SCIENCE CHINA
Mathematics

Robust low-rank data matrix approximations

FENG XingDong and HE XuMing

Citation: SCIENCE CHINA Mathematics **60**, 189 (2017); doi: 10.1007/s11425-015-0484-1

View online: http://engine.scichina.com/doi/10.1007/s11425-015-0484-1

View Table of Contents:http://engine.scichina.com/publisher/scp/journal/SCM/60/2

Published by the Science China Press

Articles you may be interested in

SCIENCE CHINA
Mathematics

Invited Articles

• ARTICLES •

February 2017 Vol. 60 No. 2: 189–200
doi: 10.1007/s11425-015-0484-1

Robust low-rank data matrix approximations

FENG XingDong[1,2] & HE XuMing[3,*]

[1]*School of Statistics and Management, Shanghai University of Finance and Economics, Shanghai* 200433*, China;*
[2]*Key Laboratory of Mathematical Economics (SUFE), Ministry of Education, Shanghai* 200433*, China;*
[3]*Department of Statistics, University of Michigan, Ann Arbor, MI* 48109*, USA*
Email: feng.xingdong@mail.shufe.edu.cn, xmhe@umich.edu

Received July 17, 2015; accepted October 17, 2016; published online November 22, 2016

Abstract We review some recent approaches to robust approximations of low-rank data matrices. We consider the problem of estimating a low-rank mean matrix when the data matrix is subject to measurement errors as well as gross outliers in some of its entries. The purpose of the paper is to make various algorithms accessible with an understanding of their abilities and limitations to perform robust low-rank matrix approximations in both low and high dimensional problems.

Keywords dimension reduction, low-rank, M-estimator, regression, robust estimator, singular value decomposition

MSC(2010) 62F03, 62F35

Citation: Feng X D, He X M. Robust low-rank data matrix approximations. Sci China Math, 2017, 60: 189–200, doi: 10.1007/s11425-015-0484-1

1 Introduction

Analysis of a large data matrix often calls for dimension reduction, and is sometimes necessitated by the need to characterize the row and column effects in a simpler structure. Low-rank representation or approximation of the data matrix is a useful and convenient approach; from latent semantic indexing in information retrieval to low-rank modeling in image analysis, researchers from a wide range of scientific areas have used low-rank matrix approximation to achieve dimension reduction. The mathematical formulation of low-rank matrix approximations is based on singular value decomposition (SVD) in matrix algebra and principal component analysis (PCA) in statistics.

As shown in [4], a low-rank subspace approximation based on the first singular values can be found by solving a least squares problem. The principal component analysis is based on the second moments as summary statistics of the data. It is then not surprising that SVD and PCA are sensitive to outliers in the data.

With a weighted least squares approach, alternating regression (or criss-cross regression) has been considered as an effective approach for low-rank approximation in earlier years [6, 15]. Chen et al. [3] have provided a robust SVD procedure with alternating regression based on an M-estimator [7] and the least trimmed squares estimator (LTS) [12]. In more recent years, researchers have considered the robust low-rank approximation problem as recovering a low-rank matrix and a sparse matrix that account for outliers. Candès et al. [2] and Zhou et al. [19] provided detailed discussions of this approach, and Agarwal

[*] Corresponding author

et al. [1] further considered the method with a more general design matrix. Following the idea of [2], She and Chen [13] discussed the general connection between robustness and nonconvexity under reduced rank regression. Along this line, Zhang et al. [17] used a penalization approach to deal with functional data matrices. Zhang and Lerman [18] proposed a M-type robust low-rank approximation procedure by using a different convex relaxation algorithm.

If the data matrix is simply a superposition of a low-rank component and a sparse component without random noise, it has been shown in [18] that we can recover the two components exactly by solving a convex problem. The problems we consider here are of a different nature, where the data matrix takes the form of

$$\boldsymbol{Y} = \sum_{l=1}^{k} \boldsymbol{\theta}_l \boldsymbol{\phi}_l^{\mathrm{T}} + \boldsymbol{E}, \tag{1.1}$$

where $\boldsymbol{Y}$ is an $n \times m$ data matrix, the vectors $\boldsymbol{\theta}_l$ and $\boldsymbol{\phi}_l$ $(l = 1, \ldots, k)$ explain the row and column effects, respectively, and the rows of $\boldsymbol{E}$ are independently distributed with mean $\mathbf{0}$ and covariance $\sigma^2 \boldsymbol{I}_m$. This model has the following features:

- The mean $E(\boldsymbol{Y})$ is a rank-k matrix, where k is typically small.
- The rows of the data matrix are independent but not necessarily identically distributed.
- The column effects $\boldsymbol{\phi}_l$ are fixed, but the row effects $\boldsymbol{\theta}_l$ are either fixed or random. If the row effects are random with mean $\boldsymbol{\mu}_l = E(\boldsymbol{\theta}_l)$ and independent components, the elements within each row of $\boldsymbol{Y}$ might be correlated through the random effects.

If we rewrite the model as

$$\boldsymbol{Y} = \sum_{l=1}^{k} \boldsymbol{\mu}_l \boldsymbol{\phi}_l^{\mathrm{T}} + \boldsymbol{E}^*,$$

the error matrix $\boldsymbol{E}^*$ has independent rows but each row has mean zero and an unspecified covariance structure. A similar model where all random variables are assumed normally distributed has been considered in [8, 9]. In applications, the column entries can be measurements at different times or locations of randomly sampled subjects. To estimate the low-rank matrix $E(\boldsymbol{Y})$ with robustness against outliers of general patterns, we review four existing algorithms in Section 2, and compare their performances in Section 3 through a small scale simulation study. The estimation procedures do not account for the correlation structure in $\boldsymbol{E}^*$, but if any statistical inference on the estimated row- or column-effects is to be carried out, additional assumptions on the model are needed as discussed in [5]. The conclusions are given in Section 4.

