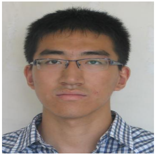
Efficient Algorithms For Low-rank Matrix Optimization

Zaiwen Wen

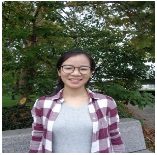
Beijing International Center For Mathematical Research
Peking University

References/Coauthors in our group or alumnus

(a) 李勇锋



(b) 刘浩洋



(c) 段雅琦

- Li Yongfeng, Liu Haoyang, Wen Zaiwen, and Yuan Yaxiang, **Low-rank Matrix Optimization Using Polynomial-filtered Subspace Extraction**, SIAM Journal on Scientific Computing, accepted
- Duan Yaqi, Wang Mengdi, Wen Zaiwen, Yuan Yaxiang; **Adaptive Low Rank Approximation for State Aggregation of Markov Chain**, SIAM Journal on Matrix Analysis and Applications, Vol 41, No. 1, pp. 244-278, 2020

Outline

1. Optimization Using Polynomial-filtered Subspace Extraction
   - Motivation
   - Polynomial Filtered Algorithms
   - Examples: PFPG and PFAM
   - Convergence Analysis
   - Numerical Results

2. Low-Nonnegative-Rank Approximation for State Aggregation
   - Motivation
   - Problem Setup
   - Low-Nonnegative-Rank Approximation
   - Numerical Results

Eigenvalue Computations in Matrix Optimization

- Consider matrix optimization problems with eigenvalue decompositions (EVD).

- Commonly used algorithm (formal):

$$x^{k+1} = \mathcal{T}(x^k, \text{EVD of } \mathcal{B}(x^k)),$$

where $\mathcal{B} \ : \ \mathcal{D} \to \mathcal{S}^n$.

Problem	EVD type
Semi-definite opt.	All positive / negative eigenvalues
Nuclear norm	r largest eigenvalues/singular values
Maximal eigenvalue opt.	Max eigenvalue in magnitude

Table: Eigenvalue computation in matrix optimization.

Application: Matrix Rank Minimization

- nuclear norm minimization:

$$\min \ \|X\|_* \ \text{ s.t. } \mathcal{A}(X) = b$$

where $\|X\|_* = \sum_i \sigma_i$ and $\sigma_i = i$th singular value of matrix X.
Linearized Bregman method:

$$\begin{aligned} V^{k+1} &:= V^k - \tau \mathcal{A}^*(\mathcal{A}(X^k) - b) \\ X^{k+1} &:= \mathrm{prox}_{\tau\mu}(V^{k+1}) \end{aligned}$$

- Unconstrained Nuclear Norm Minimization:

$$\min \ F(X) := \mu\|X\|_* + \frac{1}{2}\|\mathcal{A}(X) - b\|_2^2.$$

Proximal gradient method (g is the gradient of $\frac{1}{2}\|\mathcal{A}(X) - b\|_2^2$):

$$\begin{aligned} X^{k+1} &= \arg\min_X \ \mu\|X\|_* + \langle g^k, X - X^k\rangle + \frac{1}{2\tau}\|X - X^k\|_F^2 \\ &= \arg\min_X \ \mu\|X\|_* + \frac{1}{2\tau}\|X - (X^k - \tau g^k)\|_F^2 \end{aligned}$$

Application: Maximal Eigenvalue Problem

- Maximal eigenvalue problem:

$$\min_{x \in \mathcal{D}} \lambda_1(\mathcal{A}^*(y)).$$

- Widely used in: **phase recovery**, **blind deconvolution**, and **max-cut problems**.
- The (sub-)gradient is

$$g = \mathcal{A}(USU^T),$$

  where U spans the eigenspace of $\lambda_1(\mathcal{A}^*(x))$, $S \succeq 0$ and $\mathrm{Tr}(S) = 1$.

Application: Max cut

- Max-cut problem:

$$\max x^T Cx, \quad \text{s.t. } x_i \in \{-1, +1\}.$$

- Max-cut SDP:

$$(P) \begin{array}{ll} \max & \langle C, X \rangle, \\ \text{s.t.} & X_{ii} = 1, X \succeq 0 \end{array} \qquad (D) \begin{array}{ll} \min & n\lambda + \mathbf{1}^T y, \\ \text{s.t.} & \lambda = \lambda_{\max}(C - \mathrm{Diag}(y)) \end{array}$$

- Graphs are HUGE! Is it possible to solve huge-scale SDPs?
- **Idea**: attack the dual problem → Requires **EVD**.

Application: Nearest Correlation Matrix (NCM)

- NCM problem (primal)

$$\begin{array}{rl} \min & \frac{1}{2}\|G - X\|_F^2, \\ \text{s.t.} & X_{ii} = 1, \\ & X \succeq 0. \end{array}$$

- NCM problem (dual)

$$\min \frac{1}{2}\|\mathcal{P}_{\mathcal{K}}(G + \mathrm{Diag}(x))\|_F^2 - \mathbf{1}^T x,$$

  where $\mathcal{K}$ is the PSD matrix cone.

- The gradient is

$$\nabla F(x) = \mathcal{P}_{\mathcal{K}}(G + \mathrm{Diag}(x)) - \mathbf{1},$$

Application: ADMM for SDP

- Consider the standard SDP in the dual form:

$$\begin{aligned} \min_{y,S} \quad & b^T y, \\ \text{s.t.} \quad & S = \mathcal{A}^*(y) - C, \\ & S \succeq 0. \end{aligned}$$

- ADMM method for dual SDP

$$\begin{aligned} y^{k+1} &:= \arg\min_{y} L(y, S^k, X^k), \\ S^{k+1} &:= \arg\min_{S \succeq 0} L(y^{k+1}, S, X^k), \\ X^{k+1} &:= X^k + \mu(S^{k+1} - \mathcal{A}^*(y^{k+1}) + C). \end{aligned}$$

- To update S^k one needs all positive eigenvalues and corresponding eigenvectors.

Motivation

Possible Issues:

- At least 1 EVD/Update on a different matrix.
- High EVD cost ($\sim$ 90%).
- Not all eigenvalues are needed – all positive/negative ones, first p eigenvalues, maximal one, etc.

Room for Improvement:

- The update information usually lies in a low-dimensional subspace. Can we find the space?
- The matrices in two consecutive iterations are quite close: the eigenvalues & eigenvectors should be similar?

