SCIENCE CHINA
Mathematics

· ARTICLES ·

doi:

Robust Reduced Rank Regression in a Distributed Setting

Xi Chen[1], Weidong Liu[2,] & Xiaojun Mao[3,]

[1]*Stern School of Business, New York University, New York, New York* 10012, *U.S.A.;*
[2]*School of Mathematical Sciences and MoE Key Lab of Artificial Intelligence, Shanghai Jiao Tong University, Shanghai,* 200240, *China;*
[3]*School of Data Science, Fudan University, Shanghai* 200433, *China*

Email: xc13@stern.nyu.edu, weidongl@sjtu.edu.cn, maoxj@fudan.edu.cn

Abstract This paper studies the reduced rank regression problem, which assumes a low-rank structure of the coe cient matrix, together with heavy-tailed noises. To address the heavy-tailed noise, we adopt the quantile loss function instead of commonly used squared loss. However, the non-smooth quantile loss brings new challenges to both computation and the development of statistical properties, especially when the data is large in size and distributed across di erent machines. To this end, we rst transform the response variable and reformulate the problem into a trace-norm regularized least-squares problem, which greatly facilitates the computation. Based on this formulation, we further develop a distributed algorithm. Theoretically, we establish the convergence rate of the obtained estimator and the theoretical guarantee for rank recovery. The simulation analysis is provided to demonstrate the e ectiveness of our method.

Keywords Reduced rank regression, Distributed estimation, Quantile loss, Rank recovery

MSC(2010) 62H12

Citation: Xi Chen, Weidong Liu, Xiaojun Mao. Robust Reduced Rank Regression in a Distributed Setting. Sci China Math, 2020, , doi:

1 Introduction

Multi-response regression model is one of the most fundamental models in multi-task learning (see e.g., (Zhang and Yang, 2017)). In particular, assume that there are$q$ response variables, which correspond to q learning tasks. Given$n$ independent and identical distributed observations of$q$ response variables and p predictors, denoted by$Y_i \in \mathbb{R}^q; X_i \in \mathbb{R}^{p+1}$ (with $X_{i1} = 1$ corresponding to the intercept) for $i = 1,\ldots,n$, we consider the following multi-response regression model:

$$Y = X \quad + \quad : \tag{1.1}$$

Here,$Y = (Y_1,\ldots,Y_n)^T \in \mathbb{R}^{n \quad q}$ is the response matrix,$X = (X_1,\ldots,X_n)^T \in \mathbb{R}^{n \quad (p+1)}$ is the covariate matrix, and $\in \mathbb{R}^{(p+1) \quad q}$ is the unknown coe cient matrix. Moreover, $= (_1,\ldots,_n) \in \mathbb{R}^{n \quad q}$ are n independent and identical distributed random noise vectors. When the number of predictors$p$ or the

* Co-corresponding authors

© Science China Press and Springer-Verlag Berlin Heidelberg 2019

number of responses q is large, the number of parameters to be estimated $p(p+1) \quad q$ will be too large for the estimation task. Therefore, the low rank condition on the coe cient matrix becomes a natural assumption, and the corresponding problem is referred to as \reduced rank regression" or "low-rank multi-task learning" (see, e.g., Chen et al. (2011, 2013); Reinsel and Velu (1998); She and Chen (2017)). In particular, we assume that the coe cient has the rank r, which is much smaller than p and q.

For reduced rank regression problem, trace-norm regularized least square loss has been widely used as an objective function to learn , i.e.,

$$\underset{S \in \mathbb{R}^{(p+1) \quad q}}{\arg\min} \; kY \quad XSk_F^2 + \quad kSk \; ; \tag{1.2}$$

where $k \quad k_F$ denotes the matrix Frobenius norm and $k \quad k$ the matrix trace norm. The obtained solution S is a natural estimator of . This method requires the noise to be sub-Gaussian. However, non sub-Gaussian, or heavy-tailed empirical data has been repeatedly observed in various domains, ranging from genomics, medical imaging to economics and nance. In particular, let $_i = (_{i1}; \ldots; _{iq})$ and we only assume the density function of each $_{ij}$ (for the j-th response) exists, which is denoted by $f_j(\ )$. Without loss of generality, we assume that $\Pr(_{ij} \leqslant 0) =$ for some pre-speci ed $2 (0; 1)$ for all $i; j$. For any pre-speci ed $2 (0; 1)$, we can always shift the quantile of the distribution of $f_j(\ )$ so that $\Pr(_{ij} \leqslant 0) =$. We do not impose any moment assumption on $_{ij}$, which allows for very heavy-tailed distributions such as Cauchy distribution. To address the new challenges arising from heavy-tailed data, we adopt the quantile loss function (Koenker, 2005) as a robust loss function and reformulate the objective function as follows,

$$\underset{S \in \mathbb{R}^{(p+1) \quad q}}{\arg\min} \quad (Y \quad XS) + \quad kSk \; ; \tag{1.3}$$

where

$$(u) = u(\quad \mathbb{I}(u < 0)) = \begin{cases} u; & \text{if } u \geqslant 0 \\ (\quad 1)u; & \text{if } u < 0; \end{cases}$$

is known as the quantile loss function at the level , and $\mathbb{I}(\ )$ is the indicator function. For any given matrix A, (A) applies the function to each element of A and take the sum.

The quantile loss e ectively addresses the issue of heavy-tailed noise. However, since it is a non-smooth loss function, it brings new challenges to computation, especially when the data size is large and distributed across di erent machines. Indeed, the development of modern technology has enabled data collection of unprecedented size. Therefore, modern large-scale data usually cannot be tted into memory or are collected in a distributed environment. For example, a personal computer usually has a limited memory size in GBs; while the data stored on a hard disk could have a size in TBs. In addition, sensor network data are naturally collected by many sensors.

The goal of the paper is to develop robust estimator for the low-rank reduced regression problem in distributed settings, and to provide theoretical guarantees on both estimation and rank recovery. In terms of rank recovery, it has been pointed out that the traditional penalty function kSk usually overestimate the rank (Mao et al., 2019; Mazumder et al., 2010). By noting that X is also a low rank matrix, an e ective approach is using the penalty function $kXSk$ instead of kSk in (1.2), as proposed by Chen et al. (2013). With the penalty function $kXSk$, the computational complexity is $O(n^2q + npq + p^2q)$, which is computationally heavy when n is large. To overcome the computational challenge as well as the heavy-tailed noise issue, we rst develop a new robust low-rank estimator with the pooled data in Section 3.1. This new estimator motivates our distributed robust low-rank estimator with an e cient communication scheme, which reduces the computational complexity from $O(n^2q + npq + p^2q)$ to $O(m^2q + mpq + p^2q)$, where $m \quad n$ is the local sample size on a local machine.

