Lecture 2

Table of Contents

为什么要学习Optimization?

- 在机器学习中，训练过程通常表示成为优化问题。
  - 如给定数据集D， 寻找参数W
- 特例（Exception) 如KNN, Bayesian Learning

对于大一点的数据

我们可能会遇到各种各样的效率问题，出现的原因有哪些？

- 时间：
  - optimization method e.x. · newton method > 消耗大
    · gradient decent
  - Data Structure : Matrix/Vector ✓
- 内存：
  - may not prefer $O(N^x)$, $x \in 2, 3 \ldots k$
  - sparsity

小结

- 掌握灵活使用各类优化方法很重要。为了达到这个目的：
  - 需要能够识别出问题的类型
  - 需要懂得各种优化方法之间的区别
  - 需要懂得每种优化方法更深入的细节

Convex Optimization

- 考虑以下优化问题：

$$\min_{w \in C} f(w)$$

$C \in \mathbb{R}^d$

- 这个问题为Convex optimization问题，当：
  - 集合$\boxed{C\text{为 convex set}}$
  - 函数$\boxed{f\text{为 convex function}}$

Convex Optimization

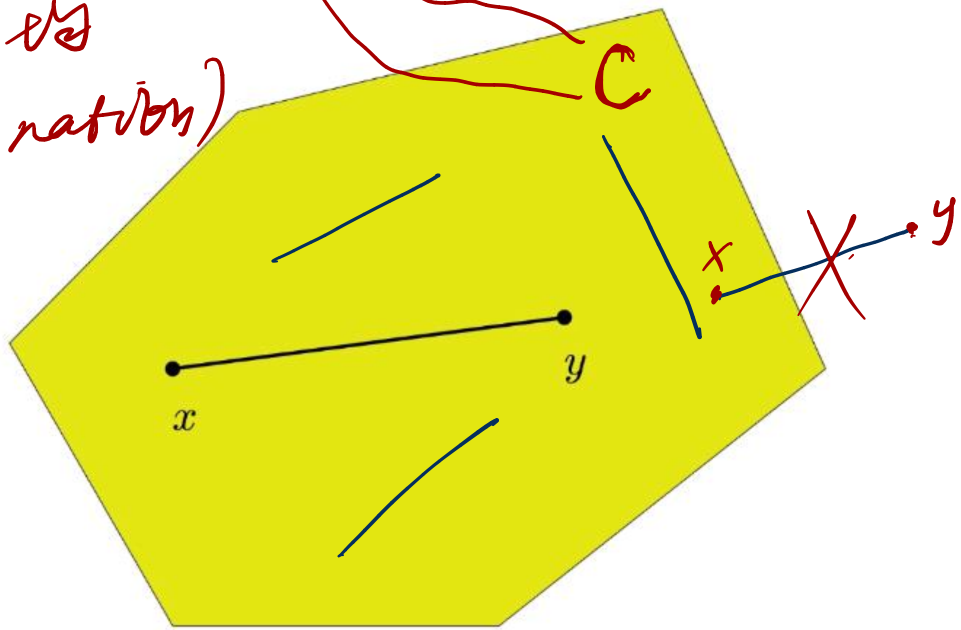
- **Key Property:** 所有的局部最优解为全局最优解
- **Convexity is a good indicator of tractability**
  - 最小化凸函数通常比较简单
  - 最小化非凸函数通常比较难

Convex Set

Convex Set (凸集)

