# 15. Algorithms Primer

## ① Unconstrained Optimization

For convexity: $\nabla f(x)^T \Delta x < 0$

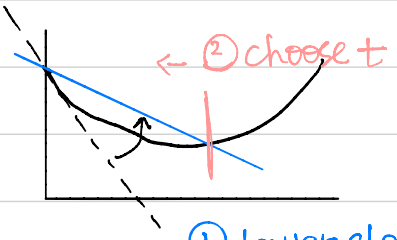
### A: Descent Method:

$$x^{k+1} = x^k + \underbrace{t^k}_{\text{step size}} \underbrace{\Delta x^k}_{\text{search direction}}, \quad f(x^{k+1}) < f(x^k)$$

↳ Line search

1) Exact line search: $t = \underset{t>0}{\operatorname{argmin}} f(x + t\Delta x)$

2) Backtracking:
1. start at $t = 1$ (full step)
2. until $f(x + t\Delta x) < f(x) + \alpha t \nabla f(x)^T \Delta x$
3. $t := \beta t$, $\beta \in (0, 1)$




① lower slope (cut through func): smaller $\alpha$

### Gradient Descent Method: $\Delta x = -\nabla f(x)$

### B: Newton's Method: Minimizing the second order approximation

→ needs twice-differentiable func.

Derivation: $\hat{f}(x+v) = f(x) + \nabla f(x)^T v + \frac{1}{2} v^T \nabla^2 f(x) v$

choose $v$ to minimize $\hat{f}$: $D\hat{f} = Df + D^2 v = 0 \Rightarrow v = -D^{-2} Df$

Steps:

1) Compute Newton Step and decrement

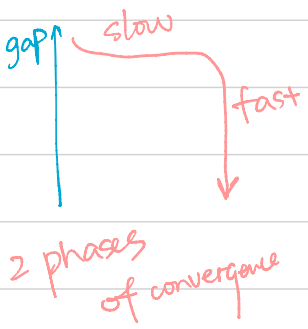
$$\Delta x_{nt}^k = -\nabla^2 f(x^k)^{-1} \nabla f(x^k), \quad \underbrace{\lambda(x^k)^2 = -\nabla f(x^k)^T \Delta x_{nt}^k}_{\text{tells how far from optimum}}$$

2) stop criterion: if $\lambda(x^k)^2/2 \leq \epsilon$

3) Line search

4) update: $x^{k+1} = x^k + t^k \Delta x_{nt}^k$

5) $k = k+1$




## ② Constrained Optimization

↗ or KKT

### A: Equality Constrained Optimization (Eliminating equality constraints)

$Ax = b \Rightarrow Fz + x_0$, $F$ is nullspace of $A$ i.e. $AF = 0$

$$\begin{cases} \min f(x) \\ \text{s.t. } Ax = b \end{cases} \longrightarrow \min \tilde{f}(z) = f(Fz + x_0)$$ is convex ∵ $f$ is assumed convex

↳ unconstrained: Newton

chain rule: $\nabla \tilde{f}(z) = F^T \nabla f(x)$

"also gradient on z is OK; in B is gradient on x" $\nabla^2 \tilde{f}(z) = F^T \nabla f(x) F$

LJ : cont'd.

## B. Gradient Projection Method (both equality or inequality constraints)

$$\min f(x) \quad \text{s.t. } x \in \mathcal{X} \quad \Rightarrow \quad x^{k+1} = [x^k - \alpha^k \nabla f(x^k)]_{\mathcal{X}}$$

projection (closest feasible point)

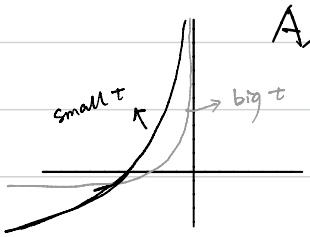
↓

another optimization prob.

## C. Interior-Point Methods (IPM)

$$\min_x f_0(x) \quad \text{s.t. } f_i(x) \le 0, \; Ax = b \quad \rightarrow \quad \min_x f_0(x) + \sum_{i=1}^m I_-(f_i(x)) \quad \text{s.t. } Ax = b$$

indicator func: $\begin{cases} 0 & \text{if } u \le 0 \\ \infty & \text{otherwise.} \end{cases}$




$-(\frac{1}{t})\log(-x)$ approximate with a smooth function

### Barrier method:

= called Interior-Point

Step:

0. Very initial point → each iteration the point is feasible
1. Start with small $t$ (at the begining, large $t$ → Newton slow converge)
2. use Newton (fast phase better), use last $x$ as initial point
3. stop at $m/t$ (duality gap) $< \epsilon$
4. increase $t$

Feasibility problem: find $x$ such that $f_i(x) \le 0, i = 1, \dots, m; Ax = b$

↓

Phase I Method: $\min_{x,s} s \quad \text{s.t. } f_i(x) \le s; Ax = b$

choose a $s > \max f_i(x)$ and decrease. if feasible $s^* < 0$, then $x^*$ feasible

initial point for this optimization problem (IPM to solve)

15. cont'd

## ③ Block Coordinate Algorithms

### A: Block Coordinate Descent

update parameters block sequentially

Feature: 1) non-increasing objective value

2) each subproblem may be much easier to solve (even closed-form)

Application: Lasso (L2-L1 optimization) via BCD

prob: $\underset{x}{\text{minimize}} \quad f(x) \triangleq \frac{1}{2}\|y - Ax\|_2^2 + \lambda\|x\|_1$

↓

closed form BCD: $\underset{x_i}{\text{minimize}} \quad f_i(x_i) \triangleq \frac{1}{2}\|\tilde{y}_i^k - a_i x_i\|_2^2 + \lambda|x_i|$

vector ×

scalar ✓

where $\tilde{y}_i^k \triangleq y - \sum_{j<i} a_j x_j^{k+1} - \sum_{j>i} a_j x_j^k$

(将矩阵运算拆成scalar形式、求偏导)

↓

closed form solution: $x_i^{k+1} = \text{soft}_\lambda(a_i^T \tilde{y}_i^k)/\|a_i\|^2$

this can also be solved by MM

### B. Jacobi Algorithm:

update parameters block in parellel

Feature: 1) no non-increasing objective value

# L9 Portfolio Optimization

## Premier on Financial Data

log-prices $y_t \triangleq \log P_t$ . follows random walk $y_t = \mu + y_{t-1} + \varepsilon_t$

simple return $R_t \triangleq \frac{P_t - P_{t-1}}{P_{t-1}} = \frac{P_t}{P_{t-1}} - 1$

log-return $r_t \triangleq y_t - y_{t-1} = \log\left(\frac{P_t}{P_{t-1}}\right) = \log(1 + R_t)$ 描绘price差分

### why log-return ?

① prices are assumed to be distributed log normally $\Rightarrow r_t$ normally distributed

② $r_t \approx R_t$ when returns are small . good approximation

③ Compound return : $\prod_i (1 + R_i) = (1+R_1)(1+R_2)\cdots(1+R_i)$

$\log \prod_i (1+R_i) = r_1 + r_2 + \cdots + r_i$ → time-additivity . normally distributed

各种model. iid最简单

i.i.d $\begin{cases} \mu_t \triangleq E[r_t \mid F_{t-1}] = \mu \\ \Sigma_t \triangleq Cov[r_t \mid F_{t-1}] = E[(r_t - \mu_t)(r_t - \mu_t)^T \mid F_{t-1}] \end{cases}$ future

未来mean.var → iid 认为 const → sample value 来代替

## Portfolio Basics

$\begin{cases} \text{portfolio } w: & 1^T w = 1 \\ \text{portfolio return}: & R_t^P \approx w^T r_t \end{cases}$

performance measures

expected return : $w^T \mu$

volatility : $\sqrt{w^T \Sigma w}$

Sharpe Ratio (SR): $SR = \frac{w^T \mu - r_f}{\sqrt{w^T \Sigma w}}$ ← risk-free rate

# L 9 cont'd.

## Markowitz's Mean-variance Portfolio (MVP)

$$\begin{bmatrix} \underset{w}{\text{maximize}} & w^T\mu - \lambda w^T\Sigma w \\ \text{s.t.} & 1^T w = 1 \end{bmatrix}$$ convex QP

