Chapter 1: Introduction

Chao Wang

SUSTech

Outline

1. Information of This Course
   - basic info.
   - main contents
   - teaching hours

2. Overview of OR & Optimization
   - examples
   - applications

Information of This Course

Basic Information

- Lecturer: Dr. Chao Wang
  - Email: wangc6@sustech.edu.cn
  - Office: School of Business Building 333
  - Office hours: Thursday 16:00 – 18:00 or by appointment
- Tutors:
  - Mr. Xinyuan Huang (Email: 12332891@mail.sustech.edu.cn)
  - Mr. Yangzhou Chen (Email: 12331110@mail.sustech.edu.cn)
  - Ms. Yujie Qin (Email: 12432277@mail.sustech.edu.cn)
  - Mr. Chengyu Wang (Email: 12443002@mail.sustech.edu.cn)
- Lecture Hours:
  - Tuesday (only even week) 10:20 - 12:10
  - Thursday 10:20 - 12:10
- Venue: FTB 107
- QQ group: 1030851115

Pre-requisites

- MA107 Advanced Linear Algebra I or MA113 Linear Algebra

References

- Main references:
  - [1] Slides for this course - This course will follow closely with the lecture slides which can be downloaded from the course website.
  - [2] Lily Pan, Operations Research, The Chinese University of Hong Kong (lecture note) 2021.
  - [3] Haoyan Liu, Jiang Hu, Yongfeng Li, Zaiwen Wen. Optimization: Modeling, algorithm and theory, (chinese), 2021.
- Other references
  - [5] Luenberger, D. G., & Ye, Y. Linear and nonlinear programming (3rd Edition). Reading, MA: Addison-wesley. 2008.
  - [6] Baomin Chen. Optimization theory and algorithm, Tsinghua university press (chinese)
  - [7] Juraj Stacho Introduction to operations research, Columbia University, lecture notes (link)
  - [8] Fred S. Hillier, Gerald J. Lieberman, Introduction to Operations Research,10th Edition,McFraw-Hill Education,2015.

Main contents

- Chapter 1. Introduction
  (ref: [ch1,2] & [ch1&ch3,3])
- Chapter 2. Basic Concepts
  (ref: [ch2,2] & [ch2,3])
- Chapter 3. Canonical Problem Forms
  (ref: [ch4,3])
- Chapter 4. Linear Programming & Simplex Method
  (ref: [ch1&ch4&ch5,2])
- Chapter 5. Duality & Sensitivity Analysis
  (ref: [ch7&ch8,2])
- Chapter 6. Optimality Condition
  (ref: [ch5,3])
- Chapter 7. Gradient-based Algorithms
  (ref: [ch6,3])

Assessment

- A project presentation (10%)
- Six assignments (20%)
- A two-hour written midterm examination (20%)
- A two-hour written final examination (50%)

Remark: The assignment is due by 00:00 am on the due date. Partially or wholly copied assignments will be penalized and/or reported as plagiarism.

Teaching hours

- Lectures: The lectures will be used to introduce concepts and develop theories and algorithms.
- Tutorials: To facilitate the understanding of the course content, there will be 10 tutorials scheduled in this term starting from the next week. Recitations will cover problems that are similar to the ones on the current problem set. You will be asked to work out solutions to the problems.
- Midterm: There will be one quiz (Midterm test). The tentative date for the test is Apr. 10. If in any doubt, check with the instructor.
- Final: There will be one final exam arranged by the university. The tentative date will be arranged in June.