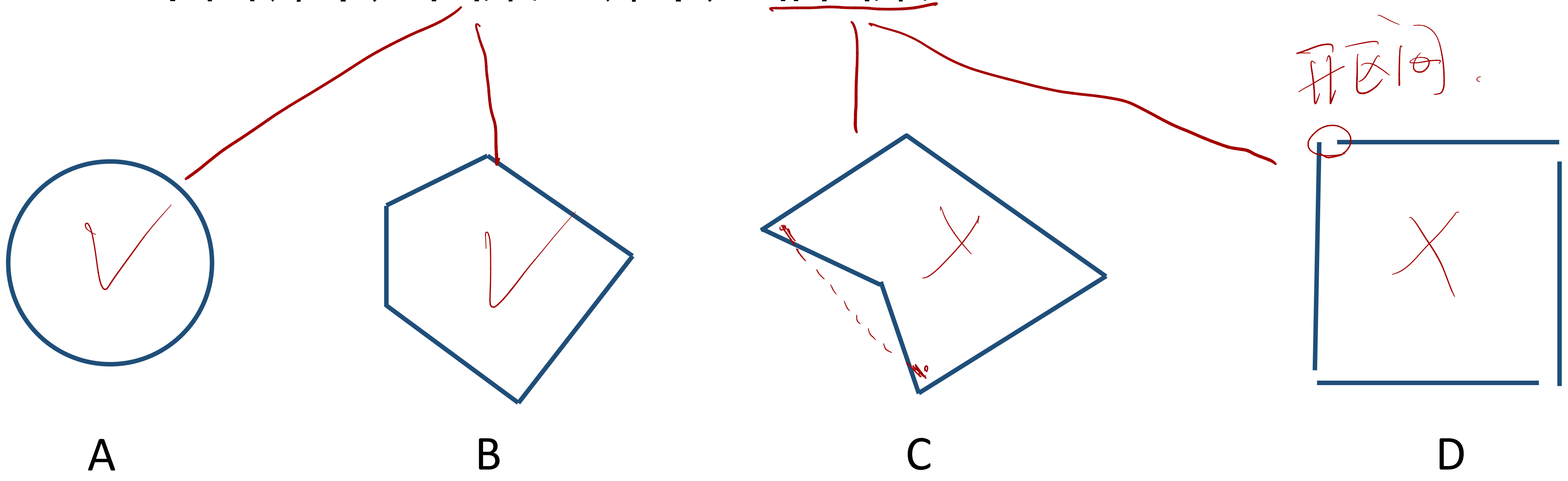
假设对于任意$x, y \in C$并且任意参数，$\boxed{\alpha \in [0,1]}$，我们有$\alpha x + (1-\alpha)y \in C$，则集合为凸集

x, y 的加权平均
(convex-combination)



Convex Set (凸集)

下面哪个是凸集？ 哪个是非凸集？



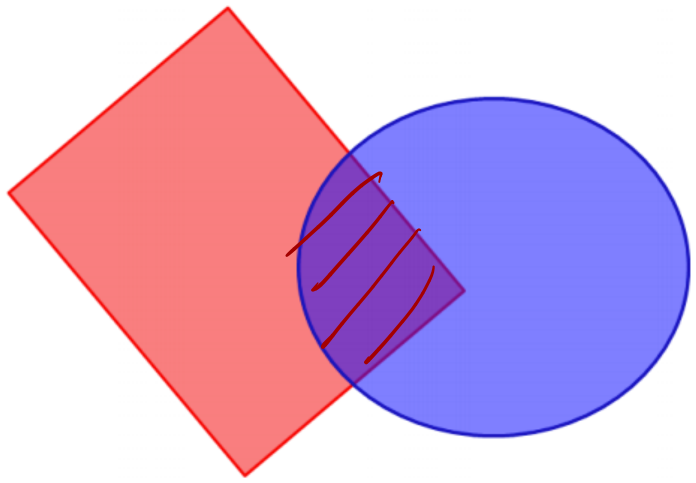
Convex Set (凸集) $\in R^d$

例子:

- Real space: R^n
- 所有正数集合R_+^n
- 范数$||x|| \leq 1$
- Affine set: 线性方程组是的所有解$Ax = b$
- Halfspace: 不等式的所有解：$Ax \leq b$

Convex Set (凸集)

两个凸集的交集也是凸集(intersection of convex set is convex)



