Median Matrix Completion: from Embarrassment to Optimality

Weidong Liu [1] **Xiaojun Mao** [2] **Raymond K. W. Wong** [3]

Abstract

In this paper, we consider matrix completion with absolute deviation loss and obtain an estimator of the median matrix. Despite several appealing properties of median, the non-smooth absolute deviation loss leads to computational challenge for large-scale data sets which are increasingly common among matrix completion problems. A simple solution to large-scale problems is parallel computing. However, embarrassingly parallel fashion often leads to inefficient estimators. Based on the idea of pseudo data, we propose a novel refinement step, which turns such inefficient estimators into a rate (near-)optimal matrix completion procedure. The refined estimator is an approximation of a regularized least median estimator, and therefore not an ordinary regularized empirical risk estimator. This leads to a non-standard analysis of asymptotic behaviors. Empirical results are also provided to confirm the effectiveness of the proposed method.

1. Introduction

Matrix completion (MC) has recently gained a substantial amount of popularity among researchers and practitioners due to its wide applications; as well as various related theoretical advances Candès & Recht (2009); Candès & Plan (2010); Koltchinskii et al. (2011); Klopp (2014). Perhaps the most well-known example of a MC problem is the Netflix prize problem (Bennett & Lanning, 2007), of which the goal is to predict missing entries of a partially observed matrix of movie ratings. Two commonly shared challenges among MC problems are high dimensionality of the matrix and a huge proportion of missing entries. For instance, Netflix data has less than 1% of observed entries of a matrix with around 5×10^5 rows and 2×10^4 customers. With technological advances in data collection, we are confronting increasingly large matrices nowadays.

Without any structural assumption on the target matrix, it is well-known that MC is an ill-posed problem. A popular and often well-justified assumption is low rankness, which however leads to a challenging and non-convex rank minimization problem (Srebro et al., 2005). The seminal works of Candès & Recht (2009); Candès & Tao (2010); Gross (2011) showed that, when the entries are observed without noise, a perfect recovery of a low-rank matrix can be achieved by a convex optimization via near minimal order of sample size, with high probability. As for the noisy setting, some earlier work (Candès & Plan, 2010; Keshavan et al., 2010; Chen et al., 2019b) focused on arbitrary, not necessarily random, noise. In general, the arbitrariness may prevent asymptotic recovery even in a probability sense.

Recently, a significant number of works (e.g. Bach, 2008; Koltchinskii et al., 2011; Negahban & Wainwright, 2011; Rohde & Tsybakov, 2011; Negahban & Wainwright, 2012; Klopp, 2014; Cai & Zhou, 2016; Fan et al., 2019; Xia & Yuan, 2019) targeted at more amenable random error models, under which (near-)optimal estimators had been proposed. Among these work, trace regression model is one of the most popular models due to its regression formulation. Assume N independent pairs $(\mathbf{X}_k, Y_k)$, for $k = 1, \ldots, N$, are observed, where $\mathbf{X}_k$'s are random design matrices of dimension $n_1 \times n_2$ and Y_k's are response variables in $\mathbb{R}$. The trace regression model assumes

$$Y_k = \mathrm{tr}\left(\mathbf{X}_k^{\mathrm{T}} \mathbf{A}_\star\right) + \epsilon_k, \qquad k = 1, \ldots, N, \tag{1.1}$$

where $\mathrm{tr}(\mathbf{A})$ denotes the trace of a matrix $\mathbf{A}$, and $\mathbf{A}_\star \in \mathbb{R}^{n_1 \times n_2}$ is an unknown target matrix. Moreover, the elements of $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_N)$ are N i.i.d. random noise variables independent of the design matrices. In MC setup, the design matrices $\boldsymbol{X}_k$'s are assumed to lie in the set of canonical bases

$$\mathcal{X} = \{\boldsymbol{e}_j(n_1)\boldsymbol{e}_k(n_2)^{\mathrm{T}} : j = 1, \ldots, n_1; k = 1, \ldots, n_2\}, \tag{1.2}$$

where $\boldsymbol{e}_j(n_1)$ is the j-th unit vector in $\mathbb{R}^{n_1}$, and $\boldsymbol{e}_k(n_2)$ is the k-th unit vector in $\mathbb{R}^{n_2}$. Most methods then apply a regularized empirical risk minimization (ERM) framework

*Equal contribution [1]School of Mathematical Sciences and MoE Key Lab of Artificial Intelligence Shanghai Jiao Tong University, Shanghai, 200240, China [2]School of Data Science, Fudan University, Shanghai, 200433, China [3]Department of Statistics, Texas A&M University, College Station, TX 77843, U.S.A.. Correspondence to: Xiaojun Mao <maoxj@fudan.edu.cn>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

with a quadratic loss. It is well-known that the quadratic loss is most suitable for light-tailed (sub-Gaussian) error, and leads to non-robust estimations. In the era of big data, a thorough and accurate data cleaning step, as part of data preprocessing, becomes virtually impossible. In this regard, one could argue that robust estimations are more desirable, due to their reliable performances even in the presence of outliers and violations of model assumptions. While robust statistics is a well-studied area with a rich history (Davies, 1993; Huber, 2011), many robust methods were developed for small data by today's standards, and are deemed too computationally intensive for big data or complex models. This work can be treated as part of the general effort to broaden the applicability of robust methods to modern data problems.

1.1. Related Work

Many existing robust MC methods adopt regularized ERM and assume observations are obtained from a low-rank-plus-sparse structure $\boldsymbol{A}_\star + \boldsymbol{S} + \boldsymbol{E}$, where the low-rank matrix $\boldsymbol{A}_\star$ is the target uncontaminated component; the sparse matrix $\boldsymbol{S}$ models the gross corruptions (outliers) locating at a small proportion of entries; and $\boldsymbol{E}$ is an optional (dense) noise component. As gross corruptions are already taken into account, many methods with low-rank-plus-sparse structure are based on quadratic loss. Chandrasekaran et al. (2011); Candès et al. (2011); Chen et al. (2013); Li (2013) considered the noiseless setting (i.e., no $\boldsymbol{E}$) with an element-wisely sparse $\boldsymbol{S}$. Chen et al. (2011) studied the noiseless model with column-wisely sparse $\boldsymbol{S}$. Under the model with element-wisely sparse $\boldsymbol{S}$, Wong & Lee (2017) looked into the setting of arbitrary (not necessarily random) noise $\boldsymbol{E}$, while Klopp et al. (2017) and Chen et al. (2020) studied random (sub-Gaussian) noise model for $\boldsymbol{E}$. In particular, it was shown in Proposition 3 of Wong & Lee (2017) that in the regularized ERM framework, a quadratic loss with element-wise ℓ_1 penalty on the sparse component is equivalent to a direct application of a Huber loss without the sparse component. Roughly speaking, this class of robust methods, based on the low-rank-plus-sparse structure, can be understood as regularized ERMs with Huber loss.

Another class of robust MC methods is based on the absolute deviation loss, formally defined in (2.1). The minimizer of the corresponding risk has an interpretation of median (see Section 2.1), and so the regularized ERM framework that applies absolute deviation loss is coined as median matrix completion (Elsener & van de Geer, 2018; Alquier et al., 2019). In the trace regression model, if the medians of the noise variables are zero, the median MC estimator can be treated as a robust estimation of $\boldsymbol{A}_\star$. Although median is one of the most commonly used robust statistics, the median MC methods have not been studied until recently. Elsener & van de Geer (2018) derived the asymptotic behavior of the trace-norm regularized estimators under both the absolute deviation loss and the Huber loss. Their convergence rates match with the rate obtained in Koltchinskii et al. (2011) under certain conditions. More complete asymptotic results have been developed in Alquier et al. (2019), which derives the minimax rates of convergence with any Lipschitz loss functions including absolute deviation loss.

