1. Introduction

- mathematical optimization
- least-squares and linear programming 最小二乘法和线性规划（均为凸问题）
- convex optimization
- example
- course goals and topics
- nonlinear optimization
- brief history of convex optimization

Mathematical optimization

(mathematical) optimization problem

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

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

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

Examples

portfolio optimization

投资组合优化

- variables: amounts invested in different assets
- constraints: budget, max./min. investment per asset, minimum return 最低回报
- objective: overall risk or return variance 降低收益率变化的范围/方差

device sizing in electronic circuits

- variables: device widths and lengths
- constraints: manufacturing limits, timing requirements, maximum area
  电路的时序要求
- objective: power consumption

data fitting

机器学习、大数据等

- variables: model parameters
- constraints: prior information, parameter limits 先验知识：如有些模型中协方差矩阵必须是半正定或者正定的
- objective: measure of misfit or prediction error, plus regularization term

设置好问题背景，明确最小化的目标，执行优化过程

Solving optimization problems

general optimization problem

- very difficult to solve

折衷/妥协

- methods involve some compromise, *e.g.*, very long computation time, or not always finding the solution (which may not matter in practice)

可能会消耗很长的计算时间，或者不是永远都能找到最优解

exceptions: certain problem classes can be solved efficiently and reliably

最小二乘法、线性规划、凸优化问题可以有效且稳定地找到最优解

- least-squares problems
- linear programming problems

最小二乘法、线性规划都是凸优化问题中的一类，属于凸优化的子类问题

- convex optimization problems

如何判断一个问题是不是最小二乘问题：
看目标函数是不是x的仿射函数的L2范数的平方，
这里放射函数是指x乘以系数再加上常数。
如果是，则属于最小二乘问题，如果不是，则不属于最小二乘问题

Least-squares

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

solving least-squares problems

后面讲的很多凸优化问题都没有闭式解析解

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

  k是行数，n是列数

- a mature technology

  在大数据应用中，k通常是样本个数，n是特征维度等

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

简称为LP

x是n行1列的向量

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

目标函数是一个线性函数

solving linear programs

- no analytical formula for solution

没有闭式解析解

- reliable and efficient algorithms and software
- computation time proportional to $n^2 m$ if $m \geq n$; less with structure

n是变量x的行数，m是约束不等式的数量

- a mature technology

一般来说不等式数量m要大于等于变量个数n

using linear programming

很多问题看起来不像线性规划问题（比如有非线性函数），但可以转化为线性规划问题

- 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)

后面会详细讲到

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:

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

凸函数的定义

从导数的角度来讲，凸函数具有非负的曲率

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

仿射函数（线性）的曲率为0，因此满足条件

- 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

约束函数及其一二阶导数

- almost a technology

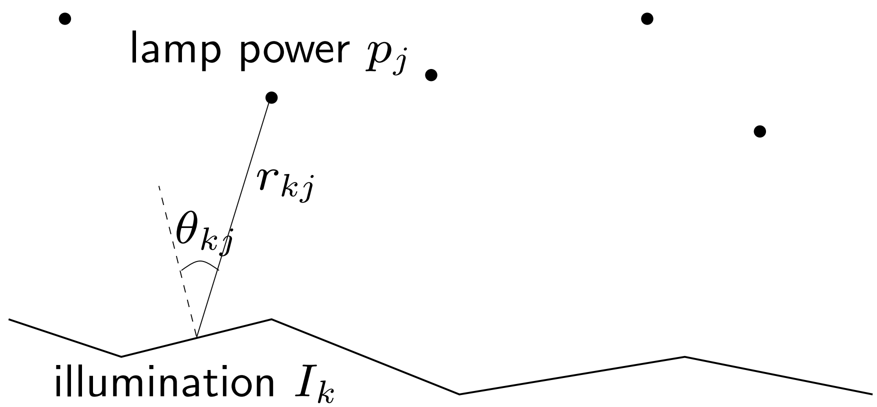
using convex optimization

实际中的函数很难辨别是不是凸函数

- often difficult to recognize
- many tricks for transforming problems into convex form 把问题转化成凸问题
- 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:

认为是非相干光，在参考面上满足线性叠加性

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

要乘以夹角的角度

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

越接近所需要的光照强度越好

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

how to solve?