Convex Set

由多个不等式构成的集合C:

$$\{w \mid g(w) \leq a\}$$

集合C为Convex set，如果函数g为凸函数

Convex Function

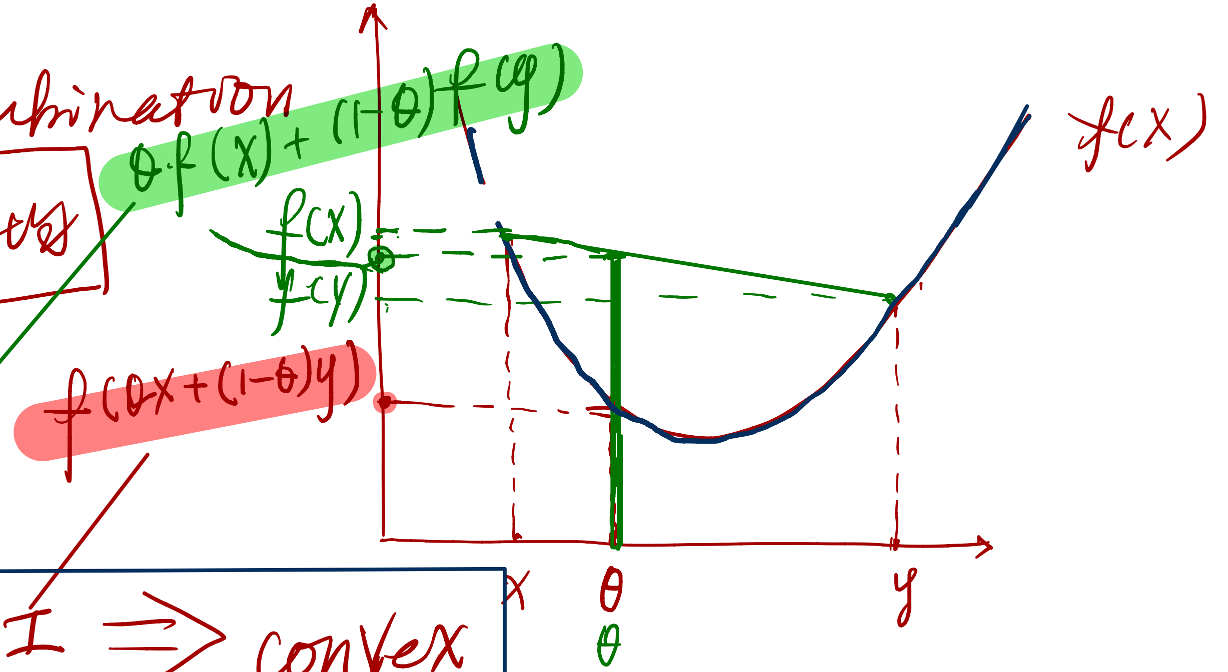
凸函数定义1

函数的定义域dom$f$为凸集，对于定义域里任意x, y，函数满足

$$f(\theta x + (1-\theta)y) \le \theta f(x) + (1-\theta)f(y) \qquad \theta \in [0,1]$$

convex combination

定义域 加权平均



绿 > 红 ⇒ convex

i.e., Convexity of Norms

Proof: All norms are convex

$$f(w) = ||w||_p$$

$p=1$: $||w||_1$: L_1 norm

$p=2$: $||w||_2$: L_2 norm

norm 定义: $||w||_p = (|w_1|^p + |w_2|^p + \cdots\cdots |w_d|^p)^{\frac{1}{p}}$

(绝对值)

$\forall\ x, y \in \mathbb{R}^d$

$f(\alpha x + (1-\alpha) y) \leq \alpha f(x) + (1-\alpha) f(x)$

$\Leftrightarrow\ ||\alpha x + (1-\alpha) y||_p \leq \alpha ||x||_p + (1-\alpha) ||y||_p$

i.e., Linear Function

- 线性函数：$f(x) = b^T x + c$

是下图.