2 Four algorithms

We review four robust low-rank approximation algorithms in this section. Each represents a distinctive approach to the problem with its own merit.

2.1 Alternating regression approach

Since the SVD can be characterized by alternating regression (or criss-cross regression) as demonstrated in [6, 15], one approach to robust SVD is to robustify each regression step. Replacing the square loss by a robust loss function does not address the robustness issue completely, because the M-estimator of regression at each step of alternating regression has a low breakdown point when the dimension of the matrix is moderately high. To achieve stronger robustness against multiple outliers, Chen et al. [3] proposed a robust SVD procedure with the aid of a high breakdown point estimator at the earlier stages of the iteration. To fix notation, denote by $\boldsymbol{R}$ the matrix that consists of $\boldsymbol{\mu}_l$ $(l = 1, \ldots, k)$ as rows, and by $\boldsymbol{C}$ the matrix that consists of ϕ_l $(l = 1, \ldots, k)$ as rows. Then we have

$$E(\boldsymbol{Y}) = \boldsymbol{R}^{\mathrm{T}} \boldsymbol{C}.$$

Two robust regression estimators will be used. In generic forms, consider linear regression of z_i on x_i $(i = 1, \dots, n)$. The first is an M-estimator that minimizes

$$\sum_{i=1}^{n} L(z_i - x_i^{\mathrm{T}}\beta)$$

over β for a robust loss function L. The second is the least trimmed squares (LTS) estimator that minimizes

$$\sum_{J} (z_i - x_i^{\mathrm{T}}\beta)^2$$

over β, where the sum is over the smallest $(1-\alpha)100\%$ of the squared residuals for some $\alpha \in (0, 0.5]$. We refer to [11] for more details about the least trimmed squares regression.

The first few steps of the algorithm aim to protect us from breakdown in the regression estimates. A sketch of the algorithm is given as follows, where the element at the i-th row and j-th column of the data matrix $\boldsymbol{Y}$ is denoted by y_{ij}.

(A1) Robust scaling. A robust scaling for the j-th column is done by

$$y_{ij} = \frac{y_{ij}}{\mathrm{MAD}_i(y_{ij})}, \tag{2.1}$$

where $\mathrm{MAD}_i(y_{ij})$ is the median absolute deviation of the j-th column, before we estimate the row and column matrices. The same scaling procedure can be done to rows, but we should not scale both rows and columns at the same time.

(A2) Initialization. $\boldsymbol{R}^{(0)}$ is generated from a given distribution (e.g., each entry of the matrix is drawn from the uniform distribution $U(0,1)$).

(A3) Sequential stability test. This step is to moderate the effect from leveraged outliers.

(a) Suppose $\boldsymbol{R}^{(0)}$ is an initial row matrix. The columns of the data matrix $\boldsymbol{Y}$ are regressed on $\boldsymbol{R}^{(0)}$ to give the M-estimator $\boldsymbol{C}_M^{(0)}$ and the LTS estimator $\boldsymbol{C}_{\mathrm{LTS}}^{(0)}$.

(b) Compare $\boldsymbol{C}_M^{(0)}$ with $\boldsymbol{C}_{\mathrm{LTS}}^{(0)}$. If the two estimates differ from each other by more than a fixed amount, let $\boldsymbol{C}_{\mathrm{LTS}}^{(0)}$ be our initial column matrix $\boldsymbol{C}^{(0)}$. Otherwise, let $\boldsymbol{C}^{(0)} = \boldsymbol{C}_M^{(0)}$.

(c) Regress rows of $\boldsymbol{Y}$ on $\boldsymbol{C}^{(0)}$ to get the M-estimator and the LTS estimator of the row matrices $\boldsymbol{R}_M^{(1)}$ and $\boldsymbol{R}_{\mathrm{LTS}}^{(1)}$, respectively. If these two estimates differ from each other by more than a fixed amount, take $\boldsymbol{R}^{(1)} = \boldsymbol{R}_{\mathrm{LTS}}^{(1)}$. Otherwise, take $\boldsymbol{R}^{(1)} = \boldsymbol{R}_M^{(1)}$.

(d) Repeat the process until both $(\boldsymbol{R}_M^{(m)}, \boldsymbol{R}_{\mathrm{LTS}}^{(m)})$ and $(\boldsymbol{C}_M^{(m)}, \boldsymbol{C}_{\mathrm{LTS}}^{(m)})$ show small differences; or a prespecified number of iterations is reached. Denote by $\boldsymbol{R}^{(m_0)}$ the row matrix estimate from the above iteration.

(A4) Alternating M regression. Perform alternating regression by minimizing

$$D(\boldsymbol{R}, \boldsymbol{C}) \triangleq \Delta(\boldsymbol{Y} - \boldsymbol{R}^{\mathrm{T}}\boldsymbol{C}), \tag{2.2}$$

with $\boldsymbol{R}^{(m_0)}$ as the starting value for $\boldsymbol{R}$, where Δ is a matrix norm based on a robust loss function L, i.e.,

$$\Delta(\boldsymbol{A}) = \sum_{i=1}^{n}\sum_{j=1}^{m} L(a_{ij}), \tag{2.3}$$

where a_{ij} is the ij-th entry of the $n \times m$ matrix $\boldsymbol{A}$.

After the lower rank matrix $\boldsymbol{R}^{\mathrm{T}}\boldsymbol{C}$ is estimated from the alternating M regression, we apply the regular SVD procedure to the matrix to obtain the robust SVD.

Several loss functions L are compared in [3], and the logistic loss is recommended for good numerical performance. This algorithm has been implemented in an R package "rsvd" (available at http://bb.shufe.edu.cn/bbcswebdav/users/2011000070/rsvd.zip) for non-commercial use. It is possible that the algorithm finds multiple solutions with different randomization in its initial step. Unless L is the square loss, the objective function in alternating M regression is not a convex function. In the empirical investigations in the next section, we take any solution from the algorithm.