Subspace Assumption

- Original method

$$x^{k+1} = \mathcal{T}(x^k, \text{EVD of } \mathcal{B}(x^k)),$$

- **Assumption: the update information lies in an eigen-subspace**

$$x^{k+1} = \mathcal{T}(x^k, \text{EVD of } \mathcal{P}_{V^k}(\mathcal{B}(x^k))).$$

- Projection of A onto a subspace spanned by **orthogonal** $V \in \mathbb{R}^{n \times r}$:

$$\mathcal{P}_V(A) = VV^T AVV^T.$$

- If V^k is known and r is small, the EVD on the projected matrix is quite simple.

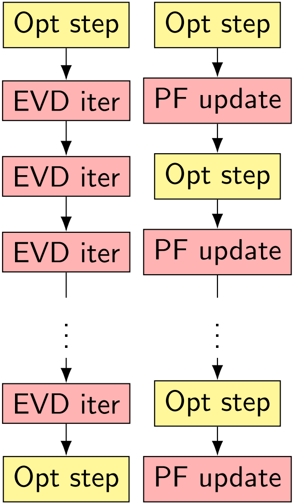
Polynomial-filtered Update

- How to approximate V^k – **By subspace extraction**.
- From any starting matrix U^k, the polynomial-filtered subspace extraction (of $\mathcal{B}(x^k)$) is:

$$U^{k+1} = \textbf{orth}(\rho(\mathcal{B}(x^k))U^k).$$

- Polynomial-filtered update (formal):

$$\begin{aligned} x^{k+1} &= \mathcal{T}(x^k, \text{EVD of } \mathcal{P}_{U^k}(\mathcal{B}(x^k))), \\ U^{k+1} &= \textbf{orth}(\rho(\mathcal{B}(x^{k+1}))U^k). \end{aligned}$$



Opt step

EVD iter

EVD iter

EVD iter

⋮

EVD iter

Opt step

Opt step

PF update

Opt step

PF update

⋮

Opt step

PF update

Composite Convex Program

- Optimization model:

$$\min F(x) + R(x).$$

- $F(x) = f \circ \lambda(\mathcal{B}(x))$ and the outer function is a **spectral function**. This means given x, first use $\mathcal{B}$ to obtain a matrix. Then the function value only depends on the eigenvalues of $\mathcal{B}(x)$.
- $R(x)$ is a convex regularizer.
- $\mathcal{B} : \mathbb{R}^m \to \mathcal{S}^n$: matrix-valued operator

$$\mathcal{B}(x) = G + \mathcal{A}^*(x),$$

where $\mathcal{A}^*(x)$ is a linear operator.

Proximal Gradient method

- The gradient of F is

$$\nabla F(x) = \mathcal{A}(\Psi(\mathcal{B}(x))).$$

  Ψ is a matrix-valued operator:

$$\Psi(X) = V\mathrm{Diag}(\nabla f(\lambda(X)))V^T.$$

  V contains all eigenvectors of X, Ψ is assumed to be Lipschitz continuous.

- *proximal mapping*:

$$\mathrm{prox}_{th}(x) := \arg\min_{u} \{h(u) + \frac{1}{2t}\|u - x\|^2\}.$$

- Proximal Gradient method:

$$x^{k+1} = \mathrm{prox}_{\tau_k R}(x^k - \tau_k \nabla F(x^k))),$$

Low-rank Assumption

Assumption

Suppose $\Omega \subset \mathbb{R}^m$ is a subset and $x^* \in \Omega$. Let $\mathcal{I}(x) \subset [n]$ be an integer set with $\mathcal{I}(x) = \{s, s+1, \ldots, t\}$. For all $x \in \Omega$, $\nabla F(x)$ has the form

$$\nabla F(x) = \mathcal{A}(\Psi(V_{\mathcal{I}} V_{\mathcal{I}}^T \mathcal{B}(x) V_{\mathcal{I}} V_{\mathcal{I}}^T)),$$

where $V_{\mathcal{I}} \in \mathbb{R}^{n \times |\mathcal{I}|}$ contains all $v_i, i \in \mathcal{I}(x)$.

- This assumption essentially means that computing f and ∇f only involves a small number of eigenvalues of $\mathcal{B}(x)$ for some $x \in \mathbb{R}^m$.
- $V_{\mathcal{I}} V_{\mathcal{I}}^T \mathcal{B}(x) V_{\mathcal{I}} V_{\mathcal{I}}^T$ is actually the projection of $\mathcal{B}(x)$ on the subspace spanned by $V_{\mathcal{I}}$. Thus another interpretation is that if one feeds either $\mathcal{B}(x)$ or the projection of $\mathcal{B}(x)$ on $V_{\mathcal{I}}$, he will get the same result.

Polynomial Filtered Proximal Gradient (PFPG) Method

- Under the low-rank assumption, the proximal gradient method is

$$x^{k+1} = \mathrm{prox}_{\tau_k R}(x^k - \tau_k \mathcal{A}(\Psi(V^k (V^k)^T \mathcal{B}(x^k) V^k (V^k)^T))),$$

- How to compute V^k (contains all eigenvalues in $\mathcal{I}(x^k)$)? By Chebyshev polynomial filters.

$$U^k = \mathbf{orth}(\rho_k(\mathcal{B}(x^k))U^{k-1}).$$

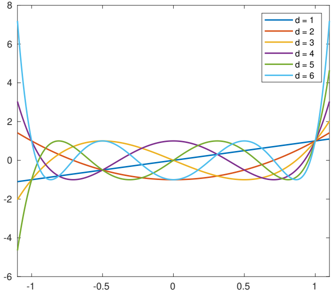
  We use U^k instead of V^k because U^k is only an approximation of V^k.

- The polynomial-filtered proximal gradient method (PFPG) is

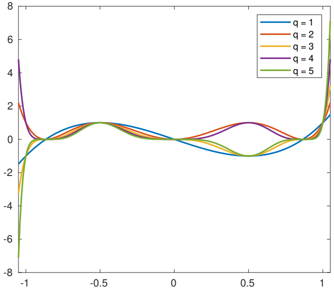
$$\begin{aligned} x^{k+1} &= \mathrm{prox}_{\tau_k R}(x^k - \tau_k \mathcal{A}(\Psi(U^k (U^k)^T \mathcal{B}(x^k) U^k (U^k)^T))), \\ U^{k+1} &= \mathbf{orth}(\rho_k^{q_k}(\mathcal{B}(x^{k+1}))U^k). \end{aligned}$$

Chebyshev Polynomials

- Bounded within $[-1, 1]$, fast growth beyond $[-1, 1]$.