i.e., Linear Function

- 线性函数：$f(x) = \underline{b^T x + c}$

$\forall\, x, y \in \mathbb{R}^d$

$$f(x) = b^T x + c$$
$$f(y) = b^T y + c$$

$$f(\alpha x + (1-\alpha) y) \le \alpha f(x) + (1-\alpha) f(y)$$

$$\Downarrow$$

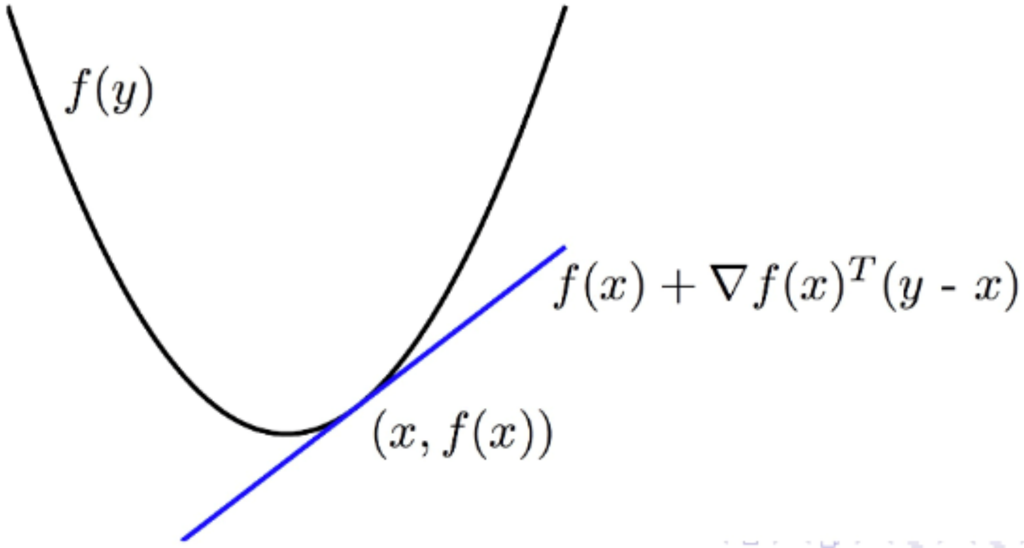
$$b^T(\alpha x + (1-\alpha) y) + c \le \alpha (b^T x + c) + (1-\alpha)(b^T y + c)$$

凸函数定义2 – 1st Order Convexity

假设$f: R^n \rightarrow R$是可导的 (differentiable)，则f 为凸函数，

当且仅当： $$f(y) \geq f(x) + \nabla(x)^T(y-x)$$

对于任意$x, y \in domf$



凸函数定义3 – 2nd order convexity

假设f: $R^n \rightarrow R$是两次可导的（twice differentiable），则f 为凸函数，当且仅当：

$$\boxed{\nabla^2 f(x)} \geq 0$$

hessian matrix

对于任意$x, y \in dom f$

$[H] \geq$ positive semi-definite

M is P.S.D

$\Rightarrow$ 定义: $V^T M \cdot V \geq 0$, $\forall V$

例: $f(x) = x^2$

$f''(x) = 2 \geq 0 \Rightarrow f(x)$ is convex.

Hessian Matrix : $\nabla^2 f(x)$

$$\nabla^2 f(x) = \begin{bmatrix} \frac{\partial}{\partial x_1 \partial x_1} f(x) & \frac{\partial}{\partial x_1 \partial x_2} f(x) & \cdots \\ \vdots & & \\ \frac{\partial}{\partial x_n \partial x_1} f(x) & \cdots & \frac{\partial}{\partial x_n \partial x_n} f(x) \end{bmatrix}$$

判断矩阵为半正定矩阵 P.S.D

- 所有特征值（eigenvalues）为非负（non-negative）
- 对于任意向量v，$v^T A v$ 为非负 ≥ 0.

$n\times 1$ $1\times n$ $n\times n$ $n\times 1$ $= 1\times 1 \geq 0$

way: $M = A\,B\,C$

diagonal matrix.

二次函数（quadratic function)

- 二次方函数（quadratic function)

$$f(x) = \frac{1}{2}x^T A x + b^T x + c$$

$$\frac{\partial f(x)}{\partial x} = Ax + b$$

对于任意A ≥0 $\Leftrightarrow$ A is p.s.D

$$\frac{\partial^2 f(x)}{\partial^2 x} = A \text{ is p.s.D}$$

hessian matrix

Operations preserve convexity

- 如果函数 f 和 g 均为凸函数，以下操作之后仍然是凸函数
  - **Non-negative scaling**: $h(w) = \underline{a}\, f(w)$ $a \geqslant 0$
    $a \leqslant 0$ concave.
  - **Sum**: $h(w) = \alpha f(w) + \beta g(w)$ $\alpha \geqslant 0$ $\beta \geqslant 0$
  - **Maximum**: $h(w) = \max\{f(w), g(w)\}$
  - **Affine map**: $h(w) = f(Aw + b)$

但是，一般情况下 $f(g(w))$ 不是凸函数

Some Convex Optimization Problems

Linear Program

convex

$$\max \quad C^T x$$ — linear function $\Rightarrow$ convex funtion

s.t. $Ax \leq$ b

$x \geq 0$

half space $\boxed{\text{convex set}}$ is convex set

1. obj. funtion is convex funtion
2. set is convex set

$\Rightarrow$ Lp is convex

Robust Regression

$$argmin_{w \in R^d} \sum_{i=1}^{n} |w^T x^i - y^i|$$

1-norm.