Tentative Teaching Plan

L=Lecture, T=Tutorial, P=Project presentation date, HW= Homework due date

Week	Date	Content	Date	Content
1			Feb. 20	L1+L2
2	Feb. 25	L3+T1	Feb. 27	L4+L5
3			Mar. 6	L6+ T2
4	Mar. 11	L7+L8	Mar. 13	L9+T3+HW1
5			**Mar. 23**	L10+L11
6	Mar. 25	L12+T4	Mar. 27	L13+L14+HW2
7			Apr. 3	L15+T5
8	Apr. 8	L16+L17	Apr. 10	Midterm+ HW3
9			Apr. 17	L18+T6
10	Apr. 22	L19+L20	Apri. 28	L21+T7+HW4
11			May 1	Holiday
12	May 6	L22+L23	May 8	L24+T8+HW5
13			May 15	L25+ L26
14	May 20	L27+T9	May 22	L28+L29 +HW6
15			May 29	L30+T10
16	Jun. 3	P	Jun. 5	P

Overview of OR & Optimization

What is OR?

- **Operations research (OR)** is an analytical method of problem-solving and decision-making that is useful in the management of organizations.
- In operations research, problems are broken down into basic components and then solved in defined steps by mathematical analysis.
- OR=Operational Research = Operations Research
  $\approx$ Management Science $\approx$ Industrial Engineering
- **Example**: Assignment problem, transportation problem, network problem, game theory, linear programming...

What is Optimization?

- **Optimization** models the goal of solving a problem in the optimal way.
- Optimization is an essential tool in life, business, and engineer.
- In general, it can be formulated as

$$\begin{array}{ll} \min & f_0(x) \\ \text{s.t.,} & f_i(x) \leq b_i, i = 1, \ldots, m. \end{array}$$

  - $x = (x_1, \ldots, x_n)$: optimization variables 决策变量
  - $f_0 : \mathbb{R}^n \to \mathbb{R}$: objective function
  - $f_i : \mathbb{R}^n \to \mathbb{R}, i = 1, \ldots, m$: constraint functions
- **Example**: linear/nonlinear optimization, convex/non-convex optimization

Optimization problem

- The general form of optimization:

$$\begin{array}{rl} \text{Min} & f(x) \\ \text{Subject to} & x \in \Omega \end{array}$$

- Suppose $x \in \mathbb{R}^n, \Omega$ is called the feasible set. 约束集合(可行域).
- If $\Omega = \mathbb{R}^n$, then the problem is called unconstrained. 无约束.
- Otherwise, the problem is called constrained. 约束.
- We can write any constrained problem in the unconstrained form

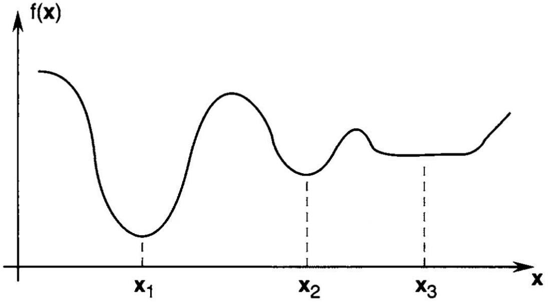
$$\min f(x) + \iota_\Omega(x),$$

where the indicator function

$$\iota_\Omega(x) = \begin{cases} 0, x \in \Omega, \\ +\infty, x \notin \Omega. \end{cases}$$

Types of solutions

- x^* is a local minimizer if there is $\epsilon > 0$ such that $f(x) \geq f(x^*)$ for all $x \in \Omega$ and $\|x - x^*\| < \epsilon$.
- x^* is a global minimizer if $f(x) \geq f(x^*)$ for all $x \in \Omega\{x^*\}$
- If " $\geq$ " is replaced with "$>$", then they are strict local minimizer and strict global minimizer, respectively.



x_1: strict global minimizer; x_2: strict local minimizer; x_3: local minimizer

Type of Problems — Linear Programming

线性.

$$\text{Max/Min} \qquad z = \sum_{j=1}^{n} c_j x_j$$

$$\text{Subject to} \qquad \sum_{j=1}^{n} a_{ij} x_j \begin{pmatrix} \geq \\ \leq \\ = \end{pmatrix} b_i, \qquad i = 1, 2, \ldots, m$$

$$x_j \begin{pmatrix} \geq 0 \\ \leq 0 \\ \text{unrestricted} \end{pmatrix}, \qquad j = 1, 2, \ldots, n.$$

Terminology 目标方程.

