CSCI5254: Convex Optimization and Its Applications

Introduction

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

Course logistics

basic information

- canvas: TBA

  course website: http://spot.colorado.edu/∼lich1539/cvxopt.html

- time and location: WM 1:25-2:40pm, DLC 1B20

  zoom: https://cuboulder.zoom.us/j/6933927360

  (will be recorded and posted)

- office hours:

  - Tue 3:00pm-5:00pm on zoom (tentative)
  - by appointment (email is most convenient)

- main textbook: Convex Optimization, Boyd & Vandenberghe
- prerequisite: calculus and linear algebra, exposure to probability
  - review Appendix A
- acknowledgement: lecture slides are adapted mainly from Dr. Stephen Boyd's lecture notes on convex optimization at Stanford University.

grading, homework, and final

- grading: 40% homework, 50% final, 10% participation
- homework:
  - 8 homework sets
  - due by midnight on Mondays
  - collaboration strongly encouraged, but write your own solutions
- final: 24-hour take-home; more detail later in the semester

Course goals and topics

goals

- recognize/formulate convex optimization problems that arise in engineering and applied science
- characterize optimal solution and understand how such problems are solved numerically
- develop skills to use tools and methods of optimization in your researches or applications

topics

- convex analysis: convex sets, functions, optimization problems
- optimization theory: linear, quadratic, semidefinite, and geometric programming; optimality conditions and duality theory

- basic applications: signal processing, control, communications, networks, statistics, machine learning, circuit design, and mechanical engineering, etc; will adapt depending on your interest and time

- some optimization algorithms: descent methods and interior-point methods

- some advanced topics: stochastic gradient algorithms, reinforcement learning, if time permits

Mathematical optimization

(mathematical) optimization problem

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

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

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

Examples

portfolio optimization

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

communications

- variables: transmission power to each user in a cell
- constraints: power budget, maximal interference to users in other cells
- objective: total or sum rate

data fitting and machine learning

- variables: model parameters

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

networks

- variables: flow rates (the source sending rate of each communication)
- constraints: link capacities
- objective: total network utility

sparse recovery

- variables: unknown sparse signal
- constraints: measurements of signal
- objective: sparsity or recovery error

Solving optimization problems

general optimization problem

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

exceptions: certain problem classes can be solved efficiently and reliably

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

"In fact the great watershed in optimization isn't between linearity and nonlinearity, but convexity and nonconvexity."

– Rockafellar, SIAM Review, 1993

Least-squares

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

solving least-squares problems

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

using least-squares

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

Linear programming

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

solving linear programs

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

using linear programming

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

Convex optimization problem

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

- objective and constraint functions are convex:

$$f_i(\lambda x + \mu y) \leq \lambda f_i(x) + \mu f_i(y)$$

  if $\lambda + \mu = 1$, $\lambda \geq 0$, $\mu \geq 0$

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

solving convex optimization problems

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

using convex optimization

- often difficult to recognize, but surprisingly many applications
- critical to efficient computation
- critical to distributed computation/decision, many implications for architecture and operation of complex networked systems
- important to learn skills to formulate problems as convex problems and explore (hidden) convexity

Nonlinear optimization

traditional techniques for general nonconvex problems involve compromises

local optimization methods (nonlinear programming)

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

global optimization methods

- find the (global) solution
- worst-case complexity is exponential with problem size

insights from convex optimization is helpful

- initialization for local optimization methods
- convex relaxation can lead to good bound, efficient algorithm, and even exact solution

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: polynomial-time interior-point methods for linear programming (Karmarkar 1984)
- late 1980s–now: polynomial-time interior-point methods for nonlinear convex optimization (Nesterov & Nemirovski 1994)

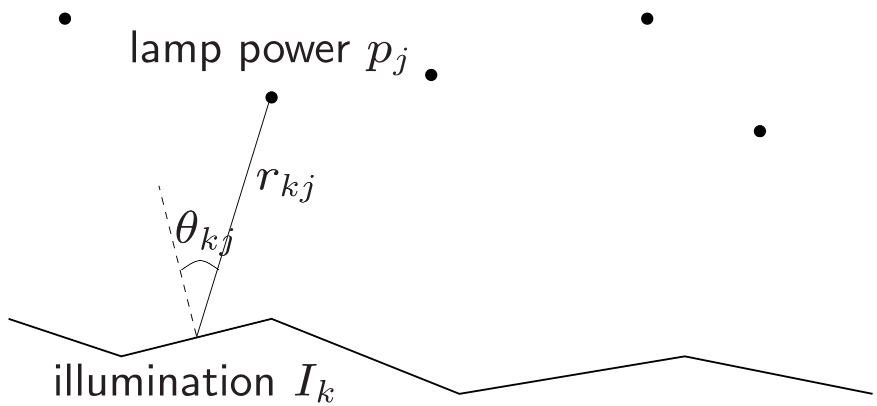
applications

- before 1990: mostly in operations research; few in engineering

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

Example

m lamps illuminating n (small, flat) patches



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

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

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

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

how to solve?

