Numerical Optimization

Spring 2022

Ye Shi

信息科学与技术学院
School of Information Science and Technology

Outline

- **Course introduction**
  - Mathematical modeling, linear and nonlinear programming, integer programming, convex analysis, optimization in machine learning
- **Optimization classification**
  - Constrained vs. unconstrained, convex vs. nonconvex, deterministic vs. stochastic, solvability vs. scalability
- **Optimization algorithms**
  - Simplex, Branch & Bound, Gradient descent, Newton method, Difference of Convex, Distributed algorithms, etc.
- **Topics and grading**
  - Theoretical foundations, machine learning approaches, and applications.

Mathematical Optimization

- The field of optimization is concerned with the study of maximization and minimization of mathematical functions. Very often the arguments of (i.e., variables or unknowns in) these functions are subject to side conditions or constraints. By virtue of its great utility in such diverse areas as data science, machine learning, engineering, economics, finance, medicine, and statistics, optimization holds an important place in the practical world and the scientific world. Indeed, as far back as the Eighteenth Century, the famous Swiss mathematician and physicist Leonhard Euler (1707-1783) proclaimed[a] that . . . nothing at all takes place in the Universe in which some rule of maximum or minimum does not appear.

  - [a]See Leonhardo Eulero, *Methodus Inviendi Lineas Curvas Maximi Minimive Proprietate Gaudentes*, Lausanne & Geneva, 1744, p. 245.

Mathematical modeling → Optimization problem → Optimization Algorithm

Mathematical Optimization

- 数学规划(Mathematical Programming)，也称数学最优化(Mathematical Optimization)，是应用数学、计算数学与运筹学的交叉领域。

数学 →

理论数学

方程、泛函、线性/非线性泛函、集合论、微分几何、黎曼几何、代数、群论、拓扑、概率论、数论……

计算数学

数值代数、数值分析、偏微分方程数值解、有限元、并行计算、数值最优化、数理逻辑……

应用数学

概率统计、数学建模、数值方法、运筹学、数学实验、计算机基础、金融数学………………

概率统计

国外运筹学发展历史

- 1908年，宾州州立大学第一次开设了工业工程课程，1909年成系。
- 1933年Cornell第一次授予工业工程博士学位。
- 20世纪二战期间：战争中的战术效率研究和试验。诞生了Operations Research行业。
- 1949年：兰德公司(RAND)，军用转民用。
- 1950–1960：第一个工业界的成功案例：shell、Exxon、BP石油公司的downstream石油冶炼中的pooling问题。
- 欧美大学的运筹系、管理科学系、工业工程系、系统科学系纷纷涌现。
- 现代计算机的诞生和发展，促使运筹学能够研究求解更加复杂的最优化问题。

中国运筹学发展历史

- 1956年第一个运筹学小组：中科学力学所，钱学森、许国志先生1955年回国成立。
- 1959年第二个运筹学小组：力学所小组+数学所小组，主要研究排队 论、非线性规划和图论，运输理论、动态规划和经济分析。
- 1963年数学所的运筹学研究室为中国科技大学应用数学系的第一届学
- 生（58 届）开设了较为系统的运筹学专业课
- 1950年末，主要研究运输问题，“打麦场的选址问题”
- 1963年，华罗庚先生带队到农村、工厂讲解基本的优化技术和统筹方法，使之用于日常的生产和生活中
- 改开后，组合优化、生产系统优化、图论和非线性规划领域长足发展

优化应用

- 运筹层面
  - ➢ 主要关注传统优化问题的表述并研发求解策略，通 常使用已经研究好的算法或者成熟软件
  - ➢ 这个层次碰到的许多问题含有线性约束和离散变量
- 工程层面
  - ➢ 将优化策略应用到具有挑战性的实际问题中
  - ➢ 这个层次的优化知识混杂了可应用方法的有效性和可靠性，主要研究内容：解的分析，求解方法失败 的诊断及恢复（可解释性）

优化算法

- 数学规划层面
  - ➢ 重点研究“优化问题和算法的设计和基本性质”
  - ➢ 核心问题：解的存在性及描述、算法的收敛性和收敛速度等
- 科学计算层面
  - ➢ 受数学性质和(为了有效和实用目的)实现的强烈影响
  - ➢ 研究问题：数值稳定性、算法步骤的病态性、计算复杂度等

学完课程后的收获

- 建模：为具有挑战性的科学问题、管理问题和工程问题研发更好的方法，提供更恰当的表述方式
- 算法：作为数学最优化的专家(应用数学和运筹学)发展更好的求解方法
- 调用求解器：作为工程师理解并运用最新最有效的优化方法求解特定的应用

数学描述与例子

数学规划问题(基本形式)

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c_i(x) = 0, \quad i \in \mathcal{E} \\ & c_i(x) \leq 0, \quad i \in \mathcal{I} \end{aligned}$$

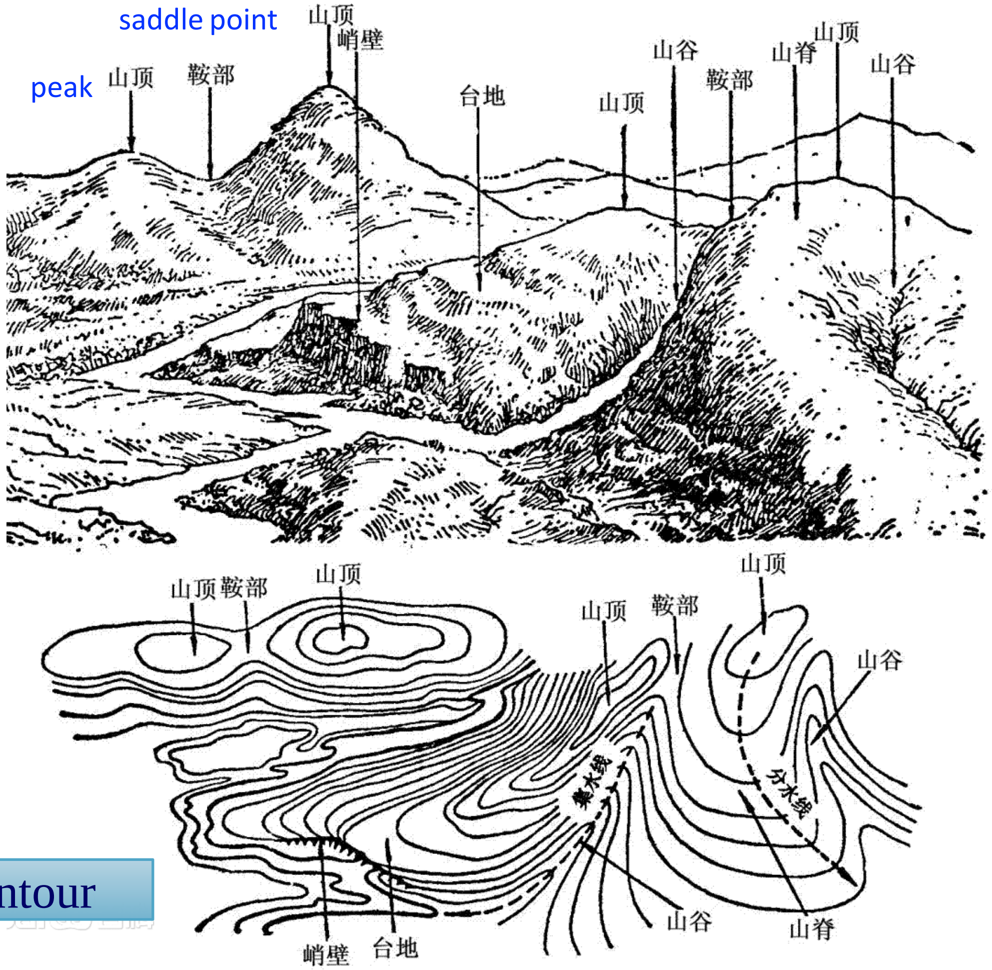
一般考虑极小化minimize
也有极大化问题maximize
分别简写为min、max

另外也有inf问题、sup问题

subject to简写为s.t.