- objective function: $\sum_{j=1}^{n} c_j x_j$
- decision variables: $x_1, x_2, \ldots, x_i, \ldots$
- constraint: e.g.: $\sum_{j=1}^{n} a_{ij} x_j \leq b_i$
- variable domains: e.g.: $x_j \geq 0$

n 商品 $\rightarrow$ a_{ij} j种商品需要i种材料个数

m 材料 $\rightarrow$ i种有b_i个

Example: A production problem

A firm produces n different goods using m different raw materials. Let b_i, $i = 1, 2, \ldots, m$, be the available amount of the ith raw material. The jth good, $j = 1, \ldots, n$, requires a_{ij} units of the ith material and results in a revenue of c_j per unit produced. The firm faces the problem of deciding how much of each good to produce in order to maximize its total revenue.

Formulation:

Let x_j, $j = 1, \ldots, n$, be the amount of jth good.

$$\begin{array}{lll} \text{Maximize} & c_1x_1 + \cdots + c_nx_n & \\ \text{subject to} & a_{i1}x_1 + \cdots + a_{in}x_n \leq b_i, & i = 1, \ldots, m, \\ & x_j \geq 0, \rightarrow \text{现实约束.} & j = 1, \ldots, n. \end{array} \tag{1}$$

$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1$

$\vdots$

$x_j \geq 0.$

m 食品 F_i b_i 价格.
n 营养 N_j c_j 最小需求.

Example: A diet problem

There are m different types of food, $F_1, \ldots, F_m$, that supply varying quantities of the n nutrients, $N_1, \ldots, N_n$, that are essential to good health. Let c_j be the minimum daily requirement of nutrient, N_j. Let b_i be the price per unit of food, F_i. Let a_{ij} be the amount of nutrient N_j contained in one unit of food F_i. The problem is to supply the required nutrients at a minimum cost.

x_i $i=1,\ldots m$ $\sum_{i=1}^{m} a_{ij} x_i \geq c_j$ $j=1,\ldots,n.$

$\min\ b_1x_1 + \cdots + b_m x_m$

Formulation:

Let x_i be the number of units of food F_i to be purchased per day. The problem can be formulated as follows:

$$
\begin{array}{rlll}
\text{Minimize} & b_1 x_1 + \cdots + b_m x_m & & \\
\text{subject to} & a_{1j} x_1 + \cdots + a_{mj} x_m \geq c_j, & j = 1, \ldots, n, & (2) \\
& x_i \geq 0, & i = 1, \ldots, m. &
\end{array}
$$

Type of problems — Linear programming

类似 数学建模.

P_i (I) 港口. $\to$ S_i (I) 生产货物

M_j (J) 市场 $\to$ r_j (J) 接收

Example: A Transportation Problem

There are I ports, or production plants, $P_1, \ldots, P_I$, that supply a certain commodity, and there are J markets, $M_1, \ldots, M_J$, to which this commodity must be shipped. Port P_i possesses an amount s_i of the commodity ($i = 1, 2, \ldots, I$), and market M_j must receive the amount of r_j of the commodity ($i = 1, \ldots, J$). Let b_{ij} be the cost of transporting one unit of the commodity from port P_i to market M_j. The problem is to meet the market requirements at minimum transportation cost.

b_{ij} $P_i \to M_j$.

