Robust Matrix Completion With Column Outliers

Feiping Nie, *Member, IEEE*, Ziheng Li, Zhanxuan Hu, Rong Wang, and Xuelong Li, *Fellow, IEEE*

***Abstract*—Matrix completion, in essence, involves recovering a low-rank matrix from a subset of its entries. Most existing methods for matrix completion neglect two significant issues. First, in several practical applications, such as collaborative filtering, some samples may be corrupted completely. However, most of the robust algorithms consider only the condition that a few components of each column have been arbitrarily contaminated. Second, many real data are not static in nature. Nevertheless, the conventional batch-based matrix completion methods cannot efficiently deal with the out-of-sample, that is, the vector completion problem. In this article, we first provide a novel robust matrix completion model and then develop an efficient optimization method that only requires conducting one time singular value decomposition for a thin matrix per iteration. Furthermore, by exploiting the essence of online matrix completion algorithms, we develop a vector completion model which can help users predict the missing values of out of sample. Numerical comparisons with traditional batch-based and online matrix completion algorithms demonstrate the benefits of the proposed method on streaming data corrupted by column outliers. Moreover, we show that our model can be used to detect outliers from incomplete information. This advantage is validated via numerous experimental results on synthetic and real-world data.**

***Index Terms*—Low rank, matrix completion, nonconvex, vector completion.**

I. Introduction

IN MANY research fields, such as collaborative filtering [1] and computer vision [2]–[5], the data matrix with missing values, in general, brings a lot of challenges to data analysis. One significant tool for solving this problem is matrix completion, which utilizes the known information to estimate the missing values of collected data.

Most of the existing algorithms for matrix completion can be roughly divided into two groups: 1) rank minimization (RM) and 2) bilinear factorization (BF), and both of them aim at finding a low rank matrix $\boldsymbol{L} \in R^{m\times n}$ that is consistent with the input data matrix $\boldsymbol{X} \in R^{m\times n}$ on known entries. Since RM is an NP-hard problem, in general, many algorithms [6]–[11] translate it as a nuclear norm minimization (NNM) problem through convex relaxation. Although NNM is a convex optimization problem with a global optimal solution, the solution provided by it may deviate from the original solution seriously [12]. In order to reduce the deviation, some algorithms replace the nuclear norm by nonconvex relaxations, for instance, the Schatten-p norm [12]–[14]; Capped norm [15]; Truncated Nuclear Norm [16], [17]; and Weighted nuclear norm [18], [19]. Extensive experimental results demonstrate the advantages of nonconvex regularizers over nuclear norm, but the theoretical guarantee over nonconvex relaxations is still missing. In addition, all of these algorithms are required by singular value decompositions (SVD), and in some recent research, such as [20], a partial SVD strategy has been deployed instead of the full SVD. However, the high computational cost of conducting SVDs per iteration still prevents the applications of RM algorithms using convex or nonconvex relaxations from large-scale data.

In order to reduce the computational cost, many algorithms [21]–[26] try to utilize the advantages of the matrix BF. All of them consider the setting that the rank of the desired matrix is known, and reformulate a nonconvex problem for matrix completion by replacing the desired matrix $\boldsymbol{L}$ with $\boldsymbol{UV}^T$, where $\boldsymbol{U} \in R^{m\times r}$ and $\boldsymbol{V} \in R^{n\times r}$ are two low rank matrices. According to the loss function used in the objective, most of the existing models can be grouped into two categories: 1) Frobenius norm-based models [1], [22], [27] and 2) ℓ_1 norm-based models [12], [28]. For the former, the practicability is limited by the fragility to non-Gaussian noise. For the latter, the practicability is limited by the fragility to parameter selection involved in the model or optimization procedure. Moreover, the above methods cannot mitigate the influence of *outliers*.[1]

For noisy data with column outliers, a natural question is *whether we can pursue the location of outliers?* Over the past decades, numerous approaches have been developed for outliers detection. RANSAC [29] is a traditional method for solving this issue, but the time consuming part of it grows exponentially with the subspace dimension. Inspired by robust PCA [30], Xu *et al.* [31] proposed an outliers pursuit method based on NNM and owned theoretical guarantee. In addition, You *et al.* [32] proposed a self-representation method for outlier detection, which combines sparse representation