To the best of our knowledge, the only existing computational algorithm of median MC in the literature is proposed by Alquier et al. (2019), which is an alternating direction method of multiplier (ADMM) algorithm developed for the quantile MC with median MC being a special case. However, this algorithm is slow and not scalable to large matrices due to the non-smooth nature of both the absolute deviation loss and the regularization term.

Despite the computational challenges, the absolute deviation loss has a few appealing properties as compared to the Huber loss. First, absolute deviation loss is tuning-free while Huber loss has a tuning parameter, which is equivalent to the tuning parameter in the entry-wise ℓ_1 penalty in the low-rank-plus-sparse model. Second, absolute deviation loss is generally more robust than Huber loss. Third, the minimizer of expected absolute deviation loss is naturally tied to median, and is generally more interpretable than the minimizer of expected Huber loss (which may vary with its tuning parameter).

1.2. Our Goal and Contributions

Our goal is to develop a robust and scalable estimator for median MC, in large-scale problems. The proposed estimator approximately solves the regularized ERM with the non-differentiable absolute deviation loss. It is obtained through two major stages — (1) a fast and simple initial estimation via embarrassingly parallel computing and (2) a refinement stage based on pseudo data. As pointed out earlier (with more details in Section 2.2), existing computational strategy (Alquier et al., 2019) does not scale well with the dimensions of the matrix. Inspired by Mackey et al. (2015), a simple strategy is to divide the target matrix into small sub-matrices and perform median MC on every sub-matrices in an *embarrassingly* parallel fashion, and then naively concatenate all estimates of these sub-matrices to form an initial estimate of the target matrix. Therefore, most computations are done on much smaller sub-matrices, and hence this computational strategy is much more scalable. However, since low-rankness is generally a global (whole-matrix) structure, the lack of communications between the computations of different sub-matrices lead to sub-optimal estimation (Mackey et al., 2015). The key innovation of this paper is a fast refinement stage, which transforms the regularized ERM with absolute deviation loss into a regularized ERM with quadratic loss, for which many fast

algorithms exist, via the idea of *pseudo data*. Motivated by Chen et al. (2019a), we develop the pseudo data based on a Newton-Raphson iteration in expectation. The construction of the pseudo data requires only a rough initial estimate (see Condition (C6) in Section 3), which is obtained in the first stage. As compared to Huber-loss-based methods (sparse-plus-low-rank model), the underlying absolute deviation loss is non-differentiable, leading to computational difficulty for large-scale problems. The proposed strategy involves a novel refinement stage to efficiently combine and improve the embarrassingly parallel sub-matrix estimations.

We are able to theoretically 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 of Alquier et al. (2019). To the best of our knowledge, this theoretical guarantee for distributed computing is the first of its kind in the literature of matrix completion.

2. Model and Algorithms

2.1. Regularized Least Absolute Deviation Estimator

Let $\mathbf{A}_\star = (A_{\star,ij})_{i,j=1}^{n_1,n_2} \in \mathbb{R}^{n_1 \times n_2}$ be an unknown high-dimensional matrix. Assume the N pairs of observations $\{(\mathbf{X}_k, Y_k)\}_{k=1}^N$ satisfy the trace regression model (1.1) with noise $\{\varepsilon_k\}_{k=1}^N$. The design matrices are assumed to be i.i.d. random matrices that take values in $\mathcal{X}$ (1.2). Let $\pi_{st} = \Pr(\mathbf{X}_k = \mathbf{e}_s(n_1)\mathbf{e}_t^{\mathrm{T}}(n_2))$ be the probability of observing (a noisy realization of) the (s,t)-th entry of $\mathbf{A}_\star$ and denote $\mathbf{\Pi} = (\pi_{1,1}, \ldots, \pi_{n_1,n_2})^{\mathrm{T}}$. Instead of the uniform sampling where $\pi_{st} \equiv \pi$ (Koltchinskii et al., 2011; Rohde & Tsybakov, 2011; Elsener & van de Geer, 2018), out setup allows sampling probabilities to be different across entries, such as in Klopp (2014); Lafond (2015); Cai & Zhou (2016); Alquier et al. (2019). See Condition (C1) for more details. Overall, $(Y_1, \mathbf{X}_1, \varepsilon_1), \ldots, (Y_N, \mathbf{X}_N, \varepsilon_N)$ are i.i.d. tuples of random variables. For notation's simplicity, we let $(Y, \mathbf{X}, \varepsilon)$ be a generic independent tuple of random variables that have the same distribution as $(Y_1, \mathbf{X}_1, \varepsilon_1)$. Without additional specification, the noise variable ε is not identifiable. For example, one can subtract a constant from all entries of $\mathbf{A}_\star$ and add this constant to the noise. To identify the noise, we assume $\mathbb{P}(\epsilon \le 0) = 0.5$, which naturally leads to an interpretation of $\mathbf{A}_\star$ as median, i.e., $A_{\star,ij}$ is the median of $Y \mid \mathbf{X} = \mathbf{e}_i(n_1)\mathbf{e}_j(n_2)^{\mathrm{T}}$. If the noise distribution is symmetric and light-tailed (so that the expectation exists), then $\mathbb{E}(\varepsilon_k) = 0$, and $\mathbf{A}_\star$ is the also the mean matrix ($A_{\star,ij} = \mathbb{E}(Y \mid \mathbf{X} = \mathbf{e}_i(n_1)\mathbf{e}_j(n_2)^{\mathrm{T}})$), which aligns with the target of common MC techniques (Elsener & van de Geer, 2018). Let f be the probability density function of the noise. For the proposed method, the required condition of f is specified in Condition (C3) of Section 3, which is fairly mild and is satisfied by many heavy-tailed distributions whose expectation may not exist.

Define a hypothesis class $\mathcal{B}(a,n,m) = \{\mathbf{A} \in \mathbb{R}^{n\times m} : \|\mathbf{A}\|_\infty \le a\}$ where $a > 0$ such that $\mathbf{A}_\star \in \mathcal{B}(a,n,m)$. In this paper, we use the absolute deviation loss instead of the common quadratic loss (e.g., Candès & Plan, 2010; Koltchinskii et al., 2011; Klopp, 2014). According to Section 4 of the Supplementary Material (Elsener & van de Geer, 2018), $\mathbf{A}_\star$ is also characterized as the minimizer of the population risk:

$$\mathbf{A}_\star = \underset{\mathbf{A}\in\mathcal{B}(a,n_1,n_2)}{\arg\min}\ \mathbb{E}\left\{\left|Y - \mathrm{tr}(\mathbf{X}^{\mathrm{T}}\mathbf{A})\right|\right\}. \tag{2.1}$$

To encourage a low-rank solution, one natural candidate is the following regularized empirical risk estimator (Elsener & van de Geer, 2018; Alquier et al., 2019):

$$\widehat{\mathbf{A}}_{\mathrm{LADMC}} = \underset{\mathbf{A}\in\mathcal{B}(a,n_1,n_2)}{\arg\min} \frac{1}{N}\sum_{k=1}^N \left|Y_k - \mathrm{tr}(\mathbf{X}_k^{\mathrm{T}}\mathbf{A})\right| + \lambda'_N \|\mathbf{A}\|_*, \tag{2.2}$$

where $\|\mathbf{A}\|_*$ denotes the nuclear norm and $\lambda'_N \ge 0$ is a tuning parameter. The nuclear norm is a convex relaxation of the rank which flavors the optimization and analysis of the statistical property (Candès & Recht, 2009).