(d) Chebyshev polynomials of the first kind $T_d(t)$



(e) Power of Chebyshev polynomial $T_3^q(t)$

- Chebyshev polynomials suppress all eigenvalues in an interval to $[-1, 1]$ while greatly amplifying the eigenvalues beyond the interval.

Evaluating the Gradient

- In practice, never evaluate $\Psi(U^k(U^k)^T\mathcal{B}(x^k)U^k(U^k)^T)$ directly by its definition.
- Evaluating Ψ on projected $\mathcal{B}(x^k)$ is essentially a Rayleigh-Ritz (RR) procedure inserted with a single ∇f evaluation.
  1. Compute $H^k = (U^k)^T\mathcal{B}(x^k)U^k$.
  2. Compute full eigenvalue decomposition on $H^k = W^k D^k (W^k)^T$.
     Note: H^k is a small matrix thus full EVD is fine.
     $(U^k W^k)D^k(U^k W^k)^T$ is just exact EVD of projected $\mathcal{B}(x)$.
  3. Feed $d^k := \mathrm{diag}(D^k)$ into ∇f to obtain a truncated $\nabla f(d^k)$.
  4. Finally evaluate
     $$(U^k W^k)\mathrm{Diag}(\nabla f(d^k))(U^k W^k)^T.$$

The Polynomial-filtered ADMM (PFAM) Method

- Consider the standard SDP:

$$\begin{array}{ll} \min & \langle C, X\rangle, \\ \text{s.t.} & \mathcal{A}X = b, X \succeq 0. \end{array}$$

- Let $F(X) = 1_{\{X \succeq 0\}}(X)$ and $G(X) = 1_{\{\mathcal{A}X=b\}}(X) + \langle C, X\rangle$.
- The Douglas-Rachford Splitting (DRS) method can be written as

$$Z^{k+1} = T_{\mathrm{DRS}}(Z^k),$$

where

$$T_{\mathrm{DRS}} = \mathrm{prox}_{tG}(2\mathrm{prox}_{tF} - I) - \mathrm{prox}_{tF} + I,$$

which is equivalent to the ADMM on the dual problem.

- The explicit forms of $\mathrm{prox}_{tF}(Z)$ and $\mathrm{prox}_{tG}(Y)$

$$\begin{aligned} \mathrm{prox}_{tF}(Z) &= \mathcal{P}_+(Z), \\ \mathrm{prox}_{tG}(Y) &= (Y + tC) - \mathcal{A}^*(\mathcal{A}\mathcal{A}^*)^{-1}(\mathcal{A}Y + t\mathcal{A}C - b), \end{aligned}$$

where $\mathcal{P}_+(Z)$ is the projection operator onto $\{X \succeq 0\}$.

PLAM

- DRS/ADMM:

$$T_{\mathrm{DRS}} = \mathrm{prox}_{tG}(2\mathrm{prox}_{tF} - I) - \mathrm{prox}_{tF} + I,$$

- The the polynomial-filtered alternating direction method of multipliers (PFAM) can be written as

$$\begin{array}{rcl} Z^{k+1} & = & \mathrm{prox}_{tG}(2\mathcal{P}_+((U^k(U^k)^T(Z^k)(U^k(U^k)^T) - Z^k) \\ & & -\mathcal{P}_+((U^k(U^k)^T(Z^k)(U^k(U^k)^T) + Z^k, \\ U^{k+1} & = & \mathrm{orth}(\rho_{k+1}^{q_{k+1}}(Z^{k+1})U^k), \end{array}$$

where $U^k \in \mathbb{R}^{n\times p}$ is an orthogonal matrix and $q_k \geq 1$ is a small integer.

Convergence Analysis of the PFPG Method

Assumption

- $\| \sin \Theta(V_{\mathcal{I}_{k+1}}^{k+1}, U^k)\|_2 < \gamma, \ \forall \ k$ with $\gamma < 1$.
- The iteration sequence are bounded, i.e., $\|x^k\|_2 \le C, \ \forall \ k$.
- The relative gap has a lower bound, i.e., $\mathcal{G}_k \ge l, \ \forall \ k$.

Conclusion

- If $\tau_k = \tau \le \frac{1}{L}$ and let $\bar{x}_K = \frac{1}{K} \sum_{k=1}^{K} x^k$ then to achieve the convergence $\lim_{K \to \infty} h(\bar{x}^K) = h(x^*)$, we only need that the degree of polynomials satisfies

$$d_k = \Omega\left(\frac{\log k}{\min\{l, 1\}}\right).$$

Convergence Analysis of the PFPG Method

Assumption

- $\|\sin\Theta(V_{\mathcal{I}_{k+1}}^{k+1}, V_{\mathcal{I}_k}^{k})\|_2 \leq c_1\|x^{k+1}-x^k\|_2$ for all k.

If the exact proximal gradient method on $h(x)$ has a linear convergence rate, i.e.,

$$\mathrm{dist}(\mathrm{prox}_{\tau R}(x^k-\tau_k\nabla F(x^k)),\mathcal{X}) \leq \nu\mathrm{dist}(x^k,\mathcal{X}), \nu \in (0,1).$$

If η_{k+1} satisfies

$$\frac{\nu+\eta_{k+1}^{q_k}c_3}{2}+\sqrt{\left(\frac{\nu+\eta_{k+1}^{q_k}c_3}{2}\right)^2+\eta_{k+1}^{q_k}(\tau_k c_2 c_4-\nu c_3)} < \rho < 1,$$

where c_2, c_3, c_4 are constants and $\eta_k = \frac{\rho_k(\lambda_{s_{p+1}}(\mathcal{B}(x^k))}{\rho_k(\lambda_{s_p}(\mathcal{B}(x^k))} < 1$ is the ratio of the $(p+1)$-th and the p-th eigenvalue of $\rho_k(\mathcal{B}(x^k))$, then the PFPG method has a linear convergence rate.