Manuscript received 21 July 2020; revised 28 January 2021; accepted 28 March 2021. Date of publication 16 June 2021; date of current version 17 October 2022. This work was supported in part by the National Key Research and Development Program of China under Grant 2018AAA0101902; in part by the Natural Science Basic Research Program of Shaanxi Program under Grant 2021JM-071; in part by the National Natural Science Foundation of China under Grant 61936014 and Grant 61772427; and in part by the Fundamental Research Funds for the Central Universities under Grant G2019KY0501. This article was recommended by Associate Editor L. Cheng. *(Corresponding author: Feiping Nie.)*

Feiping Nie, Ziheng Li, Zhanxuan Hu, and Xuelong Li are with the School of Computer Science and School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: feipingnie@gmail.com; liziheng09@gmail.com; huzhanxuan@mail.nwpu.edu.cn; li@nwpu.edu.cn).

Rong Wang is with the School of Cybersecurity and School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an 710072, China (e-mail: wangrong07@tsinghua.org.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2021.3072896.

Digital Object Identifier 10.1109/TCYB.2021.3072896

[1] In this article, outliers refer to the columns corrupted completely by noise.

with random walks on a graph. Nevertheless, all the algorithms mentioned above are only applicable for complete data. Recently, Chen *et al.* [33] proposed a robust matrix completion algorithm that recovers the intrinsic low rank component and selects the outliers from an incomplete matrix simultaneously. However, the method is based on NNM, which makes it impossible to cope with large-scale data. In addition, its performance will degenerate significantly when the input data are corrupted by numerous column outliers. In addition, the setting that input data are nonstatic has been ignored by most of the algorithms mentioned above.

According to this issue, the existing algorithms can be divided into the batch based method and online method. Recently, the research work focuses on the online low-rank matrix completion model, which processes the input data one column (row) at a time. The most significant advantage of the online algorithm is that it can handle the out-of-sample problem, that is, the vector completion problem with new sample. As presented in [34], the core of vector completion is subspace estimation and updates. That is, the missing values of a vector can be estimated when the subspace where it lies is fixed. Incremental SVD (ISVD) [35] is a useful tool for online subspace learning. Given a matrix $\boldsymbol{X} \in R^{m\times n}$ and its SVD $\boldsymbol{X} = \boldsymbol{USV}^T$, one can use ISVD to obtain the SVD of $\hat{\boldsymbol{X}} = [\boldsymbol{X} \quad \boldsymbol{x}_{n+1}]$, that is, $\hat{\boldsymbol{X}} = \hat{\boldsymbol{U}}\hat{\boldsymbol{S}}\hat{\boldsymbol{V}}^T$, where $\boldsymbol{x}_{n+1}$ is a new column vector and can be seen as out of sample. Particularly, the updated subspace is spanned by the columns of $\hat{\boldsymbol{U}}$. Considering that $\boldsymbol{x}_{n+1}$ generally has missing values, Brand [36] generalizes the ISVD to MD-ISVD (ISVD with missing data), and then applies it into recommender systems [37]. Given a subspace $\boldsymbol{U}$[2] and a new vector $\boldsymbol{x}_{n+1}$ with missing values, the projection error over known entries can be measured by

$$\varepsilon(\boldsymbol{U}, \boldsymbol{x}_{n+1}) = \left\|\boldsymbol{U}_\Omega \boldsymbol{w} - \boldsymbol{x}_{(n+1)\Omega}\right\|_2^2 \tag{1}$$

where $\boldsymbol{x}_{(n+1)\Omega}$ is a subvector consisting of the known entries of $\boldsymbol{x}$, $\boldsymbol{U}_\Omega$ is the corresponding submatrix of $\boldsymbol{U}$, and $\boldsymbol{w} = \boldsymbol{U}_\Omega^+ \boldsymbol{x}_{(n+1)\Omega}$[3] is a weight vector. Inspired by the online manifold learning, Balzano *et al.* [38] proposed a GROUSE approach, which aims to find a matrix $\hat{\boldsymbol{U}}$ with orthonormal columns to reduce the projection error (1). In particular, MD-ISVD and GROUSE are equivalent when a particular choice of step size of GROUSE is selected [39]. Both MD-ISVD and GROUSE are based on the ℓ_2 norm of vector, which makes them very sensitive to non-Gaussian noise. To improve the robustness of GROUSE, He *et al.* [40] developed a robust subspace tracking algorithm, namely, GRASTA, which uses the following model to estimate the weight vector $\boldsymbol{w}$ and sparse noisy vector $\boldsymbol{e}$, simultaneously:

$$\min_{\boldsymbol{w},\boldsymbol{e}} \ \|\boldsymbol{e}\|_1 \quad \text{s.t.} \quad \boldsymbol{U}_\Omega \boldsymbol{w} + \boldsymbol{e} = \boldsymbol{x}_{(n+1)\Omega}. \tag{2}$$