## Global minimum variance portfolio (GMVP)

$$\begin{bmatrix} \underset{w}{\text{minimize}} & w^T\Sigma w \\ \text{s.t.} & 1^T w = 1 \end{bmatrix}$$ ignores the expected return : convex QP

## Maximum Sharpe ratio portfolio (MSRP)

$$\begin{bmatrix} \underset{w}{\text{maximize}} & \dfrac{w^T\mu - r_f}{\sqrt{w^T\Sigma w}} \\ \text{s.t.} & 1^T w = 1 \end{bmatrix}$$ nonconvex

fractional programming (FP)

$$\Downarrow$$

$$\begin{bmatrix} \underset{w,t}{\text{maximize}} & t \\ \text{s.t.} & t \le \dfrac{w^T\mu - r_f}{\sqrt{w^T\Sigma w}} \\ & 1^T w = 1 \end{bmatrix} \rightleftharpoons \begin{bmatrix} \text{find} & w \\ \text{s.t.} & t\|\Sigma^{1/2} w\|_2 \le w^T\mu - r_f \\ & 1^T w = 1 \end{bmatrix}$$

Bisection + SOCP

## Fractional Programming

$$\begin{array}{lll} \underset{x}{\text{maximize}} & \dfrac{f(x)}{g(x)} & \begin{matrix} \to \text{ concave} \\ \to \text{ convex} \end{matrix} \\ \text{s.t.} & x \in \mathcal{X} & \end{array}$$

Solution

① bisection : the problem is quasi-convex ↷ epigraph form

$$\underset{x,t}{\text{maximize}} \quad t \qquad \text{s.t.} \quad t \le \frac{f(x)}{g(x)},\ x \in \mathcal{X}$$

fix $t$ →

← 1) feasible: $t\uparrow$

2) otherwise: $t\downarrow$

$$\underset{x}{\text{maximize}} \quad 0 \qquad \text{s.t.} \quad tg(x) \le f(x)$$

$g(x)$ positive; convex problem

② Dinkelbach transform

$$\underset{x}{\text{maximize}} \quad f(x) - yg(x) \qquad \text{s.t.} \quad x \in \mathcal{X}$$

$y^{(k)} = \dfrac{f(x^{(k)})}{g(x^{(k)})}$ iteratively update

# L11. MM.

## ① MM Algorithm

### MM in a Nutshell:

difficult optimization problem ⇒ successively minimize a surrogate func.

$$\underset{x}{\text{minimize}} \; f(x) \quad \text{s.t.} \; x \in \mathcal{X} \;\Rightarrow\; x^{k+1} = \underset{x \in \mathcal{X}}{\operatorname{argmin}}\, u(x, x^k)$$

rule of surrogate func.

$$\begin{cases} u(y,y) = f(y), \; \forall y \in \mathcal{X} & \text{touch at } x^k \\ u(x,y) \geq f(x), \; \forall x,y \in \mathcal{X} & \text{upper bound} \\ u'(x,y;d)\big|_{x=y} = f'(y;d), \; \forall d \text{ with } y+d \in \mathcal{X} & \text{same slope at } x^k \\ u(x,y) \text{ is continuous in } x \text{ and } y \end{cases}$$

### Applications:

(1) nonnegative least squares: $\underset{x \geq 0}{\text{minimize}} \; \|Ax-b\|_2^2$ ; $A \in \mathbb{R}_{++}^{m \times n}$, $b \in \mathbb{R}_{+}^{m}$

$$f(x) = \tfrac{1}{2}\|Ax-b\|_2^2$$

$$u(x,x^k) = f(x^k) + \nabla f(x^k)^T(x-x^k) + \tfrac{1}{2}(x-x^k)^T \Phi(x^k)(x-x^k)$$

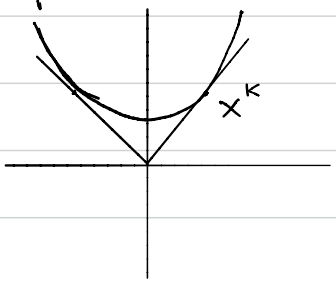
(Hessian of u) $\Phi(x^k) = \underline{\text{diag}}(\cdot\cdot) \succeq A^TA$ (Hessian of f)

$\underset{x \geq 0}{\text{minimize}}\; u(x,x^k)$ is easier because each $x_i$ is decoupled.

(2) reweighted LS for L1-norm minimization:

$\underset{x}{\text{minimize}} \; \|Ax-b\|_1$ is LP with no closed form solution → LP solver

L1 → L2: if L2-norm → LS (easy to solve)




(3) Sparse generalized eigenvalue problem:

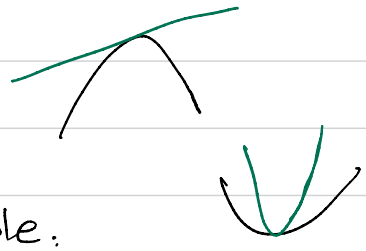
e.g. $\max_x \; x^TAx - \rho\|x\|_0$ s.t. $x^TBx = 1$ — surrogate with quadratic not linear → Generalized eigenvalue problem: $\max_x \; x^T(A-\rho_0)x$ s.t. $x^TBx = 1$

# L11 cont'd

## Construction of surrogate functions :

(1) Construction by convexity : Jensen inequality

(2) Construction by Taylor expansion:



① $k(x)$ is concave and differentiable :

$$k(x) \le k(x^k) + \nabla k(x^k)^T (x - x^k)$$

② $k(x)$ is convex and twice differentiable :

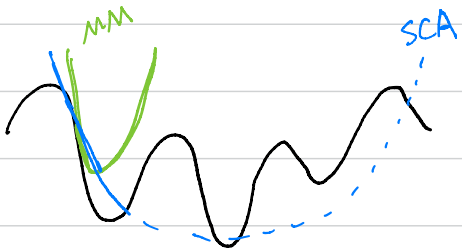
$$k(x) \le k(x^k) + \nabla k(x^k)^T (x - x^k) + \frac{1}{2}(x - x^k)^T M (x - x^k)$$

if $M - \nabla^2 k(x) \succeq 0 \ . \forall x$

## EM is a kind of MM

## Connection to Successive Convex Approximation (SCA)

△ surrogate function
- may not be convex in MM
- should be convex in SCA




- should be upper bound in MM
- doesn't need to be upper bound in SCA

△ MM can be easily extended to nonconvex $x$

△ SCA naturally has a parallel update

② Block MM : BCD + MM (with respect to that block)

↓

when finding surrogate jointly is difficult

# L12 Risk Parity Portfolio 只考虑risk分布，不考虑收益

## Problem formulation:

one number for total volatility → decompose into a sum of risk corresponding to each asset (Euler's theorem)

portfolio volatility: $\sigma(w) = \sqrt{w^T \Sigma w}$ → total

decompose: $\sigma(w) = \sum_{i=1}^{N} w_i \frac{\partial \sigma}{\partial w_i} = \sum_{i=1}^{N} w_i \cdot \frac{(\Sigma w)_i}{\sqrt{w^T \Sigma w}}$ ↗ each asset

marginal risk contribution (MRC) $MRC_i = \frac{\partial \sigma}{\partial w_i} = \frac{(\Sigma w)_i}{\sqrt{w^T \Sigma w}}$

risk contribution (RC) $RC_i = w_i \frac{\partial \sigma}{\partial w_i} = \frac{w_i (\Sigma w)_i}{\sqrt{w^T \Sigma w}}$

relative risk contribution (RRC) $RRC_i = \frac{RC_i}{\sigma(w)} = \frac{w_i (\Sigma w)_i}{w^T \Sigma w}$

"normalize RC → $\sum_{i=1}^{N} RRC_i = 1$"

Risk parity portfolio: $RC_i = \sigma(w)/N$ or $RRC_i = 1/N$

general ↓

Risk budgeting portfolio: $RRC_i = b_i \rightarrow w_i (\Sigma w)_i = b_i w^T \Sigma w$

(enable different weights on risk. RPP follows $b_i = \frac{1}{N}$)

↓ nonconvex (nonlinear constraint)

feasibility problem

## Solutions:

① assume $\Sigma$ is diagonal: $w_i^2 \sigma_i^2 = b_i \sum_{j=1}^{N} w_j^2 \sigma_j^2 \propto b_i$

$$\rightarrow w_i = \frac{\sqrt{b_i}/\sqrt{\Sigma_{ii}}}{\sum_{j=1}^{N} \sqrt{b_j}/\sqrt{\Sigma_{jj}}}$$

② vanilla convex formulation:

Let $x = w/\sqrt{w^T \Sigma w}$, then root follows: $x_i (\Sigma x)_i = b_i$

Vector Form: $\Sigma x = b/x$

considering convex function: $f(x) = \frac{1}{2} x^T \Sigma x - b^T \log(x)$

$\nabla f(x) = \Sigma x - b/x = 0$ ↔ same!!

minimization problem: $\underset{x \geq 0}{\text{minimize}} \; f(x)$ by Newton's method / BCD

Find the root ⇒ Solve a minimization problem

L12. cont'd.