Convergence Analysis of the PFAM Method

Assumption

- $\| \sin \Theta(V_{\mathcal{I}_{k+1}}^{k+1}, U^k)\|_F < \gamma, \ \forall \ k$ with $\gamma < 1$.
- The iteration sequence is bounded, i.e., $\|Z^k\|_F \leq C, \ \forall \ k$.
- The relative gap has a lower bound, i.e., $\mathcal{G}_k \geq l, \ \forall \ k$.

To achieve the convergence $\|Z^k - T_{\mathrm{DRS}}(Z^k)\|_F = o(1/\sqrt{k})$, we only need that

$$d_k = \Omega(\frac{\log k}{\min\{l, 1\}}).$$

Nearest Correlation Matrix Problem

- Find a correlation matrix X nearest to G:

$$\begin{array}{ll} \text{min} & \|X - G\|_F^2, \\ \text{s.t.} & X_{ii} = 1, X \succeq 0. \end{array}$$

n	Grad			**PFPG (Ours)**			Newton		
	time	iter	$\|g\|$	time	iter	$\|g\|$	time	iter	$\|g\|$
500	0.9	33	7.4e-08	0.7	43	1.3e-08	**0.6**	8	1.8e-07
1000	3.8	43	2.0e-08	**1.1**	54	3.0e-08	1.6	9	1.3e-07
1500	11.3	54	2.6e-08	**2.4**	65	7.1e-08	4.0	10	1.0e-07
2000	22.6	54	4.3e-08	**5.0**	76	8.3e-08	7.2	10	9.0e-08
2500	60.9	87	3.6e-08	**10.6**	120	4.2e-08	12.1	10	8.0e-08
3000	104.0	87	6.2e-08	**16.1**	129	7.8e-08	19.3	10	7.4e-08
4000	278.0	91	7.9e-08	**32.8**	142	7.3e-08	44.2	10	6.4e-08

Table: Results of Grad/PFPG/Newton on random-generated data.

Matrix Completion

- Penalized form of the matrix completion problem:

$$\min \|X\|_* + \frac{1}{2\mu}\|\mathcal{P}_\Omega(X) - \mathcal{P}_\Omega(M)\|_F^2.$$

- Use **NNLS** algorithm (essentially an accelerated proximal gradient method). The main cost is the truncated SVD of a matrix

$$A^k = \beta_1 U^k (V^k)^T - \beta_2 U^{k-1}(V^{k-1})^T - \beta_3 G^k,$$

No.	Name	(m, n)	Non-zeros	sparsity
1	jester-1	(24983, 100)	249830	10%
2	jester-2	(23500, 100)	235000	10%
3	jester-3	(24938, 100)	249380	10%
4	moive-100K	(943, 1682)	49918	3.2%
5	moive-1M	(6040, 3706)	498742	2.2%
6	moive-10M	(71567, 10677)	4983232	0.7%

Table: Matrix completion test data.

Matrix Completion (Result)

	NNLS-LANSVD				NNLS-SLRP				**PFNNLS (Ours)**			
No.	iter	svp	time	mse	iter	svp	time	mse	iter	svp	time	mse
1	26	93	10.5	1.64e-1	27	69	4.6	1.76e-1	24	84	**2.3**	1.80e-1
2	26	93	9.1	1.65e-1	26	79	4.3	1.72e-1	25	88	**2.1**	1.80e-1
3	24	83	7.1	1.16e-1	27	74	4.6	1.24e-1	24	84	**2.0**	1.30e-1
4	34	100	4.2	1.28e-1	35	100	0.8	1.26e-1	36	100	**0.6**	1.23e-1
5	50	100	40.6	1.42e-1	50	100	10.8	1.43e-1	51	100	**7.4**	1.42e-1
6	54	100	620.1	1.26e-1	57	100	179.9	1.27e-1	52	100	**92.7**	1.27e-1

Figure: Results of NNLS on real examples.

Phase Retrieval

- Phase retrieval as constrained maximal eigenvalue problem

$$\begin{aligned} \min_x \quad & \lambda_1(\mathcal{A}^*(x)), \\ \text{s.t.} \quad & \langle b, x\rangle - \|x\|_* \geq 1. \end{aligned}$$

No.	name	size	No.	name	size
1	giantbubble(L)	1200×1140	2	nebula(L)	1600×1350

Table: Image data for 2D signals.

	GAUGE				**PFGAUGE (Ours)**			
No.	time	iter	DFT	gap	time	iter	DFT	gap
1	19610.56	8	1e+06	6.0e-01	**3892.26**	6	2e+05	4.7e-06
2	21958.19	5	8e+05	1.7e-01	**4042.50**	24	1e+05	4.8e-06

Table: Phase retrieval comparisons on 2D real signal. GAUGE solver by M. P. Friedlander (essentially PG). Note: GAUGE does not converge in 18000s!

ADMM for SDP: 2-RDM Problems

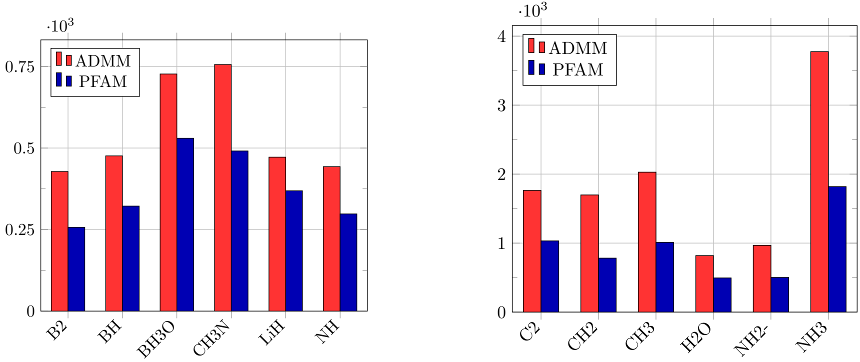


Figure: Time consumption (seconds): **ADMM** v.s. **PFAM (Ours)**.

- ADMM and PFAM attain similar accuracy.
- PFAM has $\sim 2\times$ speedup. More effective on large & low-rank data.