The alternating regression approach is designed to provide protection against leveraged outliers in the regression estimation, but the algorithmic convergence is not guaranteed, and the solutions are often non-unique. Because of the use of LTS in the earlier stages of the algorithm, the computational complexity increases quickly with the dimension of the data matrix.

2.2 Sub-sampling method

To obtain a robust estimate of the column effects in the presence of outlying rows, Feng and He [5] considered a sub-sampling approach for robust SVD of a data matrix. The algorithm proceeds as follows.

(T1) Obtain robust column estimates:

(a) select a fixed number of rows randomly from the data matrix, denoted as $\boldsymbol{Y}_p$;

(b) apply the regular SVD on the matrix $\boldsymbol{Y}_p$, and denote the first k right singular vector as $\widehat{\boldsymbol{\phi}}_l(\boldsymbol{Y}_p)$, $l = 1, \ldots, k$;

(c) find $\widehat{\boldsymbol{\theta}}_l(\boldsymbol{Y}_p)$ for $l = 1, \ldots, k$, the row parameter estimate for $\boldsymbol{\theta}$ by minimizing the objective function $\|\boldsymbol{Y} - \sum_{l=1}^k \boldsymbol{\theta}_l \widehat{\boldsymbol{\phi}}_l\|_1$, and denote the minimum by $\rho(\boldsymbol{Y}_p)$;

(d) repeat (a)–(c) for a pre-specified number of times, and find the subset $\widetilde{\boldsymbol{Y}}_p$ with the smallest value $\rho(\boldsymbol{Y}_p)$;

(e) given the initial estimate of the column parameters $\widehat{\boldsymbol{\phi}}_l(\boldsymbol{Y}_p), l = 1, \ldots, k$, choose a trimming proportion α and calculate the weights

$$s_i = 1(\hat{q}_\alpha < \|\widehat{\boldsymbol{\epsilon}}_i\|^2 \leqslant \hat{q}_{1-\alpha}), \tag{2.4}$$

where $\hat{q}_\alpha$ is the sample α quantile of $\|\widehat{\boldsymbol{\epsilon}}_i\|^2$ and

$$\widehat{\boldsymbol{\epsilon}}_i = \left(I - \sum_{l=1}^k \widehat{\boldsymbol{\phi}}_l \widehat{\boldsymbol{\phi}}_l^{\mathrm{T}}\right) \boldsymbol{y}_i;$$

(f) minimize

$$\sum_{i=1}^n s_i \left\| \boldsymbol{y}_i - \sum_{l=1}^k \theta_{il} \boldsymbol{\phi}_l \right\|_2^2$$

over $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, where θ_{il} is the i-th element of $\boldsymbol{\theta}_l$, and keep the column estimates $\widetilde{\boldsymbol{\phi}}_l$ $(l = 1, \ldots, k)$.

(T2) Obtain the i-th row estimate by minimizing

$$\sum_{j=1}^m L\left(y_{ij} - \sum_{l=1}^k \theta_{il} \widetilde{\boldsymbol{\phi}}_{jl}\right)$$

over $\boldsymbol{\theta}$ parameters for $i = 1, \ldots, n$, where $\widetilde{\boldsymbol{\phi}}_{jl}$ is the j-th element of $\widetilde{\boldsymbol{\phi}}_l$, and L is some robust loss function.

The steps (a)–(d) of the sub-sampling method aim to find a sub-matrix of rows that contains no outliers with high probability. The clean rows are then used to obtain the column effect estimates in (f). The row effects estimates are then obtained in (T2) through a robust loss function L. The consistency of the parameter estimates of this method was established in [5] in the case of $n \to \infty$. If the data matrix $\boldsymbol{Y}$ contains outliers in a majority of rows, the sub-sampling method may not work well.

2.3 Iterative re-weighted least squares

Zhang and Lerman [18] proposed a robust estimator for principal component analysis based on a constrained optimization approach. One can use this method to estimate the column effects $\boldsymbol{\phi}$ and then implement a robust regression procedure row by row as in (T2) of the sub-sampling method.

In a constrained set

$$\mathbb{C} = \{\boldsymbol{P} \in \mathbb{R}^{m \times m} : \boldsymbol{P} = \boldsymbol{P}^{\mathrm{T}}, \mathrm{tr}(\boldsymbol{P}) = 1\},$$

Zhang and Lerman [18] minimize the objective function

$$\rho_1(\boldsymbol{P}) = \sum_{i=1}^{n} \|\boldsymbol{P}\boldsymbol{y}_i\|_2,$$

where $\boldsymbol{y}_i$ is the i-th row of the data matrix $\boldsymbol{Y}$. With the constrained estimate $\widehat{\boldsymbol{P}}$, they obtain the low-rank projection space expanded by eigenvectors of the matrix $\widehat{\boldsymbol{P}}$, which is also called the geometric median subspace (GMS). By considering a convex objective function

$$\rho_2(\boldsymbol{P}) = \sum_{i=1}^{n} \|\boldsymbol{P}\boldsymbol{y}_i\|_2^2,$$

the GMS estimator can be interpreted as a robust version of an inverse covariance estimator. By [18, Theorem 10], there exists an explicit form for the solution to the optimization problem $\min_{\boldsymbol{P}\in\mathbb{C}} \rho_2(\boldsymbol{P})$, i.e.,

$$\widehat{\boldsymbol{P}}' = (\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})^{-1}/\mathrm{tr}\{(\boldsymbol{Y}^{\mathrm{T}}\boldsymbol{Y})^{-1}\}.$$

Thus, following iterative re-weighted least squares algorithm can be used with a regularization parameter λ.

(R1) Initialize the estimate $\widehat{\boldsymbol{P}}_1 = \boldsymbol{I}/m$.

