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 $5 \quad 10^5$ rows and $2 \quad 10^4$ 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 $(X_k; Y_k)$, for $k = 1; : : : ; N$, are observed, where X_k's are random design matrices of dimension $n_1 \quad n_2$ and Y_k's are response variables in $\mathbb{R}$. The trace regression model assumes

$$Y_k = \mathrm{tr}\ X_k^{\mathsf{T}} A_? + {}_k; \qquad k = 1; : : : ; N; \tag{1.1}$$

where $\mathrm{tr}(A)$ denotes the trace of a matrix A, and $A_? \ 2 \ \mathbb{R}^{n_1 \times n_2}$ is an unknown target matrix. Moreover, the elements of $= ({}_1; : : : ; {}_N)$ are N i.i.d. random noise variables independent of the design matrices. In MC setup, the design matrices X_k's are assumed to lie in the set of canonical bases

$$X = f e_j(n_1) e_k(n_2)^{\mathsf{T}} : j = 1; : : : ; n_1; k = 1; : : : ; n_2 g; \tag{1.2}$$

where $e_j(n_1)$ is the j-th unit vector in $\mathbb{R}^{n_1}$, and $e_k(n_2)$ 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, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

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 $A_? + S + E$, where the low-rank matrix $A_?$ is the target uncontaminated component; the sparse matrix S models the gross corruptions (outliers) locating at a small proportion of entries; and 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 E) with an element-wisely sparse S. Chen et al. (2011) studied the noiseless model with column-wisely sparse S. Under the model with element-wisely sparse S, Wong & Lee (2017) looked into the setting of arbitrary (not necessarily random) noise E, while Klopp et al. (2017) and Chen et al. (2020) studied random (sub-Gaussian) noise model for 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 $A_?$. 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 $\pi_{st} = \Pr(\mathbf{X}_k = \mathbf{e}_s(n_1)\mathbf{e}_t^\top(n_2))$ be the probability of observing (a noisy realization of) the $(s; t)$-th entry of $\mathbf{A}_\star$ and denote $\boldsymbol{\pi} = (\pi_{1;1}; \ldots; \pi_{n_1;n_2})^\top$. Instead of the uniform sampling where $\pi_{st} \equiv$ (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 $\mathbb{P}(\varepsilon \le 0) = 0.5$, which naturally leads to an interpretation of $\mathbf{A}_\star$ as median, i.e., $A_{\star;ij}$ is the median of $Y \mid \mathbf{X} = \mathbf{e}_i(n_1)\mathbf{e}_j(n_2)^\top$. 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 $(A_{\star;ij} = \mathbb{E}(Y \mid \mathbf{X} = \mathbf{e}_i(n_1)\mathbf{e}_j(n_2)^\top))$, 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:

$$\mathbf{A}_\star = \operatorname*{arg\,min}_{\mathbf{A} \in \mathcal{B}(a; n_1; n_2)} \mathbb{E} \left| Y - \operatorname{tr}(\mathbf{X}^\top \mathbf{A}) \right| : \tag{2.1}$$

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):

$$\widehat{\mathbf{A}}_{\text{LADMC}} = \operatorname*{arg\,min}_{\mathbf{A} \in \mathcal{B}(a; n_1; n_2)} \frac{1}{N} \sum_{k=1}^{N} \left| Y_k - \operatorname{tr}(\mathbf{X}_k^\top \mathbf{A}) \right| + \lambda_N \|\mathbf{A}\|_* ; \tag{2.2}$$

where $\|\mathbf{A}\|_*$ denotes the nuclear norm and $\lambda_N > 0$ 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 $\mathbf{A}_\star$.

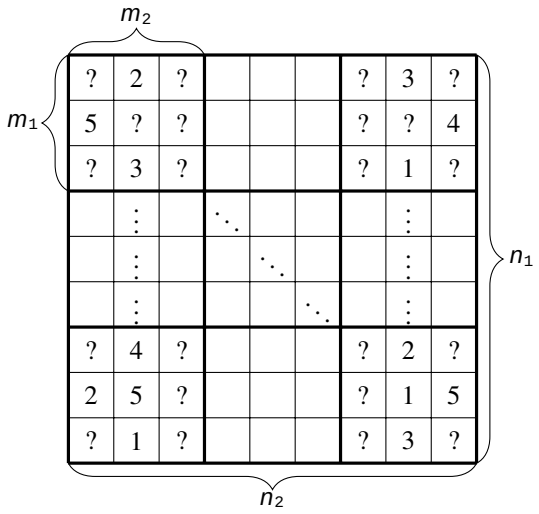

Figure 1. An example of dividing a matrix into sub-matrices.