Outline

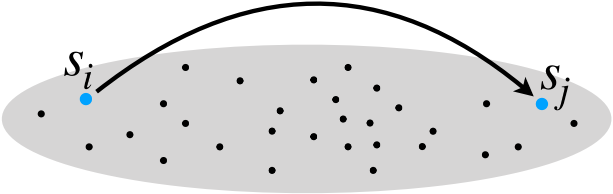
1. Optimization Using Polynomial-filtered Subspace Extraction
   - Motivation
   - Polynomial Filtered Algorithms
   - Examples: PFPG and PFAM
   - Convergence Analysis
   - Numerical Results

2. Low-Nonnegative-Rank Approximation for State Aggregation
   - Motivation
   - Problem Setup
   - Low-Nonnegative-Rank Approximation
   - Numerical Results

Motivation

- Control theory, reinforcement learning, etc.
  Model real-world systems by Markov chains.
- Markov chain with a discrete state space

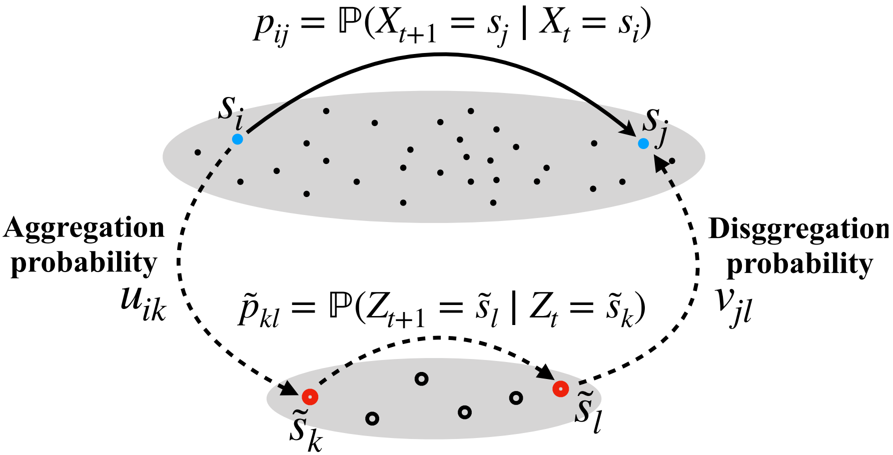
$$p_{ij} = \mathbb{P}(X_{t+1} = s_j \mid X_t = s_i)$$



$$\mathcal{S} = \{s_1, s_2, \cdots, s_d\}$$

The ambient dimention d is too large!

Model Reduction by State Aggregation



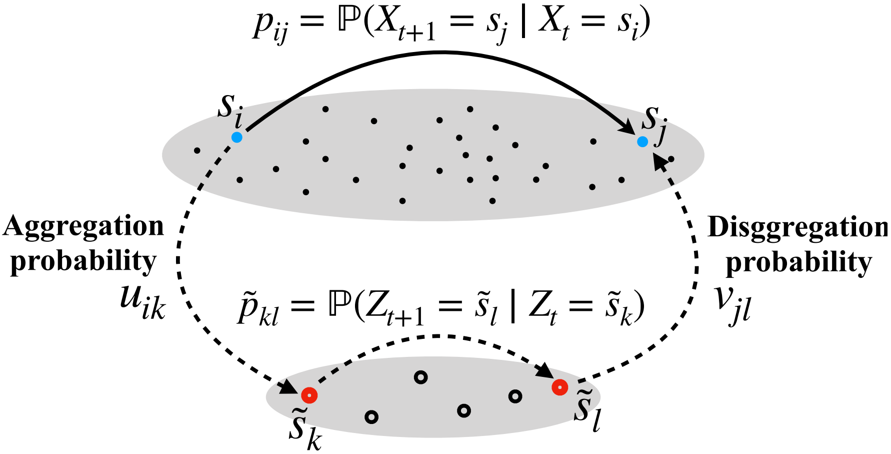
$p_{ij} = \mathbb{P}(X_{t+1} = s_j \mid X_t = s_i)$
s_i
s_j
Aggregation probability
u_{ik}
Disggregation probability
v_{jl}
$\tilde{p}_{kl} = \mathbb{P}(Z_{t+1} = \tilde{s}_l \mid Z_t = \tilde{s}_k)$
$\tilde{s}_k$
$\tilde{s}_l$

$\mathcal{S} = \{s_1, s_2, \cdots, s_d\} \mapsto$ meta-states in $\tilde{\mathcal{S}} = \{\tilde{s}_1, \cdots, \tilde{s}_r\}$

Aggregation probability $u_{ik} = \mathbb{P}(Z_t = \tilde{s}_k \mid X_t = s_i)$

Disaggregation probability $v_{jl} = \mathbb{P}(Z_{t+1} = \tilde{s}_l \mid X_{t+1} = s_j)$

Model Reduction by State Aggregation



$$p_{ij} = \mathbb{P}(X_{t+1} = s_j \mid X_t = s_i)$$

Aggregation probability u_{ik}

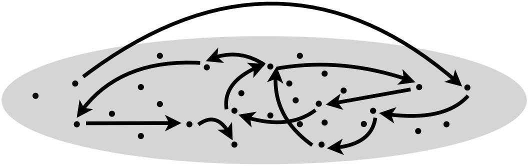
Disggregation probability v_{jl}

$$\tilde{p}_{kl} = \mathbb{P}(Z_{t+1} = \tilde{s}_l \mid Z_t = \tilde{s}_k)$$

$$p_{ij} = \sum_{k,l=1}^{r} u_{ik} \tilde{p}_{kl} v_{jl} \;\Rightarrow\; \text{Martrix form } P = U \tilde{P} V^T$$

Problem Setup

Observe a trajectory $(i_0, i_1, \cdots, i_n)$

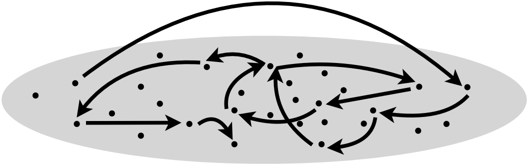


$$\mathcal{S} = \{s_1, s_2, \cdots, s_d\}$$

Driven by an unknown probability transition matrix $P^* \in \mathbb{R}^{d \times d}$

Problem Setup

Observe a trajectory $(i_0, i_1, \cdots, i_n)$



$$\mathcal{S} = \{s_1, s_2, \cdots, s_d\}$$

Driven by an unknown probability transition matrix $P^* \in \mathbb{R}^{d \times d}$

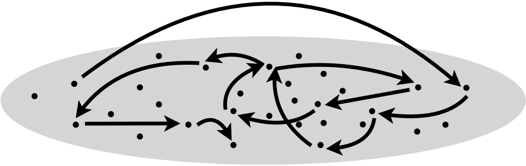
Recover

$$\longrightarrow P^* = U^* \tilde{P}^* (V^*)^T$$

(up to linear transformation).