The non-smoothness of the quantile loss function in (1.3) results in another challenge in solving (1.3). By leveraging the idea from Newton-Raphson approach based on the population loss, we create a transformed response matrix Y. Based on the transformed response, we convert the original problem in (1.3) into a trace-norm regularized least-squared problem, which makes the computation much simpler. In par-

ticular, our new estimator has a closed-form solution via singular value soft-thresholding (see Section 3.1), which greatly facilitates the computation. In distributed settings, we develop a communication e cient distributed algorithm, which runs iteratively. At each iteration, it only communicates a $(p+1)$ q-dimensional gradient information at each iteration (instead of any Hessian information).

We further establish the theoretical properties for our distributed estimator. We rst provide the rate of convergence in Frobenius norm and rank consistency for our robust low-rank estimator in the single machine setting. In the distributed setup, we show that, after a constant number of iterations, our method achieves a near-oracle rate of $\sqrt{r(p+q)\log n/ng}$ (Theorem 4.4). This rate almost matches the oracle rate $\sqrt{r(p+q)/ng}$ (upto a logarithmic factor) for reduced rank regression. Furthermore, we provide the rank recovery result of the distributed estimator. In particular, we show that the estimated rank equals the true rank of with probability tending to one. To the best of our knowledge, this is the rst rank recovery result in low-rank *distributed estimation*.

1.1 Notations

For a vector $\mathrm{v} = (v_1, \ldots, v_n)^{\mathrm{T}}$, de ne $\|\mathrm{v}\|_2 = (\sum_{i=1}^{n} v_i^2)^{1/2}$. For a given matrix $\mathrm{A} = (a_{ij}) \in \mathbb{R}^{n \times q}$, let A_i be the i-th column of A. Denote $d_i(\mathrm{A})$ and $\lambda_i(\mathrm{A})$ to be the i-th largest singular value and eigenvalue of matrix A respectively. Let $\|\mathrm{A}\| = d_{\max}(\mathrm{A})$, $\|\mathrm{A}\|_F = (\sum_{i=1}^{n}\sum_{j=1}^{q} a_{ij}^2)^{1/2}$ and $\|\mathrm{A}\|_* = \sum_{i=1}^{n\wedge q} d_i(\mathrm{A})$ be the spectral norm (operator norm), the Frobenius norm and the trace norm of a matrix A respectively. We use $C_L, C, C_0, C_1, \ldots$ to denote constants whose value may change from place to place, which do not depend on n, p, q and m.

2 Related Work

Reduced rank regression has been an important topic in statistics and machine learning, which can date back to Izenman (1975); Reinsel and Velu (1998). It has been closely connected to many multivariate methods in statistics, such as principal component analysis, matrix denoising, matrix completion (Izenman, 2013). Among recent development of reduced rank regression, Chen et al. (2013) considered a two-stage adaptive trace-norm regularization with least squared loss, which facilitates to achieve rank consistency. We also adopt the idea of adaptive trace-norm regularization but with a robust loss. Chen et al. (2011) studied the reduced rank regression with sparse left and singular vectors. She and Chen (2017) considered the reduced rank regression with a sparse outlier matrix. In fact, She and Chen (2017) mentioned the adoption of a general robust loss function. However, then it pointed out that \but such an estimator may be di cult to compute. To the best our knowledge, even is Huber's loss function, there is no algorithm ...", and thus sought for a di erent formulation on handling the outlier. Instead, She and Chen (2017) proposed a bright equivalent optimization problem with quartic form loss function. However it is not easy to nd a suitable penalty function for the outlier to induce the quantile loss in original problem. The statistical properties of the estimator obtained by the same penalty function are also unclear. Our paper directly tackles the original formulation and develops computationally e cient robust estimators for distributed settings.

The reduced rank regression is closely connected to multi-task learning. Multi-task learning can be viewed as a slight variant/generalization of our formulation, which allows di erent design matrix for di erent tasks (see the survey in Zhang and Yang (2017)). Low-rank approach, which assumes the coe cient matrix lies on a low-rank subspace, has been widely studied in multi-task learning literature (see, e.g., Ando and Zhang (2005); Argyriou et al. (2007); Pong et al. (2010) and references therein). This paper extends the existing multi-task learning by introducing non-smooth quantile loss function for heavy-tailed noises, developing an e cient distributed algorithm, and providing rank consistency theoretical results.

For modern large-scale data, traditional statistical methods, which load all the data into memory and run a certain optimization procedure, are no longer applicable due to both storage and computation issues. Therefore, distributed estimation has received a lot of attention in recent years. The averaging

divide-and-conquer (Avg-DC) approach is perhaps one of the most popular methods and has been applied to a wide range of statistical problems (see, e.g., Banerjee et al. (2019); Battey et al. (2018); Fan et al. (2019); Lee et al. (2017); Li et al. (2013); Lian and Fan (2017); Shi et al. (2018); Volgushev et al. (2019); Zhang et al. (2015); Zhao et al. (2016)). In particular, the Avg-DC method computes a local estimator on each machine only using the m local samples. Then, the final estimator is obtained by averaging all the local estimators. On one hand, the Avg-DC approach enjoys low communication cost since it only requires one-shot communication (i.e., taking the average of local estimators). On the other hand, the Avg-DC method also has several weaknesses. First, it always requires a constraint on the number of machines (i.e., the number of machines cannot be too large). The main reason is that the averaging only reduces the variance but not the bias of each local estimator. Second, for latent structured models (e.g., sparse or low-rank models), the averaging will ruin the learned structure of local estimators and the averaged estimator will no longer be sparse nor low-rank. To address these weaknesses, iterative algorithms have been developed and our approach belongs to this class. For example, Chen et al. (2019) and Chen et al. (2020) studied distributed quantile regression for low dimensional models and high-dimensional sparse models with the ℓ_1-regularization, respectively. We note that Chen et al. (2020) only investigated the vector coefficient with the l_1-regularization; and our paper studies the matrix coefficient with a low-rank structure, which is technically more challenging. Jordan et al. (2019); Shamir et al. (2014) proposed the local Newton approach, which only utilizes Hessian matrix on one local machine for efficient communication.