Due to non-differentiability of the absolute deviation loss, the objective function in (2.1) is the sum of two non-differentiable terms, rendering common computational strategies based on proximal gradient method (e.g., Mazumder et al., 2010; Wong & Lee, 2017) inapplicable. To the best of our knowledge, there is only one existing computational algorithm for (2.1), which is based on a direct application of alternating direction method of multiplier (ADMM) (Alquier et al., 2019). However, this algorithm is slow and not scalable in practice, when the sample size and the matrix dimensions are large, possibly due to the non-differentiable nature of the loss.

We aim to derive a computationally efficient method for estimating the median matrix $\mathbf{A}_\star$ in large-scale MC problems. More specifically, the proposed method consists of two stages: (1) an initial estimation via distributed computing (Section 2.2) and (2) a refinement stage to achieve near-optimal estimation (Section 2.3).

2.2. Distributed Initial Estimator

Similar to many large-scale problems, it is common to harness distributed computing to overcome computational barriers. Motivated by Mackey et al. (2015), we divide the underlying matrix into several sub-matrices, estimate each sub-matrix separately in an embarrassingly parallel fashion and then combine them to form a computationally efficient (initial) estimator of $\mathbf{A}_\star$.

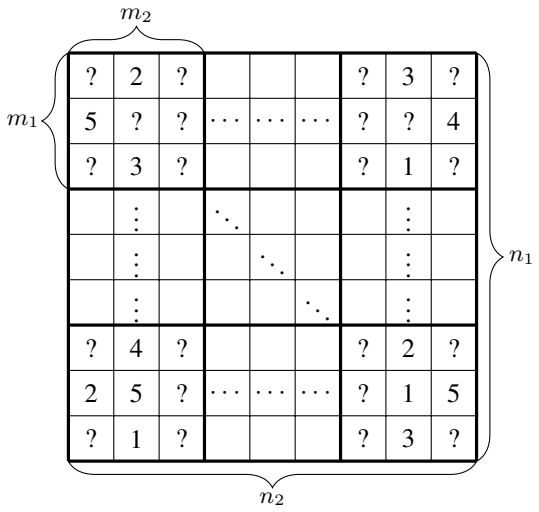

Figure 1. An example of dividing a matrix into sub-matrices.

For the convenience of notations, suppose there exist integers m_1, m_2, l_1 and l_2 such that $l_1 = n_1/m_1$ and $l_2 = n_2/m_2$. (Otherwise, the following description can be easily extended with $l_1 = \lfloor n_1/m_1 \rfloor$ and $l_2 = \lfloor n_2/m_2 \rfloor$ which leads to slightly different sizes in several sub-matrices.) We divide the row indices $1, \ldots, n_1$ into l_1 subsets evenly where each subset contains m_1 index and similarly divide the column indices $1, \ldots, n_2$ into l_2 subsets evenly. Then we obtain $l_1 l_2$ sub-matrices, denoted by $\mathbf{A}_{\star,l} \in \mathbb{R}^{m_1 \times m_2}$ for $l = 1, \ldots, l_1 l_2$. See Figure 1 for a pictorial illustration. Let $\mathbf{\Omega}_l$ be the index set of the observed entries within the l-th sub-matrix $\mathbf{A}_{\star,l}$, and N_l be the corresponding number of observed entries. Next, we apply the ADMM algorithm of Alquier et al. (2019) to each sub-matrix $\mathbf{A}_{\star,l}$ and obtain corresponding median estimator:

$$\widehat{\mathbf{A}}_{\text{LADMC},l} = \underset{\mathbf{A}_l \in \mathcal{B}(a, m_1, m_2)}{\arg\min} \frac{1}{N_l} \sum_{k \in \mathbf{\Omega}_l} \left| Y_k - \text{tr}(\mathbf{X}_{l,k}^{\mathrm{T}} \mathbf{A}_l) \right| + \lambda_{N_l,l} \left\| \mathbf{A}_l \right\|_* , \quad (2.3)$$

where $\mathbf{X}_{l,k}$ is a corresponding sub-design matrix of dimensions $m_1 \times m_2$ and $\lambda_{N_l,l} \geq 0$ is a tuning parameter. Note that the most computationally intensive sub-routine in the ADMM algorithm of Alquier et al. (2019) is (repeated applications of) SVD. For sub-matrices of dimension $m_1 \times m_2$, the computational complexity of a single SVD reduced from $\mathcal{O}(n_1^2 n_2 + n_1 n_2^2)$ to $\mathcal{O}(m_1^2 m_2 + m_1 m_2^2)$.

After we have all the $\widehat{\mathbf{A}}_{\text{LADMC},l}$ for $l = 1, \ldots, l_1 l_2$, we can put these estimators of the sub-matrices back together according to their original positions in the target matrix (see Figure 1), and form an initial estimator $\widehat{\mathbf{A}}_{\text{LADMC},0}$.

This computational strategy is conceptually simple and easily implementable. However, despite the low-rank estimations for each sub-matrix, combining them directly cannot guarantee low-rankness of $\widehat{\mathbf{A}}_{\text{LADMC},0}$. Also, the convergence rate of $\widehat{\mathbf{A}}_{\text{LADMC},0}$ is not guaranteed to be (near-)optimal, as long as m_1, m_2 are of smaller order than n_1, n_2 respectively. See Theorem 1(i) in Section 3. However, for computational benefits, it is desirable to choose small m_1, m_2. In the next section, we leverage this initial estimator and formulate a refinement stage.

2.3. The Idea of Refinement

The proposed refinement stage is based on a form of pseudo data, which leverages the idea from the Newton-Raphson iteration. To describe this idea, we start from the stochastic optimization problem (2.1). Write the loss function as $L(\mathbf{A}; \{Y, \mathbf{X}\}) = |Y - \text{tr}(\mathbf{X}^{\mathrm{T}} \mathbf{A})|$. To solve this stochastic optimization problem, the population version of the Newton-Raphson iteration takes the following form

$$\text{vec}(\mathbf{A}_1) = \text{vec}(\widehat{\mathbf{A}}_0) - \mathbf{H}(\widehat{\mathbf{A}}_0)^{-1} \mathbb{E}_{(Y,\mathbf{X})} \left[\boldsymbol{l}(\widehat{\mathbf{A}}_0; \{Y, \mathbf{X}\}) \right], \quad (2.4)$$

where $(Y, \mathbf{X})$ is defined in Section 2.1 to be independent of the data; $\text{vec}(\mathbf{A})$ is the vectorization of the matrix $\mathbf{A}$; $\widehat{\mathbf{A}}_0$ is an initial estimator (to be specified below); and $\boldsymbol{l}(\mathbf{A}; \{Y, \mathbf{X}\})$ is the sub-gradient of $L(\mathbf{A}; \{Y, \mathbf{X}\})$ with respect to $\text{vec}(\mathbf{A})$. One can show that the population Hessian matrix takes the form $\mathbf{H}(\mathbf{A}) = 2\mathbb{E}_{(Y,\mathbf{X})}(f\{\text{tr}(\mathbf{X}^{\mathrm{T}}(\mathbf{A} - \mathbf{A}_\star)\})\text{diag}(\mathbf{\Pi})$, where we recall that $\mathbf{\Pi} = (\pi_{1,1}, \ldots, \pi_{n_1,n_2})^{\mathrm{T}}$ is the vector of observation probabilities; and $\text{diag}(\cdot)$ transforms a vector into a diagonal matrix whose diagonal is the vector. Also, it can be shown that $\mathbb{E}_{(Y,\mathbf{X})}[\boldsymbol{l}(\mathbf{A}; \{Y, \mathbf{X}\})] = \mathbf{\Pi}\mathbb{E}_{(Y,\mathbf{X})}\{2\mathbb{I}\left[Y - \text{tr}(\mathbf{X}^{\mathrm{T}}\mathbf{A}) \leq 0\right] - 1\}$. Recall that $f(x)$ is the density function of the noise ϵ.

