1(x), f_0(x)) \mid x \in \mathcal{D}\}$ is set of achievable (constraint, objective) values
- **interpretation of dual function:** $g(\lambda) = \inf_{(u,t)\in\mathcal{G}} (t + \lambda u)$



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

Epigraph variation

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

Outline

Lagrangian and dual function

Lagrange dual problem

KKT conditions

Sensitivity analysis

Problem reformulations

Theorems of alternatives

Complementary slackness

- assume strong duality holds, $x^\star$ is primal optimal, $(\lambda^\star, \nu^\star)$ is dual optimal

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

- hence, the two inequalities hold with equality
- $x^\star$ minimizes $L(x, \lambda^\star, \nu^\star)$
- $\lambda_i^\star f_i(x^\star) = 0$ for $i = 1, \dots, m$ (known as **complementary slackness**):

$$\lambda_i^\star > 0 \implies f_i(x^\star) = 0, \qquad f_i(x^\star) < 0 \implies \lambda_i^\star = 0$$

Karush-Kuhn-Tucker (KKT) conditions

the **KKT conditions** (for a problem with differentiable f_i, h_i) are

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

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

if strong duality holds and x, λ, ν are optimal, they 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, then

> *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

Outline

Lagrangian and dual function

Lagrange dual problem

KKT conditions

Sensitivity analysis

Problem reformulations

Theorems of alternatives

Perturbation and sensitivity analysis

(unperturbed) optimization problem and its dual

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p \end{array} \qquad \begin{array}{ll} \text{maximize} & g(\lambda, \nu) \\ \text{subject to} & \lambda \succeq 0 \end{array}$$

perturbed problem and its dual

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \le u_i, \quad i = 1, \ldots, m \\ & h_i(x) = v_i, \quad i = 1, \ldots, p \end{array} \qquad \begin{array}{ll} \text{maximize} & g(\lambda, \nu) - u^T\lambda - v^T\nu \\ \text{subject to} & \lambda \succeq 0 \end{array}$$

- x is primal variable; u, v are parameters
- $p^\star(u, v)$ is optimal value as a function of u, v
- $p^\star(0, 0)$ is optimal value of unperturbed problem

Global sensitivity via duality

- assume strong duality holds for unperturbed problem, with $\lambda^\star$, $\nu^\star$ dual optimal
- apply weak duality to perturbed problem:

$$p^\star(u,v) \geq g(\lambda^\star,\nu^\star) - u^T\lambda^\star - v^T\nu^\star = p^\star(0,0) - u^T\lambda^\star - v^T\nu^\star$$

- **implications**
  - if $\lambda_i^\star$ large: $p^\star$ increases greatly if we tighten constraint i ($u_i < 0$)
  - if $\lambda_i^\star$ small: $p^\star$ does not decrease much if we loosen constraint i ($u_i > 0$)
  - if $\nu_i^\star$ large and positive: $p^\star$ increases greatly if we take $v_i < 0$
  - if $\nu_i^\star$ large and negative: $p^\star$ increases greatly if we take $v_i > 0$
  - if $\nu_i^\star$ small and positive: $p^\star$ does not decrease much if we take $v_i > 0$
  - if $\nu_i^\star$ small and negative: $p^\star$ does not decrease much if we take $v_i < 0$

Local sensitivity via duality

if (in addition) $p^\star(u,v)$ is differentiable at $(0,0)$, then

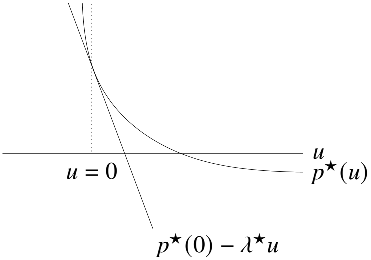
$$\lambda_i^\star = -\frac{\partial p^\star(0,0)}{\partial u_i}, \qquad \nu_i^\star = -\frac{\partial p^\star(0,0)}{\partial v_i}$$

proof (for $\lambda_i^\star$): from global sensitivity result,

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

hence, equality

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



Outline

Lagrangian and dual function

Lagrange dual problem

KKT conditions

Sensitivity analysis

Problem reformulations

Theorems of alternatives

Duality and problem reformulations

- equivalent formulations of a problem can lead to very different duals
- reformulating primal problem can be useful when 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

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

- introduce new variable y and equality constraints $y = Ax+b$

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

- dual of reformulated problem is

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

- a nontrivial, useful dual (assuming the conjugate f_0^* is easy to express)

Example: Norm approximation

- minimize $\|Ax - b\|$
- reformulate as minimize $\|y\|$ subject to $y = Ax - b$
- recall conjugate of general norm:

$$\|z\|^* = \begin{cases} 0 & \|z\|_* \leq 1 \\ \infty & \text{otherwise} \end{cases}$$

- dual of (reformulated) norm approximation problem:

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

Outline

Lagrangian and dual function

Lagrange dual problem

KKT conditions

Sensitivity analysis

Problem reformulations

Theorems of alternatives

Theorems of alternatives

- consider two systems of inequality and equality constraints
- called **weak alternatives** if no more than one system is feasible
- called **strong alternatives** if exactly one of them is feasible
- examples: for any $a \in \mathbf{R}$, with variable $x \in \mathbf{R}$,
  - $x > a$ and $x \leq a - 1$ are weak alternatives
  - $x > a$ and $x \leq a$ are strong alternatives

- a **theorem of alternatives** states that two inequality systems are (weak or strong) alternatives
- can be considered the extension of duality to feasibility problems

Feasibility problems

- consider system of (not necessarily convex) inequalities and equalities

$$f_i(x) \le 0, \quad i = 1, \ldots, m, \quad h_i(x) = 0, \quad i = 1, \ldots, p$$

- express as **feasibility problem**

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

- if system if feasible, $p^\star = 0$; if not, $p^\star = \infty$

Duality for feasibility problems

- dual function of feasibility problem is $g(\lambda, \nu) = \inf_x \left(\sum_{i=1}^m \lambda_i f_i(x) + \sum_{i=1}^p \nu_i h_i(x) \right)$
- for $\lambda \succeq 0$, we have $g(\lambda, \nu) \leq p^\star$
- it follows that feasibility of the inequality system

$$\lambda \succeq 0, \qquad g(\lambda, \nu) > 0$$

  implies the original system is infeasible
- so this is a weak alternative to original system
- it is strong if f_i convex, h_i affine, and a constraint qualification holds
- g is positive homogeneous so we can write alternative system as

$$\lambda \succeq 0, \qquad g(\lambda, \nu) \geq 1$$

Example: Nonnegative solution of linear equations

- consider system
$$Ax = b, \qquad x \succeq 0$$
- dual function is $g(\lambda, \nu) = \begin{cases} -b^T\nu & A^T\nu = \lambda \\ -\infty & \text{otherwise} \end{cases}$
- can express strong alternative of $Ax = b$, $x \succeq 0$ as
$$A^T\nu \succeq 0, \qquad b^T\nu \leq -1$$
(we can replace $b^T\nu \leq -1$ with $b^T\nu = -1$)

Farkas' lemma

- Farkas' lemma:

$$Ax \preceq 0, \quad c^T x < 0 \quad \text{and} \quad A^T y + c = 0, \quad y \succeq 0$$

  are strong alternatives

- proof: use (strong) duality for (feasible) LP

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

Investment arbitrage

- we invest x_j in each of n assets $1, \ldots, n$ with prices $p_1, \ldots, p_n$
- our initial cost is $p^T x$
- at the end of the investment period there are only m possible outcomes $i = 1, \ldots, m$
- V_{ij} is the **payoff** or final value of asset j in outcome i
- first investment is risk-free (cash): $p_1 = 1$ and $V_{i1} = 1$ for all i

- **arbitrage** means there is x with $p^T x < 0$, $Vx \geq 0$
- arbitrage means we receive money up front, and our investment cannot lose
- standard assumption in economics: the prices are such that **there is no arbitrage**

Absence of arbitrage

- by Farkas' lemma, there is no arbitrage $\Longleftrightarrow$ there exists $y \in \mathbf{R}^m_+$ with $V^T y = p$
- since first column of V is $\mathbf{1}$, we have $\mathbf{1}^T y = 1$
- y is interpreted as a **risk-neutral probability** on the outcomes $1, \ldots, m$
- $V^T y$ are the expected values of the payoffs under the risk-neutral probability
- interpretation of $V^T y = p$:

  asset prices equal their expected payoff under the risk-neutral probability

- **arbitrage theorem**: there is no arbitrage $\Leftrightarrow$ there exists a risk-neutral probability distribution under which each asset price is its expected payoff

Example

$$V = \begin{bmatrix} 1.0 & 0.5 & 0.0 \\ 1.0 & 0.8 & 0.0 \\ 1.0 & 1.0 & 1.0 \\ 1.0 & 1.3 & 4.0 \end{bmatrix}, \qquad p = \begin{bmatrix} 1.0 \\ 0.9 \\ 0.3 \end{bmatrix}, \qquad \tilde{p} = \begin{bmatrix} 1.0 \\ 0.8 \\ 0.7 \end{bmatrix}$$

- with prices p, there is an arbitrage

$$x = \begin{bmatrix} 6.2 \\ -7.7 \\ 1.5 \end{bmatrix}, \qquad p^T x = -0.2, \qquad \mathbf{1}^T x = 0, \qquad Vx = \begin{bmatrix} 2.35 \\ 0.04 \\ 0.00 \\ 2.19 \end{bmatrix}$$

- with prices $\tilde{p}$, there is no arbitrage, with risk-neutral probability

$$y = \begin{bmatrix} 0.36 \\ 0.27 \\ 0.26 \\ 0.11 \end{bmatrix} \qquad V^T y = \begin{bmatrix} 1.0 \\ 0.8 \\ 0.7 \end{bmatrix}$$

6. Approximation and fitting

Outline

Norm and penalty approximation

Regularized approximation

Robust approximation

Norm approximation

- minimize $\|Ax - b\|$, with $A \in \mathbf{R}^{m \times n}$, $m \geq n$, $\|\cdot\|$ is any norm

- **approximation**: $Ax^\star$ is the best approximation of b by a linear combination of columns of A
- **geometric**: $Ax^\star$ is point in $\mathcal{R}(A)$ closest to b (in norm $\|\cdot\|$)
- **estimation**: linear measurement model $y = Ax + v$
  - measurement y, v is measurement error, x is to be estimated
  - implausibility of v is $\|v\|$
  - given $y = b$, most plausible x is $x^\star$
- **optimal design**: x are design variables (input), Ax is result (output)
  - $x^\star$ is design that best approximates desired result b (in norm $\|\cdot\|$)

Examples

- Euclidean approximation ($\|\cdot\|_2$)
  - solution $x^\star = A^\dagger b$
- Chebyshev or minimax approximation ($\|\cdot\|_\infty$)
  - can be solved via 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 via LP

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

Penalty function approximation

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

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

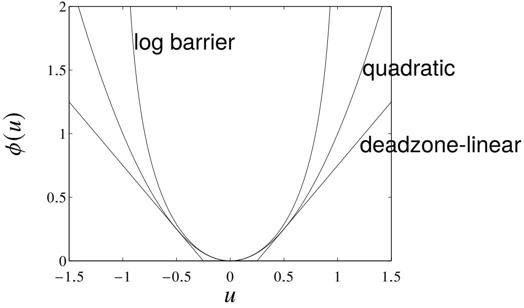
examples

- 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: histograms of residuals

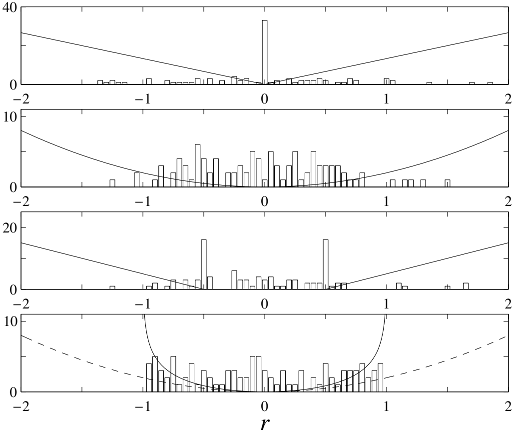
$A \in \mathbf{R}^{100\times 30}$; shape of penalty function affects distribution of residuals

absolute value $\phi(u) = |u|$

square $\phi(u) = u^2$

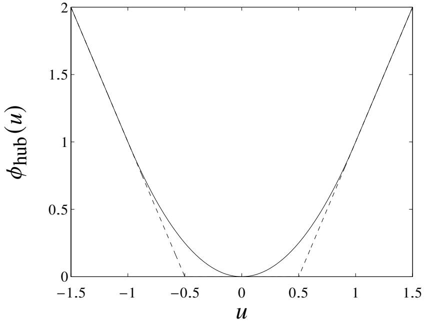
deadzone $\phi(u) = \max\{0, |u| - 0.5\}$

log-barrier $\phi(u) = -\log(1 - u^2)$



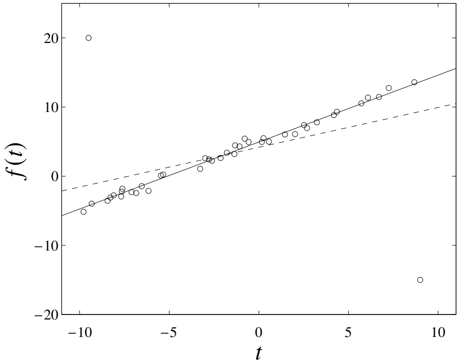
Huber penalty function

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



- linear growth for large u makes approximation less sensitive to outliers
- called a **robust penalty**

Example



- 42 points (circles) t_i, y_i, with two outliers
- affine function $f(t) = \alpha + \beta t$ fit using quadratic (dashed) and Huber (solid) penalty

Least-norm problems

- least-norm problem:

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

  with $A \in \mathbf{R}^{m \times n}$, $m \leq n$, $\|\cdot\|$ is any norm

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

Examples

- least Euclidean norm ($\|\cdot\|_2$)
  - solution $x = A^\dagger b$ (assuming $b \in \mathcal{R}(A)$)
- least sum of absolute values ($\|\cdot\|_1$)
  - can be solved via 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 yield sparse $x^\star$

Outline

Norm and penalty approximation

Regularized approximation

Robust approximation

Regularized approximation

- a bi-objective problem:

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

- $A \in \mathbf{R}^{m \times n}$, 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

- minimize $\|Ax - b\| + \gamma\|x\|$
- solution for $\gamma > 0$ traces out optimal trade-off curve
- other common method: minimize $\|Ax - b\|^2 + \delta\|x\|^2$ with $\delta > 0$
- with $\|\cdot\|_2$, called **Tikhonov regularization** or **ridge regression**

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

- can be solved as a least-squares problem

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

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

Optimal input design

- **linear dynamical system** (or **convolution system**) with impulse response h:

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

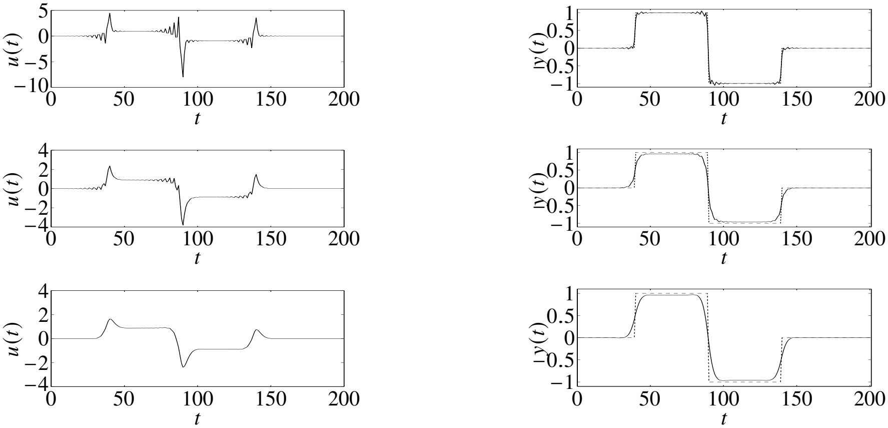
- **input design problem:** multicriterion problem with 3 objectives
  - tracking error with desired output y_{des}: $J_{\text{track}} = \sum_{t=0}^{N}(y(t) - y_{\text{des}}(t))^2$
  - input variation: $J_{\text{der}} = \sum_{t=0}^{N-1}(u(t+1) - u(t))^2$
  - input magnitude: $J_{\text{mag}} = \sum_{t=0}^{N}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

  track desired output using a small and slowly varying input signal

- **regularized least-squares formulation**: minimize $J_{\text{track}} + \delta J_{\text{der}} + \eta J_{\text{mag}}$
  - for fixed δ, η, a least-squares problem in $u(0), \ldots, u(N)$

Example

- minimize $J_{\text{track}} + \delta J_{\text{der}} + \eta J_{\text{mag}}$
- (top) $\delta = 0$, small η; (middle) $\delta = 0$, larger η; (bottom) large δ



Signal reconstruction

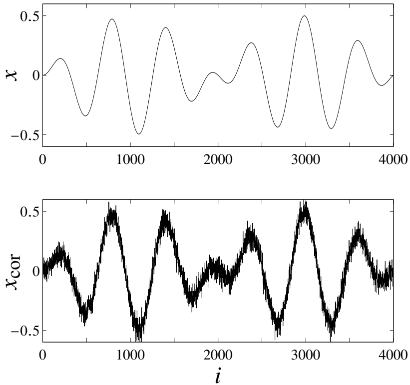
- bi-objective problem:

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

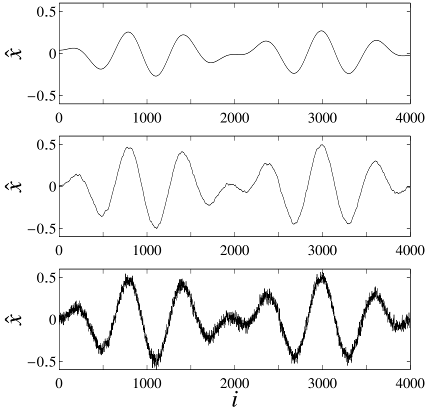
  - $x \in \mathbf{R}^n$ is unknown signal
  - $x_{\text{cor}} = x + v$ is (known) corrupted version of x, with additive noise v
  - variable $\hat{x}$ (reconstructed signal) is estimate of x
  - $\phi : \mathbf{R}^n \to \mathbf{R}$ is regularization function or smoothing objective

- **examples:**
  - quadratic smoothing, $\phi_{\text{quad}}(\hat{x}) = \sum_{i=1}^{n-1} (\hat{x}_{i+1} - \hat{x}_i)^2$
  - total variation smoothing, $\phi_{\text{tv}}(\hat{x}) = \sum_{i=1}^{n-1} |\hat{x}_{i+1} - \hat{x}_i|$

Quadratic smoothing example

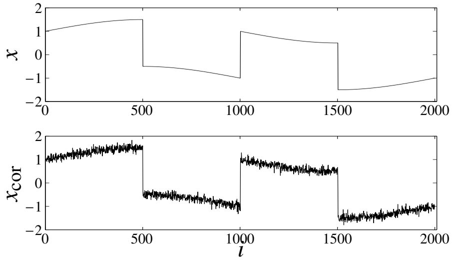


original signal x and noisy signal x_{cor}

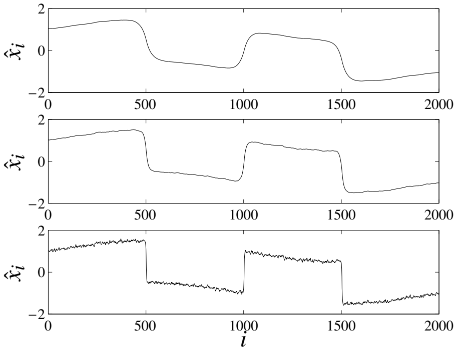


three solutions on trade-off curve
$\|\hat{x} - x_{\mathrm{cor}}\|_2$ versus $\phi_{\mathrm{quad}}(\hat{x})$

Reconstructing a signal with sharp transitions



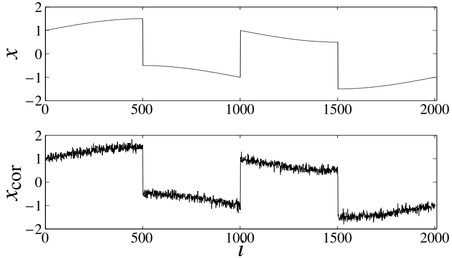
original signal x and noisy signal x_{cor}



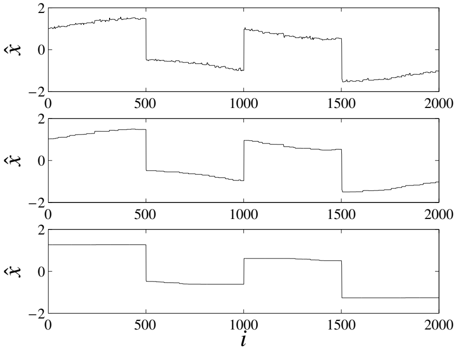
three solutions on trade-off curve
$\|\hat{x} - x_{\mathrm{cor}}\|_2$ versus $\phi_{\mathrm{quad}}(\hat{x})$

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

Total variation reconstruction



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

Outline

Norm and penalty approximation

Regularized approximation

Robust approximation

Robust approximation

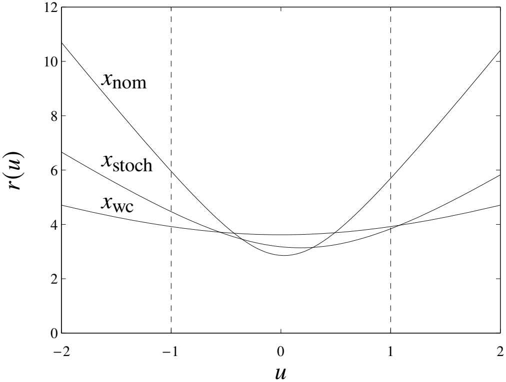
- minimize $\|Ax - b\|$ with uncertain A
- two approaches:
  - **stochastic**: assume A is random, minimize $\mathbf{E}\,\|Ax - b\|$
  - **worst-case:** set $\mathcal{A}$ of possible values of A, minimize $\sup_{A \in \mathcal{A}} \|Ax - b\|$
- tractable only in special cases (certain norms $\|\cdot\|$, distributions, sets $\mathcal{A}$)

Example

$A(u) = A_0 + uA_1$, $u \in [-1, 1]$

- x_{nom} minimizes $\|A_0 x - b\|_2^2$
- x_{stoch} minimizes $\mathbf{E}\,\|A(u)x - b\|_2^2$
  with u uniform on $[-1, 1]$
- x_{wc} minimizes $\sup_{-1 \le u \le 1} \|A(u)x - b\|_2^2$

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



Stochastic robust least-squares

- $A = \bar{A} + U$, U random, $\mathbf{E}\, U = 0$, $\mathbf{E}\, U^T U = P$
- stochastic least-squares problem: minimize $\mathbf{E}\, \|(\bar{A} + U)x - b\|_2^2$
- explicit expression for objective:

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

- hence, robust least-squares problem is equivalent to: minimize $\|\bar{A}x - b\|_2^2 + \|P^{1/2}x\|_2^2$
- for $P = \delta I$, get Tikhonov regularized problem: minimize $\|\bar{A}x - b\|_2^2 + \delta\|x\|_2^2$

Worst-case robust least-squares

- $\mathcal{A} = \{\bar{A} + u_1 A_1 + \cdots + u_p A_p \mid \|u\|_2 \leq 1\}$ (an ellipsoid in $\mathbf{R}^{m\times n}$)
- worst-case robust least-squares problem is

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

where $P(x) = \begin{bmatrix} A_1 x & A_2 x & \cdots & A_p x \end{bmatrix}$, $q(x) = \bar{A}x - b$
- from book appendix B, strong duality holds between the following problems

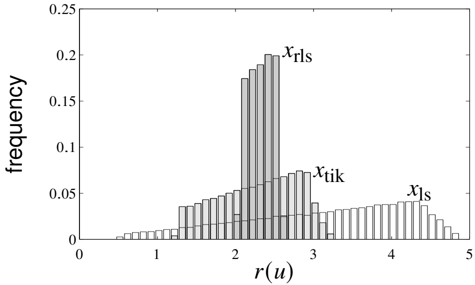
$$\begin{array}{ll} \text{maximize} & \|Pu+q\|_2^2 \\ \text{subject to} & \|u\|_2^2 \leq 1 \end{array} \qquad \begin{array}{ll} \text{minimize} & t+\lambda \\ \text{subject to} & \begin{bmatrix} I & P & q \\ P^T & \lambda I & 0 \\ q^T & 0 & t \end{bmatrix} \succeq 0 \end{array}$$

- hence, robust least-squares 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

- $r(u) = \|(A_0 + u_1 A_1 + u_2 A_2)x - b\|_2$, u uniform on unit disk
- three choices of x:
  - $x_{\rm ls}$ minimizes $\|A_0 x - b\|_2$
  - $x_{\rm tik}$ minimizes $\|A_0 x - b\|_2^2 + \delta\|x\|_2^2$ (Tikhonov solution)
  - $x_{\rm rls}$ minimizes $\sup_{A\in\mathcal{A}} \|Ax - b\|_2^2 + \|x\|_2^2$



7. Statistical estimation

Outline

Maximum likelihood estimation

Hypothesis testing

Experiment design

Maximum likelihood estimation

- **parametric distribution estimation:** choose from a family of densities $p_x(y)$, indexed by a parameter x (often denoted θ)
- we take $p_x(y) = 0$ for invalid values of x
- $p_x(y)$, as a function of x, is called **likelihood function**
- $l(x) = \log p_x(y)$, as a function of x, is called **log-likelihood function**

- **maximum likelihood estimation (MLE):** choose x to maximize $p_x(y)$ (or $l(x)$)
- a convex optimization problem if $\log p_x(y)$ is concave in x for fixed y
- not the same as $\log p_x(y)$ concave in y for fixed x, *i.e.*, $p_x(y)$ is a family of log-concave densities

Linear measurements with IID noise

linear measurement model

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

- $x \in \mathbf{R}^n$ 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

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

(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 least-squares 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 $[-a, a]$:

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

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

Logistic regression

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

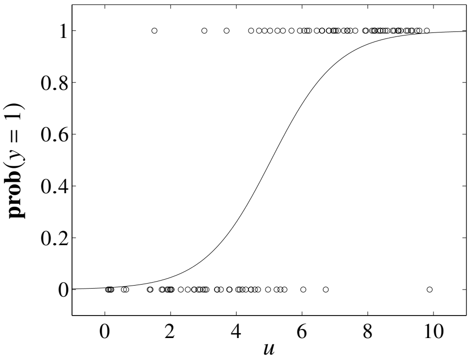
$$p = \mathbf{prob}(y = 1) = \frac{\exp(a^T u + b)}{1 + \exp(a^T u + b)}$$

- 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 $y_1 = \cdots = y_k = 1$, $y_{k+1} = \cdots = y_m = 0$):

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

concave in a, b

Example



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

Gaussian covariance estimation

- fit Gaussian distribution $\mathcal{N}(0, \Sigma)$ to observed data $y_1, \ldots, y_N$
- log-likelihood is

$$\begin{aligned} l(\Sigma) &= \frac{1}{2} \sum_{k=1}^{N} \left(-2\pi n - \log\det\Sigma - y^T \Sigma^{-1} y \right) \\ &= \frac{N}{2} \left(-2\pi n - \log\det\Sigma - \mathbf{tr}\, \Sigma^{-1} Y \right) \end{aligned}$$

  with $Y = (1/N) \sum_{k=1}^{N} y_k y_k^T$, the empirical covariance
- l is **not** concave in Σ (the $\log\det\Sigma$ term has the wrong sign)
- with no constraints or regularization, MLE is empirical covariance $\Sigma^{\text{ml}} = Y$

Change of variables

- change variables to $S = \Sigma^{-1}$
- recover original parameter via $\Sigma = S^{-1}$
- S is the **natural parameter** in an **exponential family** description of a Gaussian
- in terms of S, log-likelihood is

$$l(S) = \frac{N}{2}\left(-2\pi n + \log\det S - \mathbf{tr}\, SY\right)$$

  which is **concave**
- (a similar trick can be used to handle nonzero mean)