Problem Setup

Observe a trajectory $(i_0, i_1, \cdots, i_n)$



$$\mathcal{S} = \{s_1, s_2, \cdots, s_d\}$$

Driven by an unknown probability transition matrix $P^* \in \mathbb{R}^{d \times d}$

→

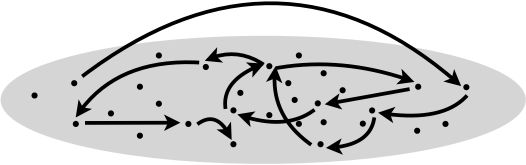
Recover

$$P^* = U^* \cancelto{I_r}{\tilde{P}^*} (V^*)^T$$

(up to linear transformation).

Problem Setup

Observe a trajectory $(i_0, i_1, \cdots, i_n)$



$$\mathcal{S} = \{s_1, s_2, \cdots, s_d\}$$

Driven by an unknown probability transition matrix $P^* \in \mathbb{R}^{d \times d}$

$\longrightarrow$

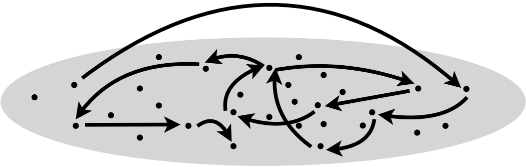
Recover

$$P^* = U^*(V^*)^T$$

(up to linear transformation).

Problem Setup

Observe a trajectory $(i_0, i_1, \cdots, i_n)$



$$\mathcal{S} = \{s_1, s_2, \cdots, s_d\}$$

Driven by an unknown probability transition matrix $P^* \in \mathbb{R}^{d \times d}$

Recover

$$\hat{P}^{(n)} \approx \hat{U}\hat{V}^T$$

(up to linear transformation).

An empirical probability transition matrix $\hat{P}^{(n)}$:

$$\hat{p}_{ij}^{(n)} := \begin{cases} \dfrac{\sum_{t=1}^{n} \mathbb{1}_{\{i_{t-1}=s_i, i_t=s_j\}}}{\sum_{t=1}^{n} \mathbb{1}_{\{i_{t-1}=s_i\}}}, & \text{if } \displaystyle\sum_{t=1}^{n} \mathbb{1}_{\{i_{t-1}=s_i\}} \geq 1, \\ 1/d, & \text{otherwise.} \end{cases}$$

Nonnegative Rank

$$\hat{P}^{(n)} \approx \hat{U}\hat{V}^T, \qquad \hat{U} \in \mathbb{R}^{d\times s}, \hat{V} \in \mathbb{R}^{d\times s}.$$

Aggregation probabilities $\hat{U}$ satisfy $\hat{U} \geq 0$, $\hat{U}\mathbf{1}_s = \mathbf{1}_d$.
Disggregation probabilities V satisfy $\hat{V} \geq 0$, $\hat{V}^T\mathbf{1}_d = \mathbf{1}_s$.
It is desirable to have $s \ll d$.

State aggregation structure ⇔ Low nonnegative rank

If $A \in \mathbb{R}_+^{d\times d}$,

$$\textbf{rank}_+(A) := \min\{m \mid A = BC^T, B \in \mathbb{R}_+^{d\times m}, C \in \mathbb{R}_+^{d\times m}\};$$

If $A \notin \mathbb{R}_+^{d\times d}$, $\textbf{rank}_+(A) := +\infty$.

Formulating an Optimization Problem

$$\text{minimize}_{X \in \mathbb{R}^{d\times d}} \quad g(X) + \chi_{\mathcal{E}}(X) + \lambda \mathbf{rank}_+(X)$$

- $g(X)$ measures the discrepancy between X and $\hat{P}^{(n)}$:

  $$g(X) := \frac{1}{2} \|\hat{\Xi}(\hat{P}^{(n)} - X)\|_F^2,$$

  where $\hat{\Xi} = \mathbf{diag}\{\hat{\xi}^{(n)}\}$, $\hat{\xi}_j^{(n)} = \frac{1}{n}\sum_{t=1}^n \mathbb{1}_{\{i_t = s_j\}}$, $j = 1, 2, \cdots, d$.

- $$\chi_{\mathcal{E}}(X) := \begin{cases} 0, & \text{if } X\mathbf{1}_d = \mathbf{1}_d, \\ +\infty, & \text{otherwise}, \end{cases} \qquad \forall X \in \mathbb{R}^{d\times d}.$$

  The implicit constraints $\chi_{\mathcal{E}}(X) < +\infty$ and $\mathbf{rank}_+(X) < +\infty$ imply that $X\mathbf{1}_d = \mathbf{1}_d$ and $X \geq 0$, forcing X to be a stochastic matrix.

- $\mathbf{rank}_+(X)$ is non-convex. ⇒ A convex surrogate function?

Atomic Norm Relaxation of **rank**

Atomic set $\mathcal{A}_* = \left\{A \in \mathbb{R}^{d\times d} \,\middle|\, \textbf{rank}(A) = 1, \|A\|_2 = 1\right\}.$

For all $X \in \mathbb{R}^{d\times d}$,

$$\textbf{rank}(X) = \min\left\{m \,\middle|\, X = \sum_{i=1}^{m} c_i A_i \text{ with } c_i \geq 0, A_i \in \mathcal{A}_*\right\}.$$

$\Downarrow$ atomic norm relaxation

Nuclear norm

$$\begin{aligned}\|X\|_* &= \min\left\{\sum_{i=1}^{m} c_i \,\middle|\, X = \sum_{i=1}^{m} c_i A_i \text{ with } c_i \geq 0, A_i \in \mathcal{A}_*\right\}\\ &= \sum_{i=1}^{d} \sigma_i(X). \quad (\sigma_i(X) \text{ is the } i\text{-th largest singular value of } X)\end{aligned}$$

Atomic Norm Relaxation of $\mathbf{rank}_+$

Atomic set $\mathcal{A}_+ = \left\{ A \in \mathbb{R}^{d\times d} \,\middle|\, \mathbf{rank}_+(A) = 1, \|A\|_2 = 1 \right\}$.