By using $\mathbf{H}(\mathbf{A}_\star) = 2f(0)\text{diag}(\mathbf{\Pi})$ in (2.4), we obtain the following approximation. When the initial estimator $\widehat{\mathbf{A}}_0$ is close to the minimizer $\mathbf{A}_\star$,

$$\begin{aligned}
&\text{vec}(\mathbf{A}_1) \approx \text{vec}(\widehat{\mathbf{A}}_0) \\
&\quad - [2f(0)\text{diag}(\mathbf{\Pi})]^{-1} \mathbb{E}_{(Y,\mathbf{X})}[\boldsymbol{l}(\widehat{\mathbf{A}}_0; \{Y, \mathbf{X}\})] \\
&= \mathbb{E}_{(Y,\mathbf{X})} \Big\{ \text{vec}(\widehat{\mathbf{A}}_0) \\
&\quad - [f(0)]^{-1} \left(\mathbb{I}\left[Y \leq \text{tr}(\mathbf{X}^{\mathrm{T}} \widehat{\mathbf{A}}_0) \right] - \frac{1}{2} \right) \mathbb{1}_{n_1 n_2} \Big\} \\
&= [\text{diag}(\mathbf{\Pi})]^{-1} \mathbb{E}_{(Y,\mathbf{X})} \Big[\text{vec}(\mathbf{X}) \Big\{ \text{vec}(\mathbf{X})^{\mathrm{T}} \text{vec}(\widehat{\mathbf{A}}_0) \\
&\quad - [f(0)]^{-1} \left(\mathbb{I}\left[Y \leq \text{tr}(\mathbf{X}^{\mathrm{T}} \widehat{\mathbf{A}}_0) \right] - \frac{1}{2} \right) \Big\} \Big] \\
&= [\text{diag}(\mathbf{\Pi})]^{-1} \mathbb{E}_{(Y,\mathbf{X})} \left(\text{vec}(\mathbf{X}) \tilde{Y}^o \right) \quad (2.5) \\
&= \{\mathbb{E}_{(Y,\mathbf{X})}[\text{vec}(\mathbf{X})\text{vec}(\mathbf{X})^{\mathrm{T}}]\}^{-1} \mathbb{E}_{(Y,\mathbf{X})} \left(\text{vec}(\mathbf{X}) \tilde{Y}^0 \right)
\end{aligned}$$

where we define the theoretical pseudo data

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

Here $\mathbb{1}_{n_1 n_2}$ denotes the vector of dimension $n_1 n_2$ with all elements equal to 1. Clearly, (2.5) is the vectorization of the solution to $\arg\min_{\mathbf{A}} \mathbb{E}_{(Y,\mathbf{X})}\{\widetilde{Y}^o - \mathrm{tr}(\mathbf{X}^{\mathrm{T}}\mathbf{A})\}^2$, where $\mathrm{tr}(\mathbf{X}^{\mathrm{T}}\mathbf{A})\} = \mathrm{vec}(\mathbf{X})^{\mathrm{T}}\mathrm{vec}(\mathbf{A})$. From this heuristic argument, we can approximate the population Newton update by a least square solution based on the pseudo data $\tilde{Y}^0$, when we start from an $\widehat{\mathbf{A}}_0$ close enough to $\mathbf{A}_\star$. Without the knowledge of $f(0)$, the pseudo data cannot be used. In the above, $f(0)$ can be easily estimated by the kernel density estimator:

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

where $K(x)$ is a kernel function which satisfies Condition (C4) and $h > 0$ is the bandwidth. For each $1 \le k \le N$, we define the actual pseudo data $\widetilde{\mathbf{Y}}$ used in our proposed procedure to be

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

and $\widetilde{\mathbf{Y}} = (\widetilde{Y}_k)$. For finite sample, regularization is imposed to estimate the high-dimensional parameter $\mathbf{A}_\star$. By using $\widetilde{\mathbf{Y}}$, one natural candidate for the estimator of $\mathbf{A}_\star$ is given by

$$\widehat{\mathbf{A}} = \underset{\mathbf{A}\in\mathcal{B}(a,n_1,n_2)}{\arg\min} \frac{1}{N}\sum_{k=1}^{N}\left(\widetilde{Y}_k - \mathrm{tr}(\mathbf{X}_k^{\mathrm{T}}\mathbf{A})\right)^2 + \lambda_N \|\mathbf{A}\|_*, \tag{2.6}$$

where $\|\cdot\|_*$ is the nuclear norm and $\lambda_N \ge 0$ is the tuning parameter. If $\widetilde{\mathbf{Y}}$ is replaced by $\mathbf{Y}$, the optimization (2.6) is a common nuclear-norm regularized empirical risk estimator with quadratic loss which has been well studied in the literature (Candès & Recht, 2009; Candès & Plan, 2010; Koltchinskii et al., 2011; Klopp, 2014). Therefore, with the knowledge of $\widetilde{\mathbf{Y}}$, corresponding computational algorithms can be adopted to solve (2.6). Note that the pseudo data are based on an initial estimator $\widehat{\mathbf{A}}_0$. In Section 3, we show that any initial estimator that fulfills Condition (C5) can be improved by (2.6), which is therefore called a refinement step. It is easy to verify that the initial estimator $\widehat{\mathbf{A}}_{\mathrm{LADMC},0}$ in Section 2.2 fulfills such condition. Note that the initial estimator, like $\widehat{\mathbf{A}}_{\mathrm{LADMC},0}$, introduces complicated dependencies among the entries of $\widetilde{\mathbf{Y}}$, which brings new challenges in analyzing (2.6), as opposed to the common estimator based on $\mathbf{Y}$ with independent entries.

From our theory (Section 3), the refined estimator (2.6) improves upon the initial estimator. Depending on how bad the initial estimator is, a single refinement step may not be good enough to achieve a (near-)optimal estimator. But this can remedied by reapplying the refinement step again and again. In Section 3, we show that a finite number of application of the refinement step is enough. In our numerical experiments, 4–5 applications would usually produce enough improvement. Write $\widehat{\mathbf{A}}^{(1)} = \widehat{\mathbf{A}}$ given in (2.6) as the estimator from the first iteration and we can construct an iterative procedure to estimate $\mathbf{A}_\star$. In particular, let $\widehat{\mathbf{A}}^{(t-1)}$ be the estimator in the $(t-1)$-th iteration. Define

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

where $K(x)$ is the same smoothing function used to estimate $f(0)$ in the first step and $h_t \to 0$ is the bandwidth for the t-th iteration. Similarly, for each $1 \le k \le N$, define

$$\widetilde{Y}_k^{(t)} = \mathrm{tr}(\mathbf{X}_k^{\mathrm{T}}\widehat{\mathbf{A}}^{(t-1)}) - \left(\widehat{f}^{(t)}(0)\right)^{-1} \times \left(\mathbb{I}\left[Y_k \le \mathrm{tr}(\mathbf{X}_k^{\mathrm{T}}\widehat{\mathbf{A}}^{(t-1)})\right] - \frac{1}{2}\right). \tag{2.7}$$

We propose the following estimator

$$\widehat{\mathbf{A}}^{(t)} = \underset{\mathbf{A}\in\mathcal{B}(a,n_1,n_2)}{\arg\min} \frac{1}{N}\sum_{k=1}^{N}\left(\widetilde{Y}_k^{(t)} - \mathrm{tr}(\mathbf{X}_k^{\mathrm{T}}\mathbf{A})\right)^2 + \lambda_{N,t}\|\mathbf{A}\|_*, \tag{2.8}$$

where $\lambda_{N,t}$ is the tunning parameter in the t-th iteration. To summarize, we list the full algorithm in Algorithm 1.