三要素

- 目标(objective)：系统性能的一种“量的度量”(利润、时间、势能)
- 变量(variables)：目标所依赖的系统的“某些可控的特征”
- 约束条件(constraints)：变量经常以某种方式受限制(如贷款利率 的量不能是负的)


saddle point
peak
山顶
鞍部
山顶
峭壁
台地
山顶
山谷
鞍部
山脊
山顶
山谷
山顶 鞍部
山顶
山顶
鞍部
山顶
山谷
集水线
分水线
峭壁
台地
山谷
山脊

Contour

3.3 二元一次不等式组与简单的线性规划问题

我们先考察生产中遇到的一个问题：

某工厂生产甲、乙两种产品，生产 1 t 甲种产品需要 A 种原料4 t、B 种原料 12 t，产生的利润为 2 万元；生产 1 t 乙种产品需要 A 种原料 1 t、B 种原料 9 t，产生的利润为 1 万元. 现有库存 A 种原料 10 t、B 种原料 60 t，如何安排生产才能使利润最大？

为理解题意，可将已知数据整理成下表：

	A 种原料(t)	B 种原料(t)	利　润(万元)
甲种产品(1 t)	4	12	2
乙种产品(1 t)	1	9	1
现有库存(t)	10	60	

设计划生产甲、乙两种产品的吨数分别为 x，y，利润为 P（万元）. 根据题意，A，B 两种原料分别不得超过 10 t 和 60 t，又产量不可能是负数，于是可得二元一次不等式组

$$\begin{cases} 4x+y\leqslant 10, \\ 12x+9y\leqslant 60, \\ x\geqslant 0, \\ y\geqslant 0, \end{cases}$$

即

$$\begin{cases} 4x+y\leqslant 10, \\ 4x+3y\leqslant 20, \\ x\geqslant 0, \\ y\geqslant 0. \end{cases}$$

因此，上述问题转化为如下的一个数学问题：在约束条件

$$\begin{cases} 4x+y\leqslant 10, \\ 4x+3y\leqslant 20, \\ x\geqslant 0, \\ y\geqslant 0 \end{cases}$$

下，求出 x，y，使利润

这是一个含有两个变量 x 和 y 的函数，称为目标函数.

$$P=2x+y$$

达到最大.

● 如何解决这个问题？

3.3.1 二元一次不等式表示的平面区域

我们分两步求解上面的问题：

第一步　研究问题中的约束条件，确定数对 (x,y) 的范围；

第二步　在第一步得到的数对 (x,y) 的范围中，找出使 P 达到最大的数对 (x,y).

先讨论第一步.

如图 3-3-1(1)，直线 l：$4x+y=10$ 将平面分成上、下两个半平面区域，直线 l 上的点的坐标满足方程 $4x+y=10$，即 $y=10-4x$，直线 l 上方的平面区域中的点的坐标满足不等式 $y>10-4x$，直线 l 下方的平面区域中的点的坐标满足不等式 $y<10-4x$.

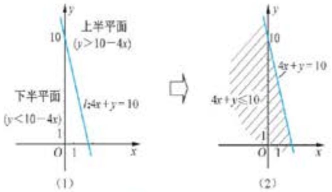


图 3-3-1

因此，$4x+y\leqslant 10$ 在平面上表示的是直线 l 及直线 l 下方的平面区域，即图 3-3-1(2) 中的阴影部分（包括边界直线 l）.

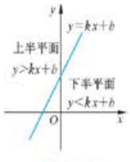


图 3-3-2

一般地，直线 $y=kx+b$ 把平面分成两个区域（图 3-3-2）：

$y>kx+b$ 表示直线上方的平面区域；

$y<kx+b$ 表示直线下方的平面区域.

思　考

对于二元一次不等式 $Ax+By+C>0$ $(A^2+B^2\neq 0)$，如何确定它所表示的平面区域？

例 1　画出下列不等式所表示的平面区域：

(1) $y>-2x+1$；　　(2) $x-y+2>0$.

解　(1)，(2) 两个不等式所表示的平面区域如图 3-3-3(1)，(2) 所示.

TOY EXAMPLE:

$$\begin{array}{ll} \text{minimize} & (x_1-2)^2+(x_2-1)^2 \\ \text{subject to} & x_1^2-x_2 \le 0 \\ & x_1+x_2 \le 2. \end{array}$$

图解法

- 画出可行域
- 做目函数的等高线
- 确定临界点

可行域(feasible region)

可行集(feasible set)

可行解/点(feasible solution/point)

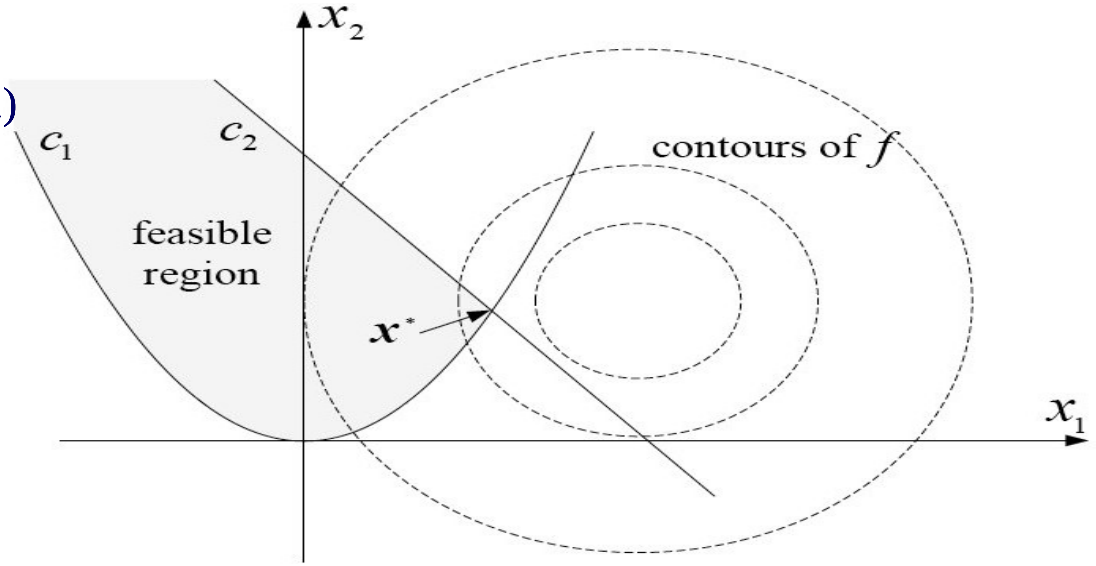
等高线/等值线(contour)