Fitting a sparse inverse covariance

- S is the **precision matrix** of the Gaussian
- $S_{ij} = 0$ means that y_i and y_j are independent, conditioned on y_k, $k \neq i, j$
- sparse S means
  - many pairs of components are conditionally independent, given the others
  - y is described by a sparse (Gaussian) Bayes network

- to fit data with S sparse, minimize convex function

$$-\log\det S + \mathbf{tr}\, SY + \lambda \sum_{i \neq j} |S_{ij}|$$

over $S \in \mathbf{S}^n$, with hyper-parameter $\lambda \geq 0$

Example

- example with $n = 4$, $N = 10$ samples generated from a sparse S^{true}

$$S^{\text{true}} = \begin{bmatrix} 1 & 0 & 0.5 & 0 \\ 0 & 1 & 0 & 0.1 \\ 0.5 & 0 & 1 & 0.3 \\ 0 & 0.1 & 0.3 & 1 \end{bmatrix}$$

- empirical and sparse estimate values of Σ^{-1} (with $\lambda = 0.2$)

$$Y^{-1} = \begin{bmatrix} 3 & 0.8 & 3.3 & 1.2 \\ 0.8 & 1.2 & 1.2 & 0.9 \\ 3.2 & 1.2 & 4.6 & 2.1 \\ 1.2 & 0.9 & 2.1 & 2.7 \end{bmatrix}, \qquad \hat{S} = \begin{bmatrix} 0.9 & 0 & 0.6 & 0 \\ 0 & 0.7 & 0 & 0.1 \\ 0.6 & 0 & 1.1 & 0.2 \\ 0 & 0.1 & 0.2 & 1.2 \end{bmatrix}.$$

- estimation errors: $\|S^{\text{true}} - Y^{-1}\|_F^2 = 49.8, \quad \|S^{\text{true}} - \hat{S}\|_F^2 = 0.2$

Outline

Maximum likelihood estimation

Hypothesis testing

Experiment design

(Binary) hypothesis testing

detection (hypothesis testing) problem
given observation of a random variable $X \in \{1, \ldots, n\}$, choose between:

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

randomized detector

- a nonnegative matrix $T \in \mathbf{R}^{2\times n}$, with $\mathbf{1}^T T = \mathbf{1}^T$
- 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

$$D = \begin{bmatrix} Tp & Tq \end{bmatrix} = \begin{bmatrix} 1 - P_{\mathrm{fp}} & P_{\mathrm{fn}} \\ P_{\mathrm{fp}} & 1 - P_{\mathrm{fn}} \end{bmatrix}$$

- 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)

- **multi-objective formulation of detector design**

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

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

Scalarization

- scalarize with weight $\lambda > 0$ to obtain

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

- 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

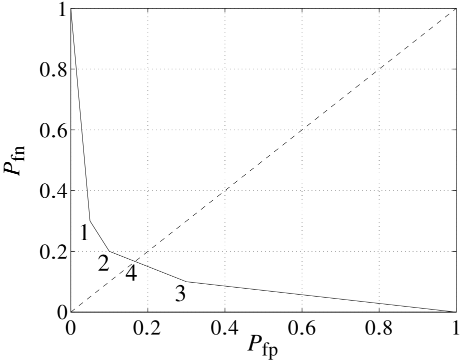
- minimize maximum of false positive and false negative probabilities

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

- an LP; solution is usually not deterministic

Example

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



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

Outline

Maximum likelihood estimation

Hypothesis testing

Experiment design

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

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

- confidence ellipsoids are given by $\{x \mid (x - \hat{x})^T E^{-1} (x - \hat{x}) \le \beta\}$
- **experiment design**: choose $a_i \in \{v_1, \dots, v_p\}$ (set of possible test vectors) to make E 'small'

Vector optimization formulation

- formulate as vector optimization problem

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

- variables are m_k, the number of vectors a_i equal to v_k
- difficult in general, due to integer constraint
- common scalarizations: minimize $\log\det E$, $\mathbf{tr}\, E$, $\lambda_{\max}(E)$, ...

Relaxed experiment design

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

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

- a convex relaxation, since we ignore constraint that $m\lambda_k \in \mathbf{Z}$
- optimal value is lower bound on optimal value of (integer) experiment design problem
- simple rounding of $\lambda_k m$ gives heuristic for experiment design problem

D-optimal design

- scalarize via log determinant

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

- interpretation: minimizes volume of confidence ellipsoids

Dual of D-optimal experiment design problem

dual problem

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

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

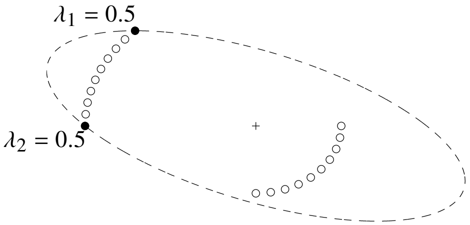
complementary slackness: for λ, W primal and dual optimal

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

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

first reformulate primal problem with new variable X:

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

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

- 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

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

change variable $W = Z/\nu$, and optimize over ν to get dual of slide 7.21

8. Geometric problems

Outline

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 $\mathcal{E}$ with $C \subseteq \mathcal{E}$
- parametrize $\mathcal{E}$ as $\mathcal{E} = \{v \mid \|Av + b\|_2 \leq 1\}$; can assume $A \in \mathbf{S}^n_{++}$
- $\mathbf{vol}\,\mathcal{E}$ is proportional to $\det A^{-1}$; to find Löwner-John ellipsoid, solve problem

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

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

- **finite set** $C = \{x_1, \ldots, x_m\}$:

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

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

Maximum volume inscribed ellipsoid

- maximum volume ellipsoid $\mathcal{E}$ with $\mathcal{E} \subseteq C$, $C \subseteq \mathbf{R}^n$ convex
- parametrize $\mathcal{E}$ as $\mathcal{E} = \{Bu + d \mid \|u\|_2 \leq 1\}$; can assume $B \in \mathbf{S}^n_{++}$
- $\mathbf{vol}\,\mathcal{E}$ is proportional to $\det B$; can find $\mathcal{E}$ 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)

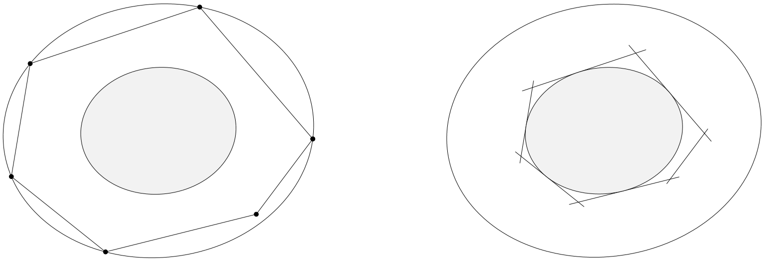
- **polyhedron** $\{x \mid a_i^T x \leq b_i,\ i = 1, \ldots, m\}$:

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

  (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 (around its center), lies inside C
- maximum volume inscribed ellipsoid, expanded by a factor n (around its center) covers C
- **example** (for polyhedra in $\mathbf{R}^2$)



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

Outline

Extremal volume ellipsoids

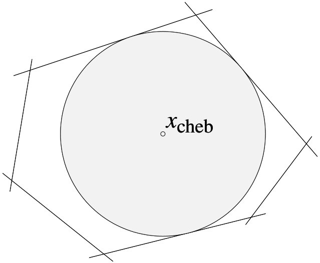
Centering

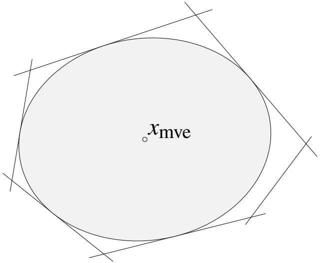
Classification

Placement and facility location

Centering

- many possible definitions of 'center' of a convex set C
- Chebyshev center: center of largest inscribed ball
  - for polyhedron, can be found via linear programming
- center of maximum volume inscribed ellipsoid
  - invariant under affine coordinate transformations





Analytic center of a set of inequalities

- the **analytic center** of set of convex inequalities and linear equations

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

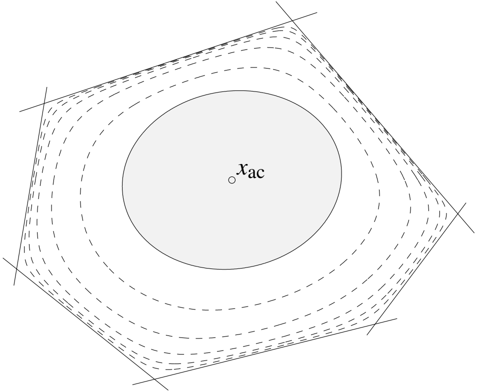
  is defined as solution of

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

- objective is called the **log-barrier** for the inequalities
- (we'll see later) analytic center more easily computed than MVE or Chebyshev center
- two sets of inequalities can describe the same set, but have different analytic centers

Analytic center of linear inequalities

- $a_i^T x \leq b_i$, $i = 1, \ldots, m$
- x_{ac} minimizes $\phi(x) = -\sum_{i=1}^m \log(b_i - a_i^T x)$
- dashed lines are level curves of ϕ



Inner and outer ellipsoids from analytic center

- we have

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

  where

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

- ellipsoid expansion/shrinkage factor is $\sqrt{m(m-1)}$
  (cf. n for Löwner-John or max volume inscribed ellpsoids)

Outline

Extremal volume ellipsoids

Centering

Classification

Placement and facility location

Linear discrimination

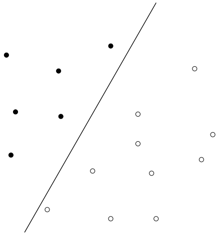
- separate two sets of points $\{x_1, \ldots, x_N\}$, $\{y_1, \ldots, y_M\}$ by a hyperplane
- *i.e.*, find $a \in \mathbf{R}^n$, $b \in \mathbf{R}$ with

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

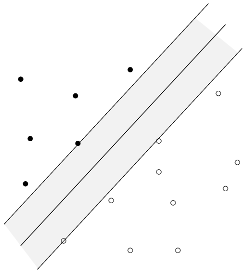
- homogeneous in a, b, hence equivalent to

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

a set of linear inequalities in a, b, *i.e.*, an LP feasibility problem



Robust linear discrimination



(Euclidean) distance between hyperplanes

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

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

to separate two sets of points by maximum margin,

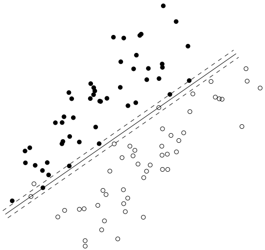
$$\begin{array}{ll} \text{minimize} & (1/2)\|a\|_2^2 \\ \text{subject to} & a^T x_i + b \geq 1, \quad i = 1, \ldots, N \\ & a^T y_i + b \leq -1, \quad i = 1, \ldots, M \end{array} \tag{2}$$

a QP in a, b

Approximate linear separation of non-separable sets

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

- an LP in a, b, u, v
- at optimum, $u_i = \max\{0, 1 - a^T x_i - b\}$, $v_i = \max\{0, 1 + a^T y_i + b\}$
- equivalent to minimizing the sum of violations of the original inequalities

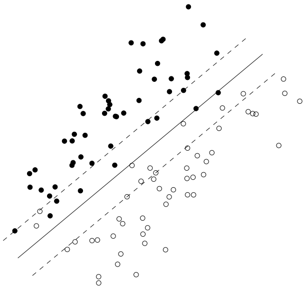


Support vector classifier

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

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

example on previous slide, with $\gamma = 0.1$:



Nonlinear discrimination

- separate two sets of points by a nonlinear function f: find $f : \mathbf{R}^n \to \mathbf{R}$ with

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

- choose a linearly parametrized family of functions $f(z) = \theta^T F(z)$
  - $\theta \in \mathbf{R}^k$ is parameter
  - $F = (F_1, \ldots, F_k) : \mathbf{R}^n \to \mathbf{R}^k$ are basis functions

- solve a set of linear inequalities in θ:

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

Examples

- **quadratic discrimination**: $f(z) = z^T P z + q^T z + r$, $\theta = (P, q, r)$
- solve LP feasibility problem with variables $P \in \mathbf{S}^n$, $q \in \mathbf{R}^n$, $r \in \mathbf{R}$

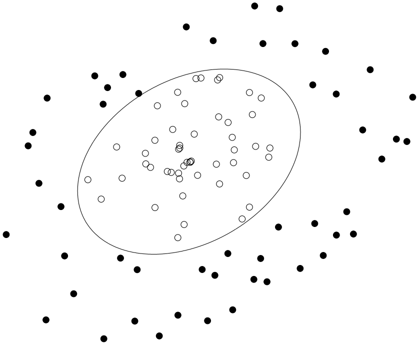
$$x_i^T P x_i + q^T x_i + r \geq 1, \qquad y_i^T P y_i + q^T y_i + r \leq -1$$

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

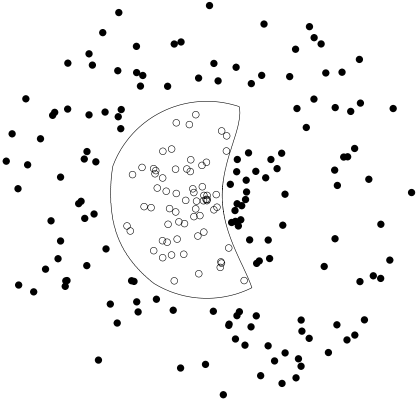
- **polynomial discrimination**: $F(z)$ are all monomials up to a given degree d
- *e.g.*, for $n = 2$, $d = 3$

$$F(z) = (1,\ z_1,\ z_2,\ z_1^2,\ z_1 z_2,\ z_2^2,\ z_1^3,\ z_1^2 z_2,\ z_1 z_2^2,\ z_2^3)$$