3. Theoretical Guarantee

To begin with, we introduce several notations. Let $m_+ = m_1 + m_2$, $m_{\max} = \max\{m_1, m_2\}$ and $m_{\min} = \min\{m_1, m_2\}$. Similarly, write $n_+ = n_1 + n_2$, $n_{\max} = \max\{n_1, n_2\}$ and $n_{\min} = \min\{n_1, n_2\}$. For a given matrix $\mathbf{A} = (A_{ij}) \in \mathbb{R}^{n_1\times n_2}$, denote $\sigma_i(\mathbf{A})$ be the i-th largest singular value of matrix $\mathbf{A}$. Let $\|\mathbf{A}\| = \sigma_1(\mathbf{A})$, $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^{n_1}\sum_{j=1}^{n_2} A_{ij}^2}$ and $\|\mathbf{A}\|_* = \sum_{i=1}^{n_{\min}}\sigma_i(\mathbf{A})$ be the spectral norm (operator norm), the infinity norm, the Frobenius norm and the trace norm of a matrix $\mathbf{A}$ respectively. Define a class of matrices $\mathcal{C}_*(n_1, n_2) = \{\mathbf{A} \in \mathbb{R}^{n_1\times n_2} : \|\mathbf{A}\|_* \le 1\}$. Denote the rank of matrix $\mathbf{A}_\star$ by $r_\star = \mathrm{rank}(\mathbf{A}_\star)$ for simplicity. With these notations, we describe the following conditions which are useful in our theoretical analysis.

(C1) For each $k = 1, \ldots, N$, the design matrix $\mathbf{X}_k$ takes value in the canonical basis $\mathcal{X}$ as defined in (1.2). There exist positive constants $\underline{c}$ and $\bar{c}$ such that for any $(s,t) \in \{1,\ldots,n_1\}\times\{1,\ldots,n_2\}$, $\underline{c}/(n_1 n_2) \le \Pr(\mathbf{X}_k = \mathbf{e}_s(n_1)\mathbf{e}_t^{\mathrm{T}}(n_2)) \le \bar{c}/(n_1 n_2)$.

Algorithm 1 Distributed Least Absolute Deviation Matrix Completion

Input: Observed data pairs $\{\mathbf{X}_k, Y_k\}$ for $k = 1, \ldots, N$, number of observations N, dimensions of design matrix $\mathbf{X}$ n_1, n_2, dimensions of sub-matrices to construct the initial estimator m_1, m_2 and the split subsets $\mathbf{\Omega}_l$ for $l = 1, \ldots, l_1 l_2$, kernel function K, a sequence of bandwidths h_t and the regularization parameters $\lambda_{N,t}$ for $t = 1, \ldots, T$.

1: Get the robust low-rank estimator of each $\mathbf{A}_{\star,l}$ by solving the minimization problem (2.3) in parallel.
2: Set $\widehat{\mathbf{A}}^{(0)}$ to be the same as the initial estimator $\widehat{\mathbf{A}}_{\text{LADMC},0}$ by putting $\widehat{\mathbf{A}}_{\text{LADMC},l}$ together.
3: **for** $t = 1, 2 \ldots, T$ **do**
4: Compute $\widehat{f}^{(t)}(0) := (Nh_t)^{-1} \sum_{k=1}^{N} K(h_t^{-1}(Y_k - \text{tr}\{\mathbf{X}_k^{\mathrm{T}} \widehat{\mathbf{A}}^{(t-1)}\}))$.
5: Construct the pseudo data $\{\widetilde{Y}_k^{(t)}\}$ by equation (2.7).
6: Plugin the pseudo data $\{\widetilde{Y}_k^{(t)}\}$ and compute the estimator $\widehat{\mathbf{A}}^{(t)}$ by solving the minimization problem (2.8).
7: **end for**

Output: The final estimator $\widehat{\mathbf{A}}^{(T)}$.

(C2) The local dimensions m_1, m_2 on each block satisfies $m_1 \geq n_1^c$ and $m_2 \geq n_2^c$ for some $0 < c < 1$. The number of observations in each block N_l are comparable for all $l = 1, \ldots, l_1 l_2$, i.e, $N_l \asymp m_1 m_2 N/(n_1 n_2)$.

(C3) The density function $f(\cdot)$ is Lipschitz continuous (i.e., $|f(x) - f(y)| \leq C_L |x - y|$ for any $x, y \in \mathbb{R}$ and some constant $C_L > 0$). Moreover, there exists a constant $c > 0$ such that $f(u) \geq c$ for any $|u| \leq 2a$. Also, $\Pr(\epsilon_k \leq 0) = 0.5$ for each $k = 1, \ldots, N$.

Theorem 1 (Alquier et al. (2019), Theorem 4.6, Initial estimator)**.** *Suppose that Conditions (C1)–(C3) hold and $\mathbf{A}_\star \in \mathcal{B}(a, n_1, n_2)$. For each $l = 1, \ldots, n_1 n_2/(m_1 m_2)$, assume that there exists a matrix with rank at most s_l in $\mathbf{A}_{\star,l} + (\rho_{s_l}/20)\mathcal{C}_*(m_1, m_2)$ where $\rho_{s_l} = C_\rho (s_l m_1 m_2)(\log(m_+)/(m_+ N_l))^{1/2}$ with the universal constant C_ρ.*

(i) Then there exist universal constants $c(\underline{c}, \overline{c})$ and C such that with $\lambda_{N_l,l} = c(\underline{c}, \overline{c})\sqrt{\log(m_+)/(m_{\min} N_l)}$, the estimator $\widehat{\mathbf{A}}_{LADMC,l}$ in (2.3) satisfies

$$\frac{1}{\sqrt{m_1 m_2}} \left\| \widehat{\mathbf{A}}_{LADMC,l} - \mathbf{A}_{\star,l} \right\|_F \leq C \min \left\{ \sqrt{\frac{s_l m_{\max} \log(m_+)}{N_l}}, \|\mathbf{A}_{\star,l}\|_*^{1/2} \left(\frac{\log(m_+)}{m_{\min} N_l} \right)^{1/4} \right\}, \quad (3.1)$$

with probability at least $1 - C\exp(-C s_l m_{\max} \log(m_+))$.

(ii) Moreover, by putting these $l_1 l_2$ estimators $\widehat{\mathbf{A}}_{LADMC,l}$ together, for the same constant C in (i), we have the initial estimator $\widehat{\mathbf{A}}_{LADMC,0}$ satisfies

$$\frac{\left\| \widehat{\mathbf{A}}_{LADMC,0} - \mathbf{A}_\star \right\|_F}{\sqrt{n_1 n_2}} \leq C \min \left\{ \sqrt{\frac{\{\sum_{l=1}^{l_1 l_2} s_l\} m_{\max} \log(m_+)}{N}}, \left(\sum_{l=1}^{l_1 l_2} \|\mathbf{A}_{\star,l}\|_*^{1/2} \right) \left(\frac{m_{\max} \log(m_+)}{n_1 n_2 N} \right)^{1/4} \right\},$$

with probability at least $1 - C\exp(\log(n_1 n_2) - C m_{\max} \log(m_+))$.

From Theorem 1, we can guarantee the convergence of the sub-matrix estimator $\widehat{\mathbf{A}}_{\text{LADMC},l}$ when $m_1, m_2 \to \infty$. For the initial estimator $\widehat{\mathbf{A}}_{\text{LADMC},0}$, under Condition (C3) and that all the sub-matrices are low-rank ($s_l \asymp 1$ for all l), we require the number of observation $N \geq C_1 (m_1 m_2)^{-1} (n_1 n_2) m_{\max} \log(m_+)$ for some constant C_1 to ensure the convergence. As for the rate of convergence, $\sqrt{(n_1 n_2) m_{\max} \log(m_+)/(N m_1 m_2)}$ is slower than the classical optimal rate $\sqrt{r_\star n_{\max} \log(n_+)/N}$ when m_1, m_2 are of smaller than n_1, n_2 respectively.

