Median Matrix Completion: from Embarrassment to Optimality

Xiaojun Mao

School of Data Science
Fudan University, China

June 15, 2020

Joint work with Dr. Weidong Liu (Shanghai Jiao Tong University, China) and Dr. Raymond K. W. Wong (Texas A&M University, U.S.A.)

1 Introduction

2 Estimations

3 Theoretical Guarantee

4 Experiments

Our Goal and Contributions

- Robust Matrix Comepletion (MC), allows heavy tails.
- Develop a ***robust*** and ***scalable*** estimator for median MC in large-scale problems.
  - A fast and simple initial estimation via embarrassingly parallel computing.
  - A refinement stage based on pseudo data.
- Theoretically, we show that this refinement stage can improve the convergence rate of the sub-optimal initial estimator to ***near-optimal*** order, as good as the computationally expensive median MC estimator.

Background: The Netflix Problem

- $n_1 \approx 480K$, $n_2 \approx 18K$.
- On average each viewer rated about 200 movies. Only 1.2% entries were observed.
- Goal: recover the true rating matrix $\boldsymbol{A}_\star$.

Robust Matrix Completion

- Low-rank-plus-sparse structure: $\boldsymbol{A}_\star + \boldsymbol{S} + \boldsymbol{E}$.
- Median matrix completion: based on the absolute deviation loss.
- Under absolute deviation loss and the Huber loss, the convergence rates of Elsener and Geer (2018) match with Koltchinskii *et al.* (2011).
- Alquier *et al.* (2019) derives the minimax rates of convergence with any Lipschitz loss functions (absolute deviation loss).

1 Introduction

2 Estimations

3 Theoretical Guarantee

4 Experiments

Trace Regression Model

- N independent pairs $(\mathbf{X}_k, Y_k)$,

$$Y_k = \mathrm{tr}\left(\mathbf{X}_k^{\mathrm{T}} \mathbf{A}_\star\right) + \epsilon_k, \qquad k = 1, \dots, N. \tag{1}$$

  - The elements of $\boldsymbol{\epsilon} = (\epsilon_1, \dots, \epsilon_N)$ are N i.i.d. random noise variables independent of the design matrices.
  - The design matrices $\boldsymbol{X}_k$:

$$\mathcal{X} = \{\boldsymbol{e}_j(n_1)\boldsymbol{e}_k(n_2)^{\mathrm{T}} : j = 1, \dots, n_1; k = 1, \dots, n_2\},$$

Regularized Least Absolute Deviation Estimator

- $\mathbf{A}_\star = (A_{\star,ij})_{i,j=1}^{n_1,n_2} \in \mathbb{R}^{n_1 \times n_2}$, $\mathbb{P}(\epsilon \le 0) = 0.5$: $A_{\star,ij}$ is the median of $Y \mid \mathbf{X}$. $\mathcal{B}(a,n,m) = \{\mathbf{A} \in \mathbb{R}^{n \times m} : \|\mathbf{A}\|_\infty \le a\}$ and $\mathbf{A}_\star \in \mathcal{B}(a,n,m)$.

- We use the absolute deviation loss:

$$\mathbf{A}_\star = \underset{\mathbf{A} \in \mathcal{B}(a,n_1,n_2)}{\arg\min} \ \mathbb{E}\left\{ \left| Y - \mathrm{tr}(\mathbf{X}^{\mathrm{T}}\mathbf{A}) \right| \right\}.$$

- To encourage a low-rank solution,

$$\widehat{\mathbf{A}}_{\mathrm{LADMC}} = \underset{\mathbf{A} \in \mathcal{B}(a,n_1,n_2)}{\arg\min} \frac{1}{N} \sum_{k=1}^{N} \left| Y_k - \mathrm{tr}(\mathbf{X}_k^{\mathrm{T}}\mathbf{A}) \right| + \lambda_N' \left\| \mathbf{A} \right\|_* .$$

- Common computational strategies based on proximal gradient method inapplicable (Sum of two non-differentiable terms).

- Alquier *et al.* (2019) use ADMM, when the sample size and the matrix dimensions are large, slow and not scalable in practice.