Formulation:

Let x_{ij} be the quantity of the commodity shipped from port P_i to market M_j. The problem can be formulated as follows:

x_{ij} $P_i \to M_j$.

$$
\begin{aligned}
&\text{Minimize} && \sum_{i=1}^{I}\sum_{j=1}^{J} x_{ij} b_{ij} && \\
&\text{subject to} && \sum_{j=1}^{J} x_{ij} \le s_i, && i = 1, \ldots, I, \\
& && \sum_{i=1}^{I} x_{ij} \ge r_j, && j = 1, \ldots, J, \\
& && x_{ij} \ge 0, && i = 1, \ldots, I \text{ and } j = 1, \ldots, J.
\end{aligned} \tag{3}
$$

Cost: $\sum_{i=1}^{I}\sum_{j=1}^{J} x_{ij} b_{ij}$

$\sum_{i=1}^{I} x_{ij} \le S_i$

$\sum_{j=1}^{J} x_{ij} \le r_j$

Properties of Linear Programming

①. 可加性

②. 数乘

1. There is a unique objective function.
唯一
2. The objective function is linear and all constraints are linear equalities or inequalities.
3. The coefficients of the decision variables in the objective function and each constraint are constant.
4. The decision variables are permitted to assume fractional as well as integer values.

Type of Problems — Nonlinear programming

- **multiobjective** or **goal programs**:Problems with more than one objective function.
- **nonlinear program**: Any optimization problem that fails to satisfy Property 2
- **dynamic program**: the coefficients is time dependent. And the problem is defined in a certain class
- **stochastic program**: the coefficients are not constant but instead are probabilistic
- **integer program** or a **mixed-integer program**: one or more of the decision variables are restricted to integer values

Type of problems — Integer Programming

Example: A local manufacturing plant runs 24 hours a day, 7 days a week. During various times throughout the day, different numbers of workers are needed to run the various machines. Below are the minimum number of people needed to safely run the plant during various times.

Times	Employees Needed
12 AM - 4 AM	8
4 AM - 8 AM	9
8 AM - 12 PM	15
12 PM - 4PM	14
4 PM - 8 PM	13
8 PM - 12 AM	11

A worker can only start working at the start of each 4-hour period, and should work for 8 hours straight. What is the minimum number of people required daily to safely run the plant?

Type of problems — Integer Programming

Formulation: To model this linearly, the key is to realize that our decision variables are not how many people work during a given 4-hour period but *how many start at the beginning of each 4-hour period.* In this way, we can keep track of who is working when.

$$x_i = \text{number of employees beginning their shift at the start of the } i\text{th 4-hour period, } i = 1, 2, \ldots, 6.$$

$$\begin{array}{llr}
\text{Minimize} & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 & \\
\text{subject to} & x_1 + x_6 \geq 8 & \text{(12AM - 4AM)} \\
& x_1 + x_2 \geq 9 & \text{(4AM - 8AM)} \\
& x_2 + x_3 \geq 15 & \text{(8AM - 12PM)} \\
& x_3 + x_4 \geq 14 & \text{(12PM - 4PM)} \\
& x_4 + x_5 \geq 13 & \text{(4PM - 8PM)} \\
& x_5 + x_6 \geq 11 & \text{(8PM - 12AM)} \\
& x_i \geq 0, \text{integer}, \quad i = 1, 2, \ldots, 6. &
\end{array} \tag{4}$$

Example: Polynomial Fitting

- Suppose the observed data is $y_i \in \mathbb{R}$ for each time point $t_i, i = 1, 2, \cdots, N$, respectively. The underlying assumption it is that there is some function of time $f : \mathbb{R} \to \mathbb{R}$ such that $y_i = f(t_i), i = 1, 2, \cdots, N$.
- How to estimate it by a polynomial of a fixed degree, say n : $p(t) = x_0 + x_1 t + x_2 t^2 + \cdots + x_n t^n$.
- How to determine the coefficients that "best" fit the data? - If were possible to exactly fit the data, then there would exist a value for the coefficient.
- If $n \ll N$, it cannot expect to fit the data perfectly and so there will be errors. It is wish to minimize the sum of the squares of the errors in the fit:

Example: Polynomial Fitting

This minimization problem has the form $\min_x \frac{1}{2}\|Vx - y\|_2^2$, where

$$y = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{pmatrix}, x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \text{ and } V = \begin{bmatrix} 1 & t_1 & t_1^2 & \cdots & t_1^n \\ 1 & t_2 & t_2^2 & \cdots & t_2^n \\ \vdots & & & & \\ 1 & t_N & t_N^2 & \cdots & t_N^n \end{bmatrix}.$$

- least squares: a standard approach in regression analysis to approximate the solution of overdetermined systems (sets of equations in which there are more equations than unknowns)

Type of Problems — Variants of least squares

Example: Structured Optimization

ℓ_1 norm regularized linear regression

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \tau\|x\|_1$$

- $\|x\|_1 = \sum_i |x_i|$ approaching $\|x\|_0$ (number of nonzero entries)
- τ is a fixed constant called regularization parameter.
- this is called LASSO in statistics, Compressed sensing in signal processing
- a non-smoothed problem but convex (will explain in the next chapter)

Type of Problems — Variants of least squares

- Ridge regression/Tikhonov regularization

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \mu\|x\|_2^2$$

- sparse regularization

$$\min_x \frac{1}{2}\|Ax - b\|_2^2 + \mu\|x\|_1$$

- Lasso/Basis pursuit

$$\min_x \|x\|_1, \text{ s.t. } \|Ax - b\|_2 \leq \epsilon$$

or

$$\min_x \|Ax - b\|_2, \text{ s.t. } \|x\|_1 \leq \sigma$$

or even under a different norm

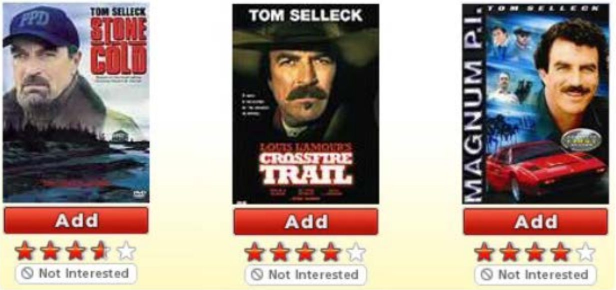
$$\min_x \|Ax - b\|_1, \text{ s.t. } \|x\|_1 \leq \sigma$$

Some applications — recommendation

Example: Recommendation System: Netflix

- Rating Movies

- Get recommendations



Example: Recommendation System: Netflix Prize

- 480,189 users rate 17,770 movies
- over 100 million ratings (1.2%)
- Data: incomplete rating matrix
- Decision: predict the missing ratings
- 1 million dollar prize

Example: Recommendation System: Netflix Prize

	Movie				
User	?	9	5	6	8
	8	8	4	9	?
	1	3	?	5	4
	8	?	10	4	1

- Rating matrix is believed to be low rank
-
$$\min_X \operatorname{rank}(X) \quad s.t. \quad X_{ij} = M_{ij}, \forall (i,j) \in \Omega$$
$$Or \quad \min_X \|X\|_* \quad s.t. \quad X_{ij} = M_{ij}, \forall (i,j) \in \Omega$$
where $\|X\|_* := \sum \sigma_i(X)$ is the nuclear norm of matrix X, and $\sigma_i(X)$ is the i-th singular value of X

Some applications — machine learning

Many problems in ML can be written as

- linear regression: $\min_{x\in\mathcal{W}} \sum_{i=1}^{N} \frac{1}{2}\left\|a_i^\top x - b_i\right\|_2^2 + \mu\|x\|_1$
- logistic regression: $\min_{x\in\mathcal{W}} \frac{1}{N}\sum_{i=1}^{N} \log\left(1+\exp\left(-b_i a_i^\top x\right)\right) + \mu\|x\|_1$
- general formulation: $\min_{w\in\mathcal{W}} \sum_{i=1}^{N} \ell\left(h\left(x, a_i\right), b_i\right) + \mu r(x)$

- The pairs (a_i, b_i) are given data, b_i is the label of the data point a_i
- $\ell(\cdot)$: measures how model fit for data points (avoids under-fitting)
- $r(x)$: regularization term (avoids over-fitting)
- $h(x, a)$: linear function or models constructed from deep neural networks