(R2) At the l-th iteration, update the estimate as

$$\widehat{\boldsymbol{P}}_l = \left(\sum_{i=1}^{n} w_i \boldsymbol{y}_i \boldsymbol{y}_i^{\mathrm{T}}\right)^{-1} \Bigg/ \mathrm{tr}\left\{\left(\sum_{i=1}^{n} w_i \boldsymbol{y}_i \boldsymbol{y}_i^{\mathrm{T}}\right)^{-1}\right\},$$

where $w_i = [\max\{\|\widehat{\boldsymbol{P}}_l \boldsymbol{y}_i\|, \lambda\}]^{-1}$.

(R3) Repeat the above iteration until some stopping criteria are satisfied.

The algorithmic convergence and complexity of this method are analyzed in [18]. The matlab code for the algorithm can be downloaded from the website https://web.math.princeton.edu/~tengz/gms.shtml. This method is computationally attractive for high dimensional data matrices, but its ability to accommodate outliers might not be as good as the first two methods reviewed here. For a variation of this approach to regularized versions of robust SVD, see [17].

2.4 Constrained optimization

Extending the work of [2] where the data matrix is assumed to be the sum of a low-rank matrix as the main feature and a sparse matrix to represent outliers, Xu et al. [16] and Zhou et al. [19] considered a more realistic structure

$$\boldsymbol{Y} = \boldsymbol{X} + \boldsymbol{Z} + \boldsymbol{U}, \tag{2.5}$$

where the matrix $\boldsymbol{X}$ is low-rank, the matrix $\boldsymbol{Z}$ is sparse, and the matrix $\boldsymbol{U}$ is the random noise. The recovery procedure uses the following constrained optimization:

$$\begin{aligned} &\min_{\boldsymbol{X},\boldsymbol{Z}} \|\boldsymbol{X}\|_* + \|\boldsymbol{Z}\|_{1,2} \\ \text{s.t.}\quad &\|\boldsymbol{Y} - \boldsymbol{X} - \boldsymbol{Z}\|_F \leqslant \delta, \end{aligned}$$

where $\|\cdot\|_*$ is the nuclear norm, $\|\cdot\|_{1,2}$ is the sum of the l_2 norm of the columns of a matrix, $\|\cdot\|_F$ is the Frobenius norm, and δ is a threshold value. A proximal gradient algorithm is used to solve the above optimization problem. We refer to [10] for a good review of proximal algorithms that can deal with non-smooth convex objective functions. The matlab code implementing the algorithm is available at the website http://guppy.mpe.nus.edu.sg/~mpexuh/code/OP.zip. The SVD of the matrix $\boldsymbol{X}$ is taken to be robust estimates of the row and column effects for $\boldsymbol{Y}$.

3 Numerical studies

We report the performance of the four procedures reviewed in the previous section in a small scale simulation study. Three sets of dimensions $(n,m)=(20,12),(n,m)=(50,42)$ and $(n,m)=(100,78)$ are used in the study.

3.1 The first study

For each dimension (n,m), we generate 5,000 data matrices from (1.1) with $k=2$, where

$$\boldsymbol{\mu}_1=(20,\dots,20)^{\mathrm{T}},\quad \boldsymbol{\mu}_2=2^{1/2}(1,-1,\dots,1,-1)^{\mathrm{T}},\quad \boldsymbol{\phi}_1=(1,\dots,1)^{\mathrm{T}}/m^{1/2}$$

and $\boldsymbol{\phi}_2=(1,-1,\dots,1,-1)^{\mathrm{T}}/m^{1/2}$, and the random effects $\boldsymbol{\theta}_1-\boldsymbol{\mu}_1$ and $\boldsymbol{\theta}_2-\boldsymbol{\mu}_2$ are generated from the multivariate normal distributions with mean $\mathbf{0}$ and covariances $4\boldsymbol{I}_m$ and $\boldsymbol{I}_m$, respectively.

To assess robustness, we generate the first two rows of the matrix from a contaminated model, in which each entry is drawn from the mixture of the normal distribution $N(0,11)$ with probability 0.1 and one of following three distributions with probability 0.9: (I) $2^{-1/2}N(0,1)$; (II) $(3/10)^{-1/2}t_5$, where t_5 is the t distribution with 5 degrees of freedom; (III) $2^{-1}(\chi_1^2-1)$, where χ_1^2 is the χ^2 distribution with 1 degree of freedom. The other rows are generated from the distribution (I), (II), or (III) without contamination. The procedures discussed in the earlier section will be denoted as

(1) “M1”: the sub-sampling method of [5].

(2) “M2”: the alternating regression method of [3].

(3) “M3”: the iterative re-weighted LS method of [18].

(4) “M4”: the constrained optimization method of [16].

For the initial step (a) of the method M1, we use 100 randomly selected subsets os size $0.3n$, and the constant $\alpha=0.1$ is used in Step (e). Two robust loss functions are used for each of the three procedures M1–M3. No robust loss function is needed in M4.

(1) “Logistic”: $L(t)=K\log\{\cosh(t/K)\}$.

(2) “Huber”:

$$L(t)=\begin{cases}2^{-1}t^2, & |t|\leqslant K,\\ K|t|-2^{-1}K^2, & |t|>K,\end{cases}$$

where the tuning parameter is chosen as $K=0.1$. Those specific choices of constants are in line with what have been used in the literature. Two performance measures will be used. They are the following:

(1) $D_1=E\|\hat{\boldsymbol{\phi}}_1-\boldsymbol{\phi}_1\|_\infty$;

(2) $D_2=E\|\hat{\boldsymbol{\phi}}_2-\boldsymbol{\phi}_2\|_\infty$;

(3) $D_3=E\|\sum_{l=1}^2\hat{\boldsymbol{\theta}}_l\hat{\boldsymbol{\phi}}_l^{\mathrm{T}}-\sum_{l=1}^2\boldsymbol{\theta}_l\boldsymbol{\phi}_l^{\mathrm{T}}\|_\infty$,

where $\|\cdot\|_\infty$ refers to the super norm. In the simulation study, the expectations are replaced by Monte Carlo averages. In the simulation study, the expectations are replaced by Monte Carlo averages.