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 $\mathbf{A}_{?,l} \in \mathbb{R}^{m_1 \times m_2}$ for $l = 1, \ldots, l_1 l_2$. See Figure 1 for a pictorial illustration. Let Ω_l be the index set of the observed entries within the l-th sub-matrix $\mathbf{A}_{?,l}$, 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 $\mathbf{A}_{?,l}$ and obtain corresponding median estimator:

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

where $\mathbf{X}_{l;k}$ is a corresponding sub-design matrix of dimensions $m_1 \times m_2$ and $\alpha_{N_l;l} > 0$ 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 $O(n_1^2 n_2 + n_1 n_2^2)$ to $O(m_1^2 m_2 + m_1 m_2^2)$.

After we have all the $\widehat{\mathbf{A}}_{\mathrm{LADMC};l}$ 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 $\widehat{\mathbf{A}}_{\mathrm{LADMC};0}$.

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 $\widehat{\mathbf{A}}_{\mathrm{LADMC};0}$. Also, the convergence rate of $\widehat{\mathbf{A}}_{\mathrm{LADMC};0}$ 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 $L(\mathbf{A}; (Y, \mathbf{X})) = |Y - \mathrm{tr}(\mathbf{X}^\top \mathbf{A})|$. To solve this stochastic optimization problem, the population version of the Newton-Raphson iteration takes the following form

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

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

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

$$\begin{aligned}
\mathrm{vec}(\mathbf{A}_1) &\approx \mathrm{vec}(\widehat{\mathbf{A}}_0) - [2f(0)\mathrm{diag}(\pi)]^{-1} \mathbb{E}_{(Y,\mathbf{X})}[\psi(\widehat{\mathbf{A}}_0; (Y, \mathbf{X}))] \\
&= \mathbb{E}_{(Y,\mathbf{X})} \mathrm{vec}(\widehat{\mathbf{A}}_0) - [f(0)]^{-1} \left\{ \mathbb{I}\{Y - \mathrm{tr}(\mathbf{X}^\top \widehat{\mathbf{A}}_0) > 0\} - \frac{1}{2} \right\} \mathbb{1}_{n_1 n_2} \\
&= [\mathrm{diag}(\pi)]^{-1} \mathbb{E}_{(Y,\mathbf{X})} \left[\mathrm{vec}(\mathbf{X}) \{ \mathrm{vec}(\mathbf{X})^\top \mathrm{vec}(\widehat{\mathbf{A}}_0) - [f(0)]^{-1} \left\{ \mathbb{I}\{Y - \mathrm{tr}(\mathbf{X}^\top \widehat{\mathbf{A}}_0) > 0\} - \frac{1}{2} \right\} \right] \\
&= [\mathrm{diag}(\pi)]^{-1} \mathbb{E}_{(Y,\mathbf{X})} [\mathrm{vec}(\mathbf{X}) Y^*] \quad (2.5) \\
&= [f\mathbb{E}_{(Y,\mathbf{X})}[\mathrm{vec}(\mathbf{X})\mathrm{vec}(\mathbf{X})^\top]\pi]^{-1} \mathbb{E}_{(Y,\mathbf{X})} [\mathrm{vec}(\mathbf{X}) Y^*]
\end{aligned}$$

where we de ne the theoretical pseudo data

$$\tilde{Y}_k^{(0)} = \mathrm{tr}(X_k^T \hat{A}^{(0)}) - [\hat{f}(0)]^{-1} \left[I\left(Y_k \le \mathrm{tr}(X_k^T \hat{A}^{(0)})\right) - \frac{1}{2} \right].$$