(C4) Assume the kernel functions $K(\cdot)$ is integrable with $\int_{-\infty}^{\infty} K(u) du = 1$. Moreover, assume that $K(\cdot)$ satisfies $K(u) = 0$ if $|u| \geq 1$. Further, assume that $K(\cdot)$ is differentiable and its derivative $K'(\cdot)$ is bounded.

(C5) The initial estimator $\widehat{\mathbf{A}}_0$ satisfies $(n_1 n_2)^{-1/2} \|\widehat{\mathbf{A}}_0 - \mathbf{A}_\star\|_F = O_{\mathsf{P}}((n_1 n_2)^{-1/2} a_N)$, where the initial rate $(n_1 n_2)^{-1/2} a_N = o(1)$.

For the notation consistency, denote the initial rate $a_{N,0} = a_N$ and define that

$$a_{N,t} = \sqrt{\frac{r_\star (n_1 n_2) n_{\max} \log(n_+)}{N}} + \frac{n_{\min}}{\sqrt{r_\star}} \left(\frac{\sqrt{r_\star} a_{N,0}}{n_{\min}} \right)^{2^t} \quad (3.2)$$

Theorem 2 (Repeated refinement)**.** *Suppose that Conditions (C1)–(C5) hold and $\mathbf{A}_\star \in \mathcal{B}(a, n_1, n_2)$. By choosing the bandwidth $h_t \asymp (n_1 n_2)^{-1/2} a_{N,t-1}$ where $a_{N,t}$ is defined as in (3.2) and taking*

$$\lambda_{N,t} = C \left(\sqrt{\frac{\log(n_+)}{n_{\min} N}} + \frac{a_{N,t-1}^2}{n_{\min}(n_1 n_2)} \right),$$

where C is a sufficient large constant, we have

$$\frac{\left\| \widehat{\mathbf{A}}^{(t)} - \mathbf{A}_\star \right\|_F^2}{n_1 n_2} = O_P \left[\max \left\{ \sqrt{\frac{\log(n_+)}{N}}, r_\star \left(\frac{n_{\max} \log(n_+)}{N} + \frac{a_{N,t-1}^4}{n_{\min}^2 (n_1 n_2)} \right) \right\} \right]. \quad (3.3)$$

When the iteration number $t = 1$, it means one-step refinement from the initial estimator $\widehat{\mathbf{A}}_0$. For the right hand side

of (3.3), it is noted that both the first term $\sqrt{\log(n_+)/N}$ and the second term $r_\star n_{\max}\log(n_+)/N$ are seen in the error bound of existing works (Elsener & van de Geer, 2018; Alquier et al., 2019). The bound has an extra third term $r_\star a_{N,0}^4/(n_{\min}^2(n_1n_2))$ due to the initial estimator. After one round of refinement, one can see that the third term $r_\star a_{N,0}^4/(n_{\min}^2(n_1n_2))$ in (3.3) is faster than $a_{N,0}^2/(n_1n_2)$, the convergence rate of the initial estimator (see Condition (C5)), because $r_\star n_{\min}^{-2}a_{N,0}^2 = o(1)$.

With the increasing of the iteration number t, Theorem 2 shows that the estimator can be refined again and again, until near-optimal rate of convergence is achieved. It can be shown that when the iteration number t exceeds certain number, i.e,

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

for some $c_0 > 0$, the second term in the term associated with $r_\star$ is dominated by the first term and the convergence rate of $\widehat{\mathbf{A}}^{(t)}$ becomes $r_\star n_{\max}N^{-1}\log(n_+)$ which is the near-optimal rate $r_\star n_{\max}N^{-1}$ (optimal up to a logarithmic factor). Note that the number of iteration t is usually small due to the logarithmic transformation.

3.1. Main Lemma and Proof Outline

For the t-th refinements, let $\xi_k^{(t)} = \widetilde{Y}_k^{(t)} - \langle \mathbf{X}_k, \mathbf{A}_\star\rangle$ be the residual of the pseudo data. Also, define the stochastic terms $\mathbf{\Sigma}^{(t)} = N^{-1}\sum_{k=1}^N \xi_k^{(t)}\mathbf{X}_k$. To provide an upper bound of $(n_1n_2)^{-1}\|\widehat{\mathbf{A}}^{(t)} - \mathbf{A}_\star\|_F^2$ in Theorem 2, we follow the standard arguments, as used in corresponding key theorems in, e.g., Koltchinskii et al. (2011); Klopp (2014). The key is to control the spectral norm of the stochastic term $\mathbf{\Sigma}^{(t)}$. A specific challenge of our setup is the dependency among the residuals $\{\xi_i^{(t)}\}$. We tackle this by the following lemma:

Lemma 1. *Suppose that Conditions (C1)–(C5) hold and $\mathbf{A}_\star \in \mathcal{B}(a, n_1, n_2)$. For any iteration $t \geq 1$, we choose the bandwidth $h_t \asymp (n_1n_2)^{-1/2}a_{N,t}$ where $a_{N,t}$ is defined as in (3.2). Then we have*

$$\left\|\mathbf{\Sigma}^{(t)}\right\| = O_P\left(\sqrt{\frac{\log(n_+)}{n_{\min}N}} + \frac{a_{N,t-1}^2}{n_{\min}(n_1n_2)}\right).$$

We now give a proof outline of Lemma 1 for $t = 1$. The same argument can be applied iteratively to achieve the repeated refinement results as shown in Lemma 1.

In our proof, we decompose the stochastic term $\Sigma^{(1)}$ into three components $\mathbf{H}_N(\widehat{\mathbf{A}}_0)$, $(N\widehat{f}(0))^{-1}\sum_{i=1}^N[\mathbf{X}_i\mathbb{I}[\epsilon_i \leq 0] - \mathbf{X}_i f(0)]$ and $\widehat{f}^{-1}(0)U_N$ where

$$\mathbf{H}_N(\mathbf{A}) = \frac{1}{N}\sum_{i=1}^N \mathbf{X}_i \mathrm{tr}\left\{\mathbf{X}_i^{\mathrm{T}}(\mathbf{A} - \mathbf{A}_\star)\right\} + \frac{\widehat{f}^{-1}(0)}{N}\sum_{i=1}^N \mathbf{X}_i\left\{f\left[\mathrm{tr}\left\{\mathbf{X}_i^{\mathrm{T}}(\mathbf{A} - \mathbf{A}_\star)\right\}\right] - f(0)\right\},$$

and $U_N = \sup_{\|\mathbf{A}-\mathbf{A}_\star\|_F \leq a_N} \|\mathbf{B}_N(\mathbf{A})\|$ with

$$\mathbf{B}_N(\mathbf{A}) = \frac{1}{N}\sum_{i=1}^N\left[\mathbf{X}_i\mathbb{I}\left[\epsilon_i \leq \mathrm{tr}\left\{\mathbf{X}_i^{\mathrm{T}}(\mathbf{A} - \mathbf{A}_\star)\right\}\right] - \mathbf{X}_i f\left(\mathrm{tr}\left\{\mathbf{X}_i^{\mathrm{T}}(\mathbf{A} - \mathbf{A}_\star)\right\}\right)\right] - \frac{1}{N}\sum_{i=1}^N\left[\mathbf{X}_i\mathbb{I}[\epsilon_i \leq 0] - \mathbf{X}_i f(0)\right].$$

Then we control their spectral norms separately.