Such a problem is solved by ADMM. Recently, an efficient method OR-RPCA based on stochastic optimization for online subspace learning has been developed [41], which is constructed for RPCA and can be easily generalized to matrix completion. In addition to the methods mentioned above, many efforts have been made for online subspace learning [42]–[46]. Nevertheless, for processing the data one column at a time, most of them will return arbitrarily poor results when input data are corrupted by outliers.

In a nutshell, the practicability of batch-based approaches is limited by its inefficiency to streaming data, and the practicability of online methods is limited by its susceptibility to column outliers. This issue leads to the second question: *whether we can complement the drawbacks of these two approaches by combining the advantages of them?*

In this article, we first provide a BF-based robust matrix completion model, which utilizes the $\ell_{2,1}$-norm to improve the robustness of the algorithm to column outliers. Then, we will generalize it to a vector completion model, which can help users cope with the out-of-sample problem. Hence, our method can be regarded as an "*online*" method (although it does not process the input data, one at a time). For conciseness, we call it "ORMC" (*online robust matrix completion*). Moreover, the similar idea can be applied into other batch-based algorithms. Ignoring the difference in coping with training data, we will present in Section III-C that all *batch-based* approaches can be implemented as *online* methods by conducting one time SVD for recovered low rank matrix $\boldsymbol{L}$. If the precision of $\boldsymbol{L}$ is high enough, the subspace spanned by the columns of $\boldsymbol{U}$, where $\boldsymbol{USV}^T = \boldsymbol{L}$, could be used to estimate the missing values of a new vector. The main contributions of this article are summarized as follows.

1) We propose a novel robust matrix completion model, which is based on $\ell_{2,1}$-norm and has three advantages: a) providing an accurate prediction for training data with column outliers; b) tracking the location of potential *outliers*; and c) learning a low-dimensional subspace for estimating the missing values of out of sample.
2) We develop an efficient optimization method to solve the nonconvex and nonsmooth models. Moreover, for a matrix $\boldsymbol{L} \in R^{m\times n}$ with rank r, the constructed method only requires conducting one time SVD for $m \times r$ instead of $m \times n$ matrix per iteration, which reduces the computational complexity from $\mathcal{O}(mn^2)$ to $\mathcal{O}(mr^2)$, $r \ll n$.
3) The adaptive weighted matrix D obtained by our method can help users to detect the location of outliers exactly from incomplete information. To the best of our knowledge, there are no matrix completion algorithms developed for dealing with this issue.
4) We provide a vector completion model that can help users estimate the missing values of out of sample. Furthermore, we show that all batch-based matrix completion methods can be implemented to deal with this issue.

The remainder of this article is organized as follows. We begin by presenting the robust matrix completion method in Section II. In Section III, we first describe the implementation

[2]In rest of this article, the orthonormal matrix $\boldsymbol{U}$ refers to the subspace spanned by its columns.

[3]Here, $\boldsymbol{A}^+$ is the Moore–Penrose pseudoinverse matrix of $\boldsymbol{A}$.

details of ORMC in Section III-A, then show the motivation of outlier detection in Section III-B, and finally, a vector completion model will be provided in Section III-C. The relevant theoretical analysis and convergence analysis are presented in Section IV. Numerical experimental results with respect to (w.r.t.) matrix completion, vector completion, and outlier detection are provided in Section V. Finally, we conclude this article and discuss the potential research directions for future work in Section VI.