3 Proposed Method

3.1 Pooled Robust Low-Rank Estimator

Recall that Θ is the coefficient matrix of interest, X denotes random covariate matrix and Y denotes the response matrix. To simplify the notations, we consider the optimization problem for each column of Θ separately at first. Inspired by the Newton-Raphson method, for each $j = 1, \ldots, q$, we consider the following stochastic optimization problem,

$$\theta_j = \arg\min_{\theta_j \in \mathbb{R}^{p+1}} E[G(\theta_j; X, Y_j)], \tag{3.1}$$

where $G(\theta_j; X, Y_j) = \rho_\tau(Y_j - X\theta_j)$ is the quantile loss function, Y_j denote the j-th column of responses and θ_j is the j-th column of the coefficient vector of interest. In (3.1), we assume that $E[G(\theta_j; X, Y_j)]$ exists only for the motivation of our new estimator. To solve this stochastic optimization problem, the population version of the Newton-Raphson iteration takes the following form

$$\theta_{1,j} = \theta_{0,j} - H(\theta_{0,j})^{-1} E[g(\theta_{0,j}; X, Y_j)], \tag{3.2}$$

where $\theta_{0,j}$ is an initial estimator, $g(\theta_j; X, Y_j) = X^{\mathrm{T}}(\mathbb{I}(Y_j - X\theta_j \leqslant 0) - \tau)$ is the subgradient of the loss function $G(\theta_j; X, Y_j)$ with respect to θ_j, and

$$H(\theta_j) = \partial E[g(\theta_j; X, Y_j)]/\partial \theta_j = E(X^{\mathrm{T}} X f_j(X(\theta_j - \theta_j^*))),$$

denotes the population Hessian matrix of $\mathbb{E}G(\theta_j; X, Y_j)$. Here, $f_j(x)$ is the density function of each noise ε_{ij}. As for the choice of the initial estimator $\theta_{0,j}$, we only require that $\theta_{0,j}$ satisfies Condition (C4) in our theory section (see Section 4). It can be shown that many low-rank estimators (e.g., the ones in Chao et al. (2016); She and Chen (2017)) satisfy this condition.

When the initial estimator $\theta_{0,j}$ is close to the true parameter θ_j^*, by using $H(\theta_j^*) = \Sigma_X f_j(0)$ in (3.2) where $\Sigma_X = E(X^{\mathrm{T}} X)$, it motivates the following iteration,

$$\theta_{1,j} = \theta_{0,j} - \Sigma_X^{-1} f_j^{-1}(0) E[g(\theta_{0,j}; X, Y_j)] \tag{3.3}$$

$$= \Sigma_X^{-1}\left(\Sigma_X b_{0;j} - f_j^{-1}(0)E[g(b_{0;j}; X; Y_j)]\right)$$
$$= \Sigma_X^{-1}E\left[X^T\left(X b_{0;j} - f_j^{-1}(0)(\mathbb{I}(Y_j \leqslant X b_{0;j}) - \tau)\right)\right]:$$

In (3.3), $f_j(0)$ can be easily estimated by the kernel density estimator:

$$\hat{f}_j(0) = \frac{1}{nh_j}\sum_{i=1}^{n} K_j\left(\frac{Y_{ij} - X_i^T b_{0;j}}{h_j}\right);$$

where $K_j(x)$ is a smoothing function which satisfies Condition (C3) and $h_j \to 0$ is the bandwidth for each $j = 1; \ldots; q$.

In (3.3), if we define a new column of response variable $\hat{Y}_j$ as

$$\hat{Y}_j = X b_{0;j} - \hat{f}_j^{-1}(0)\left(\mathbb{I}(Y_j \leqslant X b_{0;j}) - \tau\right); \tag{3.4}$$

then $\beta_{1;j} = \Sigma_X^{-1}E(X^T\hat{Y}_j)$. It is easy to see that $\beta_{1;j}$ is the best linear regression coefficient of $\hat{Y}_j$ on X, i.e., $\beta_{1;j} = \arg\min_{\beta_j \in \mathbb{R}^{p+1}} E(\hat{Y}_j - X\beta_j)^2$. Combining all the column of responses and parameters together, we have $\beta_1 = \arg\min_{\beta \in \mathbb{R}^{(p+1)\times q}} \sum_{j=1}^{q} E(\hat{Y}_j - X\beta_j)^2$. To further encourage the low-rankness of the estimator, it is natural to consider the following trace norm regularized problem,

$$\beta_{1;} = \arg\min_{\beta \in \mathbb{R}^{(p+1)\times q}} \frac{1}{2}\sum_{j=1}^{q} E(\hat{Y}_j - X\beta_j)^2 + \lambda\|\beta\|_*; \tag{3.5}$$

where $\beta_{1;}$ is low rank and can achieve a faster convergence rate than β_0.

Given a fixed $1 \leqslant j \leqslant q$, for each $1 \leqslant i \leqslant n$, define

$$\hat{Y}_{ij} = X_i^T b_{0;j} - \hat{f}_j^{-1}(0)\left(\mathbb{I}(Y_{ij} \leqslant X_i^T b_{0;j}) - \tau\right); \tag{3.6}$$

and $\hat{Y}_j = (\hat{Y}_{1j}; \ldots; \hat{Y}_{nj})^T$. One natural candidate for the estimator of the coefficient matrix could be given by

$$b = \arg\min_{\beta \in \mathbb{R}^{(p+1)\times q}} \left\{\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{q}\left(\hat{Y}_{ij} - X_i^T\beta_j\right)^2 + \lambda\|\beta\|_*\right\}. \tag{3.7}$$

It is well known that the trace norm penalty encourages low-rank solution (Bunea et al., 2011; Candès and Recht, 2009; Lu et al., 2012; Yuan et al., 2007). The reason of considering the trace norm instead of the rank norm penalty directly is that the minimization problem with rank norm penalty is NP-hard (Candès and Recht, 2009). As suggested by the Lemma 1 in Yuan et al. (2007), the solution of minimization problem (3.7) only has closed-form under the orthogonal design where $X^TX = n\mathrm{I}$, and generally does not have an explicit form. Instead of adopting an iterative algorithm for solving (3.7) that can be computationally expensive, we follow Chen et al. (2013) to adopt a modified low-rank penalty, which allows a closed-form solution with easy computation. In particular, we proposed the following pooled robust low-rank estimator:

$$b_{\text{pool}} = \arg\min_{\beta \in \mathbb{R}^{(p+1)\times q}} \left\{\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{q}\left(\hat{Y}_{ij} - X_i^T\beta_j\right)^2 + \lambda_{n;q}\|X\beta\|_*\right\}. \tag{3.8}$$

It has been shown in Cai et al. (2010); Mazumder et al. (2010) that there exists a closed-form solution of the minimization problem (3.8). Let UDV^T be the singular value decomposition (SVD) of a matrix M where $D = \mathrm{diag}(\{d_i\})$. Define the corresponding singular value soft-thresholding (SVT) operator T_c

by

$$T_c(\mathrm{M}) = \mathrm{U}\,\mathrm{diag}(f(d_i - c)_+ g)\mathrm{V}^{\mathrm{T}} \quad \text{for any } c \geqslant 0; \tag{3.9}$$

where $x_+ = \max(x; 0)$. Thus b_{pool} can be expressed by $\mathrm{b}_{\text{pool}} = n^{-1}\mathrm{X}^{\mathrm{T}} T_{n;q}(\mathrm{Y})$ where $\mathrm{b} = n^{-1}\sum_{i=1}^{n} \mathrm{X}_i \mathrm{X}_i^{\mathrm{T}}$ is the sample covariance matrix. It is noticed that the low-rankness of $T_{n;q}(\mathrm{Y})$ can lead to the low-rankness of b_{pool}.