最优解(optimal solution)

极小点(minimizer)

解(solution)

最小值(minima)



Example：田忌赛马

齐使者如梁，孙膑以刑徒阴见，说齐使。齐使以为奇，窃载与之齐。齐将田忌善而客待之。忌数与齐诸公子驰逐重射。孙子见其马足不甚相远，马有上、中、下辈。于是孙子谓田忌曰："君弟重射，臣能令君胜。"田忌信然之，与王及诸公子逐射千金。及临质，孙子曰："今以君之下驷(sì)与彼上驷，取君上驷与彼中驷，取君中驷与彼下驷。"既驰三辈毕，而田忌一不胜而再胜，卒得王千金。于是忌进孙子于威王。威王问兵法，遂以为师。——《史记》卷六十五：《孙子吴起列传第五》

解答. 用1, 2, 3分别表示上、中、下等马.

$$令 x_{ij} = \begin{cases} 1, & 田 i \to 齐王的 j 等马 \\ 0, & 否则 \end{cases}$$

$$C = (c_{ij}) = \begin{bmatrix} -1 & 1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & -1 \end{bmatrix}$$

收益矩阵(payoff matrix)

指派问题(assignment problem)

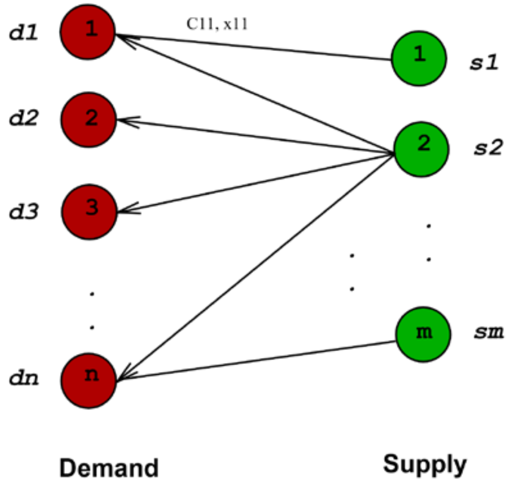
线性规划

$$\begin{aligned} \text{maximize} \quad & \sum_{i=1}^{3} \sum_{j=1}^{3} c_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{j=1}^{3} x_{ij} = 1, \; i = 1, 2, 3 \\ & \sum_{i=1}^{3} x_{ij} = 1, \; j = 1, 2, 3 \end{aligned}$$

$$x_{ij} \in \{0, 1\}, \; i, j = 1, 2, 3.$$

0-1整数线性规划

Optimal Transportation

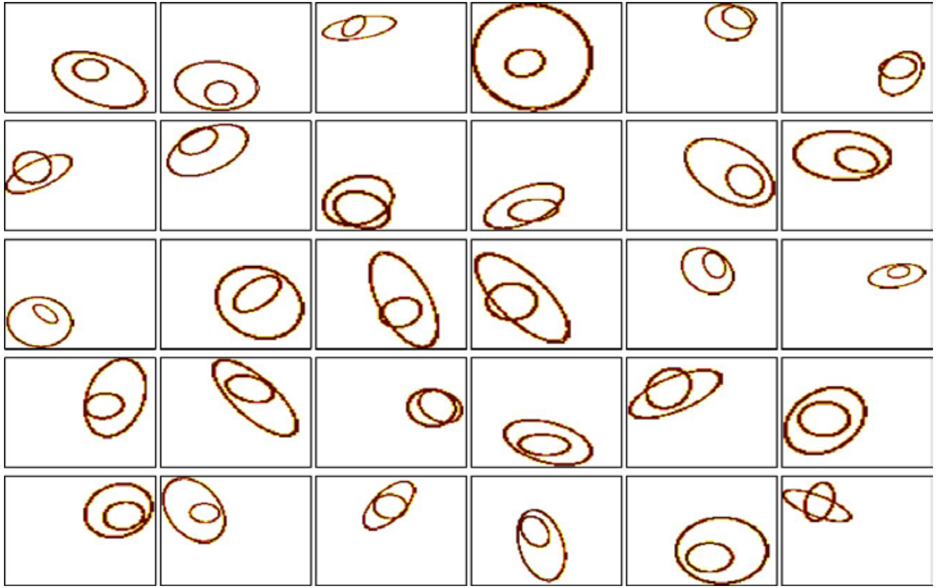


$$
\begin{aligned}
\min \quad & \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} & \\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{ij} & = s_i, \ \forall i = 1, ..., m \\
& \sum_{i=1}^{m} x_{ij} & = d_j, \ \forall j = 1, ..., n \\
& x_{ij} & \geq 0, \ \forall i, j.
\end{aligned}
$$

The minimal transportation cost is called the Wasserstein Distance (WD) between supply distribution **s** and demand distribution **d** (can be interpreted as two probability distributions after normalization). This is a linear program! The Wasserstein Barycenter Problem is to find a distribution such that the sum of its Wasserstein Distance to each of a set of other distributions would be minimized.

The Wasserstein Barycenter (Mean) Problem

What is the "mean or consensus" picture from a set of pictures:



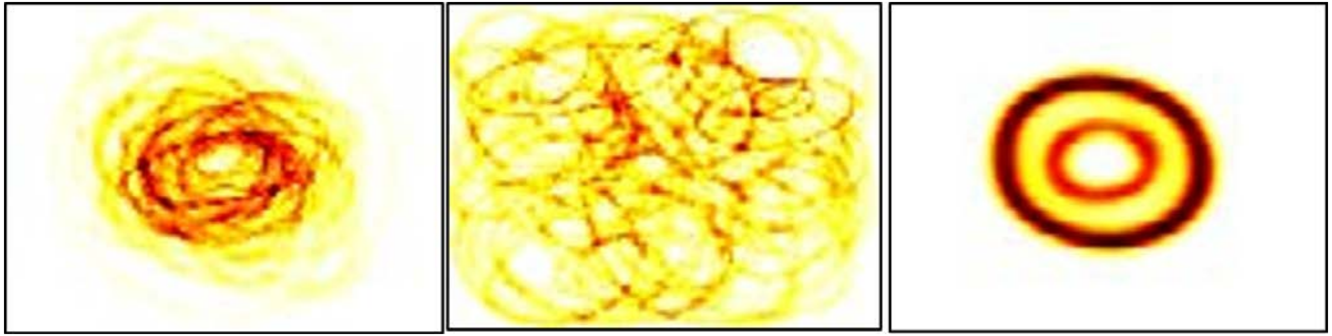


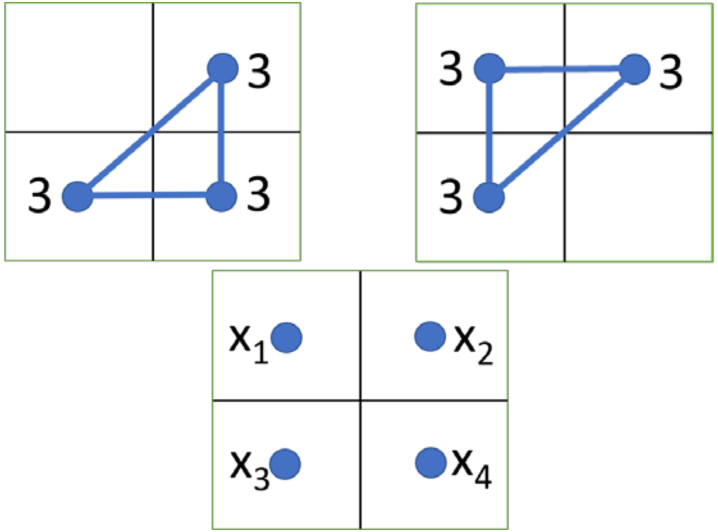
Figure 1: Mean picture constructed from the (a) Euclidean mean after re-centering images (b) Euclidean mean (c) Wasserstein Barycenter (self recenter, resize and rotate)

Euclidean Mean/Center:

$$\mathbf{x} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{a}_i, \quad \text{or} \quad \min_{\mathbf{x}} \sum_{i=1}^{n}\|\mathbf{x}-\mathbf{a}_i\|_2^2,$$

which is an unconstrained optimization, or least-squares, problem

Toy Example: the Wasserstein Barycenter (Mean)



Find distribution of $x_i,\ i = 1, 2, 3, 4$ such its WD to the left distribution plus WD to the right distribution is minimized, subject to

$$\min_{\mathbf{x}} WD_l(\mathbf{x}) + WD_r(\mathbf{x}) \qquad \text{s.t. } x_1 + x_2 + x_3 + x_4 = 9,\ x_i \geq 0,\ i = 1, 2, 3, 4.$$

This is a Hierarchy/Nonlinear optimization problem.

Model Classifications

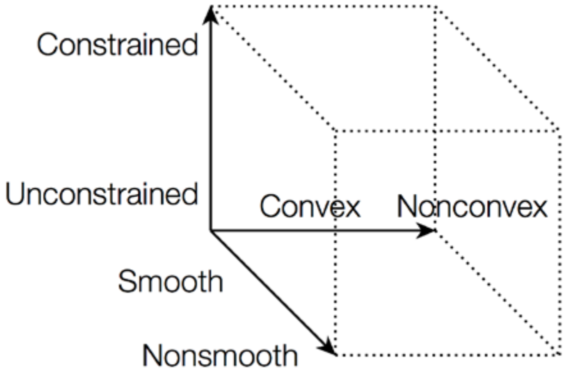
Optimization problems are generally divided into Unconstrained, Linear and Nonlinear Programming based upon the objective and constraints of the problem

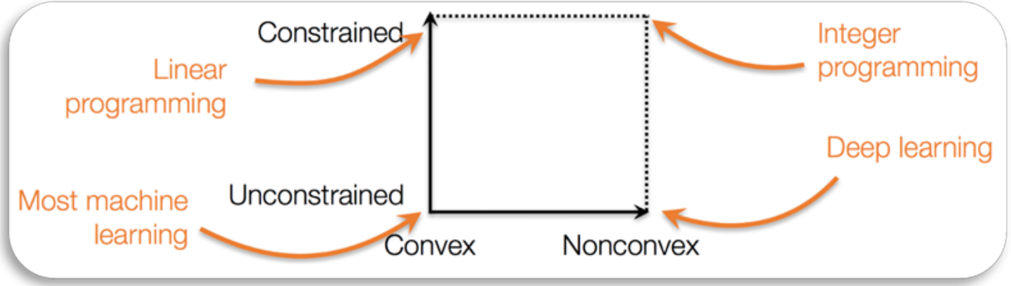
- Unconstrained Optimization: Ω is the entire space R^n
- Linear Optimization: If both the objective and the constraint functions are linear/affine
- Nonlinear Optimization: If the objective/constraints contain general nonlinear functions
- (Mixed) Integer Optimization: If the some variables are restricted to be integral
- Conic Linear Optimization: If both the objective and the constraint functions are linear/affine, but variables in a convex cone.
- Stochastic Optimization: Optimize the expected objective function with random parameters
- Fixed-Point or Min-Max Optimization: Optimization of multiple agents with zero-sum objectives

We present a few optimization examples in this lecture that we would cover through out this course.

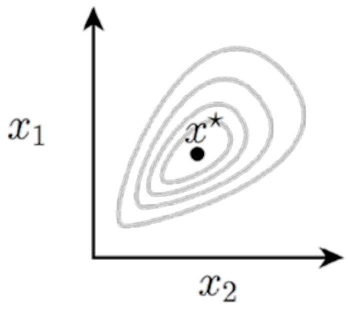
Classes of optimization problems

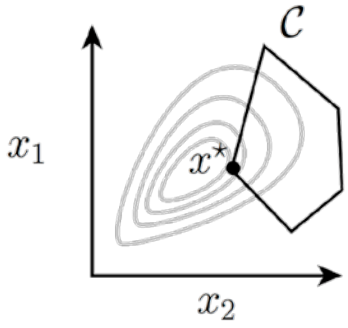
- Types of optimization problems: linear programming, nonlinear programming, integer programming, geometric programming, …





Constrained vs. unconstrained optimization

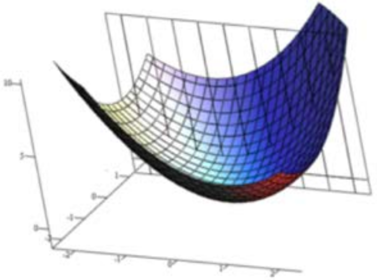


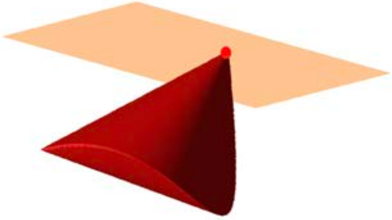


- **Unconstrained optimization:** every point $\boldsymbol{x} \in \mathbb{R}^n$ is feasible, so only focus is on minimizing $f(\boldsymbol{x})$
- **Constrained optimization:** it may be difficult to even *find* a feasible point $\boldsymbol{x} \in \mathcal{C}$

Typically leads to different classes of algorithms

Convex vs. nonconvex optimization





Convex optimization:

1) All local optima are global optima
2) Can be solved in polynomial-time

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

— R. Rockafellar '1993

$f_1(x)$

$f_2(x)$

Deterministic vs. stochastic optimization

- **Stochastic optimization**

$$\text{minimize} \quad f(\boldsymbol{x}) := \mathbb{E}[F(\boldsymbol{x}, \boldsymbol{\xi})] \quad \text{subject to} \quad \boldsymbol{x} \in \mathcal{X}$$