For $\mathbf{H}_N(\widehat{\mathbf{A}}_0)$, we first bound $|\mathbf{v}^{\mathrm{T}}\mathbf{H}_N(\widehat{\mathbf{A}}_0)\mathbf{u}|$ for fixed $\mathbf{u}$ and $\mathbf{v}$ where $\|\mathbf{u}\| = \|\mathbf{v}\| = 1$, by separating the random variables $\mathbf{X}_k$ and ϵ_k from $\widehat{\mathbf{A}}_0 - \mathbf{A}_\star$, and then applying the exponential inequality in Lemma 1 of Cai & Liu (2011). To control the spectral norm, we take supremum over $\mathbf{u}$ and $\mathbf{v}$, and the corresponding uniform bound can be derived using an $\mathcal{E}-$net argument. The same technique can be used to handle the term U_N. Therefore, we can bound the spectral norm of $\mathbf{H}_N(\widehat{\mathbf{A}}_0)$ and U_N for any initial estimator that satisfies Condition (C5).

As for the term $(N\widehat{f}(0))^{-1}\sum_{i=1}^N[\mathbf{X}_i\mathbb{I}[\epsilon_i \leq 0] - \mathbf{X}_i f(0)]$, we first note that it is not difficult to control a simplified version: $(Nf(0))^{-1}\sum_{i=1}^N[\mathbf{X}_i\mathbb{I}[\epsilon_i \leq 0] - \mathbf{X}_i f(0)]$, with $f(0)$ instead of $\widehat{f}(0)$. To control our target term, we provide Proposition A.1 in the supplementary materials which shows that $|\widehat{f}(0) - f(0)| = O_P(\sqrt{\frac{\log(n_+)}{Nh}} + \frac{a_N}{\sqrt{n_1n_2}})$.

4. Experiments

4.1. Synthetic Data

We conducted a simulation study, under which we fixed the dimensions to $n_1 = n_2 = 400$. In each simulated data, the target matrix $\mathbf{A}_\star$ was generated as $\mathbf{U}\mathbf{V}^{\mathrm{T}}$, 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 the standard normal distributions $\mathcal{N}(0,1)$ independently. Here r was set to 3. Thus $\mathbf{A}_\star = \mathbf{U}\mathbf{V}^{\mathrm{T}}$ was a low-rank matrix. The missing rate was 0.2, which corresponds to $N = 32,000$ We adopted the uniform missing mechanism where all entries had the same chance of being observed. We considered the following four noise distributions:

S1 Normal: $\epsilon \sim \mathcal{N}(0,1)$.

S2 Cauchy: $\epsilon \sim$ Cauchy$(0,1)$.

S3 Exponential: $\epsilon \sim \exp(1)$.

S4 t-distribution with degree of freedom 1: $\epsilon \sim \mathrm{t}_1$.

We note that Cauchy distribution is a very heavy-tailed distribution and its first moment (expectation) does not exist. For each of these four settings, we repeated the simulation for 500 times.

Denote the proposed median MC procedure given in Algorithm 1 by DLADMC (Distributed Least Absolute Deviations Matrix Completion). Due to Theorem 1(ii), $\|\widehat{\mathbf{A}}_{\mathrm{LADMC},0} - \mathbf{A}^\star\|_F = O_p(\sqrt{(n_1n_2)^2 m_{\max}\log(m_+)/(m_1m_2N)})$, we fixed

$$a_N = a_{N,0} = c_1\sqrt{\frac{(n_1n_2)^2 m_{\max}\log(m_+)}{m_1m_2N}},$$

where the constant $c_1 = 0.1$. From our experiences, smaller c_1 leads to similar results. As $h \asymp (n_1n_2)^{-1/2}a_N$, the bandwidth h was simply set to $h = c_2(n_1n_2)^{-1/2}a_N$, and similarly, $h_t = c_2(n_1n_2)^{-1/2}a_{N,t}$ where $a_{N,t}$ was defined by (3.2) with $c_2 = 0.1$. In addition, all the tuning parameters $\lambda_{N,t}$ in Algorithm 1 were chosen by validation. Namely, we minimized the absolute deviation loss evaluated on an independently generated validation sets with the same dimensions n_1, n_2. For the choice of the kernel functions $K(\cdot)$, we adopt the commonly used bi-weight kernel function,

$$K(x) = \begin{cases} 0, & x \le -1 \\ -\frac{315}{64}x^6 + \frac{735}{64}x^4 - \frac{525}{64}x^2 + \frac{105}{64}, & -1 \le x \le 1 \\ 0, & x \ge 1 \end{cases}.$$

It is easy to verify that $K(\cdot)$ satisfies Condition (C1) in Section 3. If we compute $e = \|\widehat{\mathbf{A}}^{(t)} - \widehat{\mathbf{A}}^{(t-1)}\|_F^2/\|\widehat{\mathbf{A}}^{(t-1)}\|_F^2$ and stop the algorithm once $e \le 10^{-5}$, it typically only requires $4-5$ iterations. We simply report the results of the estimators with $T = 4$ or $T = 5$ iterations in Algorithm 1 (depending on the noise distribution).

We compared the performance of the proposed method (DLADMC) with three other approaches:

(a) BLADMC: Blocked Least Absolute Deviation Matrix Completion $\widehat{\mathbf{A}}_{\mathrm{LADMC},0}$, the initial estimator proposed in section 2.2. 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).

The tuning parameters in these four methods were chosen based on the same validation set. We followed the selection procedure in Section 9.4 of Mazumder et al. (2010) to choose λ. Instead of fixing K to 1.5 or 2 as in Mazumder et al. (2010), we choose K by an additional pair of training and validation sets (aside from the 500 simulated datasets). We did this for every method to ensure a fair comparison. The performance of all the methods were evaluated via root mean square error (RMSE) and mean absolute error (MAE). The estimated ranks are also reported.

Table 1. The average RMSEs, MAEs, estimated ranks and their standard errors (in parentheses) of DLADMC, BLADMC, ACL and MHT over 500 simulations. The number in the first column within the parentheses represents T in Algorithm 1 for DLADMC.

(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)
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
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)
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)

From Table 1, we can see that both DLADMC and MHT produced low-rank estimators while BLADMC and ACL could not reduce the rank too much. As expected, when the noise is Gaussian, MHT performed best in terms of RMSE and MAE. Meanwhile, DLADMC and ACL were close to each other and slightly worse than MHT. It is not surprising that BLADMC was the worst due to its simple way to combine sub-matrices. As for Setting S3, ACL outperformed other three methods while the performances of DLADMC and MHT are close. For the heavy-tailed Settings S2 and S4, our proposed DLADMC performed significantly better than ACL, and MHT fails.

Moreover, to investigate whether the refinement step can be isolated from the distributed optimization, we run the refinement step on an initial matrix that is synthetically

generated by making small noises to the ground-truth matrix $\mathbf{A}_\star$, as suggested by a reviewer. We provide these results in Section B.1 of the supplementary material.