③ General nonconvex formulation: because ② only when $1^T w = 1$, $w \geq 0$

↓

measure the differences ← add more constraints

between the terms $w_i(\Sigma w)_i$ (cannot achieve $w_i(\Sigma w)_i = \phi_i w^T \Sigma w$)

$$\underset{w,\theta}{\text{minimize}} \quad \sum_{i=1}^{N} \left( w_i(\Sigma w)_i - \theta \right)^2 \qquad \cdots \text{many other versions}$$

$$\text{s.t.} \quad 1^T w = 1 \qquad \text{nonconvex}$$

# L13 Sparsity

## Optimization with Sparsity

$$\min_x \ f(x)$$
$$\text{s.t.} \quad x \in \mathcal{X}, \ \text{card}(x) \le k$$

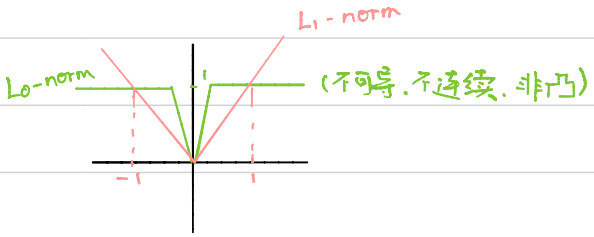
↓ NP hard. solve with convex relaxation

$L_0$-norm: $\|x\|_0$ (虽然不是norm)

card(x): number of nonzero elements

supp(x): position with nonzero values

## Algorithm for Sparsity Problems




① $L_1$-norm:
1. apply $L_1$-norm
2. set the very small elements to zero
3. re-solve the new problem with fixed zero pattern

intepretation: $L_1$-norm is convex envelope of card on $[-1, 1]$

(largest convex function as an underestimator)

## ② Iterative Reweighted $L_1$-norm Heuristic

suppose problem: $\min \|x\|_0$
$\text{s.t.} \quad x \in \mathcal{X}$

Algorithm: set $w = 1$ repeat

$\text{minimize}_x \ \|\text{Diag}(w)x\|_1, \ \text{s.t.} \ x \in \mathcal{X}$

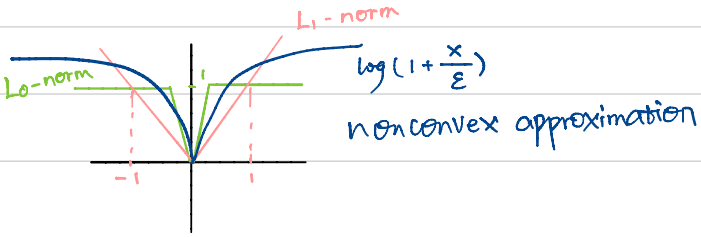
$w_i = 1/(\varepsilon + |x_i|)$

until convergence to local point

for smalle $|x_i|$, $w_i$ increases, $|x_i|$ even smaller

for lager $|x_i|$, $w_i$ decreases, allow $|x_i|$ to be larger if necessary

Derivation: ① better approximation → ② MM (Taylor, linearize)




assume $x \ge 0$

$$\sum_{i=1}^n \log(1 + x_i/\varepsilon) \approx \sum_{i=1}^n \log(1 + x_i^{(k)}/\varepsilon) + \sum_{i=1}^n \frac{x_i - x_i^{(k)}}{\varepsilon + x_i^{(k)}}$$

$$\text{minimize}_x \ \sum_{i=1}^n w_i x_i \qquad w_i = \frac{1}{\varepsilon + x_i^{(k)}}$$
$$\text{s.t.} \quad x \in \mathcal{X}$$

L13. cont'd

## Application of Sparsity:

① Compressed Sensing: $y = Ax$ with $A \in \mathbb{R}^{m \times n}$

- if $m \geq n$, and $A$ is full rank $\Rightarrow$ unique or no solution
- if $m < n$, infinite solutions

$\hat{x} = A^{+} y$ is the solution of $\hat{x} = \underset{y=Ax}{\operatorname{argmin}} \|x\|_2$

Sparsity problem: $x^{*} = \underset{y=Ax}{\operatorname{argmin}} \|x\|_0$

② Estimation with Outliers

measurements: $y_i = a_i^T x + v_i$, $v_i$ is Gaussian noise

add sparsity: $y_i = a_i^T x + v_i + w_i$

$$\begin{aligned} \underset{x,w}{\text{minimize}} \quad & \|y - Ax - w\|_2 \\ \text{s.t.} \quad & \text{card}(w) \leq k \end{aligned}$$

③ Feature Selection: choose subset of $k$ regressors that best fit $y$

$$\begin{aligned} \underset{\beta}{\text{minimize}} \quad & \|y - X^T \beta\|_2^2 \\ \text{s.t.} \quad & \text{card}(\beta) \leq k \end{aligned}$$

④ LASSO: $\hat{\beta}_{LASSO} = \underset{}{\operatorname{argmin}} \|y - X^T \beta\|_2^2 + \gamma \|\beta\|_1$

$\rightarrow$ QP, when $N$ is extremely large requires faster solution

# L14 index tracking

**Definition:** Returns of an index in $T$ days: $r^b = [r_1^b, \cdots, r_T^b]^T$

Returns of $N$ assets in $T$ days: $X = [r_1, \cdots, r_T]^T$

Company index $\rightarrow$ Design a sparse portfolio $w$ to track

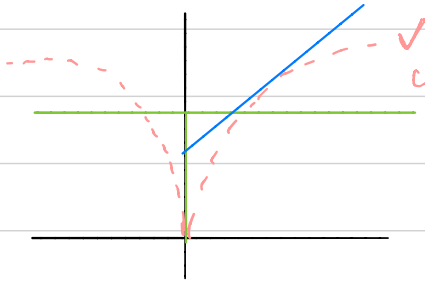
$$\begin{cases} b \geq 0 \\ b^T 1 = 1 \\ Xb = r^b \end{cases} \qquad \begin{cases} w \geq 0 \\ w^T 1 = 1 \\ Xw \approx r^b \end{cases}$$

Sparse index tracking via MM:

$$\underset{w}{\text{minimize}} \ \frac{1}{T}\|Xw - r^b\|_2^2 + \lambda \|w\|_0 \quad \text{s.t.} \quad w \in \mathcal{W}$$

$l_0$-norm approximation

Step 1: Approximation of the $l_0$-norm: $\rho_{p,\gamma}(w) = \dfrac{\log(1 + |w|/p)}{\log(1+\gamma/p)}$




$$\underset{w}{\text{minimize}} \ \frac{1}{T}\|Xw - r^b\|_2^2 + \lambda 1^T \rho_{p,u}(w) \quad \text{s.t.} \quad w \in \mathcal{W}$$

MM

Step 2: Linear approximation of $\rho_{p,\gamma}(w)$

$$\underset{w}{\text{minimize}} \ \frac{1}{T}\|Xw - r^b\|_2^2 + \lambda d_{p,u}^{(k)T} w \quad \text{s.t.} \quad w \in \mathcal{W}$$

MM

Step 3: Upper-bound the norm-square

$$\underset{w}{\text{minimize}} \ \frac{1}{T} w^T X^T X w + \left(\lambda d_{p,u}^{(k)} - \frac{2}{T} X^T r^b\right)^T w + \text{const.} \quad \text{s.t.} \quad w \in \mathcal{W}$$

there is a closed-form update algorithm

**Extension:** if error term is not $l_2$-norm, we can use MM to find a quadratic majorizer ($L_2$) first.

# L 15 - SVM

## Classification:

prob: a set of input points $x_i \in R^n$ → binary labels $y_i \in \{-1, 1\}$

↓

separate with a linear model: $\hat{y} = \beta^T x + \beta_0$

- predict "pos Class", if $\text{sign}(\hat{y}) = +1$
- predict "Neg Class", if $\text{sign}(\hat{y}) = -1$