However, it has been shown in many simulation studies that such relaxation in (3.8) usually leads to a larger estimated rank than true rank (Mao et al., 2019; Mazumder et al., 2010). To correct the estimated rank, we consider the adaptive trace norm $k \cdot k_{;\mathbf{w}}$. The adaptive trace norm of a matrix $\mathrm{A} \in \mathbb{R}^{n \times q}$ is defined as a weighted sum of its singular values: $k\mathrm{A}k_{;\mathbf{w}} = \sum_{i=1}^{n\wedge q} w_i d_i(\mathrm{A})$, where $\mathrm{w} = (w_1; \ldots; w_{n\wedge q})^{\mathrm{T}}$ is a weight coefficients vector and $d_i(\mathrm{A})$ is the i-th largest singular value of matrix A. Let $\mathrm{w}^o = (w_1^o; \ldots; w_{n\wedge q}^o)^{\mathrm{T}}$ such that $w_i^o = 1/d_i(\mathrm{A})$ if $d_i(\mathrm{A}) > 0$ and $w_i^o = 1$ if $d_i(\mathrm{A}) = 0$. We can see that $k\mathrm{A}k_{;\mathbf{w}^o} = r_{\mathbf{A}}$. Here the constant 1 can be any nonzero number, it is only used to avoid the case $w_i^o = 1/0$. Therefore, $k\mathrm{A}k_{;\mathbf{w}^o}$ is a natural candidate of matrix rank $r_{\mathbf{A}}$. We usually have no knowledge about the "oracle" weight $w_i^o = 1/d_i(\mathrm{A})$, but a good estimation would lead to a good approximation for matrix rank $r_{\mathbf{A}}$.

Denote the corresponding ith singular values of Y to be $d_i(\mathrm{Y})$. Following the idea of adaptive Lasso in Zou (2006), the "oracle" weight w_i^o can be approximated by $\mathrm{w}_i = d_i^{-\gamma}(\mathrm{Y})$ if $d_i(\mathrm{Y}) > 0$ and $\mathrm{w}_i = 1$ if $d_i(\mathrm{Y}) = 0$, where $\gamma \geqslant 0$ is a pre-specified constant. The adaptive weight γ can be chosen by cross validation or simply set as a constant (e.g., within the range of from 0.5 to 3) (Zou, 2006). Please see our numerical studies in Section 5.4 for more details. Thus we propose the adaptive robust estimator:

$$\mathrm{b}_{\text{adp-pool}} = \arg\min_{\mathbf{S} \in \mathbb{R}^{(p+1)\times q}} \left\{ \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{q} \mathrm{Y}_{ij} - \mathrm{X}_i^{\mathrm{T}}\mathrm{S}_j \Big|^2 + \gamma_{n;q}\, k\mathrm{X}\mathrm{S}k_{;\mathbf{w}}, \right\}; \tag{3.10}$$

where $\mathrm{w} = (\mathrm{w}_1; \ldots; \mathrm{w}_{n\wedge q}) \in \mathbb{R}^{n\wedge q}$ and each w_i is defined as before.

To express the solution of the minimization problem (3.10), we modified the SVT operator defined in (3.9) to be the adaptive singular value soft-thresholding (ASVT) operator $T_{\mathbf{w}}$ (for a vector w instead of a single non-negative number c) by

$$T_{\mathbf{w}}(\mathrm{M}) = \mathrm{U}\,\mathrm{diag}(f(d_i - w_i)_+ g)\mathrm{V}^{\mathrm{T}} \quad \text{for any } w_i \geqslant 0; \tag{3.11}$$

where $\mathrm{UDV}^{\mathrm{T}}$ is the SVD of a matrix M and $x_+ = \max(x; 0)$. Thus $\mathrm{b}_{\text{adp-pool}}$ can be expressed by $\mathrm{b}_{\text{adp-pool}} = n^{-1}\mathrm{b}^{-1}\mathrm{X}^{\mathrm{T}} T_{n;q\,\mathrm{w}}(\mathrm{Y})$.

3.2 Distributed Robust Low-Rank Estimator

Suppose that n samples are stored in $L = n/m$ machines and each local machine has m samples. In other words, we randomly split the data index set $\{1; 2; \ldots; n\}$ into $H_1; \ldots; H_L$ with $|H_k| = m$ and the k-th machine stores samples $\{(\mathrm{X}_i; \mathrm{Y}_i) : i \in H_k\}$. Let us define

$$\mathrm{b}_k = \frac{1}{m}\sum_{i \in H_k} \mathrm{X}_i \mathrm{X}_i^{\mathrm{T}}; \quad \text{and} \quad \mathrm{b} = \frac{1}{n}\sum_{i=1}^{n} \mathrm{X}_i \mathrm{X}_i^{\mathrm{T}} = \frac{1}{L}\sum_{k=1}^{L} \mathrm{b}_k; \tag{3.12}$$

as the sample covariance matrix on the k-th machine and the sample covariance matrix of the entire dataset, respectively. It is worthwhile noting that our algorithm does not need to explicitly compute and communicate b_k (for $k \neq 1$) (see Algorithm 1 for more details).

For each response variable $j = 1; \ldots; q$, we define the following $(p+1)$-dimensional vector for each machine k

$$\mathrm{z}_{n;kj} = \frac{1}{m}\sum_{i \in H_k} \mathrm{X}_i \mathrm{Y}_{ij};$$

and the averaged quantities across machines:

$$z_{n;j} := \frac{1}{L}\sum_{k=1}^{L} z_{n;kj} \quad \text{and} \quad Z_n = (z_{n;1};\cdots;z_{n;q}) \in \mathbb{R}^{(p+1)\times q}.$$