We see from Table 1 that the method of [18] and the constrained optimization of [16] can hardly capture the right directions under (1.1), and the other two methods are satisfactory. However, the method M2 works better for relatively large matrices in this study, probably because we fixed the number of sub-samples taken in the method M1. If the number of sub-samples is increased, the reliability of M1 improves. Parallel computing is ideal for accommodating a large number of sub-samples in the method M1. However, if outliers appear in a large number of rows, the sub-sampling method may break down. Table 2 reports the results when outliers are generated from the normal distribution $N(0,11)$ in each row with probability 0.15. In this case, the performance of the method M4 is satisfactory in most cases, and the method M2 does better when the matrix size is larger. We also note that the choice of robust loss functions (logistic or Huber’s) has little impact on the results, so only the results from the loss function “Logistic” are kept in all the tables except Table 1.

Table 1 Performance measures D_1, D_2 and D_3 in the first study as outliers appear in each of the first two rows with probability 0.1. Each entry contains the estimated performance measure and its standard error (in brackets), where $*$ refers to a value less than 0.005

			Logistic			Huber		
Error	Method	$n \times m$	D_1	D_2	D_3	D_1	D_2	D_3
Normal	M1	20×12	0.02 ($*$)	0.22 ($*$)	1.50 (0.02)	0.02 ($*$)	0.22 (0.02)	1.50 (0.02)
	M2		0.03 ($*$)	0.27 (0.02)	2.32 (0.05)	0.02 ($*$)	0.27 (0.02)	2.18 (0.05)
	M3		0.43 (0.04)	0.75 ($*$)	12.59 (0.11)	0.43 ($*$)	0.75 ($*$)	12.6 (0.11)
	M4		0.09 ($*$)	0.71 ($*$)	3.32 (0.02)	–	–	–
	M1	50×42	0.01 ($*$)	0.16 ($*$)	1.01 ($*$)	0.01 ($*$)	0.15 ($*$)	1.02 ($*$)
	M2		0.01 ($*$)	0.17 ($*$)	1.12 ($*$)	0.01 ($*$)	0.17 ($*$)	1.13 ($*$)
	M3		0.56 ($*$)	0.60 ($*$)	19.76 (0.09)	0.56 ($*$)	0.60 ($*$)	19.77 (0.09)
	M4		0.06 ($*$)	0.38 ($*$)	4.79 (0.07)	–	–	–
	M1	100×78	0.01 ($*$)	0.13 ($*$)	0.86 ($*$)	0.01 ($*$)	0.13 ($*$)	0.86 ($*$)
	M2		0.01 ($*$)	0.15 ($*$)	0.96 ($*$)	0.01 ($*$)	0.15 ($*$)	0.96 ($*$)
	M3		0.54 ($*$)	0.54 ($*$)	22.55 (0.08)	0.54 ($*$)	0.54 ($*$)	22.55 (0.08)
	M4		0.03 ($*$)	0.49 ($*$)	10.78 (0.10)	–	–	–
t	M1	20×12	0.02 ($*$)	0.23 ($*$)	1.56 (0.02)	0.02 ($*$)	0.22 ($*$)	1.54 (0.02)
	M2		0.02 ($*$)	0.24 ($*$)	2.18 (0.05)	0.02 ($*$)	0.23 ($*$)	1.95 (0.04)
	M3		0.43 ($*$)	0.77 ($*$)	12.55 (0.10)	0.43 ($*$)	0.77 ($*$)	12.55 (0.10)
	M4		0.09 ($*$)	0.71 ($*$)	3.33 (0.02)	–	–	–
	M1	50×42	0.01 ($*$)	0.15 ($*$)	0.96 ($*$)	0.01 ($*$)	0.15 ($*$)	0.97 ($*$)
	M2		0.01 ($*$)	0.14 ($*$)	0.94 ($*$)	0.01 ($*$)	0.14 ($*$)	0.94 ($*$)
	M3		0.56 ($*$)	0.60 ($*$)	19.56 (0.09)	0.56 ($*$)	0.60 ($*$)	19.56 (0.09)
	M4		0.06 ($*$)	0.39 ($*$)	4.91 (0.07)	–	–	–
	M1	100×78	0.01 ($*$)	0.13 ($*$)	0.79 ($*$)	0.01 ($*$)	0.13 ($*$)	0.79 ($*$)
	M2		0.01 ($*$)	0.12 ($*$)	0.77 ($*$)	0.01 ($*$)	0.12 ($*$)	0.77 ($*$)
	M3		0.54 ($*$)	0.54 ($*$)	22.7 (0.08)	0.54 ($*$)	0.54 ($*$)	22.7 (0.08)
	M4		0.03 ($*$)	0.48 ($*$)	10.83 (0.10)	–	–	–
χ^2	M1	20×12	0.02 ($*$)	0.23 ($*$)	1.64 (0.03)	0.02 ($*$)	0.23 ($*$)	1.66 (0.03)
	M2		0.02 ($*$)	0.18 ($*$)	1.77 (0.04)	0.02 ($*$)	0.18 ($*$)	1.75 (0.04)
	M3		0.44 ($*$)	0.77 ($*$)	13.04 (0.10)	0.44 ($*$)	0.77 ($*$)	13.05 (0.10)
	M4		0.09 ($*$)	0.70 ($*$)	3.30 (0.02)	–	–	–
	M1	50×42	0.01 ($*$)	0.17 ($*$)	0.90 ($*$)	0.01 ($*$)	0.17 ($*$)	0.90 ($*$)
	M2		0.01 ($*$)	0.10 ($*$)	0.65 ($*$)	0.01 ($*$)	0.10 ($*$)	0.65 ($*$)
	M3		0.56 ($*$)	0.60 ($*$)	19.57 (0.09)	0.56 ($*$)	0.60 ($*$)	19.57 (0.09)
	M4		0.06 ($*$)	0.38 ($*$)	4.94 (0.07)	–	–	–
	M1	100×78	0.01 ($*$)	0.14 ($*$)	0.75 ($*$)	0.01 ($*$)	0.14 ($*$)	0.75 ($*$)
	M2		0.01 ($*$)	0.08 ($*$)	0.56 ($*$)	0.01 ($*$)	0.08 ($*$)	0.56 ($*$)
	M3		0.54 ($*$)	0.55 ($*$)	22.62 (0.08)	0.54 ($*$)	0.55 ($*$)	22.62 (0.08)
	M4		0.03 ($*$)	0.48 ($*$)	10.88 (0.10)	–	–	–