II. Proposed Algorithm

A. Notation

Before continuing, we will give a brief summary of the notations used throughout this article. For a matrix $\boldsymbol{X} \in R^{m\times n}$, we denote its (i,j)th entry by $\boldsymbol{X}_{i,j}$, and the ith column and row of $\boldsymbol{X}$ by $\boldsymbol{x}_i$ and $\boldsymbol{x}^i$, respectively. We denote the set of indices of known entries and unknown entries by Ω and Ω^c, respectively, and let P_Ω be the projection operator that projects a matrix $\boldsymbol{X}$ to $P_\Omega(\boldsymbol{X})$, which is defined as

$$P_\Omega(\boldsymbol{X})_{ij} = \begin{cases} X_{ij}, & \text{if } (i,j) \in \Omega \\ 0, & \text{if } (i,j) \in \Omega^c. \end{cases} \tag{3}$$

For a vector $\boldsymbol{x}_i$, the corresponding projection operator is denoted by $P_{i\Omega}$. In addition, $\boldsymbol{x}_{i\Omega} = P_{i\Omega}(\boldsymbol{x}_i)$ and $\boldsymbol{x}_{i\Omega^c} = P_{i\Omega^c}(\boldsymbol{x}_i)$, respectively.

Besides, for the column $\boldsymbol{x}_i$ of $\boldsymbol{X}$, we suppose that $\boldsymbol{x}_i$ belongs to $\boldsymbol{X}_I$ or $\boldsymbol{X}_O$, where $\boldsymbol{X}_I$ is a matrix stacked by the inliers of $\boldsymbol{X}$ and $\boldsymbol{X}_O$ is a matrix stacked by the potential outliers. For a vector $\boldsymbol{x}$, we denote its ℓ_2 norm by $\|\boldsymbol{x}\|_2 = (\boldsymbol{x}^T\boldsymbol{x})^{1/2}$, where $\boldsymbol{x}^T$ is the transpose of $\boldsymbol{x}$. For a matrix $\boldsymbol{X}$, we denote its Frobenius norm by $\|\boldsymbol{X}\|_F = (\mathrm{tr}(\boldsymbol{X}\boldsymbol{X}^T))^{1/2}$, and the $\ell_{2,1}$ norm by $\sum_{i=1}^n \|\boldsymbol{x}_i\|_2$. In particular, $\mathcal{U}$ denotes a space spanned by the columns of $\boldsymbol{U}$.

B. Proposed Robust Matrix Completion Model

Some representative robust PCA methods have been generalized to robust matrix completion, such as IALM [9], RegL1-ALM [28], for coping with sparse noise, outliers pursuit robust matrix completion (OPRMC) [33], and for coping with column outliers. By introducing a column-sparse matrix $\boldsymbol{C}$, OPRMC reformulates the matrix completion problem as follows:

$$\min_{\boldsymbol{L},\boldsymbol{C}} \ \|\boldsymbol{L}\|_* + \lambda\|\boldsymbol{C}\|_{2,1} \quad \text{s.t. } P_\Omega(\boldsymbol{L}+\boldsymbol{C}) = P_\Omega(\boldsymbol{X}). \tag{4}$$

Solving such a problem requires conducting SVD for a full matrix at each iteration, which prevents it from coping with large scale data.

Inspired by the success of BF-based methods, we, in ORMC, translate the matrix $\boldsymbol{L}$ as a product of two low rank matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, that is, $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{V}^T$. In addition, considering the setting that the structure of the original matrix is generally approximately low rank rather than exactly low rank, we choose to optimize the unknown entries directly. The effectiveness of this strategy for decreasing the risk of overfitting has been proved in [47]. Specifically, the robust matrix completion model proposed in this article is

$$\min_{\boldsymbol{U}\in R^{m\times r},\boldsymbol{V}\in R^{n\times r},\boldsymbol{X}_{\Omega^c}} \ \left\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^T\right\|_{2,1} \tag{5}$$

where $\boldsymbol{X}_{\Omega^c}$ are the unknown entries of $\boldsymbol{X}$. Furthermore, for any invertible matrix $\boldsymbol{M}$, $\boldsymbol{U}\boldsymbol{V}^T = (\boldsymbol{U}\boldsymbol{M})(\boldsymbol{M}^{-1}\boldsymbol{V}^T)$; hence, we denote an identity matrix by $\boldsymbol{I}$ to shrink the solution space. Then, the problem (5) becomes

$$\min_{\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I},\boldsymbol{V},\boldsymbol{X}_{\Omega^c}} \ \left\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^T\right\|_{2,1} \tag{6}$$

and it can be rewritten as

$$\min_{\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I},\boldsymbol{V},\boldsymbol{X}_{\Omega^c}} \ \sum_{i=1}^n \left\|\boldsymbol{x}_i - \boldsymbol{U}(\boldsymbol{v}^i)^T\right\|_2. \tag{7}$$

Actually, the problem (6) can be solved via an IALM method just like RegL1-ALM [28], while it needs to introduce an auxiliary variable $\boldsymbol{Y}$ and select numerous reasonable parameters. In the next section, we will provide an efficient reweighted method to solve it. The most attractive advantage of this method is that no additional parameters are required, while the weighted matrix $\boldsymbol{D}$ obtained by the algorithm can help users to detect outliers.