↓

optimization:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{N} 1\{\text{sign}(\hat{y}_i) \neq y_i\} \\ \beta_0, \beta, \{\hat{y}_i\} & \\ \text{s.t.} & \hat{y}_i = \beta^T x_i + \beta_0 \;,\; \forall i \end{array}$$

→ nonconvex, nondifferentiable

## Alternatives:

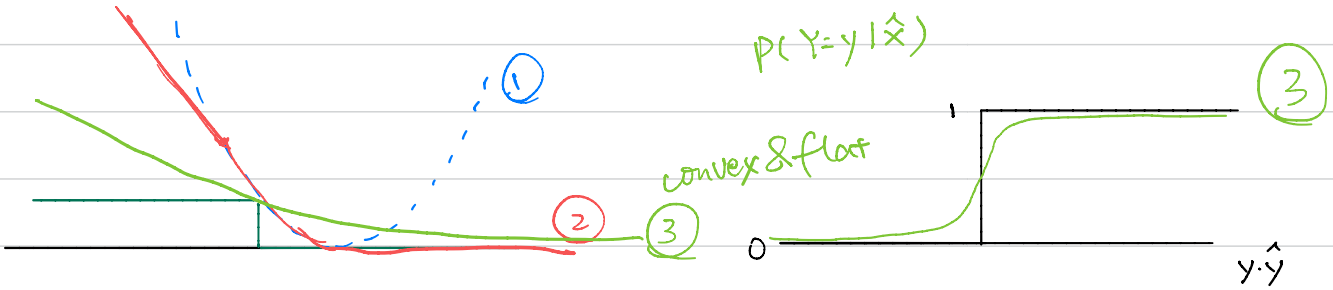
### ① Linear Regression

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{N} (\hat{y}_i - y_i)^2 \Rightarrow \sum_{i=1}^{N} (1 - y_i \hat{y}_i)^2 \\ \beta_0, \beta, \{\hat{y}_i\} & \\ \text{s.t.} & \hat{y}_i = \beta^T x_i + \beta_0 \;,\; \forall i \end{array}$$

### ② Regression with Huberized Loss

$$\phi_{hub}(x) = \begin{cases} |x|^2 & , \; |x| < M \quad \text{linear} \\ M(2|x| - M) & , \; |x| \geq M \quad \text{quadratic} \end{cases}$$

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{N} \phi_{hub\_pos}(1 - y_i \hat{y}_i)^2 \\ \beta_0, \beta, \{\hat{y}_i\} & \\ \text{s.t.} & \hat{y}_i = \beta^T x_i + \beta_0 \;,\; \forall i \end{array}$$




### ③ Logistic Model

model the probability of $y \in \{-1, 1\}$ as $P = \frac{1}{1 + e^{-y \cdot \hat{y}}}$

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^{N} \log(1 + e^{-y_i \hat{y}_i}) \\ \beta_0, \beta, \{\hat{y}_i\} & \\ \text{s.t.} & \hat{y}_i = \beta^T x_i + \beta_0 \;,\; \forall i \end{array}$$

negative log-likelihood

L15 cont'd.

# SVM:

## ① Linear Separable SVM:

Optimal Separating Hyperplane: ① Separate two classes ② Maximize the distance to closest point

signed distance: $\frac{1}{\|\beta\|}\beta^T(x - x_0) = \frac{1}{\|\beta\|}(\beta^T x + \beta_0)$ (向量內積; 平面法向量)

$\downarrow$

margin: $\frac{1}{\|\beta\|} y(\beta^T x + \beta_0)$

$$\left[\begin{array}{l}\underset{\beta_0,\beta,M}{\text{minimize}} \quad M \\ \text{s.t.} \quad \frac{1}{\|\beta\|} y_i(\beta^T x_i + \beta_0) \geq M, \ \forall i\end{array}\right] \rightarrow \left[\begin{array}{l}\underset{\beta_0,\beta}{\text{minimize}} \quad \|\beta\| \\ \text{s.t.} \quad y_i(\beta^T x_i + \beta_0) \geq 1, \ \forall i\end{array}\right]$$

## ② Linear Nonseparable SVM

$$\left[\begin{array}{l}\underset{\beta_0,\beta,\{\xi_i\}}{\text{minimize}} \quad \frac{1}{2}\|\beta\|^2 + C\sum_{i=1}^N \xi_i \quad (\|\xi\|_1) \\ \text{s.t.} \quad y_i(\beta^T x_i + \beta_0) \geq 1 - \xi_i, \ \xi_i \geq 0, \ \forall i \quad (\text{add sparsity})\end{array}\right]$$

$\Updownarrow$

$$\left[\begin{array}{l}\underset{\beta_0,\beta,\{\hat{y}_i\}}{\text{minimize}} \quad \sum_{i=1}^N [1 - y_i\hat{y}_i]^+ + \frac{\lambda}{2}\|\beta\|^2 \\ \text{s.t.} \quad \hat{y}_i = \beta^T x_i + \beta_0, \ \forall i\end{array}\right] \quad \lambda = \frac{1}{C}.$$

(hinge loss)

## ③ Nonlinear SVM

Feature Transformation: $\{R^n$: input space $\rightarrow$ $H$: feature space$\}$

$x \rightarrow h(x)$

kernel: $K(x_i, x_j) \triangleq h(x_i)^T h(x_j)$

(directly find $k$ instead of $h$)

# Lib. Low Rank

**Problem Formulation:** non-convex

$$\begin{bmatrix} \underset{X}{\text{minimize}} & \text{rank}(X) \\ \text{s.t.} & X \in C \leftarrow \text{convex set} \end{bmatrix}$$

if $X$ is diagonal, $\text{rank}(X) = \|\text{diag}(X)\|_0$

rank minimization → $L_0$-norm minimization

**Lemma:** Let $X \in R^{m \times n}$ be a given matrix. Then $\text{rank}(X) \leq r$ iff there exist $Y = Y^T \in R^{m \times m}$ and $Z = Z^T \in R^{n \times n}$ such that

$$\begin{bmatrix} Y & X \\ X^T & Z \end{bmatrix} \succeq 0, \quad \text{rank}(Y) + \text{rank}(Z) \leq 2r$$

$$\begin{bmatrix} \underset{X,Y,Z}{\text{minimize}} & \frac{1}{2}\text{rank}(\text{blkdiag}(Y, Z)) \\ \text{s.t.} & \begin{bmatrix} Y & X \\ X^T & Z \end{bmatrix} \succeq 0, \ X \in C \end{bmatrix}$$

rank of rectangular matrix → square

**Heuristics for Rank Minimization Problem** NP-hard.

Solutions can be categorized into 2 groups:

① Approximate the rank function (non-convex) with surrogate functions
- Nuclear norm
- Log-det

② Solve a sequence of rank-constrained feasibility problems
- Matrix factorization
- Rank constraint via convex iteration

# L16 cont'd.

## Method 1: Nuclear Norm

$$\begin{bmatrix} \underset{X}{\text{minimize}} & \|X\|_* \\ \text{s.t.} & X \in C \end{bmatrix} \xrightarrow{\text{if } X = X^T \succeq 0} \begin{bmatrix} \underset{X}{\text{minimize}} & \text{Tr}(X) \\ \text{s.t.} & X \in C \end{bmatrix}$$

$$\left(\|X\|_* = \sum_{i=1}^{r} \sigma_i\right)$$

**Intuition:** Nuclear norm = $L_1$-norm of singular value vector
→ sparse singular value vector
→ low rank

$\|X\|_*$ is the convex envelope of rank$(X)$ on $\{X \mid \|X\|_2 \le 1\}$

**Lemma:** For $X \in \mathbb{R}^{m \times n}$ and $t \in \mathbb{R}$, we have $\|X\|_* \le t$ iff there exist matrices $Y \in \mathbb{R}^{m \times m}$ and $Z \in \mathbb{R}^{n \times n}$ such that

$$\begin{bmatrix} Y & X \\ X^T & Z \end{bmatrix} \succeq 0, \quad \text{Tr}(Y) + \text{Tr}(Z) \le 2t$$

$$\begin{bmatrix} \underset{X,Y,Z}{\text{minimize}} & \frac{1}{2}\text{Tr}(Y+Z) \\ \text{s.t.} & \begin{bmatrix} Y & X \\ X^T & Z \end{bmatrix} \succeq 0, \; X \in C \end{bmatrix}$$

minimize nuclear norm of rectangular matrix ⇒ write in bigger form (square)
singular values → eigenvalues → trace