3.2 The second study

Here we consider the same combinations of dimensions of matrices as used in Subsection 3.1. We first generate a rank-two data matrix without outliers, denoted as $\boldsymbol{Y}_1$, by using Model (1.1) with

Table 2 Performance measures D_1, D_2 and D_3 in the first study as outliers appear in every row with probability 0.15. Each entry contains the estimated performance measure and its standard error (in brackets), where $*$ refers to a value less than 0.005

				Logistic	
Error	Method	$n \times m$	D_1	D_2	D_3
Normal	M1	20×12	0.22 ($*$)	0.81 ($*$)	38.85 (0.17)
	M2		0.11 ($*$)	0.77 ($*$)	29.59 (0.20)
	M3		0.80 ($*$)	0.83 ($*$)	47.53 (0.16)
	M4		0.17 ($*$)	0.72 ($*$)	5.78 (0.01)
	M1	50×42	0.17 ($*$)	0.65 ($*$)	32.58 (0.24)
	M2		0.03 ($*$)	0.31 ($*$)	7.00 (0.19)
	M3		0.66 ($*$)	0.68 ($*$)	57.01 (0.14)
	M4		0.12 ($*$)	0.53 ($*$)	3.70 ($*$)
	M1	100×78	0.12 ($*$)	0.53 ($*$)	13.7 (0.21)
	M2		0.01 ($*$)	0.20 ($*$)	1.97 (0.07)
	M3		0.57 ($*$)	0.58 ($*$)	61.07 (0.13)
	M4		0.09 ($*$)	0.58 ($*$)	2.79 ($*$)
t	M1	20×12	0.22 ($*$)	0.81 ($*$)	38.81 (0.17)
	M2		0.11 ($*$)	0.77 ($*$)	29.38 (0.20)
	M3		0.80 ($*$)	0.83 ($*$)	47.66 (0.17)
	M4		0.17 ($*$)	0.72 ($*$)	5.79 (0.01)
	M1	50×42	0.17 ($*$)	0.65 ($*$)	32.56 (0.24)
	M2		0.02 ($*$)	0.24 ($*$)	4.38 (0.15)
	M3		0.66 ($*$)	0.68 ($*$)	57.18 (0.13)
	M4		0.12 ($*$)	0.54 ($*$)	3.69 ($*$)
	M1	100×78	0.12 ($*$)	0.54 ($*$)	13.17 (0.20)
	M2		0.01 ($*$)	0.16 ($*$)	1.33 (0.05)
	M3		0.57 ($*$)	0.57 ($*$)	60.96 (0.14)
	M4		0.09 ($*$)	0.57 ($*$)	2.79 ($*$)
χ^2	M1	20×12	0.22 ($*$)	0.81 ($*$)	38.75 (0.17)
	M2		0.11 ($*$)	0.72 ($*$)	27.87 (0.22)
	M3		0.80 ($*$)	0.83 ($*$)	47.79 (0.17)
	M4		0.17 ($*$)	0.72 ($*$)	5.79 ($*$)
	M1	50×42	0.17 ($*$)	0.65 ($*$)	32.36 (0.24)
	M2		0.02 ($*$)	0.14 ($*$)	1.43 (0.06)
	M3		0.66 ($*$)	0.68 ($*$)	56.91 (0.13)
	M4		0.12 ($*$)	0.54 ($*$)	3.69 (0.01)
	M1	100×78	0.12 ($*$)	0.53 ($*$)	12.59 (0.20)
	M2		0.01 ($*$)	0.10 ($*$)	0.74 (0.02)
	M3		0.57 ($*$)	0.58 ($*$)	61.11 (0.14)
	M4		0.09 ($*$)	0.59 ($*$)	2.69 ($*$)

$\boldsymbol{\phi}_1 = (1,\ldots,1)^{\mathrm{T}}/m^{1/2}$, $\boldsymbol{\phi}_2 = (1,-1,\ldots,1,-1)^{\mathrm{T}}/m^{1/2}$, and the elements of the vectors $\boldsymbol{\theta}$'s and the matrix $\boldsymbol{E}$ independently generated from the standard normal distribution $N(0,1)$ and the uniform distribution $U(0,1)$, respectively. We then produce a matrix, denoted as $\boldsymbol{Y}_2$, composed of two m-dimensional rows with the elements independently generated from the uniform distribution $U(0,2)$. Let

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{Y}_1 \\ \boldsymbol{Y}_2 \end{pmatrix}.$$

Similar synthetic data are considered by Zhang and Lerman [18]. We produce 5,000 such matrices, and carry out four robust SVD procedures (M1–M4) on each matrix $\boldsymbol{Y}$. Two performance measures as defined in Subsection 3.1 are reported in Table 3, where the performance measure D_1 is applied to the mean matrix with the last two rows removed. Note that the last two rows of $\boldsymbol{Y}$ under our model are just noise.

In this setting, the performance of the method M3 has improved over the first study, because this method is more capable of handling outliers of modest sizes. The performance of robust SVD procedures M1, M2 and M4 are roughly comparable, as they all down-weight the effect of outlying rows.