By leveraging the idea from Chen et al. (2020) and encouraging the low-rankness of the estimator, we develop the following robust low-rank estimator in distributed settings:

$$\begin{aligned}
b^{(1)} &= \arg\min_{S\in\mathbb{R}^{(p+1)\times q}} \Big\{ \frac{1}{q}\sum_{j=1}^{q}\Big[\frac{1}{2m}\sum_{i\in H_1} x_i^{\top}s_j\Big]^2 - \frac{1}{q}\sum_{j=1}^{q} s_j^{\top}\big(z_{n;j} + b_1 - b_{0;j}\big) + \lambda_{n;q;1}\|X_{H_1}S\|_{*;w^{(1)}}\Big\} \\
&= \arg\min_{S\in\mathbb{R}^{(p+1)\times q}} \Big\{ \frac{1}{mq}\sum_{j=1}^{q}\Big[\frac{1}{2}s_j^{\top}X_{H_1}^{\top}X_{H_1}s_j - ms_j^{\top}\big(z_{n;j} + b_1 - b_{0;j}\big)\Big] + \lambda_{n;q;1}\|X_{H_1}S\|_{*;w^{(1)}}\Big\} \\
&= \arg\min_{S\in\mathbb{R}^{(p+1)\times q}} \Big\{ \frac{1}{2}\|X_{H_1}S - X_{H_1}b_1^{-1}\big(z_n + b_1 - b_0\big)\|_F^2 + \lambda_{n;q;1}\|X_{H_1}S\|_{*;w^{(1)}}\Big\};
\end{aligned} \tag{3.13}$$

where $X_{H_1} = (X_1;\cdots;X_m)^{\top} \in \mathbb{R}^{m\times(p+1)}$ is the covariate matrix stored in the first machine H_1, $M_1 = X_{H_1}b_1^{-1}[z_n + (b_1 - b)b_0]$, and the weight $w^{(1)}$ is chosen by $w_i^{(1)} = d_i(M_1)$ if $d_i(M_1) > 0$ and $w_i^{(1)} = 1$ if $d_i(M_1) = 0$ for $\geqslant 0$ is a pre-specified constant.

This estimator has two advantages. First, it only utilizes the Hessian information on the first machine, and thus does not require communicating any Hessian information of the size $(p+1)q)^2$. Second, due to the squared loss, it also admits a simple closed-form solution. In particular, applying the ASVT operator, we have

$$b^{(1)} = m^{-1}b_1^{-1}X_{H_1}^{\top}T_{\lambda_{n;q;1}w^{(1)}}(M_1).$$

Note that when $m = n$, we have $b^{(1)} = b_{\text{adp-pool}}$. Namely, when the data is pooled on a single machine, the proposed distributed robust estimator reduces to the pooled estimator (3.10) automatically. Given the choice of the weights $w^{(1)}$ with $w_i^{(1)} = d_i(M_1)$ if $d_i(M_1) > 0$ and $w_i^{(1)} = 1$ if $d_i(M_1) = 0$, the estimated rank of $b^{(1)}$ can be defined as

$$\hat{b}_1 = \max\Big\{ s : d_s(M_1) > \lambda_{n;q;1}^{1/(\alpha+1)} \Big\}; \tag{3.14}$$

for $0 \leqslant \lambda_{n;q;1} < d_{\hat{b}_1+1}(M_1)$.

We can further extend the estimator in (3.13) to an iteratively refined estimator. Given (3.13) as the estimator from the first iteration, it is easy to construct an iterative algorithm. In particular, let $b^{(t-1)}$ be the distributed estimator in the $(t-1)$-th iteration. Define

$$\hat{f}_j^{(t)}(0) = \frac{1}{nh_{j;t}}\sum_{i=1}^{n} K_j\Big(\frac{Y_{ij} - x_i^{\top}b_j^{(t-1)}}{h_{j;t}}\Big);$$

where $K_j(x)$ is a smoothing function which satisfies condition and $h_{j;t} \to 0$ is the bandwidth for each $j = 1;\cdots;q$ for the t-th iteration (see Section 4 for more details).

Given a fixed $1 \leqslant j \leqslant q$, for each $1 \leqslant i \leqslant n$, define

$$\tilde{Y}_{ij}^{(t)} = x_i^{\top}b_j^{(t-1)} - \hat{f}_j^{(t)}(0)^{-1}\,\mathbb{I}\big\{Y_{ij} \leqslant x_i^{\top}b_j^{(t-1)}\big\};$$

$$z_{n;kj}^{(t)} = \frac{1}{m}\sum_{i\in H_k} x_i\tilde{Y}_{ij}^{(t)}; \quad z_{n;j}^{(t)} := \frac{1}{L}\sum_{k=1}^{L} z_{n;kj}^{(t)}; \quad \text{and} \quad Z_n^{(t)} = (z_{n;1}^{(t)};\cdots;z_{n;q}^{(t)}). \tag{3.15}$$

We proposed the following estimator

$$b^{(t)} = \arg\min_{S\in\mathbb{R}^{(p+1)\times q}} \Big\{ \frac{1}{2}\|X_{H_1}S - X_{H_1}b_1^{-1}\big(z_n^{(t)} + b_1 - b^{(t-1)}\big)\|_F^2 + \lambda_{n;q;t}\|X_{H_1}S\|_{*;w^{(t)}}\Big\}. \tag{3.16}$$

Algorithm 1 Distributed Robust Low-Rank Estimator

Input: Data on local machines $\{X_i; Y_i : i \in H_k\}$ for $k = 1; \ldots; L$, quantile level , pre-speci ed constant $\geqslant 0$, a sequence of kernel functions K_j for $j = 1; \ldots; q$, a sequence of bandwidths $h_{j;g}$ for $j = 1; \ldots; q$, $g = 1; \ldots; t$ and the regularization parameters $_{n;q;g}$ for $g = 1; \ldots; t$.

Set $\mathbf{b}^{(0)}$ to be the same as the initial estimator $\mathbf{b}_0$.
for $g = 1; 2; \ldots; t$ do
  Transmit $\mathbf{b}^{(g-1)}$ to all local machines.
  for $k = 1; \ldots; L$ do
    The k-th machine computes $\hat{f}_j^{(g;k)}(0) := (mh_{j;g})^{-1} \sum_{i \in H_k} K_j(h_{j;g}^{-1}(Y_{ij} - X_i^T \mathbf{b}_j^{(g-1)}))$ for each $j = 1; \ldots; q$ and sends it back to the rst machine.
  end for
  The rst machine computes $\hat{f}_j^{(g)}(0)$ for each $j = 1; \ldots; q$ based on $\hat{f}_j^{(g)}(0) = L^{-1} \sum_{k=1}^{L} \hat{f}_j^{(g;k)}(0)$.
  Transmit $\hat{f}_j^{(g)}(0)$ to all local machines for each $j = 1; \ldots; q$.
  for $k = 1; \ldots; L$ do
    The k-th machine computes $\mathbf{b}_k \mathbf{b}^{(g-1)}$ and $z_{n;kj}^{(g)} = m^{-1} \sum_{i \in H_k} X_i \hat{Y}_{ij}^{(g)}$ based on (3.15) for each $j = 1; \ldots; q$ and sends them back to the rst machine.
  end for
  Compute the estimator $M_g = X_{H_1} \mathbf{b}_1^{-1} \{Z_n^{(g)} + (\mathbf{b}_1 - \mathbf{b})\mathbf{b}^{(g-1)}\}$, $\mathbf{w}^{(g)}$ such that $\mathbf{w}_i^{(g)} = d_i(M_g)$ if $d_i(M_g) > 0$ and $\mathbf{w}_i^{(g)} = 1$ if $d_i(M_g) = 0$ and the estimator $\mathbf{b}^{(g)} = m^{-1} \mathbf{b}_1 X_{H_1}^T T_{n;q;g} \mathbf{w}^{(g)}(M_g)$ on the rst machine.
end for

Output: The nal estimator $\mathbf{b}^{(t)}$.