Example



separation by ellipsoid



separation by 4th degree polynomial

Outline

Extremal volume ellipsoids

Centering

Classification

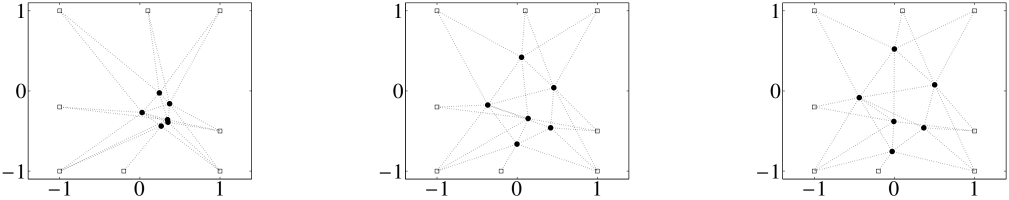
Placement and facility location

Placement and facility location

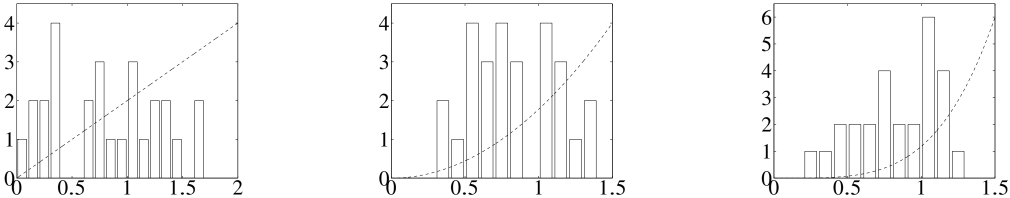
- N points with coordinates $x_i \in \mathbf{R}^2$ (or $\mathbf{R}^3$)
- some positions x_i are given; the other x_i's are variables
- for each pair of points, a cost function $f_{ij}(x_i, x_j)$
- **placement problem**: minimize $\sum_{i \neq j} f_{ij}(x_i, x_j)$
- **interpretations**
  - points are locations of plants or warehouses; f_{ij} is transportation cost between facilities i and j
  - points are locations of cells in an integrated circuit; f_{ij} represents wirelength

Example

- minimize $\sum_{(i,j)\in\mathcal{E}} h(\|x_i - x_j\|_2)$, with 6 free points, 27 edges
- optimal placements for $h(z) = z$, $h(z) = z^2$, $h(z) = z^4$



- histograms of edge lengths $\|x_i - x_j\|_2$, $(i,) \in \mathcal{E}$



B. Numerical linear algebra background

Outline

Flop counts and BLAS

Solving systems of linear equations

Block elimination

Flop count

- **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
  - simplify by keeping only the leading terms
- not an accurate predictor of computation time on modern computers, but useful as a rough estimate of complexity

Basic linear algebra subroutines (BLAS)

vector-vector operations (x, $y \in \mathbf{R}^n$) (BLAS level 1)

- inner product $x^T y$: $2n - 1$ flops ($\approx 2n$, $O(n)$)
- sum $x + y$, scalar multiplication αx: n flops

matrix-vector product $y = Ax$ with $A \in \mathbf{R}^{m\times n}$ (BLAS level 2)

- $m(2n - 1)$ flops ($\approx 2mn$)
- $2N$ 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 $A \in \mathbf{R}^{m\times n}$, $B \in \mathbf{R}^{n\times p}$ (BLAS level 3)

- $mp(2n - 1)$ flops ($\approx 2mnp$)
- 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

BLAS on modern computers

- there are good implementations of BLAS and variants (*e.g.*, for sparse matrices)
- CPU single thread speeds typically 1–10 Gflops/s (10^9 flops/sec)
- CPU multi threaded speeds typically 10–100 Gflops/s
- GPU speeds typically 100 Gflops/s–1 Tflops/s (10^{12} flops/sec)

Outline

Flop counts and BLAS

Solving systems of linear equations

Block elimination

Complexity of solving linear equations

- $A \in \mathbf{R}^{n\times n}$ is invertible, $b \in \mathbf{R}^n$
- solution of $Ax = b$ is $x = A^{-1}b$
- solving $Ax = b$, *i.e.*, computing $x = A^{-1}b$
  - almost never done by computing A^{-1}, then multiplying by b
  - for general methods, $O(n^3)$
  - (much) less if A is structured (banded, sparse, Toeplitz, ...)
  - *e.g.*, for A with half-bandwidth k ($A_{ij} = 0$ for $|i-j| > k$, $O(k^2n)$)
- it's super useful to recognize matrix structure that can be exploited in solving $Ax = b$

Linear equations that are easy to solve

- diagonal matrices: n flops; $x = A^{-1}b = (b_1/a_{11}, \ldots, b_n/a_{nn})$
- lower triangular: n^2 flops via **forward substitution**

$$\begin{array}{rcl} x_1 & := & b_1/a_{11} \\ x_2 & := & (b_2 - a_{21}x_1)/a_{22} \\ x_3 & := & (b_3 - a_{31}x_1 - a_{32}x_2)/a_{33} \\ & \vdots & \\ x_n & := & (b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{n,n-1}x_{n-1})/a_{nn} \end{array}$$

- upper triangular: n^2 flops via **backward substitution**

Linear equations that are easy to solve

- 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: for $\pi = (\pi_1, \pi_2, \ldots, \pi_n)$ a permutation of $(1, 2, \ldots, n)$

$$a_{ij} = \begin{cases} 1 & j = \pi_i \\ 0 & \text{otherwise} \end{cases}$$

  - interpretation: $Ax = (x_{\pi_1}, \ldots, x_{\pi_n})$
  - satisfies $A^{-1} = A^T$, hence cost of solving $Ax = b$ is 0 flops
  - example:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \qquad A^{-1} = A^T = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Factor-solve method for solving $Ax = b$

- factor A as a product of simple matrices (usually 2–5):

$$A = A_1 A_2 \cdots A_k$$

- *e.g.*, A_i diagonal, upper or lower triangular, orthogonal, permutation, ...
- compute $x = A^{-1}b = A_k^{-1} \cdots A_2^{-1} A_1^{-1} b$ by solving k 'easy' systems of equations

$$A_1 x_1 = b, \qquad A_2 x_2 = x_1, \qquad \ldots \qquad A_k x = x_{k-1}$$

- cost of factorization step usually dominates cost of solve step

Solving equations with multiple righthand sides

- we wish to solve
$$Ax_1 = b_1, \qquad Ax_2 = b_2, \qquad \ldots \qquad Ax_m = b_m$$
- cost: one factorization plus m solves
- called **factorization caching**
- when factorization cost dominates solve cost, we can solve a modest number of equations at the same cost as one (!!)

LU factorization

- every nonsingular matrix A can be factored as $A = PLU$ with P a permutation, L lower triangular, U upper triangular
- factorization 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).

- total cost: $(2/3)n^3 + 2n^2 \approx (2/3)n^3$ for large n

Sparse LU factorization

- for A sparse and invertible, factor as $A = P_1LUP_2$
- adding permutation matrix P_2 offers possibility of sparser L, U
- hence, less storage and 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
- often practical to solve very large sparse systems of equations

Cholesky factorization

- every positive definite A can be factored as $A = LL^T$
- L is lower triangular with positive diagonal entries
- Cholesjy factorization 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).

- total cost: $(1/3)n^3 + 2n^2 \approx (1/3)n^3$ for large n

Sparse Cholesky factorization

- for sparse positive define A, factor as $A = PLL^TP^T$
- adding permutation matrix P offers possibility of sparser L
- same as
  - permuting rows and columns of A to get $\tilde{A} = P^TAP$
  - then finding Cholesky factorization of $\tilde{A}$
- 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

Example

- sparse A with upper arrow sparsity pattern

$$A = \begin{bmatrix} * & * & * & * & * \\ * & * & & & \\ * & & * & & \\ * & & & * & \\ * & & & & * \end{bmatrix} \qquad L = \begin{bmatrix} * & & & & \\ * & * & & & \\ * & * & * & & \\ * & * & * & * & \\ * & * & * & * & * \end{bmatrix}$$

L is full, with $O(n^2)$ nonzeros; solve cost is $O(n^2)$

- reverse order of entries (*i.e.*, permute) to get lower arrow sparsity pattern

$$\tilde{A} = \begin{bmatrix} * & & & & * \\ & * & & & * \\ & & * & & * \\ & & & * & * \\ * & * & * & * & * \end{bmatrix} \qquad L = \begin{bmatrix} * & & & & \\ & * & & & \\ & & * & & \\ & & & * & \\ * & * & * & * & * \end{bmatrix}$$

L is sparse with $O(n)$ nonzeros; cost of solve is $O(n)$

LDL$^\mathsf{T}$ 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

- factorization cost: $(1/3)n^3$
- cost of solving linear equations with symmetric A by LDL$^\mathsf{T}$ 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 $\ll (1/3)n^3$

Outline

Flop counts and BLAS

Solving systems of linear equations

Block elimination

Equations with structured sub-blocks

- express $Ax = b$ in blocks as

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

  with $x_1 \in \mathbf{R}^{n_1}$, $x_2 \in \mathbf{R}^{n_2}$; blocks $A_{ij} \in \mathbf{R}^{n_i \times n_j}$
- assuming A_{11} is nonsingular, can eliminate x_1 as

$$x_1 = A_{11}^{-1}(b_1 - A_{12}x_2)$$

- to compute x_2, solve

$$(A_{22} - A_{21}A_{11}^{-1}A_{12})x_2 = b_2 - A_{21}A_{11}^{-1}b_1$$

- $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$ is the **Schur complement**

Bock elimination method

Solving linear equations by block elimination.

given a nonsingular set of linear equations 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

- for general A_{11}, $f = (2/3)n_1^3$, $s = 2n_1^2$

$$\#\text{flops} = (2/3)n_1^3 + 2n_1^2 n_2 + 2n_2^2 n_1 + (2/3)n_2^3 = (2/3)(n_1 + n_2)^3$$

  so, no gain over standard method

- block elimination is useful for structured A_{11} ($f \ll n_1^3$)
- for example, A_{11} diagonal ($f = 0$, $s = n_1$): #flops $\approx 2n_2^2 n_1 + (2/3)n_2^3$

Structured plus low rank matrices

- we wish to solve $(A+BC)x = b$, $A \in \mathbf{R}^{n\times n}$, $B \in \mathbf{R}^{n\times p}$, $C \in \mathbf{R}^{p\times n}$
- assume A has structure (*i.e.*, $Ax = b$ easy to solve)
- first **uneliminate** to write as block equations with new variable y

$$\begin{bmatrix} A & B \\ C & -I \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$$

- 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 $A + BC$ are nonsingular,

$$(A+BC)^{-1} = A^{-1} - A^{-1}B(I + CA^{-1}B)^{-1}CA^{-1}$$

Example: Solving diagonal plus low rank equations

- with A diagonal, $p \ll n$, $A + BC$ is called **diagonal plus low rank**
- for covariance matrices, called a **factor model**
- method 1: form $D = A + BC$, then solve $Dx = b$
  - storage n^2
  - solve cost $(2/3)n^3 + 2pn^2$ (**cubic** in n)
- method 2: solve $(I + CA^{-1}B)y = CA^{-1}b$, then compute $x = A^{-1}b - A^{-1}By$
  - storage $O(np)$
  - solve cost $2p^2n + (2/3)p^3$ (**linear** in n)

9. Unconstrained minimization

Outline

Terminology and assumptions

Gradient descent method

Steepest descent method

Newton's method

Self-concordant functions

Implementation

Unconstrained minimization

- unconstrained minimization problem

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

- we assume
  - f convex, twice continuously differentiable (hence $\mathbf{dom}\, f$ open)
  - optimal value $p^\star = \inf_x f(x)$ is attained at $x^\star$ (not necessarily unique)
- optimality condition is $\nabla f(x) = 0$
- minimizing f is the same as solving $\nabla f(x) = 0$
- a set of n equations with n unknowns

Quadratic functions

- convex quadratic: $f(x) = (1/2)x^T Px + q^T x + r$, $P \succeq 0$
- we can solve exactly via linear equations

$$\nabla f(x) = Px + q = 0$$

- much more on this special case later

Iterative methods

- for most non-quadratic functions, we use **iterative methods**
- these produce a sequence of points $x^{(k)} \in \mathbf{dom}\, f$, $k = 0, 1, \ldots$
- $x^{(0)}$ is the **initial point** or **starting point**
- $x^{(k)}$ is the kth **iterate**
- we hope that the method **converges**, *i.e.*,

$$f(x^{(k)}) \to p^\star, \qquad \nabla f(x^{(k)}) \to 0$$

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 $S = \{x \mid f(x) \le f(x^{(0)})\}$ 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 $f(x) \to \infty$ as $x \to \mathbf{bd}\, \mathbf{dom}\, f$
- examples of differentiable functions with closed sublevel sets:

$$f(x) = \log\left(\sum_{i=1}^{m} \exp(a_i^T x + b_i)\right), \qquad f(x) = -\sum_{i=1}^{m} \log(b_i - a_i^T x)$$

Strong convexity and implications

- f is **strongly convex** on S if there exists an $m > 0$ such that

$$\nabla^2 f(x) \succeq mI \text{ for all } x \in S$$

- same as $f(x) - (m/2)\|x\|_2^2$ is convex
- if f is strongly convex, 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
- we conclude $p^\star > -\infty$, 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, which usually you do not)

Outline

Terminology and assumptions

Gradient descent method

Steepest descent method

Newton's method

Self-concordant functions

Implementation

Descent methods

- **descent methods** generate iterates as

$$x^{(k+1)} = x^{(k)} + t^{(k)}\Delta x^{(k)}$$

  with $f(x^{(k+1)}) < f(x^{(k)})$ (hence the name)
- other notations: $x^+ = x + t\Delta x$, $x := x + t\Delta x$
- $\Delta x^{(k)}$ is the **step**, or **search direction**
- $t^{(k)} > 0$ is the **step size**, or **step length**
- from convexity, $f(x^+) < f(x)$ implies $\nabla f(x)^T\Delta x < 0$
- this means Δx is a **descent direction**

Generic descent method

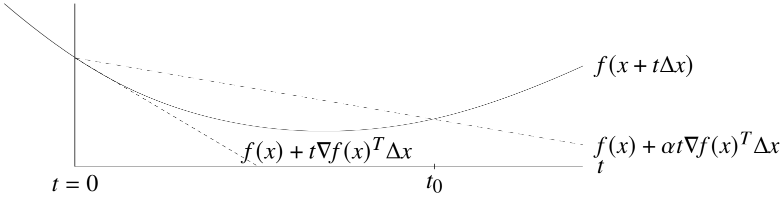
General descent method.

given a starting point $x \in \mathbf{dom}\, f$.
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: reduce t (*i.e.*, backtrack) until $t \le t_0$