1. remove "| |":

e.g. $|\alpha| = \max\{\alpha, -\alpha\}$

$\Rightarrow \underbrace{argmin}_{w \in R^d} \sum_{i=1}^{n} \max\{\underbrace{w^T x^i - y^i}, \underbrace{y^i - w^T x^i}\}$

$w \in R^d$ → convex set

linear function – convex function

↓ convex

$y^i \Rightarrow argmin_{w \in R^d, \{y_i\}} \sum_{i=1}^{n} y_i$

s.t. $y_i \geq w^T x^i - y^i$

$y_i \geq y^i - w^T x^i$

By using convexity preserving operation, we prove it is a convex operation problem

Robust Regression

Least Squares Problem

$$argmin_{w \in R^d} ||Xw - y||^2$$

Least Squares Problem

$$argmin_{w \in R^d} ||Xw - y||_2^2$$

1. $w \in R^d \Rightarrow$ convex set

2. $f(w) = ||Xw - y||_2^2$

$= (Xw-y)^T(Xw-y)$

$= (w^TX^T - y^T)(Xw-y)$

$= w^TX^TXw - y^TXw - w^TX^Ty + y^Ty$

$= w^TX^TXw - w^TX^Ty - w^TX^Ty + y^Ty$

$= w^TX^TXw - 2w^TX^Ty + y^Ty$

prove $\nabla^2 f(w) \geq 0$ (p.s.d)

$\frac{\partial^2 f(w)}{\partial w} = \underline{X^TX} \geq 0$

p.s.d.

$\boxed{v^TX^TXv \geq 0}$

$\therefore$ LS problem is convex optimization problem

General Lp-norm Loss

$$f(w) = ||Xw - y||_p$$

$p=1$: $f(w) = ||Xw-y||_1$: robust regression / L.P.

$p=2$: $f(w) = ||Xw-y||_2$: Least Square problem

convex problem

$p>2$: convex problem

$p<1$: NOT convex optimization $\Rightarrow$ NP Hard

$\boxed{\text{actually not a norm}}$

Objective of Linear SVM

scalar > 0 : convex

$$\underbrace{\sum_{i=1}^{n} \max\{0, 1 - y^i w^T x^i\}}_{\xi_i} + \frac{\lambda}{2}||w||^2$$

convex problem

const. convex

linear convex

if scalar < 0, 不能判为 convex

→ λ must be ≥ 0

$$\Rightarrow \sum_{i=1}^{n} \xi_i + \frac{\lambda}{2}||w||^2 \text{ — quadratic}$$

s.t. $\xi_i \geq 0 \quad \forall i$

$\xi_i \geq 1 - y^i w^T x^i \quad \forall i$

⇒ Quadratic Programming w/ linear constraints

convex optimization prob.

Objective of Linear SVM

$$\sum_{i=1}^{n} \max\{0, 1 - y^i w^T x^i\} + \frac{\lambda}{2} ||w||^2$$

Convex/Non-convex Use Cases

1. Classical Portfolio Optimization

given Total cash: \$1 mm on hand

stocks: 1, 2, …… m can choose

objective: Max(return) & min(risk) = max(−risk)

1) decision variable: $w_1, w_2, \cdots w_m$: 把 % 的钱投入 m stock.