  - f: loss; $\boldsymbol{x}$: parameters; $\boldsymbol{\xi}$: data samples

- **Example:** supervised machine learning (finite-sum problems)

$$\text{minimize} \quad f(\boldsymbol{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{x}) = \frac{1}{n}\sum_{i=1}^{n} \ell(b_i - \boldsymbol{a}_i^T \boldsymbol{x})$$

  - Data observations: $(\boldsymbol{a}_i, b_i) \in \mathbb{R}^d \times \mathbb{R}$; loss function: $\ell : \mathbb{R}^d \to \mathbb{R}$

- **Stochastic gradient:** $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f_{i(k)}(\boldsymbol{x}_k)$

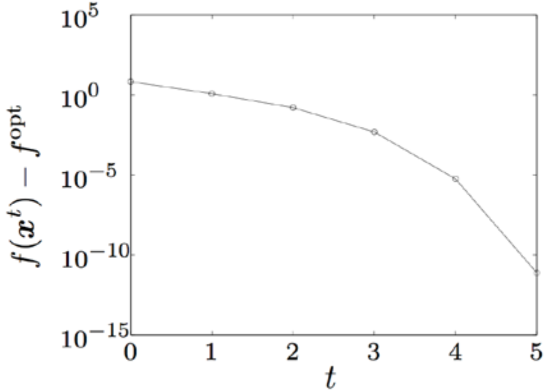
  - $i(k) \in \{1, 2, \ldots, n\}$ uniformly at random; unbiased estimate: $\mathbb{E}[\nabla f_{i(k)}] = \nabla f$

Scaling issues: solvability vs. scalability

- Polynomial-time algorithms might be *useless* in large-scale applications
- **Example:** Newton's method

$$\text{minimize}_{\boldsymbol{x}\in\mathbb{R}^n} \quad f(\boldsymbol{x})$$

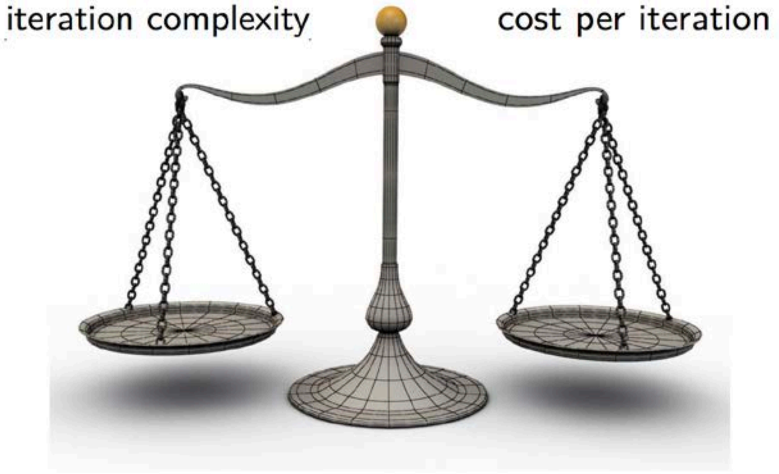
$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t - (\nabla^2 f(\boldsymbol{x}^t))^{-1}\nabla f(\boldsymbol{x}^t)$$



- Attains ϵ accuracy within $\mathcal{O}(\log\log\frac{1}{\epsilon})$ iterations; requires $\nabla^2 f(\boldsymbol{x}) \in \mathbb{R}^{n\times n}$
- A single iteration may last forever; prohibitive storage requirement

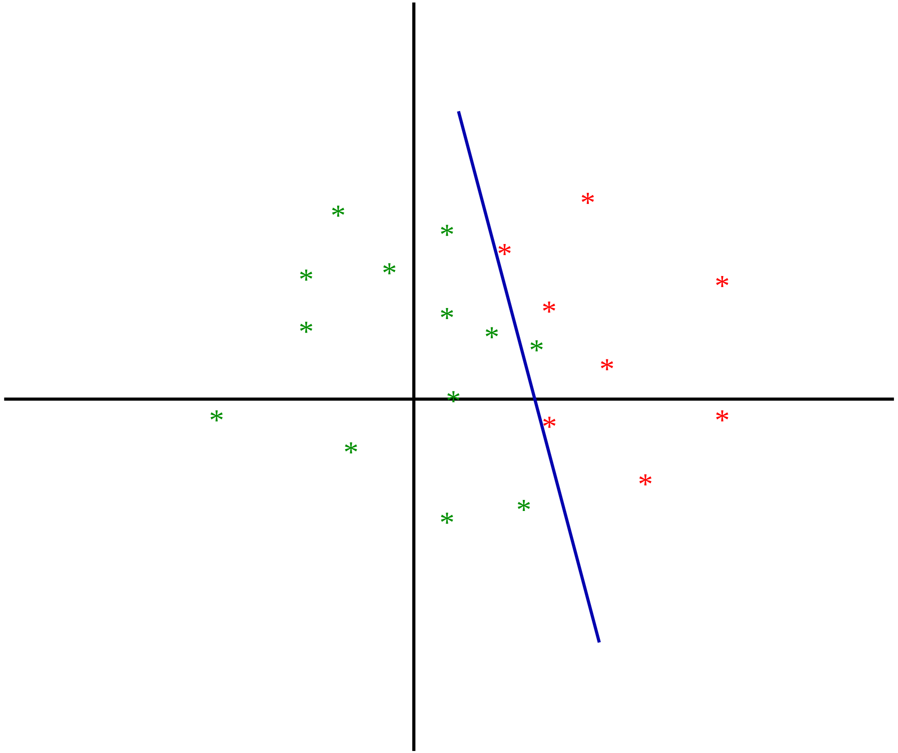
Iteration complexity vs. per-iteration cost

computational cost = iteration complexity (#iterations) x cost per iteration



Large-scale problems call for methods with *cheap iterations*

Linear Classifier: Logistic Regression and Support Vector Machine



Unconstrained Optimization: Logistic Regression I

Given two-class discrimination training data points $\mathbf{a}_i \in R^n$, according to the logistic model, the probability that it's in a class C, say in Red, is represented by a linear/affine function with slope-vector $\mathbf{x}$ and intersect scalar x_0:

$$\frac{e^{\mathbf{a}_i^T\mathbf{x}+x_0}}{1+e^{\mathbf{a}_i^T\mathbf{x}+x_0}}.$$

Thus, for some training data points, we like to determine intercept x_0 and slope vector $\mathbf{x} \in R^n$ such that

$$\frac{e^{\mathbf{a}_i^T\mathbf{x}+x_0}}{1+e^{\mathbf{a}_i^T\mathbf{x}+x_0}} = \begin{cases} 1, & \text{if } \mathbf{a}_i \in C \\ 0, & \text{otherwise} \end{cases}.$$

