

1. Introduction

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

(Thanks to Professor Stephen Boyd, Stanford University for permission to use and modify his slides)

Mathematical optimization

(mathematical) optimization problem

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

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

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

Examples

portfolio optimization

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

device sizing in electronic circuits

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

data fitting

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

Solving optimization problems

general optimization problem

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

exceptions: certain problem classes can be solved efficiently and reliably

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

Least-squares

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

solving least-squares problems

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

using least-squares

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

Linear programming

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

solving linear programs

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

using linear programming

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

Convex optimization problem

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

- objective and constraint functions are convex:

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

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

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

solving convex optimization problems

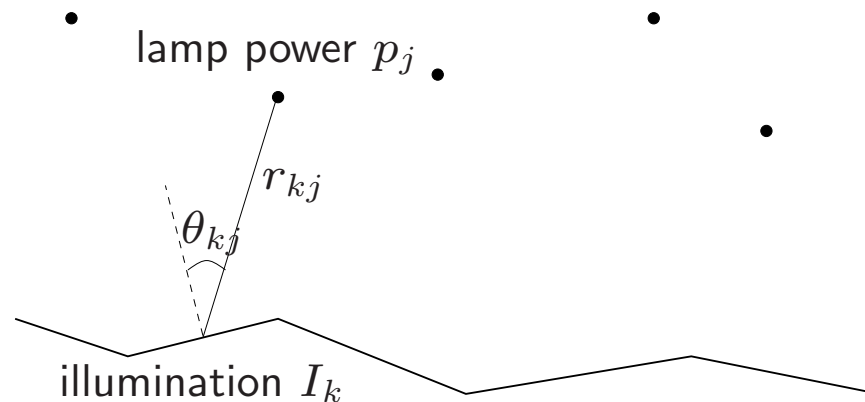
- no analytical solution
- reliable and efficient algorithms
- computation time (roughly) proportional to $\max\{n^3, n^2m, F\}$, where F is cost of evaluating f_i 's and their first and second derivatives
- almost a technology

using convex optimization

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

Example

m lamps illuminating n (small, flat) patches



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

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

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

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

how to solve?

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

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

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

3. use weighted least-squares:

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

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

4. use linear programming:

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

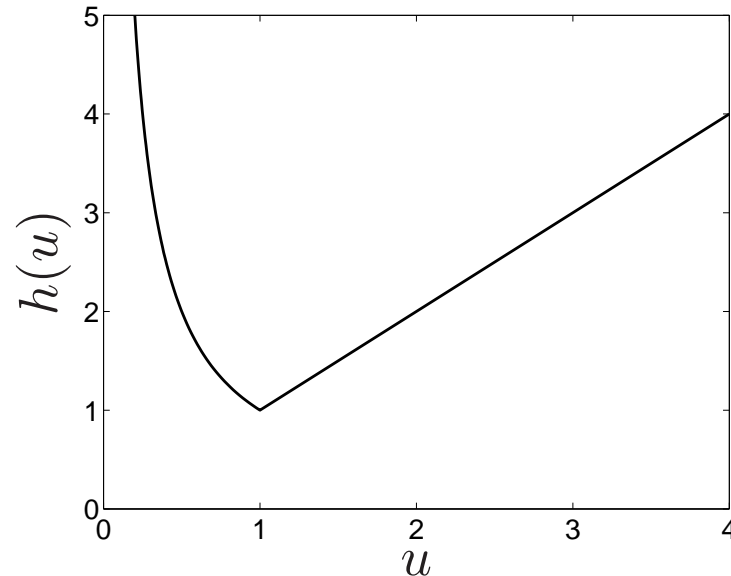
which can be solved via linear programming

of course these are approximate (suboptimal) 'solutions'