Gradient descent method

- general descent method with $\Delta x = -\nabla f(x)$

---

given a starting point $x \in \mathbf{dom}\, f$.
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,

$$f(x^{(k)}) - p^\star \leq c^k (f(x^{(0)}) - p^\star)$$

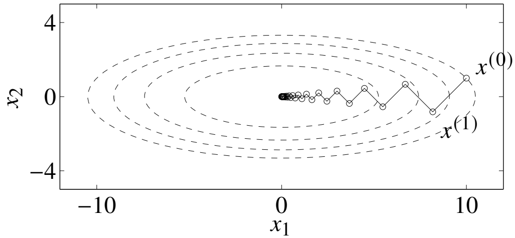
  $c \in (0,1)$ depends on m, $x^{(0)}$, line search type
- very simple, but can be very slow

Example: Quadratic function on $\mathbf{R}^2$

- take $f(x) = (1/2)(x_1^2 + \gamma x_2^2)$, with $\gamma > 0$
- with exact line search, starting at $x^{(0)} = (\gamma, 1)$:

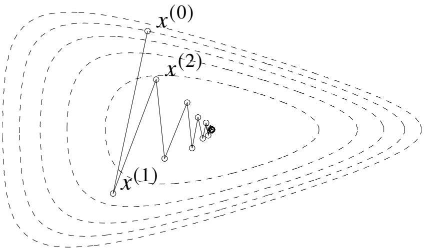
$$x_1^{(k)} = \gamma \left(\frac{\gamma - 1}{\gamma + 1}\right)^k, \qquad x_2^{(k)} = \left(-\frac{\gamma - 1}{\gamma + 1}\right)^k$$

  - very slow if $\gamma \gg 1$ or $\gamma \ll 1$
  - example for $\gamma = 10$ at right
  - called **zig-zagging**

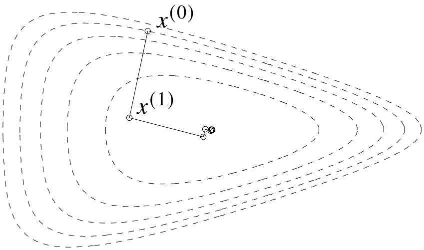


Example: Nonquadratic function on $\mathbf{R}^2$

- $f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$



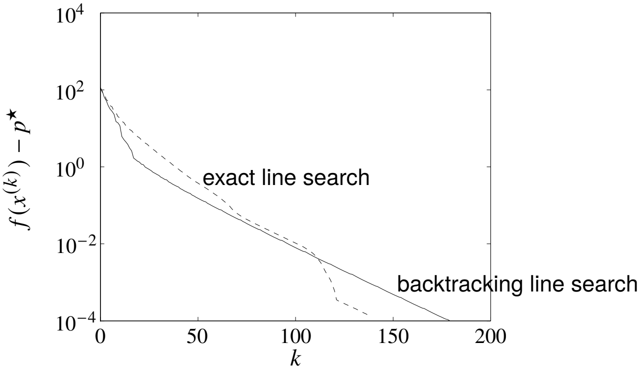
backtracking line search



exact line search

Example: 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



Outline

Terminology and assumptions

Gradient descent method

Steepest descent method

Newton's method

Self-concordant functions

Implementation

Steepest descent method

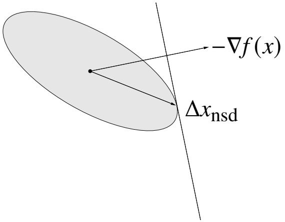
- **normalized steepest descent direction** (at x, for norm $\|\cdot\|$):

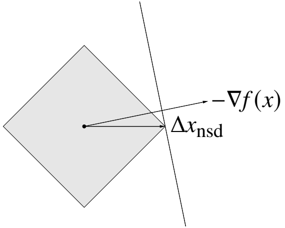
$$\Delta x_{\mathrm{nsd}} = \mathrm{argmin}\{\nabla f(x)^T v \mid \|v\| = 1\}$$

- 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**: $\Delta x_{\mathrm{sd}} = \|\nabla f(x)\|_* \Delta x_{\mathrm{nsd}}$
- satisfies $\nabla f(x)^T \Delta x_{\mathrm{sd}} = -\|\nabla f(x)\|_*^2$
- **steepest descent method**
  - general descent method with $\Delta x = \Delta x_{\mathrm{sd}}$
  - convergence properties similar to gradient descent

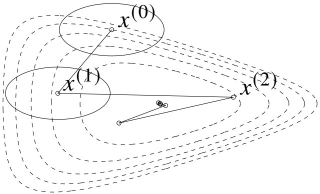
Examples

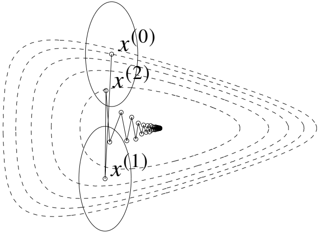
- Euclidean norm: $\Delta x_{\text{sd}} = -\nabla f(x)$
- quadratic norm $\|x\|_P = (x^T P x)^{1/2}$ ($P \in \mathbf{S}^n_{++}$): $\Delta x_{\text{sd}} = -P^{-1}\nabla f(x)$
- ℓ_1-norm: $\Delta x_{\text{sd}} = -(\partial f(x)/\partial x_i) e_i$, where $|\partial f(x)/\partial x_i| = \|\nabla f(x)\|_\infty$
- unit balls, normalized steepest descent directions for quadratic norm and ℓ_1-norm:





Choice of norm for steepest descent





- steepest descent with backtracking line search for two quadratic norms
- ellipses show $\{x \mid \|x - x^{(k)}\|_P = 1\}$
- 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

Outline

Terminology and assumptions

Gradient descent method

Steepest descent method

Newton's method

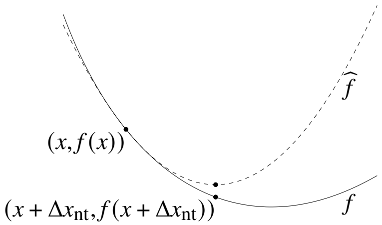
Self-concordant functions

Implementation

Newton step

- **Newton step** is $\Delta x_{\text{nt}} = -\nabla^2 f(x)^{-1} \nabla f(x)$
- **interpretation**: $x + \Delta x_{\text{nt}}$ minimizes second order approximation

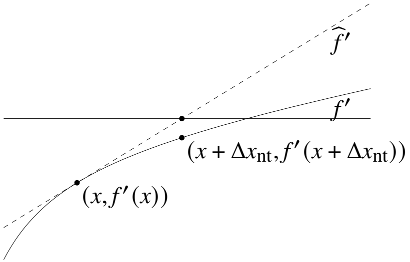
$$\widehat{f}(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v$$



Another intrepretation

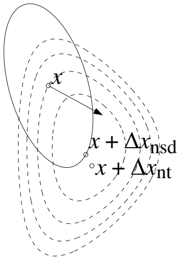
- $x + \Delta x_{\mathrm{nt}}$ solves linearized optimality condition

$$\nabla f(x+v) \approx \nabla \widehat{f}(x+v) = \nabla f(x) + \nabla^2 f(x) v = 0$$



And one more interpretation

- Δx_{nt} is steepest descent direction at x in local Hessian norm $\|u\|_{\nabla^2 f(x)} = \left(u^T \nabla^2 f(x) u\right)^{1/2}$



- 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

- **Newton decrement** is $\lambda(x) = \left(\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x)\right)^{1/2}$
- a measure of the proximity of x to $x^\star$
- gives an estimate of $f(x) - p^\star$, using quadratic approximation $\widehat{f}$:

$$f(x) - \inf_y \widehat{f}(y) = \frac{1}{2}\lambda(x)^2$$

- equal to the norm of the Newton step in the quadratic Hessian norm

$$\lambda(x) = \left(\Delta x_{\mathrm{nt}}^T \nabla^2 f(x) \Delta x_{\mathrm{nt}}\right)^{1/2}$$

- directional derivative in the Newton direction: $\nabla f(x)^T \Delta x_{\mathrm{nt}} = -\lambda(x)^2$
- affine invariant (unlike $\|\nabla f(x)\|_2$)

Newton's method

---

given a starting point $x \in \mathbf{dom}\, f$, tolerance $\epsilon > 0$.
repeat

1. **Compute the Newton step and decrement.**
   $\Delta x_{\mathrm{nt}} := -\nabla^2 f(x)^{-1} \nabla f(x); \quad \lambda^2 := \nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x)$.
2. **Stopping criterion. quit** if $\lambda^2/2 \leq \epsilon$.
3. **Line search.** Choose step size t by backtracking line search.
4. **Update.** $x := x + t\Delta x_{\mathrm{nt}}$.

- **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

$$\frac{L}{2m^2}\|\nabla f(x^{(k+1)})\|_2 \leq \left(\frac{L}{2m^2}\|\nabla f(x^{(k)})\|_2\right)^2$$

Classical convergence analysis

damped Newton phase ($\|\nabla f(x)\|_2 \geq \eta$)

- most iterations require backtracking steps
- function value decreases by at least γ
- if $p^\star > -\infty$, this phase ends after at most $(f(x^{(0)}) - p^\star)/\gamma$ 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

$$\frac{L}{2m^2}\|\nabla f(x^l)\|_2 \leq \left(\frac{L}{2m^2}\|\nabla f(x^k)\|_2\right)^{2^{l-k}} \leq \left(\frac{1}{2}\right)^{2^{l-k}}, \qquad l \geq k$$

Classical convergence analysis

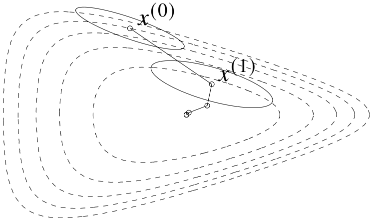
conclusion: number of iterations until $f(x) - p^\star \leq \epsilon$ is bounded above by

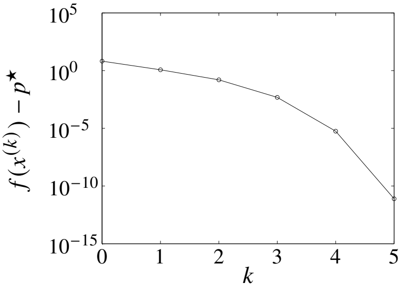
$$\frac{f(x^{(0)}) - p^\star}{\gamma} + \log_2 \log_2(\epsilon_0/\epsilon)$$

- 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)

Example: $\mathbf{R}^2$

(same problem as slide 9.13)

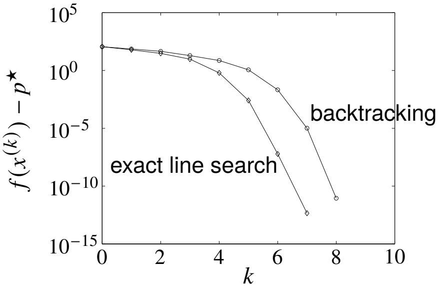


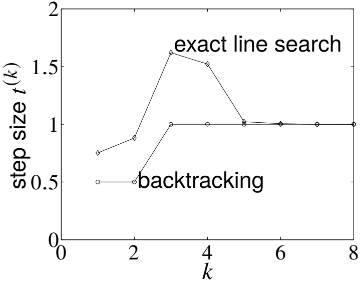


- backtracking parameters $\alpha = 0.1$, $\beta = 0.7$
- converges in only 5 steps
- quadratic local convergence

Example in $\mathbf{R}^{100}$

(same problem as slide 9.14)



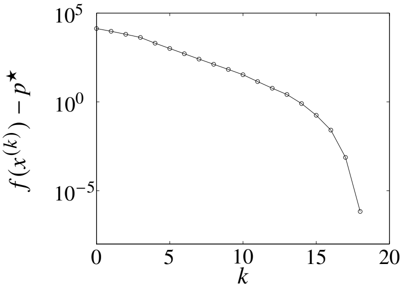


- backtracking parameters $\alpha = 0.01$, $\beta = 0.5$
- backtracking line search almost as fast as exact l.s. (and much simpler)
- clearly shows two phases in algorithm

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

Outline

Self-concordance

shortcomings of classical convergence analysis

- depends on unknown constants $(m, L, \ldots)$
- 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 **self-concordant** functions
- developed to analyze polynomial-time interior-point methods for convex optimization

Convergence analysis for self-concordant functions

definition

- convex $f : \mathbf{R} \to \mathbf{R}$ is self-concordant if $|f'''(x)| \leq 2f''(x)^{3/2}$ for all $x \in \mathbf{dom}\, f$
- $f : \mathbf{R}^n \to \mathbf{R}$ is self-concordant if $g(t) = f(x + tv)$ is self-concordant for all $x \in \mathbf{dom}\, f$, $v \in \mathbf{R}^n$

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 $f : \mathbf{R} \to \mathbf{R}$ is s.c., then $\tilde{f}(y) = f(ay + b)$ is s.c.:

$$\tilde{f}'''(y) = a^3 f'''(ay + b), \qquad \tilde{f}''(y) = a^2 f''(ay + b)$$

Self-concordant calculus

properties

- preserved under positive scaling $\alpha \geq 1$, and sum
- preserved under composition with affine function
- if g is convex with $\mathbf{dom}\, g = \mathbf{R}_{++}$ 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 $\{x \mid a_i^T x < b_i,\ i = 1, \ldots, m\}$
- $f(X) = -\log\det X$ 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)

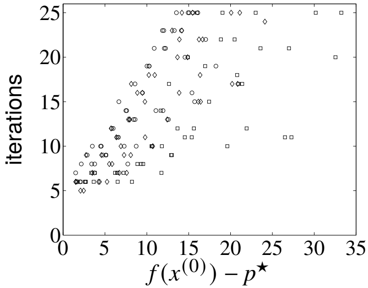
complexity bound: number of Newton iterations bounded by

$$\frac{f(x^{(0)}) - p^\star}{\gamma} + \log_2 \log_2(1/\epsilon)$$

for $\alpha = 0.1$, $\beta = 0.8$, $\epsilon = 10^{-10}$, bound evaluates to $375(f(x^{(0)}) - p^\star) + 6$

Numerical example

- 150 randomly generated instances of $f(x) = -\sum_{i=1}^{m} \log(b_i - a_i^T x)$, $x \in \mathbf{R}^n$
- ○: $m = 100$, $n = 50$; □: $m = 1000$, $n = 500$; ◇: $m = 1000$, $n = 50$



- number of iterations much smaller than $375(f(x^{(0)}) - p^\star) + 6$
- bound of the form $c(f(x^{(0)}) - p^\star) + 6$ with smaller c (empirically) valid