III. Details of Proposed Method

In this section, we first provide the implementation details of solving (7), and then give the motivation of outlier detection. Finally, we develop a vector completion model for solving the out-of-sample problem.

A. Optimization of ORMC

The problem (7) can be rewritten as

$$\min_{\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I},\boldsymbol{V},\boldsymbol{X}_{\Omega^c}} \ \Sigma_{i=1}^n f(g(\boldsymbol{v}_i, \boldsymbol{x}_{i\Omega^c}, \boldsymbol{U})) \tag{8}$$

where $f(x) = x^{(1/2)}$, $g(\boldsymbol{v}_i, \boldsymbol{x}_{i\Omega^c}, \boldsymbol{U}) = \|\boldsymbol{x}_i - \boldsymbol{U}(\boldsymbol{v}^i)^T\|_2^2$.[4] The major blocks in solving this problem include: 1) $f(x)$ is a concave function and 2) $g(x)$ has multiple variables. Here, we utilize an efficient reweighted method to solve it, which has been used to cope with the RPCA problem [48]. More details about the reweighted method can be find in [49]. Specifically, in the $(k+1)$th iteration, we aim to solve the following subproblem:

$$\min_{\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I},\boldsymbol{V},\boldsymbol{X}_{\Omega^c}} \ \sum_{i=1}^n d_i^k \left\|\boldsymbol{x}_i - \boldsymbol{U}\left(\boldsymbol{v}^i\right)^T\right\|_2^2 \tag{9}$$

where

$$d_i^k = \frac{1}{2\left\|\boldsymbol{x}_i^k - \boldsymbol{U}^k((\boldsymbol{v}^i)^k)^T\right\|_2} \tag{10}$$

can be seen as the weight of $\boldsymbol{x}_i$ in the $(k+1)$th iteration. Furthermore, (9) can be written as

$$\min_{\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I},\boldsymbol{V},\boldsymbol{X}_{\Omega^c}} \ \left\|\left(\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^T\right)\boldsymbol{D}^k\right\|_F^2 \tag{11}$$

[4]Note that $\|\boldsymbol{x}_i - \boldsymbol{U}(\boldsymbol{v}^i)^T\|_2^2 + \varepsilon^2$ is generally used in practice, where ε is *eps* in MATLAB.

where $\boldsymbol{D}^k$ is a diagonal matrix whose ith diagonal element is $(d_i^k)^{1/2}$. For brevity, we remove the superscript of $\boldsymbol{D}$. As the weighted matrix $\boldsymbol{D}$ is fixed, (11) can be rewritten as

$$\min_{\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I},\hat{\boldsymbol{V}},\hat{\boldsymbol{X}}_{\Omega^c}} \left\|\hat{\boldsymbol{X}}-\boldsymbol{U}\hat{\boldsymbol{V}}^{\boldsymbol{T}}\right\|_F^2 \tag{12}$$

where $\boldsymbol{D}$ is a diagonal matrix, and $\hat{\boldsymbol{V}}^T=\boldsymbol{V}^T\boldsymbol{D}$, $\hat{\boldsymbol{X}}=\boldsymbol{X}\boldsymbol{D}$, and $\hat{\boldsymbol{X}}_{\Omega^c}=(\boldsymbol{X}\boldsymbol{D})_{\Omega^c}$. For problem (12) that has three block variables, we cannot obtain the optimal solutions of them directly. A natural idea is using an alternating minimization (AM) method to update each one while fix residuals. Specifically, in the $(k+1)$th iteration

Update $\boldsymbol{U}$: Fixing $\hat{\boldsymbol{V}},\hat{\boldsymbol{X}}_{\Omega^c}$, we update $\boldsymbol{U}$ by solving

$$\min_{\boldsymbol{U}^T\boldsymbol{U}=\boldsymbol{I}} \left\|\hat{\boldsymbol{X}}^k-\boldsymbol{U}\left(\hat{\boldsymbol{V}}^k\right)^T\right\|_F^2. \tag{13}$$