Let $M_t = X_{H_1} \mathbf{b}_1^{-1} \{Z_n^{(t)} + (\mathbf{b}_1 - \mathbf{b})\mathbf{b}^{(t-1)}\}$ and further modify the adaptive estimator by choosing the weights to be $\mathbf{w}_i^{(t)} = d_i(M_t)$ if $d_i(M_t) > 0$ and $\mathbf{w}_i^{(t)} = 1$ if $d_i(M_t) = 0$. Given the choice of the weights $\mathbf{w}^{(t)}$ with $\mathbf{w}_i^{(t)} = d_i(M_t)$ if $d_i(M_t) > 0$ and $\mathbf{w}_i^{(t)} = 1$ if $d_i(M_t) = 0$, the estimated rank of $\mathbf{b}^{(t)}$ is

$$\mathbf{b}_t = \max\left\{ s : d_s(M_t) > \frac{1}{n;q;t}^{(\ +1)} \right\}; \tag{3.17}$$

for $0 \leqslant {}_{n;q;t} < d_1^{+1}(M_t)$. It is worthwhile noting that by comparing with the estimator proposed in Chen et al. (2013), the computational time complexity of $\mathbf{b}^{(t)}$ reduced a lot from $O(n^2q + npq + p^2q)$ to $O(t(m^2q + mpq + p^2q))$, especially when the number of iterations t is small (see (4.5) for a lower bound on t).

As for the input initial estimator $\mathbf{b}_0$ in algorithm 1, similarly, we only require that $\mathbf{b}_0$ satis es the condition (C4) (see Section 4).

4 Theoretical Properties

We rst assume the following regularity conditions and discuss the theoretical results in the distributed setting.

(C1) For each $j = 1; \ldots; q$, the density function $f_j(\ )$ is Lipschitz continuous (i.e., $|f_j(x) - f_j(y)| \leqslant C_L |x - y|$ for any $x; y \in \mathbb{R}$ and some constant $C_L > 0$). Moreover, we assume $f_j(0) \geqslant c > 0$ for some constant c and $\mathbb{P}(\ _{ij} \leqslant 0) = \ $.

(C2) There exists constants $c_1 > 0$ and $c_2 > 0$ such that $c_1 \leqslant \min(\ \mathbf{x}) \leqslant \max(\ \mathbf{x}) \leqslant c_2$.

(C3) For each $j = 1; \ldots; q$, assume the kernel functions $K_j(\ )$ is integrable with $\int_{-1}^{1} K_j(u)du = 1$. Moreover, assume that $K_j(\ )$ satis es $K_j(u) = 0$ if $|u| \geqslant 1$. Further, assume that $K_j(\ )$ is di erentiable and its derivative $K_j^0(\ )$ is bounded.

(C4) The initial estimator $\mathbf{b}_0$ satis es $k\mathbf{b}_0 - k_F = O_{\mathbb{P}}(a_n)$, where the initial rate $a_n = o(1)$ and $p \log n = o(na_n)$.

(C5) We assume that X_i satis es the sub-Gaussian condition $\sup_{j_2=1} E \exp(t(\ ^T X_i)^2) \leqslant C_1$ for some constants $t > 0$ and C_1.

(C6) The local sample size m on each machine satisfies $m \geqslant n^c$ for some $0 < c < 1$. The number of predictor p fulfills that $\sqrt{r}(p + \log m)/m\sqrt{n} = o(1)$. The number of response variables q satisfies $q \leqslant C_2 a_n^{-2}$ for some constant C_2.

Condition (C1) assumes the smoothness of the noise density functions $f_j(\cdot)$. We highlight that we do not make any moment assumption on noise and thus allow very heavy-tailed noises. The eigenvalue condition in Condition (C2) is commonly adopted for the covariance matrix. Condition (C3) presents the assumptions of the kernel functions $K_j(\cdot)$ for each $j = 1, \ldots, q$. One can easily choose a kernel function that satisfies Condition (C3) (e.g., see Section 5). Condition (C4) is a condition on the convergence rate of the initial estimator $\widehat{\Theta}_0$ provided in Algorithm 1. Condition (C5) contains a standard sub-Gaussian assumption on random covariate X_i. We also note that the sub-Gaussian assumption can be extended to some polynomial moment condition with similar proof technique. The difference is not fundamental. We adopt the sub-Gaussian assumption in favor of the presentation of theoretical results. Condition (C6) includes some mild conditions on local sample size m, the number of predictors p, and the number of response variables q. It is noted that we only require the number of predictor p to be smaller order of m/r, which is quite mild when the coefficient matrix Θ is exactly low-rank with $r = O(1)$. As we always have $r \leqslant p + 1$, in the worst case is, the number of predictor p is the smaller order of $m^{1/2}$, which is still easy to be satisfied in many applications as the local sample size m is growing. The constraint on the number of response variables q is also mild as we can easily obtain an initial estimator with the initial rate $a_n = o(1)$, which allows q to diverge.

In the distributed setting, we first investigate the performance for one iteration/refinement (see Theorems 4.1 and 4.2) and then provide the results for multiple iterations (see Theorems 4.3 and 4.4).

Theorem 4.1. *Assume that conditions (C1)–(C6) hold. We choose the bandwidths $h_j \asymp a_n$ and take*

$$\lambda_{n,q,1} = C_0 \left\{ m^{1/2} \left(\frac{(p+q)\log n}{n} \right)^{1/2} + a_n \left(\frac{p + \log m}{m} \right)^{1/2} + a_n^2 \right\}^{\nu+1}, \tag{4.1}$$

with a sufficient large C_0. Suppose that $d_r(\Theta) > C_1 m^{1/2} \lambda_{n,q,1}^{1/(\nu+1)}$ for a sufficient large constant C_1. Then we have

$$Pr(\widehat{r}_1 = r) \to 1,$$

as $m, n \to \infty$ where $\widehat{r}_1$ is the estimated rank given in (3.14).

It is worthwhile noting that the condition $d_r(\Theta) > C_1 m^{1/2} \lambda_{n,q,1}^{1/(\nu+1)}$ concerns about the signal strength relative to the noise level. Theorem 4.1 shows that the proposed estimator $\widehat{r}^{(1)}$ is able to identify the correct rank of Θ with probability tending to 1 as $m, n \to \infty$.