Outline

Terminology and assumptions

Gradient descent method

Steepest descent method

Newton's method

Self-concordant functions

Implementation

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

$$H = LL^T, \qquad \Delta x_{\mathrm{nt}} = -L^{-T}L^{-1}g, \qquad \lambda(x) = \|L^{-1}g\|_2$$

- cost $(1/3)n^3$ flops for unstructured system
- cost $\ll (1/3)n^3$ if H is sparse, banded, or has other structure

Example

- $f(x) = \sum_{i=1}^n \psi_i(x_i) + \psi_0(Ax+b)$, with $A \in \mathbf{R}^{p\times n}$ dense, $p \ll n$
- Hessian has low rank plus diagonal structure $H = D + A^T H_0 A$
- 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 (block elimination): factor $H_0 = L_0 L_0^T$; write Newton system as

$$D\Delta x + A^T L_0 w = -g, \qquad L_0^T A \Delta x - w = 0$$

eliminate Δx from first equation; compute w and Δx from

$$(I + L_0^T A D^{-1} A^T L_0) w = -L_0^T A D^{-1} g, \qquad D\Delta x = -g - A^T L_0 w$$

cost: $2p^2 n$ (dominated by computation of $L_0^T A D^{-1} A^T L_0$)

10. Equality constrained minimization

Outline

Equality constrained minimization

Newton's method with equality constraints

Infeasible start Newton method

Implementation

Equality constrained minimization

- equality constrained smooth minimization problem:

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

- we assume
  - f convex, twice continuously differentiable
  - $A \in \mathbf{R}^{p \times n}$ with $\mathbf{rank}\, A = p$
  - $p^\star$ is finite and attained

- **optimality conditions:** $x^\star$ is optimal if and only if there exists a $\nu^\star$ such that

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

Equality constrained quadratic minimization

- $f(x) = (1/2)x^T Px + q^T x + r$, $P \in \mathbf{S}^n_+$
- $\nabla f(x) = Px + q$
- optimality conditions are a **system of linear equations**

$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^\star \\ \nu^\star \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}$$

- coefficient matrix is called KKT matrix
- KKT matrix is nonsingular if and only if

$$Ax = 0, \quad x \neq 0 \qquad \Longrightarrow \qquad x^T Px > 0$$

- equivalent condition for nonsingularity: $P + A^T A \succ 0$

Eliminating equality constraints

- represent feasible set $\{x \mid Ax = b\}$ as $\{Fz + \hat{x} \mid z \in \mathbf{R}^{n-p}\}$
  - $\hat{x}$ is (any) **particular solution** of $Ax = b$
  - range of $F \in \mathbf{R}^{n \times (n-p)}$ is nullspace of A ($\mathbf{rank}\, F = n - p$ and $AF = 0$)
- **reduced or eliminated problem**: minimize $f(Fz + \hat{x})$
- an unconstrained problem with variable $z \in \mathbf{R}^{n-p}$
- from solution $z^\star$, obtain $x^\star$ and $\nu^\star$ as

$$x^\star = Fz^\star + \hat{x}, \qquad \nu^\star = -(AA^T)^{-1} A \nabla f(x^\star)$$

Example: Optimal resource allocation

- allocate resource amount $x_i \in \mathbf{R}$ to agent i
- agent i cost if $f_i(x_i)$
- resource budget is b, so $x_1 + \cdots + x_n = b$
- resource allocation problem is

$$\begin{array}{ll} \mbox{minimize} & f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n) \\ \mbox{subject to} & x_1 + x_2 + \cdots + x_n = b \end{array}$$

- eliminate $x_n = b - x_1 - \cdots - x_{n-1}$, *i.e.*, choose

$$\hat{x} = be_n, \qquad F = \left[\begin{array}{c} I \\ -\mathbf{1}^T \end{array}\right] \in \mathbf{R}^{n\times(n-1)}$$

- reduced problem: minimize $f_1(x_1) + \cdots + f_{n-1}(x_{n-1}) + f_n(b - x_1 - \cdots - x_{n-1})$

Outline

Equality constrained minimization

Newton's method with equality constraints

Infeasible start Newton method

Implementation

Newton step

- Newton step Δx_{nt} of f at feasible x is given by solution v of

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = \begin{bmatrix} -\nabla f(x) \\ 0 \end{bmatrix}$$

- Δx_{nt} solves second order approximation (with variable v)

$$\begin{array}{ll} \text{minimize} & \widehat{f}(x+v) = f(x) + \nabla f(x)^T v + (1/2) v^T \nabla^2 f(x) v \\ \text{subject to} & A(x+v) = b \end{array}$$

- Δx_{nt} equations follow from linearizing optimality conditions

$$\nabla f(x+v) + A^T w \approx \nabla f(x) + \nabla^2 f(x) v + A^T w = 0, \qquad A(x+v) = b$$

Newton decrement

- Newton decrement for equality constrained minimization is

$$\lambda(x) = \left(\Delta x_{\rm nt}^T \nabla^2 f(x) \Delta x_{\rm nt}\right)^{1/2} = \left(-\nabla f(x)^T \Delta x_{\rm nt}\right)^{1/2}$$

- gives an estimate of $f(x) - p^\star$ using quadratic approximation $\widehat{f}$:

$$f(x) - \inf_{Ay=b} \widehat{f}(y) = \lambda(x)^2/2$$

- directional derivative in Newton direction:

$$\left.\frac{d}{dt} f(x + t\Delta x_{\rm nt})\right|_{t=0} = -\lambda(x)^2$$

- in general, $\lambda(x) \neq \left(\nabla f(x)^T \nabla^2 f(x)^{-1} \nabla f(x)\right)^{1/2}$

Newton's method with equality constraints

---

given starting point $x \in \mathbf{dom}\, f$ with $Ax = b$, tolerance $\epsilon > 0$.

repeat

1. Compute the Newton step and decrement Δx_{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.* $x := x + t\Delta x_{\mathrm{nt}}$.

- a feasible descent method: $x^{(k)}$ feasible and $f(x^{(k+1)}) < f(x^{(k)})$
- affine invariant

Newton's method and elimination

- reduced problem: minimize $\tilde{f}(z) = f(Fz + \hat{x})$
  - variables $z \in \mathbf{R}^{n-p}$
  - $\hat{x}$ satisfies $A\hat{x} = b$; $\mathbf{rank}\, F = n - p$ and $AF = 0$
- (unconstrained) Newton's method for $\tilde{f}$, started at $z^{(0)}$, generates iterates $z^{(k)}$
- iterates of Newton's method with equality constraints, started at $x^{(0)} = Fz^{(0)} + \hat{x}$, are

$$x^{(k)} = Fz^{(k)} + \hat{x}$$

- hence, don't need separate convergence analysis

Outline

Equality constrained minimization

Newton's method with equality constraints

Infeasible start Newton method

Implementation

Newton step at infeasible points

- with $y = (x, \nu)$, write optimality condition as $r(y) = 0$, where
$$r(y) = (\nabla f(x) + A^T \nu, Ax - b)$$
is **primal-dual residual**
- consider $x \in \mathbf{dom} f$, $Ax \neq b$, *i.e.*, x is infeasible
- linearizing $r(y) = 0$ gives $r(y + \Delta y) \approx r(y) + Dr(y)\Delta y = 0$:
$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x_{\mathrm{nt}} \\ \Delta \nu_{\mathrm{nt}} \end{bmatrix} = - \begin{bmatrix} \nabla f(x) + A^T \nu \\ Ax - b \end{bmatrix}$$
- $(\Delta x_{\mathrm{nt}}, \Delta \nu_{\mathrm{nt}})$ is called **infeasible** or **primal-dual** Newton step at x

Infeasible start Newton method

---

given starting point $x \in \mathbf{dom} f$, 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 $\|r(x + t\Delta x_{\text{nt}}, \nu + t\Delta \nu_{\text{nt}})\|_2 > (1 - \alpha t)\|r(x, \nu)\|_2, \quad t := \beta t.$
3. *Update.* $x := x + t\Delta x_{\text{nt}}$, $\nu := \nu + t\Delta \nu_{\text{nt}}$.

until $Ax = b$ and $\|r(x, \nu)\|_2 \le \epsilon$.

- not a descent method: $f(x^{(k+1)}) > f(x^{(k)})$ is possible
- directional derivative of $\|r(y)\|_2$ in direction $\Delta y = (\Delta x_{\text{nt}}, \Delta \nu_{\text{nt}})$ is

$$\frac{d}{dt} \|r(y + t\Delta y)\|_2 \bigg|_{t=0} = -\|r(y)\|_2$$

Outline

Equality constrained minimization

Newton's method with equality constraints

Infeasible start Newton method

Implementation

Solving KKT systems

- feasible and infeasible Newton methods require solving KKT system

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = - \begin{bmatrix} g \\ h \end{bmatrix}$$

- in general, can use LDL$^\mathrm{T}$ factorization
- or elimination (if H nonsingular and easily inverted):
  - solve $AH^{-1}A^Tw = h - AH^{-1}g$ for w
  - $v = -H^{-1}(g + A^Tw)$

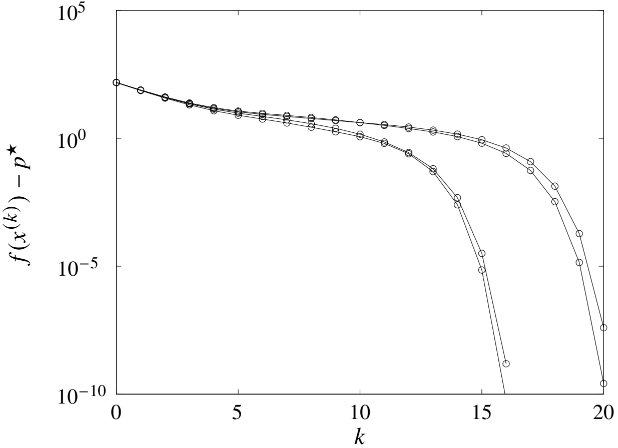
Example: Equality constrained analytic centering

- **primal problem:** minimize $-\sum_{i=1}^{n} \log x_i$ subject to $Ax = b$
- **dual problem:** maximize $-b^T \nu + \sum_{i=1}^{n} \log(A^T \nu)_i + n$
  - recover $x^\star$ as $x_i^\star = 1/(A^T \nu)_i$
- three methods to solve:
  - Newton method with equality constraints
  - Newton method applied to dual problem
  - infeasible start Newton method

  these have **different requirements for initialization**
- we'll look at an example with $A \in \mathbf{R}^{100 \times 500}$, different starting points

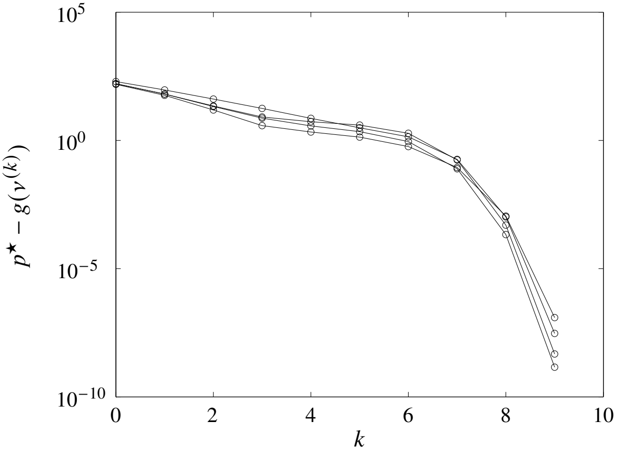
Newton's method with equality constraints

- requires $x^{(0)} \succ 0$, $Ax^{(0)} = b$



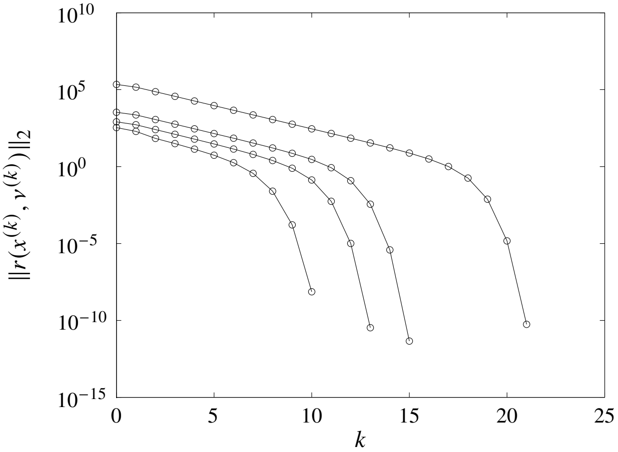
Newton method applied to dual problem

- requires $A^T\nu^{(0)} \succ 0$



Infeasible start Newton method

- requires $x^{(0)} \succ 0$



Complexity per iteration of three methods is identical

- for feasible Newton method, 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 $A\,\mathbf{diag}(x)^2 A^T w = b$
- for Newton system applied to dual, solve $A\,\mathbf{diag}(A^T\nu)^{-2}A^T\Delta\nu = -b + A\,\mathbf{diag}(A^T\nu)^{-1}\mathbf{1}$
- for infeasible start Newton method, use block elimination to solve KKT system

$$\begin{bmatrix} \mathbf{diag}(x)^{-2} & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta\nu \end{bmatrix} = \begin{bmatrix} \mathbf{diag}(x)^{-1}\mathbf{1} - A^T\nu \\ b - Ax \end{bmatrix}$$

  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

Example: Network flow optimization

- directed graph with n arcs, $p + 1$ nodes
- x_i: flow through arc i; ϕ_i: strictly convex flow cost function for arc i
- **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 incidence matrix** $A \in \mathbf{R}^{p\times n}$ is $\tilde{A}$ with last row removed
- $\mathbf{rank}\, A = p$ if graph is connected
- flow conservation is $Ax = b$, $b \in \mathbf{R}^p$ is (reduced) source vector
- **network flow optimization problem**: minimize $\sum_{i=1}^n \phi_i(x_i)$ subject to $Ax = b$

KKT system

- KKT system is

$$\begin{bmatrix} H & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = - \begin{bmatrix} g \\ h \end{bmatrix}$$