For all $X \in \mathbb{R}_+^{d\times d}$,

$$\mathbf{rank}_+(X) = \min\left\{ m \,\middle|\, X = \sum_{i=1}^{m} c_i A_i \text{ with } c_i \geq 0, A_i \in \mathcal{A}_+ \right\}.$$

$$\Downarrow \quad \text{“atomic norm relaxation”}$$

Atomic regularizer

$$\Omega(X) = \inf\left\{ \sum_{i=1}^{m} c_i \,\middle|\, X = \sum_{i=1}^{m} c_i A_i \text{ with } c_i \geq 0, A_i \in \mathcal{A}_+ \right\}$$

$$= \inf\left\{ \sum_{j=1}^{s} \|U_j\|_2 \|V_j\|_2 \,\middle|\, X = UV^T \text{ with } U, V \in \mathbb{R}_+^{d\times s} \right\}.$$

Optimal factorization (w.r.t. Ω):

a factorization $X = UV^T$ that achieves the infimum.

Convexifed Formulation

$$\text{minimize}_{X\in\mathbb{R}^{d\times d}} \quad f_\lambda(X) := g(X) + \chi_{\mathcal{E}}(X) + \lambda\Omega(X) \qquad (1)$$

Theorem (Sufficient and necessary conditions for global optimality)

$\hat{X}$ is globally optimal for (1) with an optimal factorization $\hat{X} = \hat{U}\hat{V}^T$ iff $\exists \mu \in \mathbb{R}^d$ s.t.

$$\begin{cases} \mathbf{u}^T\left(\mu\mathbf{1}_d^T - \nabla g(\hat{X})\right)\mathbf{v} \le \lambda, \quad \forall \mathbf{u},\mathbf{v} \in \mathbb{R}_+^d \textit{ with } \|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1, \\ \left[\mu\mathbf{1}_s^T - \nabla g(\hat{X})\hat{V}\right]_+ = \lambda\hat{U}\mathbf{diag}\left\{\frac{\|\hat{V}_j\|_2}{\|\hat{U}_j\|_2}\right\}_{j=1}^s, \\ \left[\mathbf{1}_d\mu^T\hat{U} - \left(\nabla g(\hat{X})\right)^T\hat{U}\right]_+ = \lambda\hat{V}\mathbf{diag}\left\{\frac{\|\hat{U}_j\|_2}{\|\hat{V}_j\|_2}\right\}_{j=1}^s. \end{cases}$$

- $\Omega(X)$ does not have an explicit form.

Factorized Optimization Model

$$\mathrm{minimize}_{U,V\in\mathbb{R}^{d\times s}} \quad F_\lambda(U,V) := g(UV^T) + \lambda\sum_{j=1}^{s}\|U_j\|_2\|V_j\|_2, \tag{2}$$

$$s.t. \qquad U\mathbf{1}_s = \mathbf{1}_d,\, V^T\mathbf{1}_d = \mathbf{1}_s, \quad U\geq 0, V\geq 0$$

- s: the rank of model, a parameter to be adjusted.
- When s is sufficiently large, the factorized optimization model (2) is equivalent to the convexifed problem (1).

Theorem (KKT conditions of (2))

Suppose that $\left(\hat{U},\hat{V}\right)$ is a local solution to (2). Then, $\exists\mu\in\mathbb{R}^d$ s.t.

$$\begin{cases} \left[\mu\mathbf{1}_s^T - \nabla g(\hat{X})\hat{V}\right]_+ = \lambda\hat{U}\mathbf{diag}\left\{\dfrac{\|\hat{V}_j\|_2}{\|\hat{U}_j\|_2}\right\}_{j=1}^{s} \\ \left[\mathbf{1}_d\mu^T\hat{U} - \left(\nabla g(\hat{X})\right)^T\hat{U}\right]_+ = \lambda\hat{V}\mathbf{diag}\left\{\dfrac{\|\hat{U}_j\|_2}{\|\hat{V}_j\|_2}\right\}_{j=1}^{s} \end{cases}.$$

Global Optimality Certificate

Sufficient and necessary conditions for global optimality of (1):

$$\begin{cases} \boxed{\mathbf{u}^T\big(\mu\mathbf{1}_d^T - \nabla g(\hat{X})\big)\mathbf{v} \le \lambda, \quad \forall \mathbf{u},\mathbf{v} \in \mathbb{R}_+^d \text{ with } \|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1,} \\ \left[\mu\mathbf{1}_s^T - \nabla g(\hat{X})\hat{V}\right]_+ = \lambda\hat{U}\mathbf{diag}\left\{\frac{\|\hat{V}_j\|_2}{\|\hat{U}_j\|_2}\right\}_{j=1}^s, \\ \left[\mathbf{1}_d\mu^T\hat{U} - \left(\nabla g(\hat{X})\right)^T\hat{U}\right]_+ = \lambda\hat{V}\mathbf{diag}\left\{\frac{\|\hat{U}_j\|_2}{\|\hat{V}_j\|_2}\right\}_{j=1}^s. \end{cases}$$

KKT conditions for local solutions to (2):

$$\begin{cases} \left[\mu\mathbf{1}_s^T - \nabla g(\hat{X})\hat{V}\right]_+ = \lambda\hat{U}\mathbf{diag}\left\{\frac{\|\hat{V}_j\|_2}{\|\hat{U}_j\|_2}\right\}_{j=1}^s, \\ \left[\mathbf{1}_d\mu^T\hat{U} - \left(\nabla g(\hat{X})\right)^T\hat{U}\right]_+ = \lambda\hat{V}\mathbf{diag}\left\{\frac{\|\hat{U}_j\|_2}{\|\hat{V}_j\|_2}\right\}_{j=1}^s. \end{cases}$$

General Idea of the Algorithm

1. Solve the factorized optimization model (2) and obtain a local solution $(\hat{U}, \hat{V})$.

2. Calculate a vector $\mu \in \mathbb{R}^d$ according to the KKT conditions:

$$\mu_i = \left(\nabla g(\hat{X})\hat{V}\right)_{ij} + \lambda \hat{u}_{ij} \frac{\|\hat{V}_j\|_2}{\|\hat{U}_j\|_2}, \quad \text{for any } j \text{ such that } \hat{u}_{ij} > 0.$$

3. Determine whether μ satisfies

$$\mathbf{u}^T\left(\mu \mathbf{1}_d^T - \nabla g(\hat{X})\right)\mathbf{v} \leq \lambda, \quad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}^d \text{ with } \|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1.$$

Remaining Issues

- How to verify the global optimality certificate?

$$\mathbf{u}^T\left(\mu\mathbf{1}_d^T - \nabla g(\hat{X})\right)\mathbf{v} \leq \lambda, \quad \forall \mathbf{u}, \mathbf{v} \in \mathbb{R}_+^d \text{ with } \|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1.$$

- How to refine a local solution $(\hat{U}, \hat{V})$ if it does not represent a global minimum to the convexified problem?