Then the probability to give a "right classification answer" for all training data points is

$$\left(\prod_{\mathbf{a}_i \in C} \frac{e^{\mathbf{a}_i^T\mathbf{x}+x_0}}{1+e^{\mathbf{a}_i^T\mathbf{x}+x_0}}\right)\left(\prod_{\mathbf{a}_i \notin C} \frac{1}{1+e^{\mathbf{a}_i^T\mathbf{x}+x_0}}\right)$$

Logistic Regression II

Therefore, we like to maximize the probability when deciding intercept x_0 and slope vector $\mathbf{x} \in R^n$

$$\left(\prod_{\mathbf{a}_i \in C} \frac{e^{\mathbf{a}_i^T \mathbf{x} + x_0}}{1 + e^{\mathbf{a}_i^T \mathbf{x} + x_0}}\right)\left(\prod_{\mathbf{a}_i \notin C} \frac{1}{1 + e^{\mathbf{a}_i^T \mathbf{x} + x_0}}\right) = \left(\prod_{\mathbf{a}_i \in C} \frac{1}{1 + e^{-\mathbf{a}_i^T \mathbf{x} - x_0}}\right)\left(\prod_{\mathbf{a}_i \notin C} \frac{1}{1 + e^{\mathbf{a}_i^T \mathbf{x} + x_0}}\right),$$

which is equivalently to maximize

$$-\left(\sum_{\mathbf{a}_i \in C} \ln(1 + e^{-\mathbf{a}_i^T \mathbf{x} - x_0})\right) - \left(\sum_{\mathbf{a}_i \notin C} \ln(1 + e^{\mathbf{a}_i^T \mathbf{x} + x_0})\right).$$

Or

$$\min_{x_0, \mathbf{x}} \quad \left(\sum_{\mathbf{a}_i \in C} \ln(1 + e^{-\mathbf{a}_i^T \mathbf{x} - x_0})\right) + \left(\sum_{\mathbf{a}_i \notin C} \ln(1 + e^{\mathbf{a}_i^T \mathbf{x} + x_0})\right).$$

This is an unconstrained optimization problem, where the objective is a convex function of decision variables: intercept x_0 and slope vector $\mathbf{x} \in R^n$.

Data Classification: Supporting Vector Machine

Similar to logistic regression, suppose we have two-class discrimination training data points. Another powerful discrimination method is the Supporting Vector Machine (SVM).

Let the first class, say in Red, data points i be denoted by $\mathbf{a}_i \in R^d$, $i = 1, ..., n_1$ and the second class data points j be denoted by $\mathbf{b}_j \in R^d$, $j = 1, ..., n_2$. We like to find a hyperplane to separate the two classes:

$$
\begin{array}{rl}
\text{minimize} & \beta + \mu \|\mathbf{x}\|^2 \\
\text{subject to} & \mathbf{a}_i^T \mathbf{x} + x_0 + \beta \geq 1, \ \forall i, \\
 & \mathbf{b}_j^T \mathbf{x} + x_0 - \beta \leq -1, \ \forall j, \\
 & \beta \geq 0,
\end{array}
$$

where μ is a fixed positive regularization parameter.

This is a constrained quadratic program (QP). If $\mu = 0$, then it is a linear program (LP)!

Graph Realization and Sensor Network Localization

Given a graph $G = (V, E)$ and sets of non–negative weights, say $\{d_{ij} : (i, j) \in E\}$, the goal is to compute a realization of G in the Euclidean space $\mathbf{R}^d$ for a given low dimension d, where the distance information is preserved.

More precisely: given anchors $\mathbf{a}_k \in \mathbf{R}^d$, $d_{ij} \in N_x$, and $\hat{d}_{kj} \in N_a$, find $\mathbf{x}_i \in \mathbf{R}^d$ such that

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = d_{ij}^2, \ \forall\, (i,j) \in N_x,\ i < j,$$
$$\|\mathbf{a}_k - \mathbf{x}_j\|^2 = \hat{d}_{kj}^2, \ \forall\, (k,j) \in N_a,$$

This is a set of Quadratic Equations.

Does the system have a localization or realization of all $\mathbf{x}_j$'s? Is the localization unique? Is there a certification for the solution to make it reliable or trustworthy? Is the system partially localizable with certification?

It can be relaxed to SOCP (change "=" to "$\leq$") or SDP.

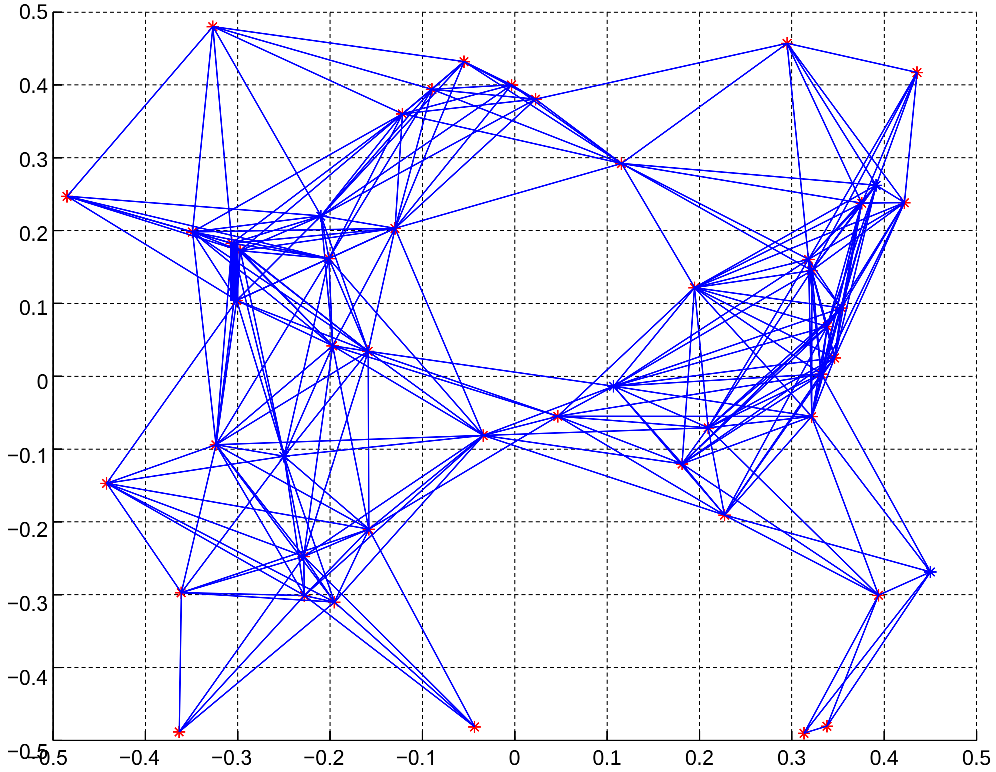


Figure 4: 50-node 2-D Sensor Localization.