1. use uniform power: $p_j = p$, vary p

2. use least-squares:

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

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

3. use weighted least-squares:

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

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

4. use linear programming:

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

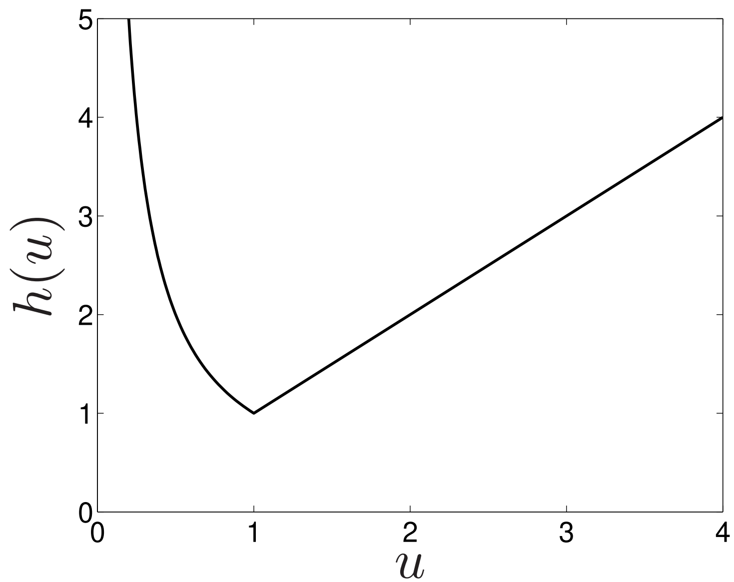
   which can be solved via linear programming

of course these are approximate (suboptimal) 'solutions'

5. use convex optimization: problem is equivalent to

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

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



f_0 is convex because maximum of convex functions is convex

exact solution obtained with effort $\approx$ modest factor $\times$ least-squares effort additional constraints: does adding 1 or 2 below complicate the problem?

1. no more than half of total power is in any 10 lamps
2. no more than half of the lamps are on ($p_j > 0$)

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

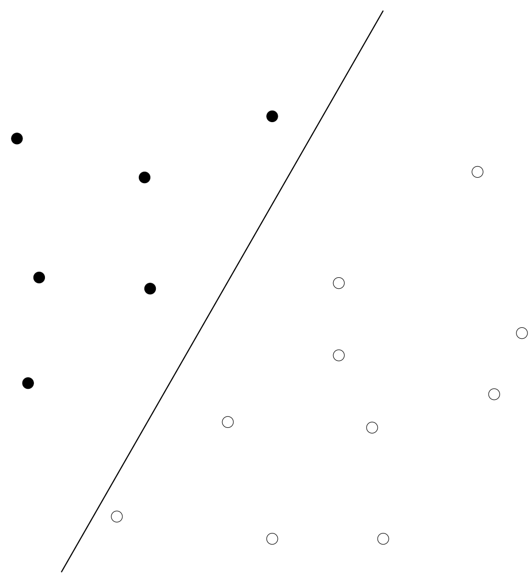
1. no more than half of total power is in any 10 lamps
2. no more than half of the lamps are on ($p_j > 0$)

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

Example: Linear discrimination

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

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



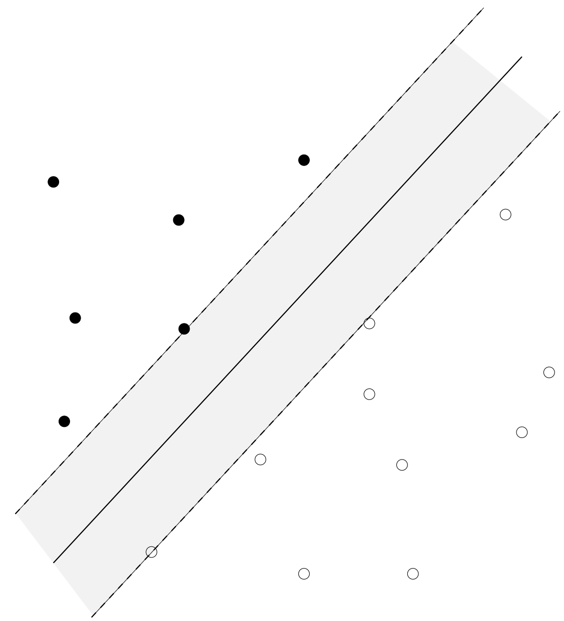
homogeneous in a, b, hence equivalent to

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

a set of linear inequalities in a, b

Robust linear discrimination

(Euclidean) distance between hyperplanes



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

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

to separate two sets of points by maximum margin,

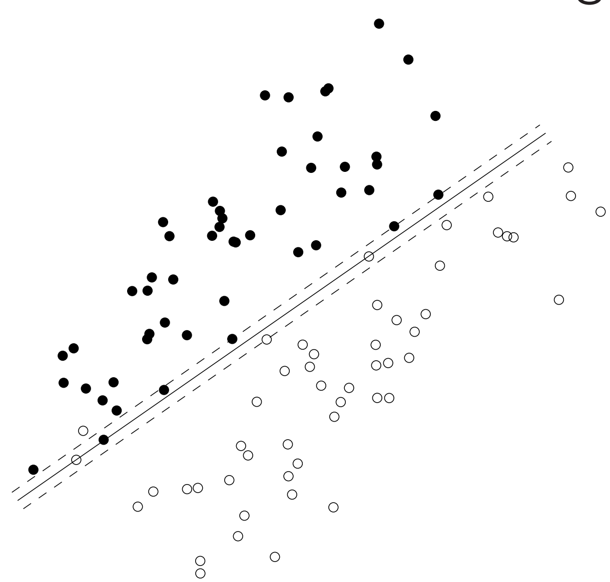
$$\begin{array}{ll} \text{minimize} & (1/2)\|a\|_2 \\ \text{subject to} & a^T x_i + b \geq 1, \quad i = 1, \ldots, N \\ & a^T y_i + b \leq -1, \quad i = 1, \ldots, M \end{array} \qquad (1)$$

(after squaring objective) a QP in a, b

Approximate linear separation of non-separable sets

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

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



Support vector classifier

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

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

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