3.3 The third study

In this study, we consider a case where outliers appear in the space orthogonal to the space expanded by column vectors $\boldsymbol{\phi}_l, l = 1, \ldots, k$ of (1.1), which is also discussed by She et al. [14]. The simulation model can be formulated as

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{\Phi}^{\mathrm{T}} + \boldsymbol{T}\boldsymbol{\Phi}_{\perp}^{\mathrm{T}} + \boldsymbol{E}, \tag{3.1}$$

where $\boldsymbol{U}$ and $\boldsymbol{\Phi}$ are the first k left and right singular vectors of an $n \times m$ matrix with each element independently drawn from the standard normal distribution, respectively, the matrix $\boldsymbol{\Phi}_{\perp}$ is composed of all of the rest right singular vectors, $\boldsymbol{D}$ is a $k \times k$ diagonal matrix, and $\boldsymbol{T}$ is an $n \times (m-k)$ matrix of the form

$$\begin{pmatrix} 2 & 2 & \cdots & 2 \\ 2 & 2 & \cdots & 2 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{pmatrix}.$$

Table 3 Performance measures D_1, D_2 and D_3 in the second study where two rows are contaminated with bounded noise. The standard errors of all the entries are less than 0.005

			Logistic	
Method	$n \times m$	D_1	D_2	D_3
M1		0.09	0.18	1.01
M2	20×12	0.10	0.20	1.03
M3		0.26	0.44	1.23
M4		0.08	0.37	0.96
M1		0.04	0.12	0.92
M2	50×42	0.04	0.16	0.98
M3		0.14	0.43	1.24
M4		0.03	0.10	0.77
M1		0.02	0.09	0.86
M2	100×78	0.03	0.13	0.93
M3		0.10	0.35	1.16
M4		0.02	0.08	0.73

Table 4 Performance measures D_1, D_2 and D_3 under logistic loss in the third study where $\boldsymbol{D} = \mathrm{Diag}\{60, 40\}$ ($k = 2$) and $\boldsymbol{D} = \mathrm{Diag}\{60, 40, 20, 10\}$ ($k = 4$), respectively. Each entry contains the estimated performance measure and its standard error (in brackets), where $*$ refers to a value less than 0.005

				$k=2$			$k=4$	
Error	Method	$n \times m$	D_1	D_2	D_3	D_1	D_2	D_3
Normal	M1	20×12	0.06 ($*$)	0.65 ($*$)	25.59 (0.17)	0.10 ($*$)	0.71 ($*$)	59.25 (0.17)
	M2		0.06 ($*$)	0.68 ($*$)	27.73 (0.16)	0.09 ($*$)	0.74 ($*$)	60.04 (0.18)
	M4		0.22 ($*$)	0.77 ($*$)	6.72 ($*$)	0.22 ($*$)	0.77 ($*$)	38.10 (0.011)
	M1	50×42	0.08 ($*$)	0.63 ($*$)	6.60 (0.07)	0.08 ($*$)	0.65 ($*$)	43.01 (0.06)
	M2		0.12 ($*$)	0.60 ($*$)	8.90 (0.12)	0.12 ($*$)	0.61 ($*$)	46.53 (0.12)
	M4		0.12 ($*$)	0.50 ($*$)	4.13 ($*$)	0.12 ($*$)	0.50 ($*$)	38.13 (0.01)
	M1	100×78	0.06 ($*$)	0.56 ($*$)	5.18 (0.06)	0.06 ($*$)	0.57 ($*$)	42.72 (0.07)
	M2		0.12 ($*$)	0.49 ($*$)	7.72 (0.13)	0.12 ($*$)	0.49 ($*$)	45.86 (0.14)
	M4		0.09 ($*$)	0.48 ($*$)	3.46 (0.01)	0.09 ($*$)	0.49 ($*$)	39.55 ($*$)
t	M1	20×12	0.06 ($*$)	0.66 ($*$)	25.75 (0.16)	0.10 ($*$)	0.71 ($*$)	59.22 (0.17)
	M2		0.06 ($*$)	0.69 ($*$)	27.63 (0.16)	0.09 ($*$)	0.73 ($*$)	59.89 (0.19)
	M4		0.22 ($*$)	0.76 ($*$)	6.71 ($*$)	0.22 ($*$)	0.77 ($*$)	38.10 (0.01)
	M1	50×42	0.08 ($*$)	0.64 ($*$)	6.69 (0.08)	0.08 ($*$)	0.68 ($*$)	43.24 (0.06)
	M2		0.12 ($*$)	0.64 ($*$)	10.59 (0.15)	0.12 ($*$)	0.65 ($*$)	47.71 (0.14)
	M4		0.12 ($*$)	0.51 ($*$)	4.15 ($*$)	0.12 ($*$)	0.50 ($*$)	38.12 (0.01)
	M1	100×78	0.06 ($*$)	0.58 ($*$)	5.38 (0.07)	0.07 ($*$)	0.60 ($*$)	43.24 (0.06)
	M2		0.12 ($*$)	0.52 ($*$)	8.73 (0.15)	0.12 ($*$)	0.53 ($*$)	46.48 (0.15)
	M4		0.09 ($*$)	0.48 ($*$)	3.46 (0.01)	0.09 ($*$)	0.49 ($*$)	39.55 ($*$)
χ^2	M1	20×12	0.06 ($*$)	0.66 ($*$)	26 (0.16)	0.10 ($*$)	0.71 ($*$)	59.29 (0.17)
	M2		0.06 ($*$)	0.68 ($*$)	27.5 (0.16)	0.10 ($*$)	0.74 ($*$)	59.95 (0.19)
	M4		0.22 ($*$)	0.76 ($*$)	6.72 ($*$)	0.22 ($*$)	0.77 ($*$)	38.08 (0.01)
	M1	50×42	0.10 ($*$)	0.67 ($*$)	6.86 (0.08)	0.10 ($*$)	0.70 ($*$)	43.35 (0.07)
	M2		0.17 ($*$)	0.62 ($*$)	10.41 (0.15)	0.17 ($*$)	0.64 ($*$)	48.12 (0.15)
	M4		0.12 ($*$)	0.51 ($*$)	4.17 ($*$)	0.12 ($*$)	0.50 ($*$)	38.13 (0.01)
	M1	100×78	0.29 ($*$)	0.46 ($*$)	7.06 (0.12)	0.30 ($*$)	0.47 ($*$)	44.86 (0.13)
	M2		0.27 ($*$)	0.47 ($*$)	9.31 (0.16)	0.28 ($*$)	0.48 ($*$)	47.54 (0.16)
	M4		0.09 ($*$)	0.48 ($*$)	3.51 (0.01)	0.09 ($*$)	0.49 ($*$)	39.55 ($*$)