5. use convex optimization: problem is equivalent to

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

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



f_0 is convex because maximum of convex functions is convex

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

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

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

answer: with (1), still easy to solve; with (2), extremely difficult

Another Example: Robust LP

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

- assume a_i is Gaussian with mean \bar{a}_i , covariance Σ_i ($a_i \sim \mathcal{N}(\bar{a}_i, \Sigma_i)$)
- Is this problem hard or easy?
- answer: for $\eta > \frac{1}{2}$ it is easy!, for $\eta \leq \frac{1}{2}$ VERY hard!!
- moral: (untrained) intuition doesn't always work; without the proper background very easy problems can appear quite similar to very difficult problems

Reconstruction of the early Universe as a convex optimization problem

Y. Brenier,¹ U. Frisch,^{2,3*} M. Hénon,² G. Loeper,¹ S. Matarrese,⁴
R. Mohayaee,² A. Sobolevskii^{2,5}

¹ CNRS, UMR 6621, Université de Nice-Sophia-Antipolis, Parc Valrose, 06108 Nice Cedex 02, France

² CNRS, UMR 6529, Observatoire de la Côte d’Azur, BP 4229, 06304 Nice Cedex 4, France

³ Institute for Advanced Study, Einstein Drive, Princeton, NJ 08540, USA

⁴ Dipartimento di Fisica ‘G. Galilei’ and INFN, Sezione di Padova, via Marzolo 8, 35131-Padova, Italy

⁵ Department of Physics, M. V. Lomonossov Moscow University, Leninskie Gory, 119992 Moscow, Russia

11 April 2003

ABSTRACT

We show that the deterministic past history of the Universe can be uniquely reconstructed from the knowledge of the present mass density field, the latter being inferred from the 3D distribution of luminous matter, assumed to be tracing the distribution of dark matter up to a known bias. Reconstruction ceases to be unique below those scales – a few Mpc – where multi-streaming becomes significant. Above $6 h^{-1}$ Mpc we propose and implement an effective Monge–Ampère–Kantorovich method of unique reconstruction. At such scales the Zel’dovich approximation is well satisfied and reconstruction becomes an instance of optimal mass transportation, a problem which goes back to Monge (1781). After discretization into N point masses one obtains an assignment problem that can be handled by effective algorithms with not more than $O(N^3)$ time complexity and reasonable CPU time requirements. Testing against N -body cosmological simulations gives over 60% of exactly reconstructed points.

We apply several interrelated tools from optimization theory that were not used in cosmological reconstruction before, such as the Monge–Ampère equation, its relation to the mass transportation problem, the Kantorovich duality and the auction algorithm for optimal assignment. Self-contained discussion of relevant notions and techniques is provided.

Key words: cosmology: theory – large-scale structure of the Universe – hydrodynamics

Course goals and topics

goals

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

topics

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

Nonlinear optimization

traditional techniques for general nonconvex problems involve compromises

local optimization methods (nonlinear programming)

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

global optimization methods

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

these algorithms are often based on solving convex subproblems

Brief history of convex optimization

theory (convex analysis): ca1900–1970

algorithms

- 1947: simplex algorithm for linear programming (Dantzig)
- 1960s: early interior-point methods (Fiacco & McCormick, Dikin, . . .)
- 1970s: ellipsoid method and other subgradient methods
 - Kantorovich wins Nobel Prize in Economics in 1975 (Dantzig doesn't!)
 - Shor develops the ellipsoid algorithm
 - A 25 year old Georgian Mathematician named Leonid Khachian proves that ellipsoid algorithm solves LPs with polynomial time complexity. Appears on the cover of NY Times
- 1980s: polynomial-time interior-point methods for linear programming
 - interior-point polynomial time algorithm for LP (Karmarkar 1984).
Works Better than Simplex!

- Nemirovsky and Yudin prove that information complexity of convex programs is far better than general nonlinear programs.
- late 1980s–now: polynomial-time interior-point methods for nonlinear convex optimization (Nesterov & Nemirovski 1994)
 - Nesterov and Nemirovsky extend Karmarkar’s approach to general convex problems.

Applications

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