This is the famous orthogonal procrustes problem (OPP), and the optimal solution is provided by the SVD of matrix $\boldsymbol{M}=\hat{\boldsymbol{X}}^k\hat{\boldsymbol{V}}^k$. Suppose

$$\boldsymbol{P}\boldsymbol{\Sigma}\boldsymbol{Q}^T=\boldsymbol{M} \tag{14}$$

is the SVD of $\boldsymbol{M}$. The variable $\boldsymbol{U}$ is updated by

$$\boldsymbol{U}^{k+1}=\boldsymbol{P}\boldsymbol{Q}^T. \tag{15}$$

Update $\boldsymbol{V}$: For variable $\boldsymbol{V}$, we first obtain the optimal solution of $\hat{\boldsymbol{V}}$ by solving

$$\min_{\hat{\boldsymbol{V}}} \left\|\hat{\boldsymbol{X}}^k-\boldsymbol{U}^{k+1}\hat{\boldsymbol{V}}^T\right\|_F^2. \tag{16}$$

Since $\boldsymbol{U}^{k+1}$ is an orthogonal matrix, the optimal solution of (16) is

$$\hat{\boldsymbol{V}}^{k+1}=\left(\hat{\boldsymbol{X}}^k\right)^T\boldsymbol{U}^{k+1}. \tag{17}$$

Naturally, the variable $\boldsymbol{V}$ can be updated by

$$\boldsymbol{V}^{k+1}=\left(\left(\hat{\boldsymbol{V}}^{k+1}\right)^T\boldsymbol{D}^{-1}\right)^T=(\boldsymbol{X})^T\boldsymbol{U}^{k+1} \tag{18}$$

where $\boldsymbol{D}^{-1}$ is the inverse matrix of $\boldsymbol{D}$.

Update $\boldsymbol{X}_{\Omega^c}$: For $\boldsymbol{X}_{\Omega^c}$, we first calculate the optimal solution of $\hat{\boldsymbol{X}}_{\Omega^c}$ by solving

$$\min_{\hat{\boldsymbol{X}}_{\Omega^c}} \left\|\hat{\boldsymbol{X}}-\boldsymbol{U}^{k+1}\left(\hat{\boldsymbol{V}}^{k+1}\right)^T\right\|_F^2. \tag{19}$$

Obviously

$$\hat{\boldsymbol{X}}_{\Omega^c}^{k+1}=\boldsymbol{T}_{\Omega^c} \tag{20}$$

where $\boldsymbol{T}=\boldsymbol{U}^{k+1}(\hat{\boldsymbol{V}}^{k+1})^T$. Furthermore, $\boldsymbol{X}_{\Omega^c}$ can be updated by

$$\boldsymbol{X}_{\Omega^c}^{k+1}=\left(\hat{\boldsymbol{X}}^{k+1}\boldsymbol{D}^{-1}\right)_{\Omega^c}. \tag{21}$$

In fact, (21) is also the optimal solution of

$$\min_{\boldsymbol{X}_{\Omega^c}} \left\|\boldsymbol{X}-\boldsymbol{U}^{k+1}\left(\boldsymbol{V}^{k+1}\right)^T\right\|_{2,1}. \tag{22}$$

Algorithm 1 Solving ORMC via Reweighted Method

Input: the matrix $\boldsymbol{X}$ with missing values, the rank r of desired low rank matrix, the maximum number of iterations $N=1000$.

Initialization: Initialize all elements of $\boldsymbol{X}_{\Omega^c}$ by 0. Let $\boldsymbol{V}_0=\boldsymbol{V}_1\boldsymbol{S}_1$, where $\boldsymbol{U}_1\boldsymbol{S}_1\boldsymbol{V}_1^T=svd(\boldsymbol{X})$. In addition, the weighted matrix $\boldsymbol{D}_0=\boldsymbol{I}$ is an identity matrix.

for $k=1:N$ **do**

- Update $\boldsymbol{U}$ by Eq. (15);
- Update $\boldsymbol{V}$ by Eq. (18);
- Update $\boldsymbol{X}_{\Omega^c}$ by (21);
- Update the ith diagonal element of weighted matrix $\boldsymbol{D}$ by $d_i^{\frac{1}{2}}$, where d_i is calculated by (10).
- Convergence checking .

end for

Output: the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$, the weighted matrix $\boldsymbol{D}$ and the recovered matrix $\boldsymbol{X}$.