Distributed Initial Estimator

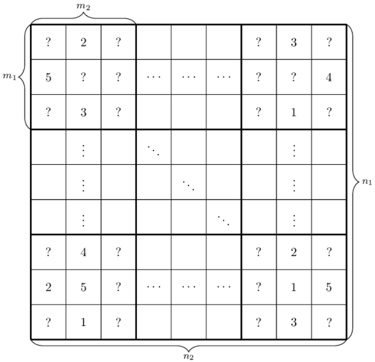

Figure: An example of dividing a matrix into sub-matrices.

$$\widehat{\mathbf{A}}_{\mathrm{LADMC},l} = \mathop{\arg\min}_{\mathbf{A}_l \in \mathcal{B}(a,m_1,m_2)} \frac{1}{N_l} \sum_{k \in \mathbf{\Omega}_l} \left| Y_k - \operatorname{tr}(\mathbf{X}_{l,k}^{\mathrm{T}} \mathbf{A}_l) \right| + \lambda_{N_l,l} \left\| \mathbf{A}_l \right\|_* .$$

The Idea of Refinement

- $L(\mathbf{A};\{Y,\mathbf{X}\}) = |Y - \mathrm{tr}(\mathbf{X}^{\mathrm{T}}\mathbf{A})|$. The Newton-Raphson iteration:

$$\mathrm{vec}(\mathbf{A}_1) = \mathrm{vec}(\widehat{\mathbf{A}}_0) - \mathbf{H}(\widehat{\mathbf{A}}_0)^{-1}\mathbb{E}_{(Y,\mathbf{X})}\left[\boldsymbol{l}(\widehat{\mathbf{A}}_0;\{Y,\mathbf{X}\})\right],$$

where $\widehat{\mathbf{A}}_0$ is an initial estimator; $\boldsymbol{l}(\mathbf{A};\{Y,\mathbf{X}\})$ is the sub-gradient and $\mathbf{H}(\mathbf{A})$ is the Hessian matrix.

- When $\widehat{\mathbf{A}}_0$ is close to the minimizer $\mathbf{A}_\star$,

$$\begin{aligned}
&\mathrm{vec}(\mathbf{A}_1) \approx \mathrm{vec}(\widehat{\mathbf{A}}_0) - [2f(0)\mathrm{diag}(\boldsymbol{\Pi})]^{-1}\mathbb{E}_{(Y,\mathbf{X})}[\boldsymbol{l}(\widehat{\mathbf{A}}_0;\{Y,\mathbf{X}\})] \\
&= \mathbb{E}_{(Y,\mathbf{X})}\left\{\mathrm{vec}(\widehat{\mathbf{A}}_0) - [f(0)]^{-1}\left(\mathbb{I}\left[Y \le \mathrm{tr}(\mathbf{X}^{\mathrm{T}}\widehat{\mathbf{A}}_0)\right] - \frac{1}{2}\right)\mathbb{1}_{n_1 n_2}\right\} \\
&= \{\mathbb{E}_{(Y,\mathbf{X})}[\mathrm{vec}(\mathbf{X})\mathrm{vec}(\mathbf{X})^{\mathrm{T}}]\}^{-1}\mathbb{E}_{(Y,\mathbf{X})}\left(\mathrm{vec}(\mathbf{X})\,\tilde{Y}^0\right)
\end{aligned}$$

where $\boldsymbol{\Pi} = (\pi_{1,1},\ldots,\pi_{n_1,n_2})^{\mathrm{T}}$, $\pi_{st} = \Pr(\mathbf{X}_k = \mathbf{e}_s(n_1)\mathbf{e}_t^{\mathrm{T}}(n_2))$, and the theoretical pseudo data

$$\widetilde{Y}^o = \mathrm{tr}(\mathbf{X}^{\mathrm{T}}\widehat{\mathbf{A}}_0) - [f(0)]^{-1}\left(\mathbb{I}\left[Y \le \mathrm{tr}(\mathbf{X}^{\mathrm{T}}\widehat{\mathbf{A}}_0)\right] - \frac{1}{2}\right).$$

The First Iteration Refinement Details

- $\mathrm{vec}(\mathbf{A}_1) \approx \arg\min_{\mathbf{A}} \mathbb{E}_{(Y,\mathbf{X})}\{\widetilde{Y}^o - \mathrm{tr}(\mathbf{X}^{\mathrm{T}}\mathbf{A})\}^2$.
- Choice of the initial estimator: $\widehat{\mathbf{A}}_0$ satisfies certain rate Condition.
- $K(x)$: kernel function; $h > 0$: the bandwidth.

$$\widehat{f}(0) = \frac{1}{Nh}\sum_{k=1}^{N} K\left(\frac{Y_k - \mathrm{tr}(\mathbf{X}_k^{\mathrm{T}}\widehat{\mathbf{A}}_0)}{h}\right).$$

- Let $\widetilde{\mathbf{Y}} = (\widetilde{Y}_k)$, denote

$$\widetilde{Y}_k = \mathrm{tr}(\mathbf{X}_k^{\mathrm{T}}\widehat{\mathbf{A}}_0) - [\widehat{f}(0)]^{-1}\left(\mathbb{I}\left[Y_k \le \mathrm{tr}(\mathbf{X}_k^{\mathrm{T}}\widehat{\mathbf{A}}_0)\right] - \frac{1}{2}\right).$$