## Method 2: Log-det

$$\begin{bmatrix} \underset{X}{\text{minimize}} & \log\det(X + \delta I) \\ \text{s.t.} & X \in C \end{bmatrix}$$

**Intuition:** determinant = product of singular values

$$\begin{cases} \log\det(X+\delta I) = \sum_i \log(\sigma_i(X+\delta I)) \\ \text{rank}(X) = \|\sigma(X)\|_0 \end{cases}$$

⇒ log det can be seen as a surrogate function of card(s). but it is concave

↓

Iterative linearization & minimization (MM)

General for nonsquare X

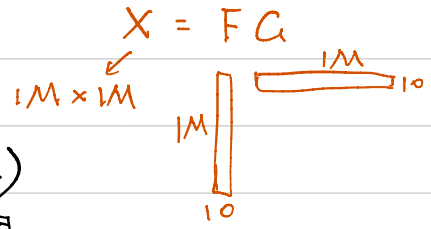
$$\begin{bmatrix} \underset{X,Y,Z}{\text{minimize}} & \log\det(\text{blkdiag}(Y,Z) + \delta I) \\ \text{s.t.} & \begin{bmatrix} Y & X \\ X^T & Z \end{bmatrix} \succeq 0, \; X \in C \end{bmatrix}$$

# Lib cont'd.

## Method 3: Matrix Factorization based Method.

< Previous 2 methods are not for big data (Large matrix) >

Low-rank Factorization: $X = FC$




$X = FC$ 本质上还是rank，对于 $(X, F, C)$ 是nonconvex；但对于 $(X, F)$ 或 $(X, C)$ 是convex

当含一个变量 fixed.

[We assume $X$ is convex, $F$ (or $C$) is affined by $X$ and both convex]

≈ BCD: $(F, C)$

↑ ↑

optimize alternatively

## Method 4: Rank Constraint via Convex iteration

semidefinite rank-constrained feasibility problem:

find $X$

s.t. $X \in C$, $X \succeq 0$ } convex prob.

$\text{rank}(X) \le r$ → nonconvex

$= \sum_{i=r+1}^{n} \lambda_i(X)$ = sum of smallest $n-r$ $\lambda_i(X)$

$$\left[\begin{array}{ll} \underset{X}{\text{minimize}} & \text{Tr}(W^*X) \\ \text{s.t.} & X \in C \\ & X \succeq 0 \end{array}\right] \rightleftharpoons \left[\begin{array}{ll} \underset{W}{\text{minimize}} & \text{Tr}(WX^*) \\ \text{s.t.} & \underset{①}{0} \preceq W \preceq \underset{②}{I} \\ & \text{Tr}(W) = n-r \end{array}\right]$$

$0 \preceq W \preceq I$ → $0 \le \lambda_i(W) \le 1$

$\text{Tr}(W) = n-r$ → $\sum \lambda_i(W) = n-r$

Intuition: align $X$ with nullspace of $W^*$

Intuition: Considering $X^*$ is semidefinite. $\lambda_1(X) \ge \lambda_2(X) \cdots$

(1) if only constraint ①: all $\lambda_i(W) = 0$

(2) if constraint ①, ③: $\lambda_n(W) = n-r$ $\lambda_{1\sim n-1}(W) = 0$

(3) if constraint ①②③: $\underbrace{\lambda_{r+1}(W) = \lambda_{r+2}(W) = \cdots = \lambda_n(W) = 1}_{n-r}$, $\lambda_{1\sim r}(W) = 0$

Minimize the smallest $n-r$ $\lambda_i(X)$ ⇌ Find the smallest $n-r$ $\lambda_i(X)$

↓

To smallest $n-r$ $\lambda_i(X) = 0$ ⇒ $\text{rank}(X) \le r$

# L17 Robust Optimization

## Robust Optimization:

Problem formulation: some problem contains parameters $\theta$ that are typically estimated in practice ($\hat{\theta}$) $\Rightarrow x^*(\hat{\theta}) \neq x^*(\theta)$

Several ways to make the problem robust to parameters errors:

manageable
① stochastic robust optimization (involving expectations) — 假设有概率分布 (普遍情况)
② worst-case robust optimization 太悲观
③ chance programming or chance robust optimization best but difficult (nonconvex)

## Stochastic robust optimization: Expectations

Instead of using approximated function $f(x;\hat{\theta})$, use $E_\theta[f(x;\theta)]$.

e.g. ① model the estimated value as $\theta = \hat{\theta} + \delta$, $\delta \sim N(0, \sigma)$

② consider $f(x;\hat{\theta}) = (\hat{c}^T x)^2$

$$\rightarrow E_\theta[f(x;\theta)] = E_\delta[((\hat{c}+\delta)^T x)^2]$$
$$= E_\delta[x^T \hat{c}\hat{c}^T x + x^T \delta\delta^T x]$$
$$= (\hat{c}^T x)^2 + x^T Q x \quad \text{regularizer}$$

Convexity: if $f(x;\theta)$ convex, then expectation (nonnegative sum) is convex

## Worst-case robust optimization

Assume that true parameter lies in an uncertainty region centered around the estimated value: $\theta \in \mathcal{U}$

关键: ① $\mathcal{U}$ 好解 ② $\mathcal{U}$ 不大

sephere region: $\mathcal{U} = \{\theta \mid \|\theta - \hat{\theta}\|_2 \leq \delta\}$

box region: $\mathcal{U} = \{\theta \mid \|\theta - \hat{\theta}\|_\infty \leq \delta\}$

elliptical region: $\mathcal{U} = \{\theta \mid (\theta-\hat{\theta})^T S^{-1} (\theta-\hat{\theta}) \leq \delta^2\}$

# L.17 cont'd

e.g. ① consider a sphere uncertainty region $\mathcal{U}=\{c \mid \|c-\hat{c}\|_2 \le \delta\}$

② consider $f(x;\hat{\theta}) = (\hat{c}^T x)^2$ if the function is the objective to be minimized or it is a constraint of the form $f(x;\hat{\theta}) \le 0$ ($\ge$) (maximized)

→ worst case: $\max_{c\in\mathcal{U}} |c^T x| = \max_{\|e\|\le\delta} |(\hat{c}+e)^T x|$

$\le \max_{\|e\|\le\delta} (|\hat{c}^T x| + |e^T x|)$

$\le |\hat{c}^T x| + \delta\|x\|$ ← the upper bound can achieve (maximum)

Cauchy-Schwarz inequality $|u^T v| \le \|u\|\|v\|$

equal iff $u=\alpha v$

要minimize的objective，考虑它worst case则是 min max

要是 constraint $\le 0$，考虑它的 worst case 是 max $\le 0$

一般要求 max 有闭式解，否则问题会复杂。

convexity: if $\mathcal{C}$ is convex set, $f$ is convex, then pointwise maximum is convex (一系列不同C的function)

## Application:

### 1) Robust Beamforming in Wireless Communications

sol 1: (no robust)

$$\max_w \text{SINR} = \frac{|w^H a|^2}{w^H R_{in} w} \quad \text{(signal power)} \quad \text{(inference \& noise power)}$$

a & $R_{in}$ are parameters required to be estimated

nonconvex / nonconcave but has hidden convexity

$$= \frac{w^H a a^H w}{w^H R_{in} w} = \frac{\tilde{w}^H R_{in}^{-1/2} a a^H R_{in}^{-1/2} \tilde{w}}{\tilde{w}^H \tilde{w}}$$

rank 1 matrix

$\tilde{w} = R_{in}^{1/2} w$ ; let it be 1 (normalize)