1. use uniform power: $p_j = p$, vary p
2. use least-squares: 方法一

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

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

3. use weighted least-squares: 方法二

以最大功率的一半作为分界线

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

   iterativel adjust weights w_j until $0 \le p_j \le p_{\text{max}}$ 在机器学习中，后一项称为正则化
   Boyd认为交叉验证的效果比正则化更好

4. use linear programming: 方法三 这个目标函数不是最小化sum error，而是最小化fractional error

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

最小化照明差异的最大项，很神奇的想法

天然的线性不等式

   which can be solved via linear programming

of course these are approximate (suboptimal) 'solutions'

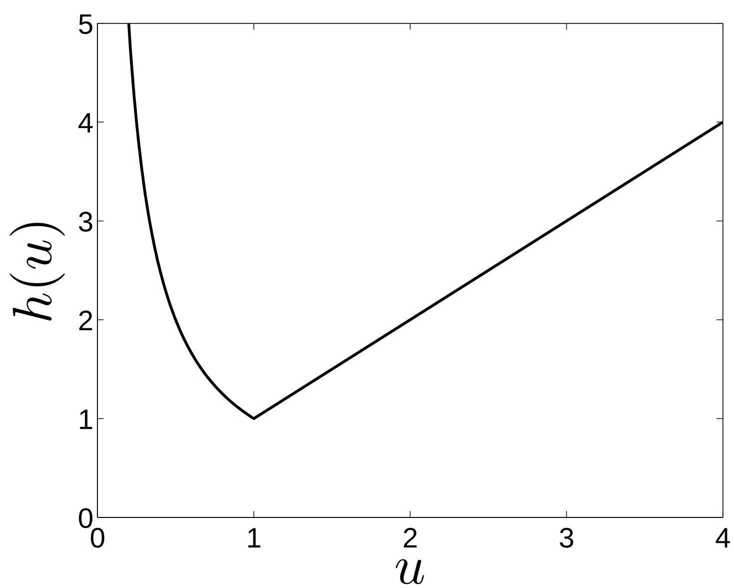
以上都属于近似解

5. use convex optimization: problem is equivalent to

牛逼！

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

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



如果I_k比I_des大，则保持不变，最小化f等价于将I_k缩小到I_des

而如果I_k比I_des小，则取倒数，最小化f等价于将I_k放大到I_des

由于凸函数之和还是凸函数，所以max h(u)仍是凸函数

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?

在任意10盏灯中，每盏灯的功率不超过总功率的一半

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

第二个问题可以用启发式方法来解决

不超过一半的灯是开着的

- 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

直觉并不一定总是work的，特别是在连续型凸优化问题中没有学过convexity等知识的情况下

Course goals and topics

goals

辨别/表达问题，转化为凸优化问题

1. recognize/formulate problems (such as the illumination problem) as convex optimization problems

设计中等大小问题的程序 有很多专业的人写了相关的程序

2. develop code for problems of moderate size (1000 lamps, 5000 patches)
3. characterize optimal solution (optimal power distribution), give limits of performance, etc.

学过统计学和机器学习的人，对于指数族函数，用于正确估计的参数不是协方差，而是逆协方差。正确的估计不是一对向量的平均值，而是逆协方差乘以均值。

topics

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

Nonlinear optimization

需要一定的折衷妥协

traditional techniques for general nonconvex problems involve compromises

非线性规划，又称为NLP

local optimization methods (nonlinear programming)

ILP是指整数线性规划，即Integer Linear Programming

- find a point that minimizes f_0 among feasible points near it
- fast, can handle large problems
- require initial guess 比如梯度下降法对初值的要求很高
- provide no information about distance to (global) optimum 无法确认与全局最优解的距离，把这件事看成尽力而为的事

global optimization methods

- find the (global) solution 需要更大的计算时间，甚至是指数级别的时间复杂度
- worst-case complexity grows exponentially with problem size

把一个非凸问题分解为多个凸问题，从而作为子程序

these algorithms are often based on solving convex subproblems

Brief history of convex optimization

theory (convex analysis): 1900–1970

algorithms

时间上与现代数字计算机的发展相一致

- 1947: simplex algorithm for linear programming (Dantzig) LP的单纯形法
- 1970s: ellipsoid method and other subgradient methods 椭球法和次梯度法
- 1980s & 90s: polynomial-time interior-point methods for convex optimization (Karmarkar 1984, Nesterov & Nemirovski 1994) 多项式时间的内点法
- 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, . . .) 如Fisher分析等
- since 2000s: machine learning and statistics statistical estimate