- $H = \mathbf{diag}(\phi_1''(x_1), \ldots, \phi_n''(x_n))$, positive diagonal
- solve via elimination:

$$AH^{-1}A^T w = h - AH^{-1}g, \qquad v = -H^{-1}(g + A^T w)$$

- sparsity pattern of $AH^{-1}A^T$ is given by graph connectivity

$$\begin{aligned} (AH^{-1}A^T)_{ij} \neq 0 \quad &\Longleftrightarrow \quad (AA^T)_{ij} \neq 0 \\ &\Longleftrightarrow \quad \text{nodes } i \text{ and } j \text{ are connected by an arc} \end{aligned}$$

Analytic center of linear matrix inequality

- minimize $-\log\det X$ subject to $\mathbf{tr}(A_iX) = b_i,\ \ i = 1, \ldots, p$
- optimality conditions

$$X^\star \succ 0, \qquad -(X^\star)^{-1} + \sum_{j=1}^{p} \nu_j^\star A_i = 0, \qquad \mathbf{tr}(A_iX^\star) = b_i, \quad i = 1, \ldots, p$$

- Newton step ΔX at feasible X is defined by

$$X^{-1}(\Delta X)X^{-1} + \sum_{j=1}^{p} w_j A_i = X^{-1}, \qquad \mathbf{tr}(A_i\Delta X) = 0, \quad i = 1, \ldots, p$$

- follows from linear approximation $(X + \Delta X)^{-1} \approx X^{-1} - X^{-1}(\Delta X)X^{-1}$
- $n(n+1)/2 + p$ variables ΔX, w

Solution by block elimination

- eliminate ΔX from first equation to get $\Delta X = X - \sum_{j=1}^{p} w_j X A_j X$
- substitute ΔX in second equation to get

$$\sum_{j=1}^{p} \mathbf{tr}(A_i X A_j X) w_j = b_i, \quad i = 1, \ldots, p$$

- a dense positive definite set of linear equations with variable $w \in \mathbf{R}^p$
- form and solve this set of equations to get w, then get ΔX from equation above

Flop count

- find Cholesky factor L of X $\quad (1/3)n^3$
- form p products $L^T A_j L$ $\quad (3/2)pn^3$
- form $p(p+1)/2$ inner products $\mathbf{tr}((L^T A_i L)(L^T A_j L))$ to get coefficent matrix $\quad (1/2)p^2n^2$
- solve $p \times p$ system of equations via Cholesky factorization $\quad (1/3)p^3$
- flop count dominated by $pn^3 + p^2n^2$
- cf. naïve method, $(n^2 + p)^3$

11. Interior-point methods

Outline

Inequality constrained minimization

Logarithmic barrier and central path

Barrier method

Phase I methods

Complexity analysis

Generalized inequalities

Inequality constrained minimization

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

we assume

- f_i convex, twice continuously differentiable
- $A \in \mathbf{R}^{p \times n}$ with $\mathbf{rank}\, A = p$
- $p^\star$ is finite and attained
- problem is strictly feasible: there exists $\tilde{x}$ with

$$\tilde{x} \in \mathbf{dom}\, f_0, \qquad f_i(\tilde{x}) < 0, \quad i = 1, \ldots, m, \qquad A\tilde{x} = b$$

hence, strong duality holds and dual optimum is attained

Examples

- LP, QP, QCQP, GP
- entropy maximization with linear inequality constraints

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n x_i \log x_i \\ \text{subject to} & Fx \preceq g, \quad Ax = b \end{array}$$

  with $\mathbf{dom}\, f_0 = \mathbf{R}^n_{++}$

- 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)

Outline

Inequality constrained minimization

Logarithmic barrier and central path

Barrier method

Phase I methods

Complexity analysis

Generalized inequalities

Logarithmic barrier

- reformulation via **indicator function**:

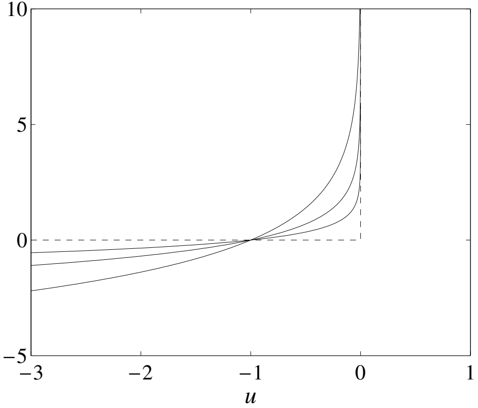
$$\begin{array}{ll} \text{minimize} & f_0(x) + \sum_{i=1}^m I_-(f_i(x)) \\ \text{subject to} & Ax = b \end{array}$$

  where $I_-(u) = 0$ if $u \leq 0$, $I_-(u) = \infty$ otherwise
- **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 $t \to \infty$

- $-(1/t)\log u$ for three values of t, and $I_-(u)$



Logarithmic barrier function

- log barrier function for constraints $f_1(x) \leq 0, \ldots, f_m(x) \leq 0$

$$\phi(x) = -\sum_{i=1}^{m} \log(-f_i(x)), \qquad \mathbf{dom}\, \phi = \{x \mid f_1(x) < 0, \ldots, f_m(x) < 0\}$$

- convex (from composition rules)
- twice continuously differentiable, with derivatives

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

Central path

- for $t > 0$, define $x^\star(t)$ as the solution of

$$\begin{array}{ll} \text{minimize} & tf_0(x) + \phi(x) \\ \text{subject to} & Ax = b \end{array}$$

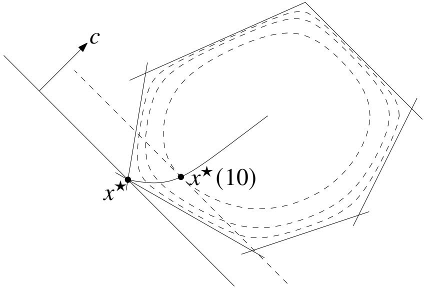
(for now, assume $x^\star(t)$ exists and is unique for each $t > 0$)

- central path is $\{x^\star(t) \mid t > 0\}$

example: central path for an LP

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

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



Dual points on central path

- $x = x^\star(t)$ if there exists a w such that

$$t\nabla f_0(x) + \sum_{i=1}^m \frac{1}{-f_i(x)} \nabla f_i(x) + A^T w = 0, \qquad Ax = b$$

- therefore, $x^\star(t)$ minimizes the Lagrangian

$$L(x, \lambda^\star(t), \nu^\star(t)) = f_0(x) + \sum_{i=1}^m \lambda_i^\star(t) f_i(x) + \nu^\star(t)^T (Ax - b)$$

  where we define $\lambda_i^\star(t) = 1/(-tf_i(x^\star(t))$ and $\nu^\star(t) = w/t$
- this confirms the intuitive idea that $f_0(x^\star(t)) \to p^\star$ if $t \to \infty$:

$$p^\star \geq g(\lambda^\star(t), \nu^\star(t)) = L(x^\star(t), \lambda^\star(t), \nu^\star(t)) = f_0(x^\star(t)) - m/t$$

Interpretation via KKT conditions

$x = x^\star(t)$, $\lambda = \lambda^\star(t)$, $\nu = \nu^\star(t)$ satisfy

1. primal constraints: $f_i(x) \le 0$, $i = 1, \ldots, m$, $Ax = b$
2. dual constraints: $\lambda \succeq 0$
3. approximate complementary slackness: $-\lambda_i f_i(x) = 1/t$, $i = 1, \ldots, m$
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)

$$\text{minimize} \quad tf_0(x) - \textstyle\sum_{i=1}^m \log(-f_i(x))$$

- **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 $F_i(x) = (1/f_i(x))\nabla f_i(x)$
- forces balance at $x^\star(t)$:

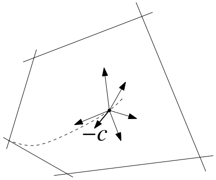
$$F_0(x^\star(t)) + \sum_{i=1}^m F_i(x^\star(t)) = 0$$

Example: LP

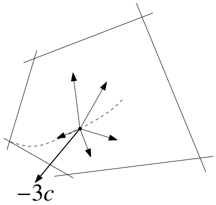
- minimize $c^T x$ subject to $a_i^T x \le b_i$, $i = 1, \ldots, m$, with $x \in \mathbf{R}^n$
- objective force field is constant: $F_0(x) = -tc$
- constraint force field decays as inverse distance to constraint hyperplane:

$$F_i(x) = \frac{-a_i}{b_i - a_i^T x}, \qquad \|F_i(x)\|_2 = \frac{1}{\mathbf{dist}(x, \mathcal{H}_i)}$$

where $\mathcal{H}_i = \{x \mid a_i^T x = b_i\}$



$t = 1$



$t = 3$

Outline

Inequality constrained minimization

Logarithmic barrier and central path

Barrier method

Phase I methods

Complexity analysis

Generalized inequalities

Barrier method

---

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

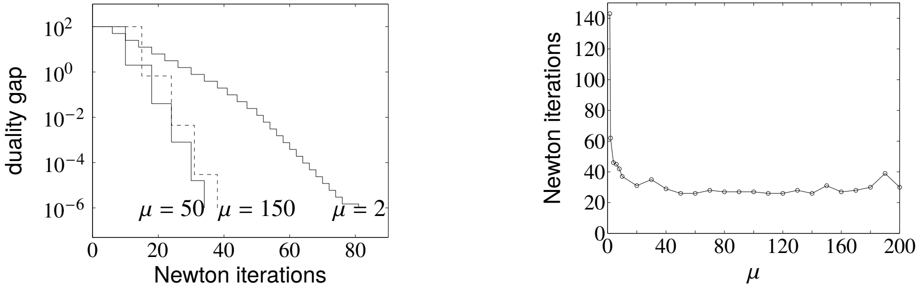
repeat

1. *Centering step.* Compute $x^\star(t)$ by minimizing $tf_0 + \phi$, subject to $Ax = b$.
2. *Update.* $x := x^\star(t)$.
3. *Stopping criterion.* **quit** if $m/t < \epsilon$.
4. *Increase* t. $t := \mu t$.

- terminates with $f_0(x) - p^\star \leq \epsilon$ (stopping criterion follows from $f_0(x^\star(t)) - p^\star \leq m/t$)
- 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: $\mu = 10$ or 20
- several heuristics for choice of $t^{(0)}$

Example: Inequality form LP

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

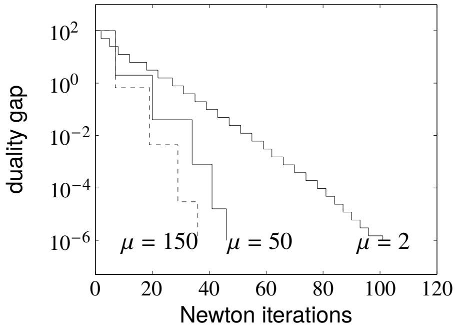


- starts with x on central path ($t^{(0)} = 1$, duality gap 100)
- terminates when $t = 10^8$ (gap 10^{-6})
- total number of Newton iterations not very sensitive for $\mu \geq 10$

Example: Geometric program in convex form

($m = 100$ inequalities and $n = 50$ variables)

$$\begin{array}{ll} \text{minimize} & \log\left(\sum_{k=1}^{5} \exp(a_{0k}^T x + b_{0k})\right) \\ \text{subject to} & \log\left(\sum_{k=1}^{5} \exp(a_{ik}^T x + b_{ik})\right) \leq 0, \quad i = 1, \ldots, m \end{array}$$

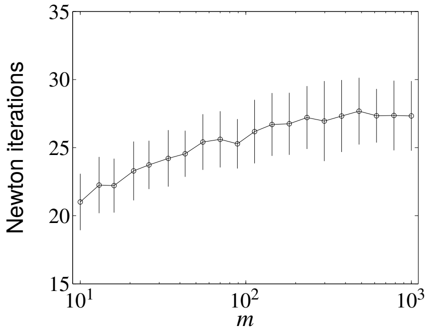


Family of standard LPs

($A \in \mathbf{R}^{m\times 2m}$)

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

$m = 10, \ldots, 1000$; for each m, solve 100 randomly generated instances



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

Outline

Inequality constrained minimization

Logarithmic barrier and central path

Barrier method

Phase I methods

Complexity analysis

Generalized inequalities

Phase I methods

- barrier method needs strictly feasible starting point, *i.e.*, x with

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

- (like the infeasible start Newton method, more sophisticated interior-point methods do not require a feasible starting point)
- **phase I** method forms an optimization problem that
  - is itself strictly feasible
  - finds a strictly feasible point for original problem, if one exists
  - certifies original problem as infeasible otherwise
- **phase II** uses barrier method starting from strictly feasible point found in phase I

Basic phase I method

- introduce slack variable s in **phase I problem**

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

  with optimal value $\bar{p}^\star$
  - if $\bar{p}^\star < 0$, original inequalities are strictly feasible
  - if $\bar{p}^\star > 0$, original inequalities are infeasible
  - $\bar{p}^\star = 0$ is an ambiguous case

- start phase I problem with
  - any $\tilde{x}$ in problem domain with $A\tilde{x} = b$
  - $s = 1 + \max_i f_i(\tilde{x})$

Sum of infeasibilities phase I method

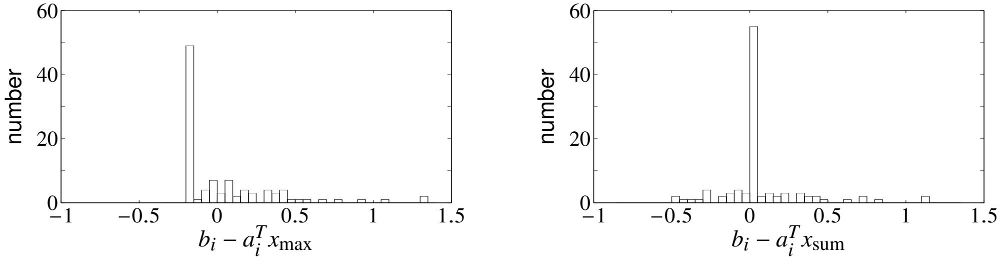
- minimize **sum** of slacks, not max:

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

- will find a strictly feasible point if one exists
- for infeasible problems, produces a solution that satisfies many (but not all) inequalities
- can weight slacks to set **priorities** (in satifying constraints)