Matrix Representation of SNL and SDP Relaxation

Let $X = [\mathbf{x}_1\, \mathbf{x}_2 \ldots \mathbf{x}_n]$ be the $d \times n$ matrix that needs to be determined and $\mathbf{e}_j$ be the vector of all zero except 1 at the jth position. Then

$$\mathbf{x}_i - \mathbf{x}_j = X(\mathbf{e}_i - \mathbf{e}_j) \quad \text{and} \quad \mathbf{a}_k - \mathbf{x}_j = [I \;\; X](\mathbf{a}_k; -\mathbf{e}_j)$$

so that

$$\|\mathbf{x}_i - \mathbf{x}_j\|^2 = (\mathbf{e}_i - \mathbf{e}_j)^T X^T X(\mathbf{e}_i - \mathbf{e}_j)$$

$$\|\mathbf{a}_k - \mathbf{x}_j\|^2 = (\mathbf{a}_k; -\mathbf{e}_j)^T [I \;\; X]^T [I \;\; X](\mathbf{a}_k; -\mathbf{e}_j) =$$

$$(\mathbf{a}_k; -\mathbf{e}_j)^T \begin{pmatrix} I & X \\ X^T & X^T X \end{pmatrix} (\mathbf{a}_k; -\mathbf{e}_j).$$

Or, equivalently,

$$
\begin{aligned}
&(\mathbf{e}_i - \mathbf{e}_j)^T Y (\mathbf{e}_i - \mathbf{e}_j) = d_{ij}^2, \ \forall\, i, j \in N_x, \ i < j, \\
&(\mathbf{a}_k; -\mathbf{e}_j)^T \begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} (\mathbf{a}_k; -\mathbf{e}_j) = \hat{d}_{kj}^2, \ \forall\, k, j \in N_a, \\
&Y = X^T X.
\end{aligned}
$$

Relax $Y = X^T X$ to $Y \succcurlyeq X^T X$, which is equivalent to matrix inequality:

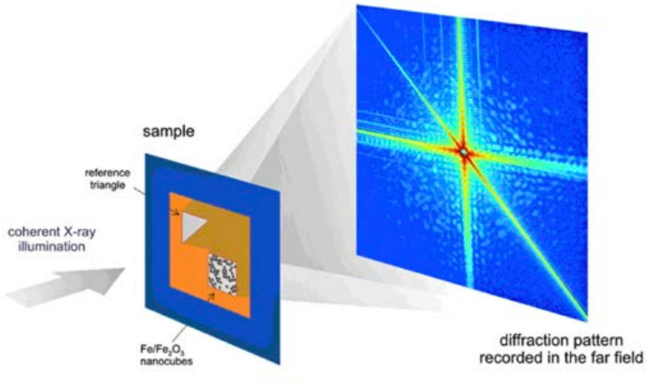
$$
\begin{pmatrix} I & X \\ X^T & Y \end{pmatrix} \succeq \mathbf{0}.
$$

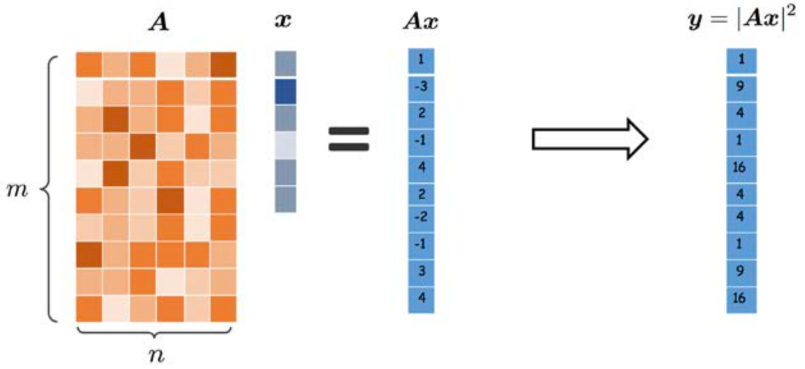
This matrix has rank at least d; if it's d, then $Y = X^T X$, and the converse is also true.

The problem is now an SDP problem.

Quadratic model

- **Phase retrieval:** recover signal from intensity (missing phase)





- Recover $\boldsymbol{z}^\natural \in \mathbb{R}^n$ from m random quadratic measurements

$$\text{find} \quad \boldsymbol{z} \qquad \text{subject to} \quad y_r = |\langle \boldsymbol{a}_r, \boldsymbol{z} \rangle|^2, \quad r = 1, 2, \ldots, m$$

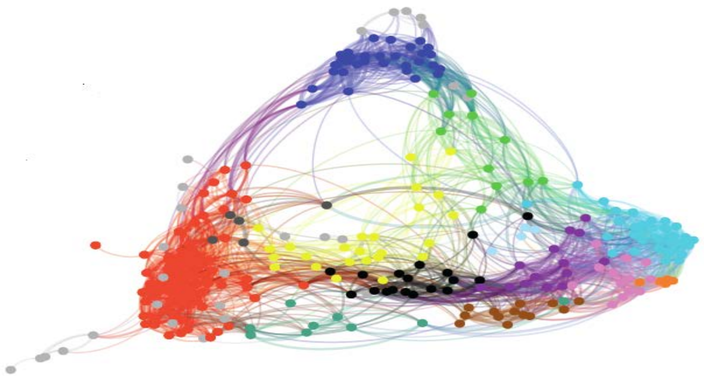
Low-rank model

Fig. credit: Candès

- Given partial samples Ω of a low-rank matrix $M^\natural$, fill in missing entries

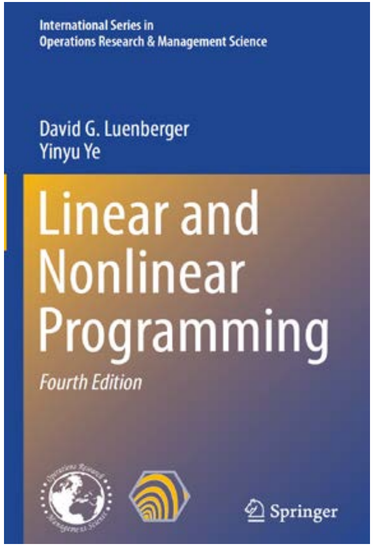
$$\underset{\boldsymbol{M} \in \mathbb{C}^{m \times n}}{\text{minimize}} \quad \text{rank}(\boldsymbol{M}) \quad \text{subject to} \quad Y_{i,k} = M_{i,k}, \quad (i,k) \in \Omega$$

Topics and Grading



Reference Books

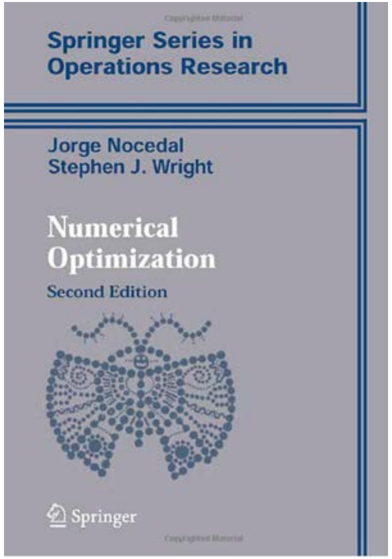
- **Main topics:** linear programing, unconstrained problems, constrained minmization



Linear and Nonlinear Programing, by David G. Luenberger and Yinyu Ye, Springer, 2015.