Theorem 4.2. *Assume that conditions (C1)–(C6) hold and choose the bandwidths $h_j \asymp a_n$. Further suppose that $d_r(\Theta) > C_1 m^{1/2} \lambda_{n,q,1}^{1/(\nu+1)}$ for a sufficient large C_1. By taking $\lambda_{n,q,1}$ as given in (4.1), we have*

$$\|\widehat{\Theta}^{(1)} - \Theta\|_F = O_{\mathbb{P}}\left(\left(\frac{r(p+q)\log n}{n} \right)^{1/2} + a_n \left(\frac{r(p+\log m)}{m} \right)^{1/2} + a_n^2 r^{1/2} \right). \tag{4.2}$$

Note that the first term $\sqrt{r(p+q)\log n/mn}$ in the right hand of equation (4.2) matches the near-oracle rate obtained by Chao et al. (2016) under the classical standalone machine setting with all samples stored in a single machine. It is easy to see that this rate improves a lot from a_n because $a_n r^{1/2} = o(1)$. Thus our one round estimator $\widehat{\Theta}^{(1)}$ narrows the gap between the near-oracle rate and initial rate a_n.

Theorems 4.3 and 4.4 below show that the multi-iteration estimator $\widehat{\Theta}^{(t)}$ for t-th iterations is also able to identify the correct rank of Θ and iteratively refine the convergence rate. Let us define $a_{n,0} = a_n$, which is the initial rate, and

$$a_{n,t} = \left(\frac{r(p+q)\log n}{n} \right)^{1/2} + a_{n,0} \left(\frac{r(p+\log m)}{m} \right)^{t/2} + r^{1/2}(a_{n,0} r^{1/2})^{2^t} \quad t \geqslant 1. \tag{4.3}$$

As we will prove in Theorem 4.4 below, $a_{n,t}$ is the convergence rate of the obtained distributed estimator $\hat{b}^{(t)}$ for the t-th iteration.

Theorem 4.3. *Assume that conditions (C1){(C6) hold. For any iteration $t \geqslant 1$, we choose the bandwidths $h_{j,t} \asymp a_{n,t-1}$ and take*

$$\lambda_{n,q,t} = C_0 m^{1/2} \left(\frac{(p+q)\log n}{n} + a_{n,t-1} \frac{p+\log m}{m} + a_{n,t-1}^2 \right)^{\phi+1}; \tag{4.4}$$

with a sufficient large C_0. Suppose that $d_r(\Theta) > C_1 m^{1/2} \lambda_{n,q,t}^{1/(\phi+1)}$ for a sufficient large constant C_1. Then we have

$$Pr(\hat{b}_t = r) \to 1,$$

as $m, n \to \infty$ where $\hat{b}_t$ is the estimated rank given in (3.17).

By a recursive application of Theorem 4.2, we provide the convergence rate for the multi-iteration estimator $\hat{b}^{(t)}$ in Theorem 4.4 for $t \geqslant 1$ iterations. Theorem 4.4 shows that an iterative refinement of the initial estimator will reduce the mean squared error of the our distributed estimator and achieve a near-oracle rate $r(p+q)\log n/ng^{1/2}$ after a certain number of iterations.

Theorem 4.4. *Assume that conditions (C1){(C6) hold and choose the bandwidths $h_{j,t} \asymp a_{n,t-1}$. Further suppose that $d_r(\Theta) > C_1 m^{1/2} \lambda_{n,q,t}^{1/(\phi+1)}$ for a sufficient large constant C_1. By taking $\lambda_{n,q,t}$ as given in* (4.4)*, we have*

$$\|\hat{b}^{(t)}\Theta - \Theta\|_F = O_{\mathbb{P}}(a_{n,t}),$$

where $a_{n,t}$ is defined in (4.3).

It can be shown that when the iteration number t is relatively large,

$$t \geqslant \frac{\log(r(p+q)\log n) - \log n - 2\log a_{n,0}}{c_0 \log(r(p+\log m)) - c_0 \log m}, \quad \text{for some } c_0 > 0; \tag{4.5}$$

the second and third term in (4.3) is dominated by the first term, and the convergence rate of $\hat{b}^{(t)}$ becomes $r(p+q)\log n/ng^{1/2}$ which is a near-oracle rate of $r(p+q)/ng^{1/2}$ upto a logarithmic factor. Note that under the local sample size constrain in condition (C5), we guarantee that the number of iterations t is of constant order. In practice, we provide a small tolerance parameter and stop the algorithm once the changes of estimation in each step is smaller than the tolerance parameter (see Section 5).

5 Simulation Studies

In our simulation studies, we consider the linear model

$$Y_{ij} = (X\Theta)_{ij} + \varepsilon_{ij}, \qquad i = 1, 2, \ldots, n \quad \text{and} \quad j = 1, 2, \ldots, q,$$

where each $X_i = (1, X_{i,1}, \ldots, X_{i,p})$ is a $(p+1)$-dimensional covariate vector and $(X_{i,1}, \ldots, X_{i,p})$s are drawn independent and identical distributed from a multivariate normal distribution $N(0, \Sigma_X)$. The covariance matrix Σ_X is constructed by $\Sigma_{ij} = 0.5^{|i-j|}$ for $1 \leqslant i, j \leqslant p$ to enforce correlation structure among covariates. We fix the number of predictors $p = 100$ and the number of response variables $q = 100$ and choose the loss function to be the QR loss with quantile level $\tau = 0.3$. Note that other choices of τ lead to similar results in the experiment. The low-rank coefficient matrix Θ is generated by the product of two random matrices $U \in \mathbb{R}^{(p+1)\times r}$ and $V \in \mathbb{R}^{r\times q}$. Thus the true coefficient $\Theta = UV^{\mathrm{T}}$ is a low-rank matrix of rank r. The entries of matrices U and V are all generated from the uniform distribution Unif$(0, 1)$ independently where r is fixed to be $2, 5$, or 10. For each experiment, we repeat 500 independent

simulations and report averaged results. The standard deviations for these 500 repetitions are relatively small and thus omitted for better illustration. We consider the following three noise distributions:

(a) Normal: the noise $_{ij}$ N (0; 1).

(b) Cauchy: the noise $_{ij}$ Cauchy(0; 1).

(c) Exponential: the noise $_{ij}$ exp(1).

We note that Cauchy distribution is a very heavy-tailed distribution and its rst moment does not exist.

Denote the proposed distributed robust low-rank estimator given in Algorithm 1 by Dist RLRE (Distributed Robust Low-Rank Estimator). For the initial estimator b_0, we use the classical trace-norm regularized estimator but with the data only on the rst machine, i.e.,

$$\mathrm{b}_0 = \arg\min_{\mathbf{S}\in\mathbb{R}^{(p+1)\times q}} \left\{ \frac{1}{mq} \sum_{i\in H_1} \sum_{j=1}^{q} \left(Y_{ij} - X_i^{\mathrm{T}} \mathbf{S}_j \right) + {}_0 \|\mathbf{S}\|_* \right\}. \tag{5.1}$$

The computation of the initial estimator is usually fast in practice since it only involves local samples on one machine, instead of entire samples.