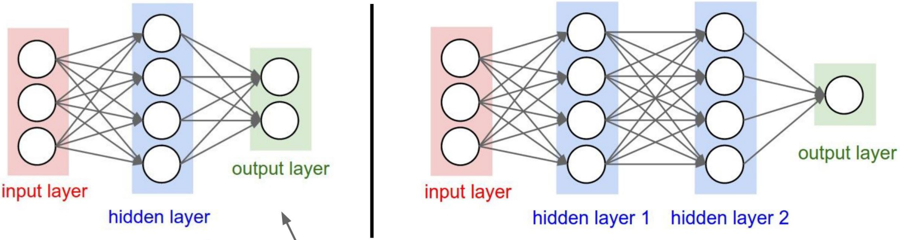
Some applications — machine learning

Given:

- Some dataset of (x, y)
- a score function:
  - Linear: $s = f(x; W) \stackrel{e.g}{=} Wx$
  - 2-layer neural network: $s = f(x; W) \stackrel{e.g}{=} W_2 \max(0, W_1 x)$
- a loss function:
  - Softmax: $L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$
  - SVM $L_i = \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$
  - Full loss: $L = \frac{1}{N} \sum_{i=1}^{N} L_i + R(W)$

Some applications — machine learning

Neural networks: Architectures



"2-layer Neural Net", or
"1-hidden-layer Neural Net"

"3-layer Neural Net", or
"2-hidden-layer Neural Net"

"Fully-connected" layers

Some applications — machine learning

Convolutional network (AlexNet)

卷积神经网络

input image

weights

loss

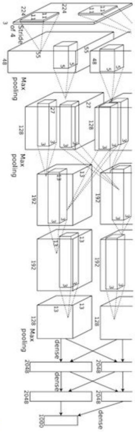


Figure copyright Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, 2012. Reproduced with permission.

Some applications — image reconstruction

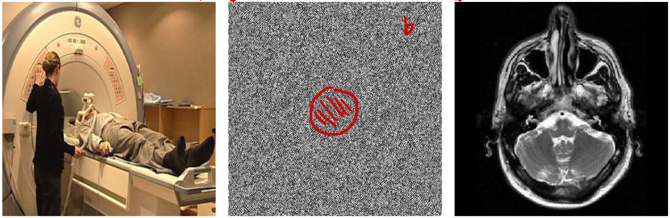
Example: MRI: Magnetic Resonance Imaging

$Au=b$

$\boxed{\quad A \quad}$

$I u = I b.$

$f \qquad f^{-1}$



(a) MRI Scan (b) Fourier Coefficients (c) Image

Is it possible to cut the scan time into half?

Some applications — image reconstruction

Example: MRI: Magnetic Resonance Imaging

- MR images often have sparse sparse representations under some wavelet transform Φ
- Solve

$$\min_u \|\Phi u\|_1 + \frac{\mu}{2}\|Ru - b\|^2$$

R : partial discrete Fourier transform

(a) full sampling

(b) 39% sampling,

Summary

- Optimization problems underlie nearly everything we do in Machine Learning and Statistics. In other courses, you learn how to:

translate  into $P : \min_{x \in D} f(x)$

Conceptual idea *Optimization problem*

- This course:
  - Model: How to simplify P into a simpler form?
  - Algorithm: How to solve P?
  - Analysis: why this is a good skill to have

Recall: Main contents

- Chapter 1. Introduction (overview)
- Chapter 2. Basic Concepts (warm-up)
- Chapter 3. Canonical Problem Forms (warm-up)
- Chapter 4. Linear Programming & Simplex Method (model & algorithm)
- Chapter 5. Duality & Sensitivity Analysis (analysis)
- Chapter 6. Optimality Condition (analysis)
- Chapter 7. Gradient-based Algorithms (algorithm)