- By using $\widetilde{\mathbf{Y}}$, one natural estimator is given by

$$\widehat{\mathbf{A}} = \mathop{\arg\min}_{\mathbf{A}\in\mathcal{B}(a,n_1,n_2)} \frac{1}{N}\sum_{k=1}^{N}\left(\widetilde{Y}_k - \mathrm{tr}(\mathbf{X}_k^{\mathrm{T}}\mathbf{A})\right)^2 + \lambda_N \|\mathbf{A}\|_*.$$

The t-th Iteration Refinement Details

- Let $h_t \to 0$ is the bandwidth for the t-th iteration,

$$\widehat{f}^{(t)}(0) = \frac{1}{Nh_t} \sum_{k=1}^{N} K\left(\frac{Y_k - \mathrm{tr}(\mathbf{X}_k^{\mathrm{T}} \widehat{\mathbf{A}}^{(t-1)})}{h_t} \right).$$

- Similarly, for each $1 \leq k \leq N$, define

$$\widetilde{Y}_k^{(t)} = \mathrm{tr}(\mathbf{X}_k^{\mathrm{T}} \widehat{\mathbf{A}}^{(t-1)}) - \left(\widehat{f}^{(t)}(0) \right)^{-1} \left(\mathbb{I}\left[Y_k \leq \mathrm{tr}(\mathbf{X}_k^{\mathrm{T}} \widehat{\mathbf{A}}^{(t-1)}) \right] - \frac{1}{2} \right).$$

- We propose the following estimator

$$\widehat{\mathbf{A}}^{(t)} = \underset{\mathbf{A} \in \mathcal{B}(a, n_1, n_2)}{\arg\min} \frac{1}{N} \sum_{k=1}^{N} \left(\widetilde{Y}_k^{(t)} - \mathrm{tr}(\mathbf{X}_k^{\mathrm{T}} \mathbf{A}) \right)^2 + \lambda_{N,t} \left\| \mathbf{A} \right\|_* .$$

1 Introduction

2 Estimations

3 Theoretical Guarantee

4 Experiments

Notations

- $n_+ = n_1 + n_2$, $n_{\max} = \max\{n_1, n_2\}$ and $n_{\min} = \min\{n_1, n_2\}$. Denote $r_\star = \mathrm{rank}(\mathbf{A}_\star)$.
- In additional to some regular conditions, the initial estimator $\widehat{\mathbf{A}}_0$ satisfies $(n_1 n_2)^{-1/2}\|\widehat{\mathbf{A}}_0 - \mathbf{A}_\star\|_F = O_{\mathrm{P}}((n_1 n_2)^{-1/2} a_N)$, where the initial rate $(n_1 n_2)^{-1/2} a_N = o(1)$.
- Denote the initial rate $a_{N,0} = a_N$ and define that

$$a_{N,t} = \sqrt{\frac{r_\star (n_1 n_2) n_{\max} \log(n_+)}{N}} + \frac{n_{\min}}{\sqrt{r_\star}} \left(\frac{\sqrt{r_\star} a_{N,0}}{n_{\min}} \right)^{2^t}.$$

Convergence Results of Repeated Refinement Estimator

Theorem (Repeated refinement)

Suppose that certain regular conditions hold and $\mathbf{A}_\star \in \mathcal{B}(a, n_1, n_2)$. By choosing h_t and $\lambda_{N,t}$ to be certain orders, we have

$$\frac{\left\|\widehat{\mathbf{A}}^{(t)} - \mathbf{A}_\star\right\|_F^2}{n_1 n_2} = O_P\left[\max\left\{\sqrt{\frac{\log(n_+)}{N}}, r_\star\left(\frac{n_{\max}\log(n_+)}{N} + \frac{a_{N,t-1}^4}{n_{\min}^2(n_1 n_2)}\right)\right\}\right].$$

$$t \geq \log\left\{\frac{\log(r_\star^2 n_{\max}^2 \log(n_+)) - \log(n_{\min} N)}{c_0 \log(r_\star a_{N,0}^2) - 2c_0 \log(n_{min})}\right\} / \log(2), \quad \text{for some} \quad c_0 > 0,$$

- The convergence rate of $\widehat{\mathbf{A}}^{(t)}$ becomes $r_\star n_{\max} N^{-1} \log(n_+)$ which is the near-optimal rate $r_\star n_{\max} N^{-1}$ upto a logarithmic factor.
- Under certain condition, t is of constant order.