Since $\|\mathrm{b}_0 - \|_F = O_{\mathbb{P}}(\sqrt{r(p+q)}(\log p + \log q)/mg^{1/2})$ (Chao et al., 2016), we x

$$a_n = a_{n;0} = c_1\sqrt{r(p+q)}(\log p + \log q)/mg^{1/2};$$

where $c_1 = 0.1$. Other choices of c_1 lead to similar results as long as the value of c_1 is not too extreme. The bandwidth h_j is simply set to be $h_j = a_n = c_1\sqrt{r(p+q)}(\log p + \log q)/mg^{1/2}$ and $h_{j;t} = a_{n;t-1}$ where $a_{n;t}$ is de ned by (4.3).

All the regularization parameters $_{n;q;t}$ in Algorithm 1 are chosen by validation. In particular, we minimize the quantile loss on an independently generated validation dataset with the sample size n. For the choice of the kernel functions $K_j(\cdot)$, we adopt the commonly used bi-weight kernel function for each response variable j, $K_j(\cdot)$ $K(\cdot)$ and

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

It is easy to verify that all $K_j(\cdot)$ satisfy the condition (C1). We also note that other choices of $K_j(\cdot)$ provide similar results. We use the tolerance parameter $e_r = 10^{-4}$ as the stopping criterion. In Algorithm 1, at t-th iteration, we compute $e = \|\mathrm{b}^{(t)} - \mathrm{b}^{(t-1)}\|_F^2 / \|\mathrm{b}^{(t-1)}\|_F^2$ and stop the algorithm once $e \leqslant e_r$. Typically, it only requires 4 5 iterations to stop.

We compare the proposed Dist RLRE with other two benchmark approaches. One is the averaging divide-and-conquer (Avg-DC) which computes the low-rank estimator proposed by Chao et al. (2016) on each local machine and combines the local estimators by taking the average. The other is the standard squared loss low-rank estimator with trace norm penalty on a single machine with pooled data proposed by Chen et al. (2013). We denote it by CDC following the rst letters of the last names of authors. As for the choice of in CDC, we simply set it to be 2 which is suggested in Chen et al. (2013). The performance of the three methods is evaluated by root mean square error (RMSE) and the estimated rank.

5.1 E ect of the QR Loss Under Heavy-Tailed Noise

We study the e ect of the QR loss with trace norm penalty in the presence of heavy-tailed noise. We compare the proposed Dist RLRE with Pooled RLRE (the estimator in (3.8)), Avg-DC, and CDC. We vary the sample size $n = \{5000, 10000, 15000, 20000\}$ and compute the root mean square errors and estimated ranks for each of the methods. We x the local sample size $m = 1000$, the number of predictors $p = 100$, the number of response variables $q = 100$ and the true rank of coe cient matrix $r = 10$. The results for all three types of noises are reported in Table 1.

Table 1 The RMSEs and estimated ranks of the Dist RLRE, Pooled RLRE, Avg-DC, and CDC under different sample size n. Local sample size $m = 1000$ and true rank $r = 10$. Noises are generated from different distributions.

Noise	n	Dist RLRE		Pooled RLRE		Avg-DC		CDC	
		RMSE	rank	RMSE	rank	RMSE	rank	RMSE	rank
Norm	5000	0.0155	10.38	0.0098	15.34	0.0487	53.58	0.0125	14.27
	10000	0.0166	10.01	0.0070	18.36	0.0411	100.00	0.0083	22.91
	15000	0.0126	10.11	0.0057	22.49	0.0409	100.00	0.0072	25.19
	20000	0.0129	10.00	0.0050	24.49	0.0470	100.00	0.0060	35.08
Cauchy	5000	0.0245	11.26	0.0163	19.53	0.0530	100.00	65.0576	17.79
	10000	0.0249	11.56	0.0116	27.51	0.0483	100.00	52.4385	28.77
	15000	0.0174	11.90	0.0096	37.53	0.0483	100.00	50.9524	39.10
	20000	0.0189	11.90	0.0087	35.72	0.0467	100.00	54.6778	47.66
Exp	5000	0.0118	11.21	0.0046	13.25	0.0243	100.00	0.0144	13.74
	10000	0.0115	11.31	0.0033	17.89	0.0170	100.00	0.0094	24.26
	15000	0.0102	11.38	0.0027	20.96	0.0174	100.00	0.0082	25.93
	20000	0.0138	10.99	0.0023	18.04	0.0162	100.00	0.0068	36.27

It is noted that the squared-loss estimator with trace norm penalty CDC uses the pooled data in a single machine. Thus when the noise is normal, it is expected that CDC has smaller root mean square errors. When the noise has a slightly heavier tail (Exponential noise), the proposed Dist RLRE outperforms CDC when n is small and becomes slightly worse with the increase of sample size n. In the case of heavy-tailed noise (Cauchy noise), CDC completely fails with a much larger RMSE. For all three types of noises, Dist RLRE is much better than Avg-DC in terms of both RMSE and rank recovery. The Dist RLRE achieves robust rank consistency while Avg-DC completely ruins the low-rank structure. As for CDC, the estimated rank can be much larger than the true rank.

We also report the results of our proposed pooled robust low-rank estimator (Pooled RLRE) in Table 1. We note that Pooled RLRE has a similar performance as CDC for normal noise and outperforms for heavier tail models (Exponential and Cauchy noises).

5.2 Effect of Sample Size and Local Sample Size

We study the performance of different approaches by varying the total sample size n and local sample size m. We fix the number of predictors $p = 100$, the number of response variables $q = 100$, and the true rank of coefficient matrix $r = 10$. Additionally, we vary the sample size $n = \{12000, 18000, 24000\}$ and the local sample size $m = \{1000, 1500, 2000\}$ and report the results under different types of noises in Tables 2–4.

Table 2 The RMSEs and estimated ranks of the Dist RLRE, Avg-DC, and CDC under different sample size n and different local sample size m. Noises are generated from Normal distribution.

Method	n	12000			18000			24000		
	m	1000	1500	2000	1000	1500	2000	1000	1500	2000
Dist RLRE	RMSE	0.0171	0.0114	0.0079	0.0158	0.0097	0.0068	0.0178	0.0096	0.0068
	rank	10.01	11.70	11.66	10.10	11.75	11.74	10.23	11.88	11.77
Avg-DC	RMSE	0.0493	0.0472	0.0460	0.0483	0.0449	0.0449	0.0362	0.0387	0.0413
	rank	100.00	81.92	61.04	100.00	100.00	90.02	100.00	100.00	100.00
CDC	RMSE	0.0081	0.0081	0.0081	0.0070	0.0070	0.0070	0.0054	0.0054	0.0054
	rank	22.82	20.94	22.68	30.68	32.43	31.82	38.52	40.65	29.64