Reference Books

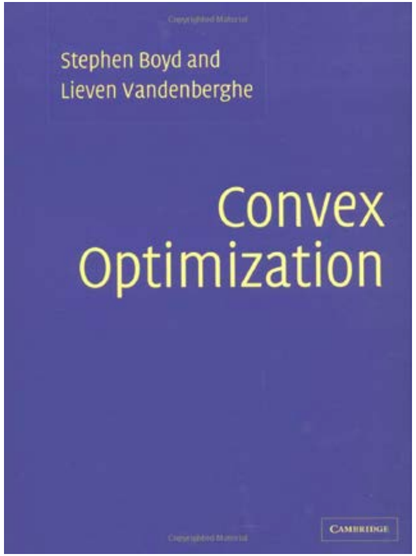
- **Main topics:** Newton method, interior-point methods, quasi-Newton methods



Numerical Optimization, by J. Nocedal and S. Wright, Springer-Verlag, 2006.

Reference Books

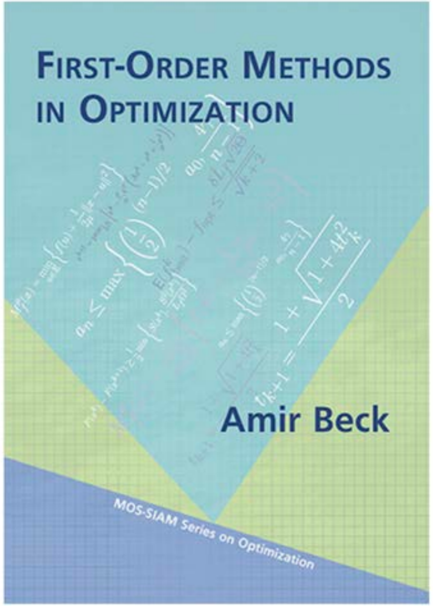
- **Main topics:** convex sets, convex functions, convex problems, Lagrange duality and KKT conditions, disciplined convex programming



Convex Optimization, by S. Boyd and L. Vandenberghe, Cambridge University Press, 2003.

Reference Books

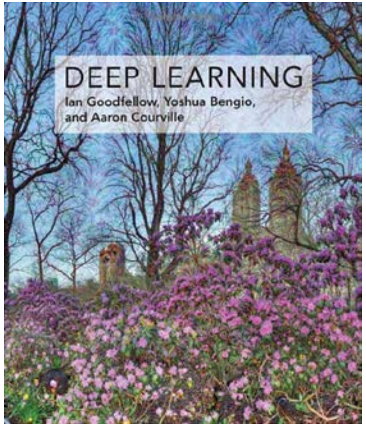
- **Main topics:** gradient methods, subgradient methods, proximal methods



First-order Methods in Optimization, by A. Beck, MOS-SIAM Series on Optimization, 2017.

Machine learning for optimization

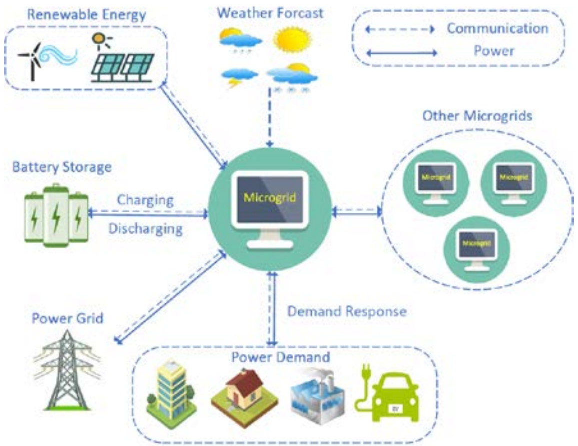
- **Main topics:** sparse optimization (deep neural networks), mixed integer nonlinear programming (imitation learning), nonconvex optimization (statistical learning), multiple objective optimization (active learning)



Deep Learning, by I. Goodfellow, Y. Bengio and A. Courville, MIT Press, 2016.

Applications in Smart Grid

- **Main topics:** Sparse and low rank optimization (Optimal power flow), mixed integer linear/nonlinear programming (Electric vehicle charging/discharging, Demand response, PMU placement)



[1] Ye Shi, Hoang D. Tuan, Hoang Tuy and Steven W. Su, "Global Optimization for Optimal Power Flow over Transmission Networks", Journal of Global Optimization, vol. 69, pp. 745-760, 2017.

[2] Ye Shi, Hoang D. Tuan, Andrey V. Savkin, Trung Q. Duong and H. Vincent Poor, "Model Predictive Control for Smart Grids with Multiple Electric-Vehicle Charging Stations", IEEE Transaction on Smart Grid, vol. 10, pp. 2127-2136, 2019.

[3] Ye Shi, Hoang D. Tuan, Trung Q. Duong, H. Vincent Poor and Andrey V. Savkin, "PMU Placement Optimization for Efficient State Estimation in Smart Grid", IEEE Journal on Selected Areas in Communications, vol. 38, no. 1, pp. 71-83, 2020.

Grading

- **Homework (20%):** 4-6 homework assignments
- **Quiz (10%):** several times in class
- **Course project (20%):**
  - either individually or in groups of two/three
  - list of topics; final report & slides
- **Mid-term Exam (20%)**
- **Final Exam (30%)**

Grading

- **Regrade Requests**
- If you feel you deserved a better grade on an assignment, you may submit a regrade request by email within **3 days** of the grade release. Your request should briefly summarize why you feel the original grade was unfair. Your TA will re-evaluate your assignment as soon as possible, and then issue a decision.
- **Late Policy**
- All students have 4 free late days for the quarter.
- You may use up to 2 late days per assignment with no penalty.
- You may use late days for the assignments, project.
- Once you have exhausted your free late days, we will deduct a late penalty of 25% per additional late day.
  - For example: you submit A1 one day late, submit A2 three days late, and submit A3 two days late. You receive no penalty for A1, and exhaust one of your free late days. For A2 the first two late days exhaust two of your free late days; the third day late incurs a 25% penalty. For A2 the first late day exhausts your final free late day; the second late day incurs a 25% penalty.

Course information

- **Instructor:** Ye Shi (http://faculty.sist.shanghaitech.edu.cn/faculty/shiye)
  - Email: shiye@shanghaitech.edu.cn
  - Office location: Room 1A-404A, SIST Building
  - Office hours: Office hours: Wednesday 14: 30-15: 30 (or by appointments)
- **TAs:**
  - Haixiang Sun (sunhx@shanghaitech.edu.cn)
  - Office location: Room 1A-405, SIST Building
  - Office hours: Office hours: Thursday 19: 00-20: 00 (or by appointments)

Course information

- Use WeChat as the main mode of electronic communication; please post (and answer) questions there!

该二维码7天内(2月20日前)有效，重新进入将更新

- Course website: BlackBoard

https://elearning.shanghaitech.edu.cn:8443/webapps/blackboard/execute/modulepage/view?course_id=_2778_1&cmp_tab_id=_6738_1&editMode=true&mode=cpview

Thanks