$w_i \in [0,1]$ $\sum_{i=1}^{m} w_i = 1$

λ: control return 和 risk 权重

即 λ↑ 注重 return

λ↓ 注重 risk

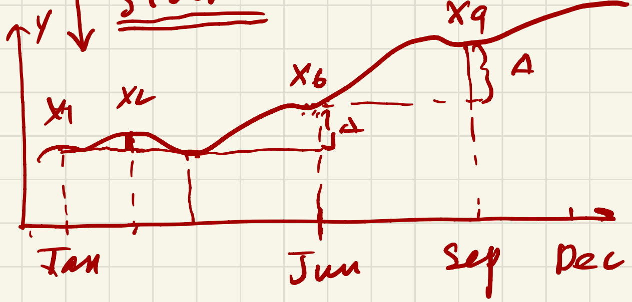
2) objective function: minimize(− return + λ risk)

S: random variable: each stock's return

$S_i \sim N(\mu_i, \sigma_i^2)$: S_i normal distribution

avg. return risk

$S_1 \sim N(\underline{\mu_1}, \underline{\sigma_1^2})$ → stock 1



$\therefore \mu_1 = \sum_{i=1}^{t} y_i / t$

$\sigma = std\{y_i\}$: 波动率

$$\boxed{\begin{array}{l} S_1 \sim N(\underline{\mu_1}, \underline{\sigma_1^2}) \\ S_2 \sim N(\underline{\mu_2}, \underline{\sigma_2^2}) \\ \vdots \qquad\qquad \vdots \\ S_m \sim N(\mu_m, \sigma_m^2) \end{array}}$$

$\therefore w_1 S_1 + w_2 S_2 + \cdots w_m S_m$

$= \sum_{i=1}^{m} w_i S_i \sim N(\underline{\mu'}, \underline{\sigma'^2})$

portfolio return

portfolio risk

guassian dist.

$\Rightarrow \mu' = \sum_{i=1}^{m} w_i \mu_i$

$\sigma'^2 = \sum_{i=1}^{m} w_i w_j \sigma_{ij}$ (cov)

$\therefore$ objective:

$\Downarrow$

$$\text{minimize} \; -\sum_{i=1}^{m} w_i \mu_i + \lambda \sum_{i=1}^{m} w_i w_j \sigma_{ij}$$

$$W = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{bmatrix}, \quad \mu = \begin{bmatrix} \mu_1 \\ \vdots \\ \mu_m \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \sigma_1^2, \sigma_1\sigma_2 & \cdots & \sigma_1\sigma_m \\ & & \vdots \\ \sigma_m\sigma_1 & \cdots & \sigma_m^2 \end{bmatrix}$$

$$\boxed{\begin{aligned} \therefore \; & \text{minimize} \; -\mu^T w + \lambda W^T \Sigma W \\ & \text{s.t.} \; \sum_{i=1}^{m} w_i = 1, \quad w_i \geq 0 \end{aligned}}$$

~~判~~断:

1) convex set (✓)

2) objective function

→ quadratic (✓)

$$\nabla^2 f(W) = \lambda \Sigma$$

$\lambda \geq 0$, Σ p.s.D ≥ 0

∴ Convex optimization problem

└→ Quadratic programming w/ Linear constraint

finance domain

└→ Mean-Variance portfolio optimization.

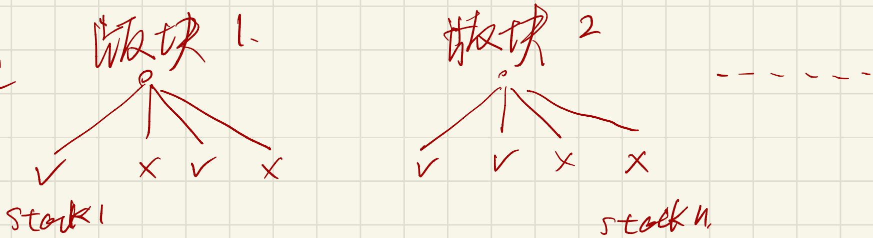
improvement:

A. sparsity: 3000 stocks → 50 (only select)

2950 = 0.

B. $w_i \geq 0.2$

C.



D. buy/sell ⇒ brokerage fee ← minimize

E. other constraints

put A, B, C, D, E to above model

this could cause "non-convex"

1. Classical Portfolio Optimization

2.Set Cover Problem

到今没有好解.

假设我们有个全集U (Universal Set), 以及m个子集合 $S_1, S_2 \ldots, S_m$, 目标是要寻找最少的集合，使得集合的union等于U.