1 Introduction

2 Estimations

3 Theoretical Guarantee

4 Experiments

Synthetic Data Generation

- $\mathbf{A}_\star = \mathbf{U}\mathbf{V}^{\mathrm{T}}$, where the entries of $\mathbf{U} \in \mathbb{R}^{n_1 \times r}$ and $\mathbf{V} \in \mathbb{R}^{n_2 \times r}$ were all drawn from $\mathcal{N}(0, 1)$ independently.
- Set $r = 3$, chose $n_1 = n_2$: 400, repeat 500 times.
- The missing rate was 0.2, we adopted the uniform missing mechanism.
- Four noise distributions:
  - S1 Normal: $\epsilon \sim \mathcal{N}(0, 1)$.
  - S2 Cauchy: $\epsilon \sim \mathrm{Cauchy}(0, 1)$.
  - S3 Exponential: $\epsilon \sim \exp(1)$.
  - S4 t-distribution with degree of freedom 1: $\epsilon \sim \mathrm{t}_1$.
- Cauchy distribution is a very heavy-tailed distribution and its first moment (expectation) does not exist.

Comparison Methods

(a) BLADMC: Blocked Least Absolute Deviation Matrix Completion $\widehat{\mathbf{A}}_{\mathrm{LADMC},0}$. Number of row subsets $l_1 = 2$, number of column subsets $l_2 = 2$.

(b) ACL: Least Absolute Deviation Matrix Completion with nuclear norm penalty based on the computationally expensive ADMM algorithm proposed by Alquier *et al.* (2019).

c) MHT: The squared loss estimator with nuclear norm penalty proposed by Mazumder *et al.* (2010).

Simulation Results for Noise Distribution S1 and S2

Table: The average RMSEs, MAEs, estimated ranks and their standard errors (in parentheses) of DLADMC, BLADMC, ACL and MHT.

(T)		DLADMC	BLADMC
S1(4)	RMSE	0.5920 (0.0091)	0.7660 (0.0086)
	MAE	0.4273 (0.0063)	0.5615 (0.006)
	rank	52.90 (2.51)	400 (0.00)
S2(5)	RMSE	0.9395 (0.0544)	1.7421 (0.3767)
	MAE	0.6735 (0.0339)	1.2061 (0.1570)
	rank	36.49 (7.94)	272.25 (111.84)
(T)		ACL	MHT
S1(4)	RMSE	0.5518 (0.0081)	0.4607 (0.0070)
	MAE	0.4031 (0.0056)	0.3375 (0.0047)
	rank	400 (0.00)	36.89 (1.79)
S2(5)	RMSE	1.8236 (1.1486)	106.3660 (918.5790)
	MAE	1.2434 (0.5828)	1.4666 (2.2963)
	rank	277.08 (170.99)	1.25 (0.50)

Simulation Results for Noise Distribution S3 and S4

Table: The average RMSEs, MAEs, estimated ranks and their standard errors (in parentheses) of DLADMC, BLADMC, ACL and MHT.

(T)		DLADMC	BLADMC
S3(5)	RMSE	0.4868 (0.0092)	0.6319 (0.0090)
	MAE	0.3418 (0.0058)	0.4484 (0.0057)
	rank	66.66 (1.98)	400 (0.00)
S4(4)	RMSE	1.1374 (0.8945)	1.6453 (0.2639)
	MAE	0.8317 (0.7370)	1.1708 (0.1307)
	rank	47.85 (13.22)	249.16 (111.25)
(T)		ACL	MHT
S3(5)	RMSE	0.4164 (0.0074)	0.4928 (0.0083)
	MAE	0.3121 (0.0054)	0.3649 (0.0058)
	rank	400 (0.00)	37.91 (1.95)
S4(4)	RMSE	1.4968 (0.6141)	98.851 (445.4504)
	MAE	1.0792 (0.3803)	1.4502 (1.1135)
	rank	237.05 (182.68)	1.35 (0.71)

MovieLens 100K Results

Table: The RMSEs, MAEs and estimated ranks.

		DLADMC	BLADMC	ACL	MHT
RawA	RMSE	0.9235	0.9451	0.9258	0.9166
	MAE	0.7233	0.7416	0.7252	0.7196
	rank	41	530	509	57
	t	254.33	65.64	393.40	30.16
RawB	RMSE	0.9352	0.9593	0.9376	0.9304
	MAE	0.7300	0.7498	0.7323	0.7280
	rank	51	541	521	58
	t	244.73	60.30	448.55	29.60
OutA	RMSE	1.0486	1.0813	1.0503	1.0820
	MAE	0.8568	0.8833	0.8590	0.8971
	rank	38	493	410	3
	t	255.25	89.65	426.78	10.41
OutB	RMSE	1.0521	1.0871	1.0539	1.0862
	MAE	0.8616	0.8905	0.8628	0.9021
	rank	28	486	374	6
	t	260.79	104.97	809.26	10.22

Thank you!