$\Rightarrow \tilde{w} \propto R_{in}^{-1/2} a$ ← 唯一非零 eigenvector

$$\max_x \; x^T A x \quad \text{s.t.} \; \|x\| = 1$$

x 找 A 最大 eigenvalue 对应的 eigenvector

(最小对应 min 问题)

# L17 cont'd

sol 2: (no robust)

$$\min_w \ w^H R_{in} w \quad s.t. \quad |w^H a| = 1 \quad \Leftrightarrow \quad \min_w \ w^H R_{in} w \quad s.t. \quad w^H a = 1 \quad \Big\} \text{QP. KKT}$$

$|w^H a| = 1$ → let it to be real

△ $w^H = w e^{j\phi}$ 不多1个维度 (对每个a有对应的$\phi$. 不适用于 robust 讨论一系列 a)

sol 2 : add robust

$$\mathcal{A} = \{c \mid c = \hat{a} + e \ , \ \|e\| \le \varepsilon\}$$

Problem: $\min_w \ w^H R w$ s.t. $|w^H a| = 1 \ , \ \forall a \in \mathcal{A}$ → $\min_w \ w^H R w$ s.t. $|w^H a| \ge 1 \ , \ \forall a \in \mathcal{A}$

一族 constraint

worst-case → $\min_a |w^H a| \ge |w^H \hat{a}| - \varepsilon \|w\|$

↓

只有 $\hat{a}$, 此时可用 $w^H = w e^{j\phi}$ trick

## 2) Robust Portfolio Optimization

(例如 LP/QP → SOCP)

加上 robustness, 形式上会变复杂

### ① Robust Global Maximum Return Portfolio Optimization

add regularizer

$$\max_w \ w^T \mu \ \ s.t. \ \ 1^T w = 1 \quad \Rightarrow \quad \max_w \ \min_{\mu \in \mathcal{U}_\mu} w^T \mu \ \ s.t. \ \ 1^T w = 1 \quad \Rightarrow \quad \max_w \ w^T \hat{\mu} - \kappa \|S^{1/2} w\|_2 \ \ s.t. \ \ 1^T w = 1$$

LP

(assume $\mathcal{U}_\mu = \{\mu = \hat{\mu} + \kappa S^{1/2} u \mid \|u\|_2 \le 1\}$)

SOCP (epigraph form)

### ② Robust Global Minimum Variance Portfolio Optimization

$$\min_w \ w^T \Sigma w \ \ s.t. \ \ 1^T w = 1 \quad \Rightarrow \quad \min_w \ \max_{\Sigma \in \mathcal{U}_\Sigma} w^T \Sigma w \ \ s.t. \ \ 1^T w = 1 \quad \Rightarrow \cdots \Rightarrow \quad \min_w \ \|\hat{X} w\|_2 + \delta_X \|w\|_2 \ \ s.t. \ \ 1^T w = 1$$

($\le t_1$, $\le t_2$)

QP

SOCP (epigraph form)

↓

Matrix version robustness

$$\hat{\Sigma} = \frac{1}{T} X^T X \ , \ X = \hat{X} + \Delta \ , \ \mathcal{U}_X = \{X \mid \|X - \hat{X}\|_F \le \delta_X\}$$

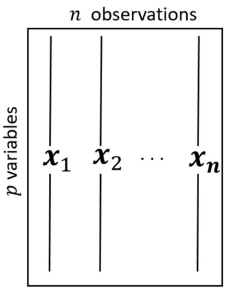
QP 变形, 都加上平方 (试了才知道)

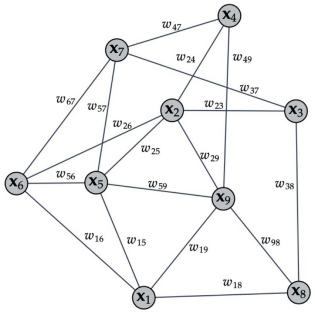
$$\min_w \ \|\hat{X} w\|_2^2 + \delta_X \|w\|_2^2 = w^T (\hat{X}^T \hat{X} + \delta_X I) w \quad s.t. \quad 1^T w = 1$$

# L 18 Graph Learning

## Graph Basics:

data matrix $X =$
$[x_1, x_2, \cdots, x_n] \in R^{p \times n}$




(有多种形式，一般对称且对角为0，非PSD)

weight adjacency matrix W

Degree matrix D contains the degrees $d_1, \cdots, d_p$ along the diagonal

$D = Diag(W1)$ i.e. $d_i = \sum_{j=1}^{p} W_{ij}$

Laplacian of a graph: $L = D - W$
① symmetric & PSD
② # zero eigenvalues = # connected components
③ measure the smoothness of that vector

$x^T L x = \frac{1}{2} \sum_{ij} W_{ij} (x_i - x_j)^2$ ← weighted with the graph weights

## Learning Graph from Data

learn W or L both OK.

### Method 1: Similarity function based (Learn W)

Intuition: decide the weight between two nodes based on their similarity

Various measures of similarity:

- Thresholded Euclidean distance graph: $\|x_i - x_j\|^2 \leq \gamma$, $W_{ij} = 0/1$
- Gaussian graph: $W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$
- k-NN graph
- Feature correlation graph: $W_{ij} = x_i^T x_j$
- Self-tuned Gaussian graph: $W_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma_i \sigma_j}\right)$ distance($x_j$, $x_j$'s kNN)

# L is cont'd

## Method 2: Smooth signal based

Intuition: find a $L$ that minimizes $\sum_{i=1}^{n} x_i^T L x_i = Tr(X^T L X)$

obj: $\underset{L}{\text{minimize}}\ Tr(X^T L X) + \gamma h(L)$ → regularization (all convex)

$\|L\|_1$ : sparsity, $\|L\|_F^2$ : energy

$-\log\det(L)$ : volume (∵ det is volume)

convex QP

$$\begin{aligned} \underset{L}{\text{minimize}}\ & Tr(X^T L X) + \gamma \|L\|_{F,\text{off}}^2 \quad \text{(off-diagonal)} \\ \text{s.t.}\ & \operatorname{diag}(L) = 1 \quad \text{(可选, 控制 degree)} \\ & L1 = 0,\ L_{ij} = L_{ji} \le 0 \quad \text{(Laplacian 性质)} \end{aligned}$$

all linear

写成 L

⇓

Linear + QP

$$\begin{aligned} \underset{L}{\text{minimize}}\ & Tr(WZ) + \gamma \|W\|_F^2 \\ \text{s.t.}\ & W1 = 1 \\ & \operatorname{diag}(W) = 0,\ W = W^T \ge 0 \end{aligned}$$

写成 W

$Z_{ij} = \|x_i - x_j\|^2$

solution: ignore $W = W^T$, the problem decomposes into each row;

at the end symmetrize by $W \leftarrow \frac{1}{2}(W + W^T)$.

$$\begin{aligned} \underset{L}{\text{minimize}}\ & z_i^T w_i + \gamma \|w_i\|^2 \\ \text{s.t.}\ & w_i^T 1 = 1,\ w_{ii} = 0,\ w_i \ge 0 \end{aligned}$$

→ KKT waterfilling

or

relax $w_i^T 1 = 1$

$$\begin{aligned} \underset{w}{\text{minimize}}\ & \tfrac{1}{2} z_i^T w_i - \alpha \log(1^T w_i) + \tfrac{\beta}{2}\|w_i\|^2 \\ \text{s.t.}\ & w_{ii} = 0,\ w_i \ge 0 \end{aligned}$$

L18 cont'd.

## Extension: Learning a k-connected graph

Goal: Given an initial noisy adjacency matrix $W_0$, infer a k-connected graph

= k smallest eigenvalues are zero

Define the Laplacian operator: $L(W) = Diag(W1) - W$

problem formulation (nonconvex)

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & \|W - W_0\|_F^2 \\ \text{s.t.} \quad & W1 = 1 \;,\; diag(W) = 0 \;,\; W = W^T \geq 0 \\ & rank(L(W)) = p - k \end{aligned}$$

relax constraint

$\lambda_1(L(W)) = \cdots = \lambda_k(L(W)) = 0$

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & \|W - W_0\|_F^2 + \beta \textstyle\sum_{i=1}^{k} \lambda_i(L(W)) \\ \text{s.t.} \quad & W1 = 1 \;,\; diag(W) = 0 \;,\; W = W^T \geq 0 \end{aligned}$$

Knowing that $\sum_{i=1}^{k} \lambda_i(X) = \left[\begin{array}{ll} \min_{F \in \mathbb{R}^{p \times k}} & Tr(F^T X F) \text{ (nonconvex for } (F, X)) \\ \text{s.t.} & F^T F = I \text{ (nonconvex)} \end{array}\right]$

BCD

1) fix F: convex

2) fix W: nonconvex ←

Li 1 beamforming

$$\begin{aligned} \underset{W}{\text{minimize}} \quad & \|W - W_0\|_F^2 + \beta\, Tr(F^T L(W) F) \\ \text{s.t.} \quad & W1 = 1 \;,\; diag(W) = 0 \;,\; W = W^T \geq 0 \\ & F^T F = I \end{aligned}$$

F 取 L(W) 最小 k 个 λi 对应 eigenvector

## Method 3: i.i.d model based (MLE based)

Details can be found in handout.