例子： U = {1,2,3,4,5}, S: {S_1={1,2,3}, S_2={2,4}, S_3={1,3}, S_4={4}, S_5={3,4}, S_6={4,5}}，最少的集合为：{1,2,3}, {4,5}，集合个数为2.

$S_1 \cup S_6 = U$

例子： U = {1,2,3,4,5}，S: {S_1={1,2,3}, S_2={2,4}, S_3={1,3}, S_4={4}, S_5={3,4}, S_6={4,5}}，最少的集合为：{1,2,3}{4,5}，集合个数为2.

Approach 1: Exhaustive Search

brute force

例子： U = {1,2,3,4,5}，S: {S_1={1,2,3}, S_2={2,4}, S_3={1,3}, S_4={4}, S_5={3,4}, S_6={4,5}}，最少的集合为：{1,2,3}, {4,5}，集合个数为2.

$S_1 \neq U$

$S_2 \neq U \Rightarrow$

⋮

$S_6 \neq U$

$S_1 \cup S_2 \neq U$

$S_1 \cup S_3 \neq U$

⋮

$S_1 \cup S_6 = U$ ✓

$\Rightarrow$ powerset (2^m)

Approach 2: Greedy Search

局部最优
和全局最优.

例子： U = {1,2,3,4,5}，S: {S_1={1,2,3}, S_2={2,4}, S_3={1,3}, S_4={4}, S_5={3,4}, S_6={4,5}}，最少的集合为：{1,2,3}, {4,5}，集合个数为2.

Strategy: 尽量选择更多元素.

设 $\alpha = \Phi$

$S_1 \quad S_2 \quad \cdots\cdots \quad S_6 \Longrightarrow \alpha = \{S_1\}$

3 elements ✓　　2 elements　　2 elements

$S_1 \cup S_2 = 4$

$S_1 \cup S_3 = 3$

$S_1 \cup S_5 = 4$

$\boxed{S_1 \cup S_6 = \boxed{5}}$ ✓ 最大 $\Longrightarrow \alpha = \{S_1, S_6\} = U$

Mathematical Formulation

例子： U = {1,2,3,4,5}，S: {S_1={1,2,3}, S_2={2,4}, S_3={1,3}, S_4={4}, S_5={3,4}, S_6={4,5}}，最少的集合为：{1,2,3}, {4,5}，集合个数为2.

Decision Variable: 选或不选第i个 S_i

: $x_i \in \{0,1\}$

objective function: $\min \sum_{i=1}^{n} x_i$

every element in U: included in

$1 \rightarrow S_1$: $x_1 \geq 1$ $\quad i: 1 \in S_i$

$2 \rightarrow S_1, S_2$ $x_1 + x_2 \geq 1$

$3 \rightarrow S_1, S_3, S_5$: $x_1 + x_3 + x_5 \geq 1$

$4 \rightarrow S_2, S_4, S_5$: $x_2 + x_4 + x_5 \geq 1$

$5 \rightarrow S_6$: $x_6 \geq 1$

s.t. $\sum_{i: e \in S_i} x_i \geq 1, \quad \forall e \in U$

$x_i \in \{0,1\}$

$$\text{minimise} \sum_{i=1}^{n} x_i$$

$$\text{s.t.} \sum_{i:e\in S_i} x_i \geq 1,$$

$$x_i \in \{0,1\}$$

$\Rightarrow$

1) convex set (X)

$\because x_i \in \{0,1\} \neq$ convex

$\therefore$ NOT a convex optimization problem

Instead, is integer programming

NP-H $\Leftarrow$ exact optimization (linear)

Relaxing = Approximation Solution

$\Downarrow$

relaxing by $x_i \in [0,1]$. linear programming

Approx. Exact

$n \times$ Exact

gap

1.5 倍 exact

2 倍 exact

⋮

TakeAway:

- convex set
- convex function
- convex problem
- math formulation for usecase

Is it Convex?

$$minimize \sum_{i=1}^{m} x_i$$

$$s.t. \sum_{i:e \in s_i} x_i \geq 1$$

$$x_i \in \{0,1\} \quad i = 1, \dots, m$$