The elements of the error matrix $\boldsymbol{E}$ are generated from one of three distributions (I) $2^{-1/2}N(0,1)$; (II) $(3/10)^{-1/2}t_5$; (III) $2^{-1}(\chi_1^2 - 1)$ as described in the first study. For each combination of the dimensions n and m, we generated 5,000 matrices from (3.1). Since it costs more than 10 minutes for each generated matrix with the method M3, we have to terminate the related computation based on this approach. The comparisons among other methods are reported in Table 4 where datasets are generated from (3.1) with the rank $k = 2$ and $k = 4$ for the matrix $\boldsymbol{D}$, respectively.

We still consider the rank-2 matrix approximation regardless of the true rank k. As indicated in Table 4, the performance of the method M4 is clearly better than other methods based on the measure of the difference between the estimated and the true mean matrices, although larger matrix size can shrink such differences. Since the setting of (3.1) is similar to those considered by Candès et al. [2] where a matrix is composed of a low-rank matrix and a sparse matrix, we are not surprised to observe better performance of the method M4, especially when the matrix is large. When the true rank k in (3.1) is larger than what we have thought, the performance of all methods in recovering the low-rank matrix $\boldsymbol{UD\Phi}^{\mathrm{T}}$ is not as satisfactory as that of the case where we correctly specify the rank k.

4 Conclusions

Different algorithms to estimate robust low-rank data matrices have been developed in recent years. The iterative re-weighted least squares method uses the geometric median subspace to provide a fast and scalable algorithm for high dimensional matrices with outliers of modest sizes. Gross outliers that occur in a small number of rows can be well accommodated by the sub-sampling method, and the sub-sampling algorithm is especially suited for parallel computing. The robust alternating regression approach is capable of handling more general patterns of outliers, but it is computationally difficult to scale and there is no assurance of algorithmic convergence. For matrices composed of low-rank and sparse matrices, the constrained approach can better recover the low-rank structures.

The computational complexity of the algorithms depends on the dimension of the data matrix. The alternating regression approach and the sub-sampling approach are probably the least scalable with the dimension. The former uses the LTS estimator that is expensive to compute well, and the sub-sampling method requires a larger number of sub-samples to handle a larger number of outlying rows. The iterative re-weighted least squares method is computationally simpler but can run into problems of convergence when ill-conditioned matrices are present in the iterations. The constrained optimization method is computationally more stable then the other methods. Although the empirical study reported in the present paper is limited in scope, we hope that it will stimulate further developments and comparisons in robust low-rank matrix approximations, both in theory and in computation.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant No. 11571218), the State Key Program in the Major Research Plan of National Natural Science Foundation of China (Grant No. 91546202), Program for Changjiang Scholars and Innovative Research Team in Shanghai University of Finance and Economics (Grant No. IRT13077), and Program for Innovative Research Team of Shanghai University of Finance and Economics.

References

1 Agarwal A, Negahban S, Wainwright M J. Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. Ann Statist, 2012, 40: 1171–1197

2 Candès E J, Li X, Ma Y, et al. Robust principal component analysis? J ACM, 2011, 58: 1–73

3 Chen C, He X, Wei Y. Lower rank approximation of matrices based on fast and robust alternating regression. J Comput Graph Statist, 2008, 17: 186–200

4 Eckart C, Young C. The approximation of one matrix by another of low rank. Psychometrika, 1936, 1: 211–218

5 Feng X, He X. Statistical inference based on robust low-rank data matrix approximation. Ann Statist, 2014, 42: 190–210

6 Gabriel K R, Zamir S. Lower rank approximation of matrices by least squares with any choice of weights. Technometrics, 1979, 21: 489–498

7 Huber P J. Robust estimation of a location parameters. Ann Math Statist, 1964, 35: 73–101

8 Johnstone I M. On the distribution of the lalarge eigenvalue in principal component analysis. Ann Statist, 2001, 29: 295–327

9 Johnstone I M, Lu A Y. On consistency and sparsity for principal components analysis in high dimension. J Amer Statist Assoc, 2009, 104: 682–693

10 Parikh N, Boyd S. Proximal algorithms. Found Trends Optim, 2013, 1: 123–231

11 Rousseeuw P J. Least median squares regression. J Amer Statist Assoc, 1984, 79: 871–880

12 Rousseeuw P J. Multivariate estimation with high breakdown point. In: Mathematical Statistics and Applications, vol. B. Dordrecht: Reidel, 1985, 283–297

13 She Y, Chen K. Robust reduced rank regression. ArXiv:1509.03938, 2015

14 She Y, Li S, Wu D. Robust orthogonal complement principal component analysis. J Amer Statist Assoc, 2016, 514: 41–64

15 Verboon P, Heiser W J. Resistent lower rank approximation of matrices by iterative majorization. Comput Statist Data Anal, 1994, 18: 457–467

16 Xu H, Caramanis C, Sanghavi S. Robust PCA via outlier pursuit. IEEE Trans Inform Theory, 2012, 58: 3047–3064

17 Zhang L, Shen H, Huang J. Robust regularized singular value decomposition with application to mortality data. Ann Appl Statist, 2013, 7: 1540–1561

18 Zhang T, Lerman G. A novel M-estimator for robust PCA. J Mach Learn Res, 2014, 15: 749–808

19 Zhou Z, Li X, Wright J, et al. Stable principal component pursuit. IEEE Internat Symp Inform Theory, 2010, 41: 1518–1522