- A subroutine to solve the factorized optimization model?

Criteria to Determine Global Optimality

- Exact stopping rule: Gradient projection method to solve

$$\begin{array}{ll} \mathrm{maximize}_{\mathbf{u},\mathbf{v}} & \mathbf{u}^T\left(\mu\mathbf{1}_d^T - \nabla g(\hat{X})\right)\mathbf{v} \\ s.t. & \|\mathbf{u}\|_2 = 1, \|\mathbf{v}\|_2 = 1, \\ & \mathbf{u} \geq 0, \mathbf{v} \geq 0. \end{array}$$

  If the objective value $\leq (1+\varepsilon_{\mathrm{Exa}})\lambda$, then $\hat{X}$ is considered to be global optimal.

- Early stopping rule: Define a function

$$\varphi(\mathbf{v}) := \left\| \left[\left(\mu\mathbf{1}_d^T - \nabla g(\hat{X})\right)\mathbf{v}\right]_+ \right\|_2 \text{ for } \mathbf{v} \in \mathbb{R}_+^d \text{ with } \|\mathbf{v}\|_2 = 1.$$

  Global optimality certificate $\Leftrightarrow L_\lambda = \left\{\mathbf{v} \,\middle|\, \varphi(\mathbf{v}) > \lambda\right\} = \emptyset$.

  Test vectors $\{\bar{\mathbf{v}}_k\}_{k=1}^N \overset{i.i.d.}{\sim} \mathrm{Uniform}\left(\left\{\mathbf{v} \in \mathbb{R}_+^d \,\middle|\, \|\mathbf{v}\|_2 = 1\right\}\right)$.
  If $\varphi(\bar{\mathbf{v}}_k) \leq \lambda$ for each k, then we say $\hat{X}$ is global optimal.

Successive Refinements to Escape from Local Solutions

- Appending a New Column:

  If $(\hat{U}, \hat{V})$ is not globally optimal, there exist $\bar{\mathbf{v}}$ and $\bar{\mathbf{u}}$ such that

  $$\bar{u}^T\left(\mu\mathbf{1}_d^T - \nabla g(\hat{X})\right)\bar{v} > \lambda, \quad \bar{\mathbf{u}}, \bar{\mathbf{v}} \geq 0, \ \|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1.$$

  Intuitively, $\bar{\mathbf{u}}\bar{\mathbf{v}}^T$ approximates the negative subgradient directions of f_λ at $\hat{X}$.

Theorem (Escaping local minima)

Take

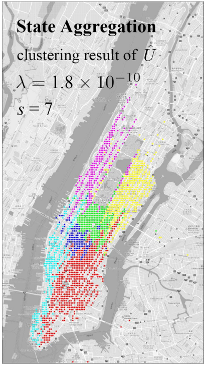
$$\bar{U} = \left[\mathbf{diag}\{\mathbf{1}_d - \kappa\bar{\mathbf{u}}\}\hat{U}, \kappa\bar{\mathbf{u}}\right], \quad \bar{V} = \left[\hat{V}, (\bar{\mathbf{v}}^T\mathbf{1}_d)^{-1}\bar{v}\right]$$

for some sufficiently small $\kappa > 0$.

Then $F_\lambda(\bar{U}, \bar{V}) < F_\lambda(\hat{U}, \hat{V})$.

Experiments with Manhattan Taxi Data

- Partition Manhattan transportation network into different regions.
- Datasets: 1.1×10^7 NYC Yellow cab trips in January 2016. Each record includes passenger pick-up and drop-off information (coordinates, time, etc.) of one trip. The movements of taxis are nearly memoryless. We divide the map into a fine grid and merge the locations in the same cell into one state. Each trip is a sampled one-step state transition between cells.
- Each point in the map represents a valid state of the Markov chain. The figures in one pair have exactly the same number of regions, where the left one is produced by the state aggregation model and the right one is provided by the SVD-based method. In some figures, there are less than s regions appearing on the map, because some points are plotted beyond the boundaries.

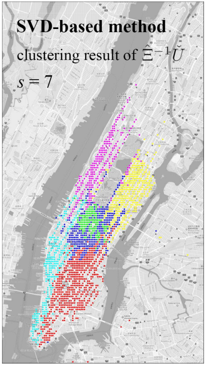


State Aggregation

clustering result of $\hat{U}$

$\lambda = 1.8 \times 10^{-10}$

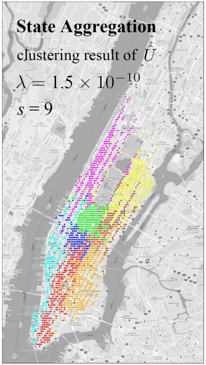
$s = 7$



SVD-based method

clustering result of $\hat{\Xi}^{-1}\check{U}$

$s = 7$

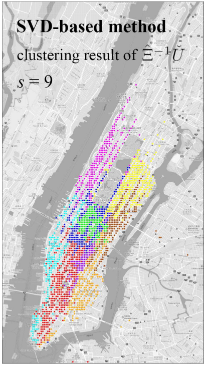


State Aggregation

clustering result of $\hat{U}$

$\lambda = 1.5 \times 10^{-10}$

$s = 9$



SVD-based method

clustering result of $\hat{\Xi}^{-1}\check{U}$

$s = 9$

Contact Information

Many Thanks For Your Attention!

- 北大课程：大数据分析中的算法，华文慕课回放 http://bicmr.pku.edu.cn/~wenzw/bigdata2020.html
- 教材：刘浩洋, 户将, 李勇锋，文再文，最优化计算方法http://bicmr.pku.edu.cn/~wenzw/optbook.html
- Looking for Ph.D students and Postdoc
  Competitive salary as U.S and Europe
- http://bicmr.pku.edu.cn/~wenzw
- E-mail: wenzw@pku.edu.cn
- Office phone: 86-10-62744125