Here $1_{n_1 n_2}$ 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 $\arg\min_A E_{(Y,X)} \{ \tilde{Y}^{(0)} - \mathrm{tr}(X^T A) \}^2$, where $\mathrm{tr}(X^T A) = \mathrm{vec}(X)^T \mathrm{vec}(A)$. 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 $\hat{A}^{(0)}$ close enough to A^*. 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:

$$\hat{f}(0) = \frac{1}{Nh} \sum_{k=1}^{N} K\left(\frac{Y_k - \mathrm{tr}(X_k^T \hat{A}^{(0)})}{h} \right);$$

where $K(x)$ is a kernel function which satis es Condition (C4) and $h > 0$ is the bandwidth. For each $1 \le k \le N$, we de ne the actual pseudo data used in our proposed procedure to be

$$\tilde{Y}_k = \mathrm{tr}(X_k^T \hat{A}^{(0)}) - [\hat{f}(0)]^{-1} \left[I\left(Y_k \le \mathrm{tr}(X_k^T \hat{A}^{(0)})\right) - \frac{1}{2} \right];$$

and $\tilde{Y} = (\tilde{Y}_k)$. For nite sample, regularization is imposed to estimate the high-dimensional parameter A^*. By using $\tilde{Y}$, one natural candidate for the estimator of A^* is given by

$$\hat{A} = \arg\min_{A \in \mathcal{B}(a; n_1, n_2)} \frac{1}{N} \sum_{k=1}^{N} \left\{ \tilde{Y}_k - \mathrm{tr}(X_k^T A) \right\}^2 + \lambda_N \|A\|_*; \quad (2.6)$$

where $\|\cdot\|_*$ is the nuclear norm and $\lambda_N > 0$ is the tuning parameter. If $\tilde{Y}$ is replaced by 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 $\tilde{Y}$, corresponding computational algorithms can be adopted to solve (2.6). Note that the pseudo data are based on an initial estimator $\hat{A}^{(0)}$. In Section 3, we show that any initial estimator that ful lls Condition (C5) can be improved by (2.6), which is therefore called a re nement step. It is easy to verify that the initial estimator $\hat{A}_{\mathrm{LADMC};0}$ in Section 2.2 ful lls such condition. Note that the initial estimator, like $\hat{A}_{\mathrm{LADMC};0}$, introduces complicated dependencies among the entries of $\tilde{Y}$, which brings new challenges in analyzing (2.6), as opposed to the common estimator based on Y with independent entries.

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

$$\hat{f}^{(t)}(0) = \frac{1}{Nh_t} \sum_{k=1}^{N} K\left(\frac{Y_k - \mathrm{tr}(X_k^T \hat{A}^{(t-1)})}{h_t} \right);$$

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

$$\tilde{Y}_k^{(t)} = \mathrm{tr}(X_k^T \hat{A}^{(t-1)}) - [\hat{f}^{(t)}(0)]^{-1} \left[I\left(Y_k \le \mathrm{tr}(X_k^T \hat{A}^{(t-1)})\right) - \frac{1}{2} \right]. \quad (2.7)$$

We propose the following estimator

$$\hat{A}^{(t)} = \arg\min_{A \in \mathcal{B}(a; n_1, n_2)} \frac{1}{N} \sum_{k=1}^{N} \left\{ \tilde{Y}_k^{(t)} - \mathrm{tr}(X_k^T A) \right\}^2 + \lambda_{N,t} \|A\|_*; \quad (2.8)$$

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 $A = (A_{ij}) \in \mathbb{R}^{n_1 \times n_2}$, denote $\sigma_i(A)$ be the i-th largest singular value of matrix A. Let $\|A\|_* = \sum_{i=1}^{n_1 \wedge n_2} \sigma_i(A)$, $\|A\|_F = \sqrt{\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} A_{ij}^2}$ and $\|A\| = \sigma_1(A) = \max_{i=1}^{n_{\min}} \sigma_i(A)$ be the spectral norm (operator norm), the in nity norm, the Frobenius norm and the trace norm of a matrix A respectively. De ne a class of matrices $\mathcal{C}(n_1, n_2) = \{A \in \mathbb{R}^{n_1 \times n_2} : \|A\|_\infty \le 1\}$. Denote the rank of matrix A^* by $r^* = \mathrm{rank}(A^*)$ 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 X_k takes value in the canonical basis as de ned in (1.2). There exist positive constants $\underline{c}$ and $\bar{c}$ such that for any $(s,t) \in \{1, \ldots, n_1\} \times \{1, \ldots, n_2\}$, $\underline{c}/(n_1 n_2) \le \Pr(X_k = e_s(n_1) e_t^T(n_2)) \le \bar{c}/(n_1 n_2)$.