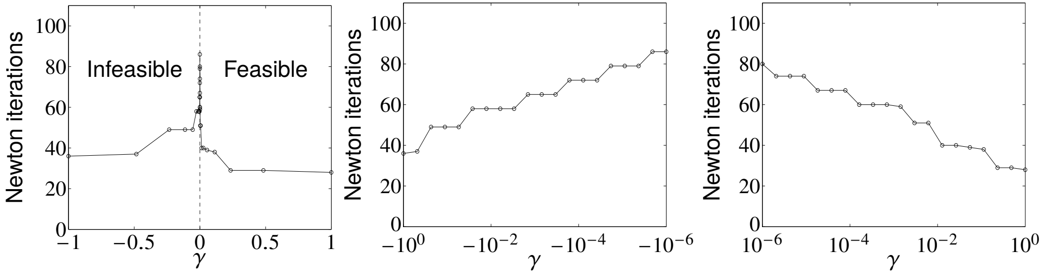
Example

- 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 inequalities



Example: Family of linear inequalities

- $Ax \preceq b + \gamma \Delta b$; strictly feasible for $\gamma > 0$, infeasible for $\gamma < 0$
- use basic phase I, terminate when $s < 0$ or dual objective is positive
- number of iterations roughly proportional to $\log(1/|\gamma|)$



Outline

Inequality constrained minimization

Logarithmic barrier and central path

Barrier method

Phase I methods

Complexity analysis

Generalized inequalities

Number of outer iterations

- in each iteration duality gap is reduced by exactly the factor μ
- **number of outer (centering) iterations** is exactly

$$\left\lceil \frac{\log(m/(\epsilon t^{(0)}))}{\log \mu} \right\rceil$$

plus the initial centering step (to compute $x^\star(t^{(0)})$)

- we will bound **number of Newton steps per centering iteration** using self-concordance analysis

Complexity analysis via self-concordance

same assumptions as on slide 11.2, 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

- we compute $x^+ = x^\star(\mu t)$, by minimizing $\mu t f_0(x) + \phi(x)$ starting from $x = x^\star(t)$
- from self-concordance theory,

$$\#\text{Newton iterations} \leq \frac{\mu t f_0(x) + \phi(x) - \mu t f_0(x^+) - \phi(x^+)}{\gamma} + c$$

- γ, c are constants (that depend only on Newton algorithm parameters)
- we will bound numerator $\mu t f_0(x) + \phi(x) - \mu t f_0(x^+) - \phi(x^+)$
- with $\lambda_i = \lambda_i^\star(t) = -1/(t f_i(x))$, we have $-f_i(x) = 1/(t\lambda_i)$, so

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

so

$$\phi(x) - \phi(x^+) = \sum_{i=1}^m \left(\log(t\lambda_i) + \log(-f_i(x^+))\right) = \sum_{i=1}^m \log(-\mu t \lambda_i f_i(x^+)) - m \log \mu$$

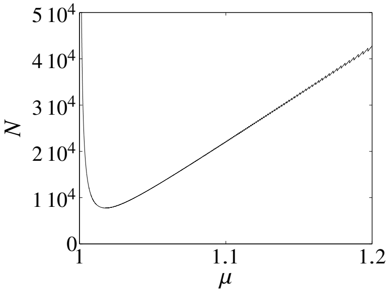
using $\log u \leq u - 1$ we have $\phi(x) - \phi(x^+) \leq -\mu t \sum_{i=1}^m \lambda_i f_i(x^+) - m - m \log \mu$, so

$$
\begin{aligned}
&\mu t f_0(x) + \phi(x) - \mu t f_0(x^+) - \phi(x^+) \\
&\quad \leq \quad \mu t f_0(x) - \mu t f_0(x^+) - \mu t \sum_{i=1}^m \lambda_i f_i(x^+) - m - m \log \mu \\
&\quad = \quad \mu t f_0(x) - \mu t \left(f_0(x^+) + \sum_{i=1}^m \lambda_i f_i(x^+) + \nu^T (Ax^+ - b) \right) - m - m \log \mu \\
&\quad = \quad \mu t f_0(x) - \mu t L(x^+, \lambda, \nu) - m - m \log \mu \\
&\quad \leq \quad \mu t f_0(x) - \mu t g(\lambda, \nu) - m - m \log \mu \\
&\quad = \quad m(\mu - 1 - \log \mu)
\end{aligned}
$$

using $L(x^+, \lambda, nu) \geq g(\lambda, \nu)$ in second last line and $f_0(x) - g(\lambda, \nu) = m/t$ in last line

Total number of Newton iterations

$$\#\text{Newton iterations} \le N = \left\lceil \frac{\log(m/(t^{(0)}\epsilon))}{\log \mu} \right\rceil \left(\frac{m(\mu - 1 - \log \mu)}{\gamma} + c \right)$$



N versus μ for typical values of γ, c;

$m = 100$, initial duality gap $\frac{m}{t^{(0)}\epsilon} = 10^5$

- confirms trade-off in choice of μ
- in practice, #iterations is in the tens; not very sensitive for $\mu \ge 10$

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 and larger

Outline

Inequality constrained minimization

Logarithmic barrier and central path

Barrier method

Phase I methods

Complexity analysis

Generalized inequalities

Generalized inequalities

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

- f_0 convex, $f_i : \mathbf{R}^n \to \mathbf{R}^{k_i}$, $i = 1, \ldots, m$, convex with respect to proper cones $K_i \in \mathbf{R}^{k_i}$
- we assume
  - f_i twice continuously differentiable
  - $A \in \mathbf{R}^{p \times n}$ with $\mathbf{rank}\, A = p$
  - $p^\star$ is finite and attained
  - problem is strictly feasible; hence strong duality holds and dual optimum is attained
- examples of greatest interest: SOCP, SDP

Generalized logarithm for proper cone

$\psi : \mathbf{R}^q \to \mathbf{R}$ is **generalized logarithm** for proper cone $K \subseteq \mathbf{R}^q$ if:

- $\mathbf{dom}\,\psi = \mathbf{int}\,K$ 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 $K = \mathbf{S}^n_+$: $\psi(Y) = \log\det Y$, with degree $\theta = n$
- second-order cone $K = \{y \in \mathbf{R}^{n+1} \mid (y_1^2 + \cdots + y_n^2)^{1/2} \leq y_{n+1}\}$:

$$\psi(y) = \log(y_{n+1}^2 - y_1^2 - \cdots - y_n^2) \quad \text{with degree } (\theta = 2)$$

Properties

- (without proof): for $y \succ_K 0$,

$$\nabla\psi(y) \succeq_{K^*} 0, \qquad y^T \nabla\psi(y) = \theta$$

- nonnegative orthant $\mathbf{R}^n_+$: $\psi(y) = \sum_{i=1}^n \log y_i$

$$\nabla\psi(y) = (1/y_1, \ldots, 1/y_n), \qquad y^T \nabla\psi(y) = n$$

- positive semidefinite cone $\mathbf{S}^n_+$: $\psi(Y) = \log\det Y$

$$\nabla\psi(Y) = Y^{-1}, \qquad \mathbf{tr}(Y\nabla\psi(Y)) = n$$

- second-order cone $K = \{y \in \mathbf{R}^{n+1} \mid (y_1^2 + \cdots + y_n^2)^{1/2} \le y_{n+1}\}$:

$$\nabla\psi(y) = \frac{2}{y_{n+1}^2 - y_1^2 - \cdots - y_n^2} \begin{bmatrix} -y_1 \\ \vdots \\ -y_n \\ y_{n+1} \end{bmatrix}, \qquad y^T \nabla\psi(y) = 2$$

Logarithmic barrier and central path

logarithmic barrier for $f_1(x) \preceq_{K_1} 0, \ldots, f_m(x) \preceq_{K_m} 0$:

$$\phi(x) = -\sum_{i=1}^{m} \psi_i(-f_i(x)), \qquad \mathbf{dom}\,\phi = \{x \mid f_i(x) \prec_{K_i} 0,\ i = 1, \ldots, m\}$$

- ψ_i is generalized logarithm for K_i, with degree θ_i
- ϕ is convex, twice continuously differentiable

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

$$\begin{array}{ll} \text{minimize} & tf_0(x) + \phi(x) \\ \text{subject to} & Ax = b \end{array}$$

Dual points on central path

$x = x^\star(t)$ 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, $x^\star(t)$ minimizes Lagrangian $L(x, \lambda^\star(t), \nu^\star(t))$, where

$$\lambda_i^\star(t) = \frac{1}{t}\nabla\psi_i(-f_i(x^\star(t))), \qquad \nu^\star(t) = \frac{w}{t}$$

- from properties of ψ_i: $\lambda_i^\star(t) \succ_{K_i^*} 0$, with duality gap

$$f_0(x^\star(t)) - g(\lambda^\star(t), \nu^\star(t)) = (1/t)\sum_{i=1}^m \theta_i$$

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: $x^\star(t)$ minimizes $tc^T x - \log\det(-F(x))$; hence

$$tc_i - \mathbf{tr}(F_i F(x^\star(t))^{-1}) = 0, \quad i = 1, \ldots, n$$

- dual point on central path: $Z^\star(t) = -(1/t)F(x^\star(t))^{-1}$ is feasible for

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

- duality gap on central path: $c^T x^\star(t) - \mathbf{tr}(GZ^\star(t)) = p/t$

Barrier method

---

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

repeat

1. *Centering step.* Compute $x^\star(t)$ by minimizing $tf_0 + \phi$, subject to $Ax = b$.
2. *Update.* $x := x^\star(t)$.
3. *Stopping criterion.* **quit** if $(\sum_i \theta_i)/t < \epsilon$.
4. *Increase* t. $t := \mu t$.

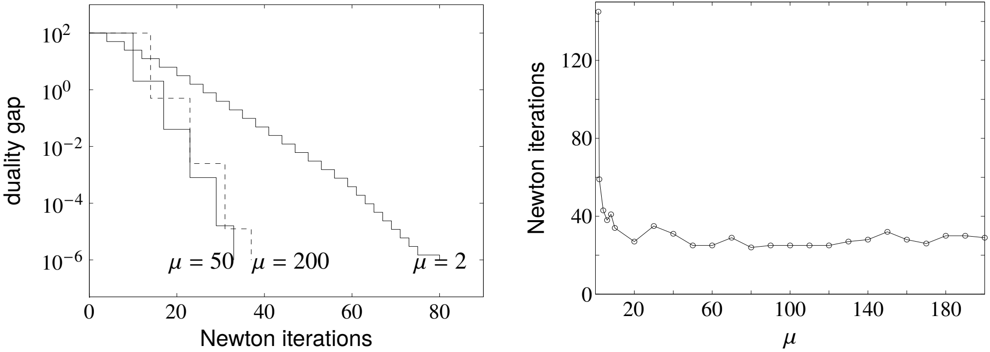
- only difference is duality gap m/t on central path is replaced by $\sum_i \theta_i/t$
- number of outer iterations:

$$\left\lceil \frac{\log((\sum_i \theta_i)/(\epsilon t^{(0)}))}{\log \mu} \right\rceil$$

- complexity analysis via self-concordance applies to SDP, SOCP

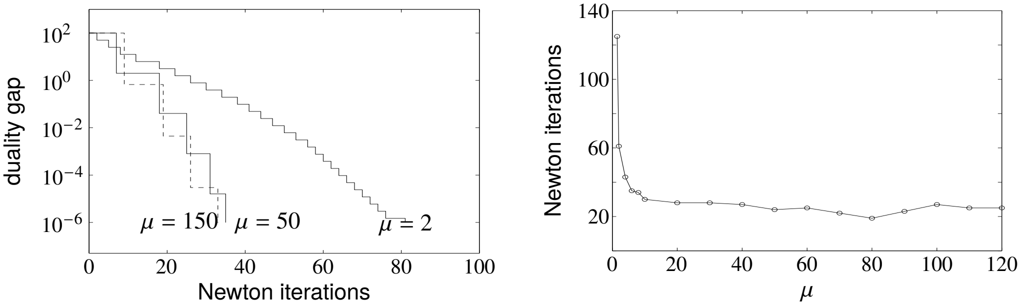
Example: SOCP

(50 variables, 50 SOC constraints in $\mathbf{R}^6$)



Example: SDP

(100 variables, LMI constraint in $\mathbf{S}^{100}$)

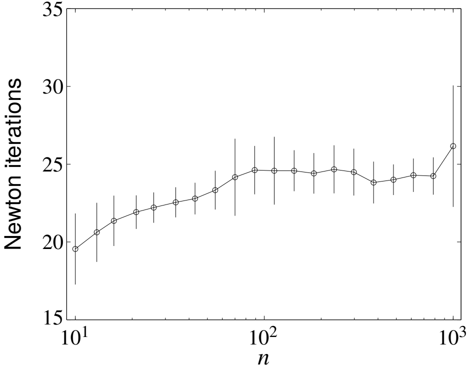


Example: Family of SDPs

($A \in \mathbf{S}^n$, $x \in \mathbf{R}^n$)

$$\begin{array}{ll} \text{minimize} & \mathbf{1}^T x \\ \text{subject to} & A + \mathbf{diag}(x) \succeq 0 \end{array}$$

$n = 10, \ldots, 1000$; 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, and κ, 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

12. Conclusions

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
- solution methods with polynomial worst-case complexity theory (with self-concordance)

Practical consequences of convexity

(most) **convex problems can be solved globally and efficiently**

- interior-point methods require $20 - 80$ steps in practice
- 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)
- high-quality solvers (some open-source) are available
- high level modeling tools like `CVXPY` ease modeling and problem specification

How to use convex optimization

to use convex optimization in some applied context

- use rapid prototyping, approximate modeling
  - start with simple models, small problem instances, inefficient solution methods
  - if you don't like the results, no need to expend further effort on more accurate models or efficient algorithms
- work out, simplify, and interpret optimality conditions and dual
- even if the problem is quite nonconvex, you can use convex optimization
  - in subproblems, *e.g.*, to find search direction
  - by repeatedly forming and solving a convex approximation at the current point

Further topics

some topics we didn't cover:

- methods for very large scale problems
- subgradient calculus, convex analysis
- localization, subgradient, proximal and related methods
- distributed convex optimization
- applications that build on or use convex optimization

these are all covered in EE364b.

Related classes

- EE364b — convex optimization II (Pilanci)
- EE364m — mathematics of convexity (Duchi)
- CS261, CME334, MSE213 — theory and algorithm analysis (Sidford)
- AA222 — algorithms for nonconvex optimization (Kochenderfer)
- CME307 — linear and conic optimization (Ye)