# L19~20 SDP relaxation

## BPSK Signal Detection Problem

channel matrix

Linear observation : $y = Hs + w$ → gaussian noise

received signal ↙ ↘ BPSK

problem formulation :

$$\begin{bmatrix} \underset{s}{\text{minimize}} & \|y - Hs\|_2^2 \\ \text{s.t.} & s \in \{\pm 1\}^n \end{bmatrix} \quad \begin{matrix} \rightarrow \text{convex} \\ \rightarrow \text{nonconvex} \end{matrix}$$

## Convex Relations (Relax 盡可能是 convex)

Type 1 : Box relaxation $\quad -1 \le s \le 1$

equivalent ↓

△ Type 2 : SDP relaxation

homogeneous form :

$$\begin{bmatrix} \underset{x}{\text{minimize}} & x^T L x \\ \text{s.t.} & x \in \{\pm 1\}^{n+1} \\ & \cancel{x_{n+1} = 1} \end{bmatrix} \rightarrow \text{can be removed}$$

$$L = \begin{bmatrix} H^TH & -H^Ty \\ -y^TH & y^Ty \end{bmatrix} \qquad x = \begin{bmatrix} s \\ 1 \end{bmatrix}$$

↓ equivalent

define $X = xx^T$ .

$X_{ii} = x_i^2 = 1$

$$\begin{bmatrix} \underset{X,x}{\text{minimize}} & \text{Tr}(LX) \\ \text{s.t.} & \text{diag}(X) = 1_{n+1} \\ & X = xx^T \end{bmatrix} \quad \begin{matrix} \rightarrow \text{linear} \\ \rightarrow \text{linear} \\ \rightarrow \text{nonconvex for } (X,x) \end{matrix}$$

↓ relaxation

$X = xx^T$ equals to $X \succeq 0$ and $\text{rank}(X) = 1$ ← nonconvex ;

remove rank constraint

$$\begin{bmatrix} \underset{X,x}{\text{minimize}} & \text{Tr}(LX) \\ \text{s.t.} & \text{diag}(X) = 1_{n+1} \\ & X \succeq 0 \end{bmatrix} \quad \text{SDP}$$

## Reconstruct Binary Solutions

Goal : Get $X$, reconstruct $x(s)$. If $\text{rank}(X) = 1$ then it means there is no relaxation (solved), which is not the common case.

✓ Method 1 : Simple Quantization $\quad X \rightarrow s \rightarrow \text{sign}(s)$

✓✓ Method 2 : Eigenvalue Decomposition take the eigenvector with the largest $\lambda$ as s

✓✓✓ Method 3 : Randomization $\quad X = xx^T$ ; use $X$ as the covariance matrix to generate random points, and keep the $\hat{x}$ with minimum objective value

L 19 ~ 20 cont'd

# Application: Multiuser Downlink Beamforming

problem formulation

$$\begin{aligned} \text{minimize}_{\{w_l\}} \quad & \sum_{l=1}^{L} \|w_l\|^2 \quad \rightarrow \text{overall transmitted power} \\ \text{s.t.} \quad & \frac{w_m^H R_{mm} w_m}{\sum_{l\neq m} w_l^H R_{ml} w_l + \sigma_m^2} \geq \rho_m \rightarrow \text{SINR constraint} \end{aligned}$$

## Method 1: SOCP formulation

trick: $w_m^H = w_m \cdot e^{j\phi}$ to make $w_m^H$ real

SOCP

$$\begin{aligned} \text{minimize}_{\{w_l\}} \quad & \sum_{l=1}^{L} \|w_l\|^2 \\ \text{s.t.} \quad & w_m^H h_m \geq \rho_m \|\tilde{R}_m^{1/2} \tilde{w}_m\| \\ & w_m^H h_m \geq 0 \end{aligned}$$

assume $R_{mm} = h_m h_m^H$

add $w_m^H h_m \geq 0$

## Method 2: SDP relaxation

define the rank-1 matrix: $X_l = w_l w_l^H$

notation: $w_l^H A w_l = Tr(w_l^H A w_l) = Tr(A w_l w_l^H) = Tr(A X_l) \triangleq A \bullet X_l$

SDP

$$\begin{aligned} \text{minimize}_{X_1 \cdots X_L} \quad & \sum_{l=1}^{L} I \bullet X_l \\ \text{s.t.} \quad & R_{mm} \bullet X_m - \rho_m \sum_{l\neq m} R_{ml} \bullet X_l \geq \rho_m \sigma_m^2 \quad \forall m \\ & X_l \succeq 0 \\ & \boxed{\text{rank}(X_l) = 1} \quad \text{ignore it} \rightarrow \text{convex problem} \end{aligned}$$

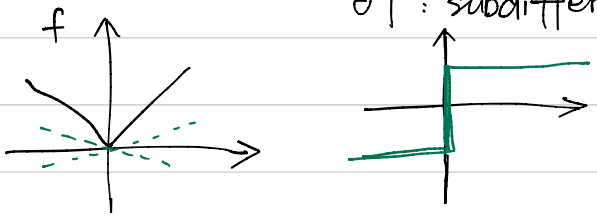
transfer nonconvexity in constraint into rank

# L21 Primal Dual Decomposition

**Subgradient** definition: $g$ is a subgradient of $f$ (not necessarily convex) at $x$ if

$$f(y) \geq f(x) + g^T(y-x)$$

support hyperplane.




→ $\partial f$ : subdifferential : a set of subgradients

**optimal conditions** Unconstrained Case : $\nabla f(x^*) = 0 \rightarrow 0 \in \partial f(x^*)$

Constrained Case : $\min f_0(x)$ s.t. $f_i(x) \leq 0$ $\xrightarrow{\text{(KKT. cond. 3)}}$ $0 \in \partial f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \partial f_i(x^*)$

**Subgradient Method** minimize a nondifferentiable convex function $f$

similar to gradient descent

Step 1 : $x^{(k+1)} = x^{(k)} - d_k g^{(k)}$ subgradient

Step 2 : $f_{best} = \min_{i=1 \cdots k} f(x^{(i)})$

not guarantee to decrease each step (keep the best)

guarantee to converge to a neighbourhood.

**Projected Subgradient Method** for constrained optimization

$\min f(x)$ s.t. $x \in X$ $\Rightarrow$ $x^{(k+1)} = [x^{(k)} - d_k g^{(k)}]_X$

step + project

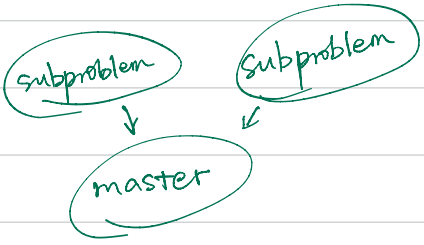
**Primal Decomposition** (for coupling variable)

problem: $\min_{x,y} f_1(x_1, y) + f_2(x_2, y)$ s.t. $x_1 \in X_1$ , $x_2 \in X_2$ , $y \in Y$

Step 1 : subproblem : fixed $y$

$$\begin{cases} f_1^*(y) = \min_{x_1 \in X_1} f_1(x_1, y) \\ f_2^*(y) = \min_{x_2 \in X_2} f_2(x_2, y) \end{cases}$$

Step 2 : master (primal) problem : $\min_{y \in Y} f_1^*(y) + f_2^*(y)$



# L21 cont'd.

To solve the master problem:

1. bisection (if $y$ is scalar)
2. gradient or Newton method (if $f_i^*$ differentiable)
3. subgradient, cutting-plane, or ellipsoid method "key"
   - ↳ projected subgradient method: $y(k+1) = [y(k) - d_k(s_1(k) + s_2(k))]_Y$

Calculate subgradients (free):

Lemma 1) Let $f^*(y)$ be the optimal value of the convex problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{s.t.} \quad & h_i(x) \le y_i \end{aligned}$$

A subgradient of $f^*(y)$ is $-\lambda^*(y)$

Dual Decomposition (coupling constraints)

problem:
$$\begin{aligned} \min_{x} \quad & f_1(x_1) + f_2(x_2) \\ \text{s.t.} \quad & x_1 \in X_1, \; x_2 \in X_2 \\ & h_1(x_1) + h_2(x_2) \le h_0 \end{aligned}$$

Partial Lagrangian $L(x, \lambda) = f_1(x_1) + f_2(x_2) + \lambda^T(h_1(x_1) + h_2(x_2) - h_0)$

↓

Max the dual function
$$\begin{aligned} g(\lambda) &= \inf_{x \in X} L(x, \lambda) \\ &= \inf_{x_1 \in X_1} \{f_1(x_1) + \lambda^T h_1(x_1)\} \\ &\quad + \inf_{x_2 \in X_2} \{f_2(x_2) + \lambda^T h_2(x_2)\} - \lambda^T h_0 \end{aligned}$$

Step 1: subproblem: fixed $y$

$$\begin{cases} g_1(\lambda) = \underset{x_1 \in X_1}{\text{minimize}} \; f_1(x_1) + \lambda^T h_1(x_1) \\ g_2(\lambda) = \underset{x_2 \in X_2}{\text{minimize}} \; f_2(x_2) + \lambda^T h_2(x_2) \end{cases}$$

Step 2: master dual problem: $\underset{\lambda \ge 0}{\text{maximize}} \; g_1(\lambda) + g_2(\lambda) - \lambda^T h_0$

→ projected subgradient method. $\lambda(k+1) = [\lambda(k) + d_k(s_1(k) + s_2(k) - h_0)]^+$

Calculate subgradients (free):

Lemma: Let $g(\lambda)$ be the dual function corresponding to the problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f_0(x) \\ \text{s.t.} \quad & h_i(x) \le 0, \quad i = 1, \cdots, m \end{aligned}$$

A subgradient of $g(\lambda)$ is $h(x^*(\lambda))$

# L21 cont'd.

benefit: no need for projection

(ADMM → dual decomposition → subgradient (solve) → non-differentiable)

## ADMM (Alternating Direction Method of Multipliers)

Background:

problem: $\begin{bmatrix} \underset{x}{\text{minimize}} & f(x) \\ \text{s.t.} & Ax = b \end{bmatrix}$

- Lagrangian: $L(x,y) = f(x) + y^T(Ax-b)$
- dual function: $g(y) = \inf_x L(x,y)$
- dual problem: $\underset{y}{\text{maximize}}\ g(y)$
- recover: $x^* = \underset{x}{\text{argmin}}\ L(x,y^*)$

Dual Ascent Idea: $x^{k+1} := \underset{x}{\text{argmin}}\ L(x,y^k)$

$y^{k+1} := y^k + \rho^k \nabla g(y^k) = y^k + \rho^k(Ax^{k+1}-b)$

## ADMM:

problem: $\begin{bmatrix} \underset{x,z}{\text{minimize}} & f(x)+g(z) \\ \text{s.t.} & Ax+Bz=c \end{bmatrix}$

Augmented Lagrangian:

$$L_\rho(x,z,y) = f(x)+g(z)+\langle y, Ax+Bz-c\rangle + \frac{\rho}{2}\|Ax+Bz-c\|_F^2$$

add regularization to dual decomposition (faster convergence) 控制住 feasible set 靠

ADMM:

$$x^{k+1} := \underset{x}{\text{argmin}}\ L_\rho(x,z^k,y^k)$$
$$z^{k+1} := \underset{z}{\text{argmin}}\ L_\rho(x^{k+1},z,y^k)$$
$$y^{k+1} := y^k + \rho(Ax^{k+1}+Bz^{k+1}-c)$$

no need for projection

Convergence: iterates approach feasibility: $Ax^k+Bz^k-c \to 0$

(assume convex f,g) objective approaches optimal value: $f(x^k)+g(z^k) \to p^*$

ok for non-differentiable

speed: slow → high accuracy; efficient → moderate accuracy

but can extend to nonconvex

L21 cont'd.

Example 1: Robust PCA

convex but not differentiable

$$\begin{bmatrix} \underset{L,S}{\text{minimize}} & \|L\|_* + \lambda \|S\|_1 \\ \text{s.t.} & L + S = M \end{bmatrix}$$

Example 2: Graphical Lasso

Precision matrix estimation from Gaussian samples

$$\begin{bmatrix} \underset{\Theta}{\text{minimize}} & \underbrace{-\log\det\Theta + \langle \Theta, S \rangle}_{\text{neg. log likelihood}} + \lambda \|\Theta\|_1 \\ \text{s.t.} & \Theta \succ 0 \end{bmatrix}$$

引入slack variable 方便构造ADMM

$$\begin{bmatrix} \underset{\Theta}{\text{minimize}} & -\log\det\Theta + \langle \Theta, S \rangle + \lambda \|\Psi\|_1 \\ \text{s.t.} & \Theta \succ 0,\ \Theta = \Psi \end{bmatrix}$$

# Summary : Algorithms

## Unconstrained Optimization:

1. Descent Method ; Gradient Descent Method
2. Newton's Method.

## Constrained Optimization:

1. Equality Constrained Optimization

general:
2. Gradient Projection Method.
3. Interior - Point Method. (IPM)

block:
4. Block Coordinate Descent (BCD) sequential
5. Jacobi Algorithm parellel

difficult problem:
6. Majorization - Minimization (MM)
7. Successive Convex Approximation (SCA) parallel

block:
8. Block Majorization - Minimization (BMM) sequential

coupling:
9. Primal Decomposition
10. Dual Decomposition
11. Alternating Direction Method of Multipliers (ADMM)

# Extension: PCA and eigenvalue

## What is PCA ?

PCA tries to find the direction such that the projection of the data on it has the highest possible variance.

Sample covariance matrix : $S = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^T = X^T X/n$

which is a PSD matrix with an eigenvalue decomposition : $S = Q\Lambda Q^T$

$(\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0)$

PCA seeks to solve a sequence of optimization problem,

the first is : $\left[\text{maximize}_{u} \ \frac{u_1^T S u_1}{u_1^T u_1}\right], u \in R^p$

常见形式 (见于LiT)

$=$

$\left[\begin{array}{ll}\text{maximize}_{u} & u_1^T S u_1 \\ \text{s.t.} & u_1^T u_1 = 1\end{array}\right]$

$u^T S u = u^T Q\Lambda Q^T u = w^T \Lambda w = \sum_{i=1}^{p} \lambda_i w_i^2$

$\|w\|_2 = \|Q^T u\|_2 = \|u\|_2 = 1$ (rotation)

$\rightarrow \left[\begin{array}{ll}\text{maximize}_{w} & \sum_{i=1}^{p} \lambda_i w_i^2 \\ \text{s.t.} & \sum_{i=1}^{p} w_i^2 = 1\end{array}\right] \rightarrow w = (1, 0, \cdots 0)^T = e_1$

$u_1 = Qw = q_1$

$\therefore$ the first principal component is the eigenvector w.r.to. the largest eigenvalue.

the remaining components are found by solving the sequence of optimization problems :

$\left[\begin{array}{ll}\text{maximize}_{u} & u_i^T S u_i \\ \text{s.t.} & u_i^T u_i = 1 \\ & u_i^T u_j = 0 \ . \ \forall 1 \leq j \leq i\end{array}\right]$

$\rightarrow u_i = q_i$

note: rank and # eigenvalue — if A is an nxn matrix, rank(A) + null(A) = n

$\because$ null(A) is the dimension of the kernel of the matrix (v)

$Av = 0 = 0v$

The kernel of A is precisely the eigenspace corresponding to eigenvalue 0

$\therefore$ rank(A) = n − eigenspace of A corresponding to eigenvalue 0