4.2. Real-World Data

We tested various methods on the MovieLens-100K[1] dataset. This data set consists of 100,000 movie ratings provided by 943 viewers on 1682 movies. The ratings range from 1 to 5. To evaluate the performance of different methods, we directly used the data splittings from the data provider, which splits the data into two sets. We refer them to as RawA and RawB. Similar to Alquier et al. (2019), we added artificial outliers by randomly changing 20% of ratings that are equal to 5 in the two sets, RawA and RawB, to 1 and constructed OutA and OutB respectively. To avoid rows and columns that contain too few observations, we only keep the rows and columns with at least 20 ratings. The resulting target matrix $\mathbf{A}_\star$ is of dimension 739×918. Before we applied those four methods as described in Section 4.1, the data was preprocessed by a bi-scaling procedure (Mazumder et al., 2010). For the proposed DLADMC, we fixed the iteration number to 7. It is noted that the relative error stopping criterion (in Section 4.1) did not result in a stop within the first 7 iteration, where 7 is just a user-specified upper bound in the implementation. To understand the effect of this bound, we provided additional analysis of this upper bound in Section 2.2 of the supplementary material. Briefly, our conclusion in the rest of this section is not sensitive to this choice of upper bound. The tuning parameters for all the methods were chosen by 5-fold cross-validations. The RMSEs, MAEs, estimated ranks and the total computing time (in seconds) are reported in Table 2. For a fair comparison, we recorded the time of each method in the experiment with the selected tuning parameter.

It is noted that under the raw data RawA and RawB, both the proposed DLADMC and the least absolute deviation estimator ACL performed similarly as the least squares estimator MHT. BLADMC lost some efficiency due to the embarrassingly parallel computing. For the dataset with outliers, the proposed DLADMC and the least absolute deviation estimator ACL performed better than MHT. Although DLADMC and ACL had similar performance in terms of the RMSEs and MAEs, DLADMC required much lower computing cost.

Suggested by a reviewer, we also performed an experiment with a bigger data set (MovieLens-1M dataset: 1,000,209 ratings of approximately 3,900 movies rated by 6,040 users.) However, ACL is not scalable, and, due to time limitations, we stopped the fitting of ACL when the running time of ACL exceeds five times of the proposed DLADMC. In our analysis, we only compared the remaining methods. The conclusions were similar as in the smaller MoviewLens-100K dataset. The details are presented in Section B.2 of the supplementary material.

[1] https://grouplens.org/datasets/movielens/100k/

Table 2. The RMSEs, MAEs and estimated ranks of DLADMC, BLADMC, ACL and MHT under dimensions $n_1 = 739$ and $n_2 = 918$.

		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

5. Conclusion

In this paper, we address the problem of median MC and obtain a computationally efficient estimator for large-scale MC problems. We construct the initial estimator in an embarrassing parallel fashion and refine it through regularized least square minimizations based on pseudo data. The corresponding non-standard asymptotic analysis are established. This shows that the proposed DLADMC achieves the (near-)oracle convergence rate. Numerical experiments are conducted to verify our conclusions.

Acknowledgments

Weidong Liu's research is supported by National Program on Key Basic Research Project (973 Program, 2018AAA0100704), NSFC Grant No. 11825104 and 11690013, Youth Talent Support Program, and a grant from Australian Research Council. Xiaojun Mao's research is supported by Shanghai Sailing Program 19YF1402800. Raymond K.W. Wong's research is partially supported by the National Science Foundation under Grants DMS-1806063, DMS-1711952 (subcontract) and CCF-1934904.

References

Alquier, P., Cottet, V., and Lecué, G. Estimation bounds and sharp oracle inequalities of regularized procedures with lipschitz loss functions. *The Annals of Statistics*, 47 (4):2117–2144, 2019.

Bach, F. R. Consistency of trace norm minimization. *Journal of Machine Learning Research*, 9(Jun):1019–1048, 2008.

Bennett, J. and Lanning, S. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, pp. 35, 2007.

Cai, T. T. and Liu, W. Adaptive thresholding for sparse covariance matrix estimation. *Journal of the American Statistical Association*, 106(494):672–684, 2011.

Cai, T. T. and Zhou, W.-X. Matrix completion via max-norm constrained optimization. *Electronic Journal of Statistics*, 10(1):1493–1525, 2016.

Candès, E. J. and Plan, Y. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

Candès, E. J. and Recht, B. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

Candès, E. J. and Tao, T. The power of convex relaxation: Near-optimal matrix completion. *Information Theory, IEEE Transactions on*, 56(5):2053–2080, 2010.

Candès, E. J., Li, X., Ma, Y., and Wright, J. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3): 11, 2011.

Chandrasekaran, V., Sanghavi, S., Parrilo, P. A., and Willsky, A. S. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

Chen, X., Liu, W., Mao, X., and Yang, Z. Distributed high-dimensional regression under a quantile loss function. *arXiv preprint arXiv:1906.05741*, 2019a.

Chen, Y., Xu, H., Caramanis, C., and Sanghavi, S. Robust matrix completion and corrupted columns. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 873–880, 2011.

Chen, Y., Jalali, A., Sanghavi, S., and Caramanis, C. Low-rank matrix recovery from errors and erasures. *IEEE Transactions on Information Theory*, 59(7):4324–4337, 2013.

Chen, Y., Chi, Y., Fan, J., Ma, C., and Yan, Y. Noisy matrix completion: Understanding statistical guarantees for convex relaxation via nonconvex optimization. *arXiv preprint arXiv:1902.07698*, 2019b.

Chen, Y., Fan, J., Ma, C., and Yan, Y. Bridging convex and nonconvex optimization in robust pca: Noise, outliers, and missing data. *arXiv preprint arXiv:2001.05484*, 2020.

Davies, P. L. Aspects of robust linear regression. *The Annals of statistics*, 21(4):1843–1899, 1993.

Elsener, A. and van de Geer, S. Robust low-rank matrix estimation. *The Annals of Statistics*, 46(6B):3481–3509, 2018.

Fan, J., Gong, W., and Zhu, Z. Generalized high-dimensional trace regression via nuclear norm regularization. *Journal of Econometrics*, 212(1):177–202, 2019.

Gross, D. Recovering low-rank matrices from few coefficients in any basis. *Information Theory, IEEE Transactions on*, 57(3):1548–1566, 2011.

Huber, P. J. *Robust statistics*. Springer, 2011.

Keshavan, R. H., Montanari, A., and Oh, S. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(69):2057–2078, 2010.

Klopp, O. Noisy low-rank matrix completion with general sampling distribution. *Bernoulli*, 20(1):282–303, 2014.

Klopp, O., Lounici, K., and Tsybakov, A. B. Robust matrix completion. *Probability Theory and Related Fields*, 169 (1-2):523–564, 2017.

Koltchinskii, V., Lounici, K., and Tsybakov, A. B. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *The Annals of Statistics*, 39(5):2302–2329, 2011.

Lafond, J. Low rank matrix completion with exponential family noise. In *Conference on Learning Theory*, pp. 1224–1243, 2015.

Li, X. Compressed sensing and matrix completion with constant proportion of corruptions. *Constructive Approximation*, 37(1):73–99, 2013.

Mackey, L., Talwalkar, A., and Jordan, M. I. Distributed matrix completion and robust factorization. *Journal of Machine Learning Research*, 16(28):913–960, 2015.

Mazumder, R., Hastie, T., and Tibshirani, R. Spectral regularization algorithms for learning large incomplete matrices. *Journal of Machine Learning Research*, 11(80): 2287–2322, 2010.

Negahban, S. and Wainwright, M. J. Estimation of (near) low-rank matrices with noise and high-dimensional scaling. *The Annals of Statistics*, 39(2):1069–1097, 2011.

Negahban, S. and Wainwright, M. J. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13 (53):1665–1697, 2012.

Rohde, A. and Tsybakov, A. B. Estimation of high-dimensional low-rank matrices. *The Annals of Statistics*, 39(2):887–930, 2011.

Srebro, N., Rennie, J., and Jaakkola, T. S. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pp. 1329–1336, 2005.

Wong, R. K. W. and Lee, T. C. M. Matrix completion with noisy entries and outliers. *Journal of Machine Learning Research*, 18(147):1–25, 2017.

Xia, D. and Yuan, M. Statistical inferences of linear forms for noisy matrix completion. *arXiv preprint arXiv:1909.00116*, 2019.