The entire procedure is outlined in Algorithm 1, where the convergence criteria are

$$\frac{|L_k-L_{k+1}|}{L_k}\le 10^{-6} \tag{23}$$

where $L_k=\|\boldsymbol{X}^k-\boldsymbol{U}^k(\boldsymbol{V}^k)^T\|_{2,1}$. In Section IV, we will prove that Algorithm 1 will decrease the objective value of (6) per iteration.

Complexity Analysis: We now discuss the time complexity of the proposed method ORMC. For Algorithm 1, in addition to simple matrix multiplications, the main time consuming is updating $\boldsymbol{U}$, which requires performing SVD for an $m\times r$ thin matrix, and the computational complexity is $\mathcal{O}(mr^2)$, which is far less than the NNM-based method even when partial SVD with $\mathcal{O}(mnr)$ is used.

B. Outlier Detection

Given a subspace $\mathcal{U}$, the projection of a complete vector $\boldsymbol{x}$ onto $\mathcal{U}$ is denoted by $\boldsymbol{U}\boldsymbol{v}$, and the corresponding reconstruction error is denoted by

$$e=\|\boldsymbol{x}-\boldsymbol{U}\boldsymbol{v}\|_2^2. \tag{24}$$

Suppose $\boldsymbol{U}$ is an orthogonal matrix, we have $\boldsymbol{v}=\boldsymbol{U}^T\boldsymbol{x}$. Furthermore, we achieve

$$e=\left\|(\boldsymbol{I}-\boldsymbol{U}\boldsymbol{U}^T)\boldsymbol{x}\right\|_2^2. \tag{25}$$

A reasonable assumption is that *Outlier* has larger reconstruction error comparing with *Inlier*. Before generalizing this assumption to the case that $\boldsymbol{x}_i$ is incomplete, we give a theorem presented by [50], which involves the following two significant definitions.

Definition 1: For an r-dimensional subspace $\mathcal{U}\in R^m$ spanned by the columns of the orthogonal matrix $\boldsymbol{U}$, its coherence is defined by

$$\mu(\mathcal{U}):=\frac{m}{r}\max_j\left\|P_{\mathcal{U}}e_j\right\|_2^2 \tag{26}$$

where $P_{\mathcal{U}}=\boldsymbol{U}\boldsymbol{U}^T$ represents the projection operator onto subspace $\mathcal{U}$, and $\{e_j\}$ represents a standard basis.

Definition 2: For a vector $\boldsymbol{x} = \boldsymbol{x}_{\mathcal{U}} + \boldsymbol{x}_{\mathcal{U}^\perp}$, where $\boldsymbol{x}_{\mathcal{U}} \in \mathcal{U}$ and $\boldsymbol{x}_{\mathcal{U}^\perp} \in \mathcal{U}^\perp$, we let $\mu(\boldsymbol{x})$ denote the coherence of subspace spanned by $\boldsymbol{x}$. Exactly

$$\mu(\boldsymbol{x}) = \frac{m\|\boldsymbol{x}\|_\infty^2}{\|\boldsymbol{x}\|_2^2} \tag{27}$$

where $\|\boldsymbol{x}\|_\infty$ is the absolute maximum value of elements in $\boldsymbol{x}$.

Suppose $\boldsymbol{x}_\Omega \in R^m$ is a vector with k nonzero entries sampled from $\boldsymbol{x}$, $P_{\mathcal{U}_\Omega} = \boldsymbol{U}_\Omega \boldsymbol{U}_\Omega^T$ is a projection operator, where $\boldsymbol{U}_\Omega$ is a submatrix stacked by the rows of $\boldsymbol{U}$ whose indices correspond to the known entries of $\boldsymbol{x}$. The following theorem [50] holds.

Theorem 1: Let $\delta > 0$ and the number of known entries be $k \geq 8/3m\mu(\mathcal{U})\log(2m/\delta)$. Then, the following two inequality holds with probability at least $1 - 4\delta$:

$$\frac{k(1-\alpha) - m\mu(\mathcal{U})\frac{(1+\beta)^2}{1-\gamma}}{n}\|\boldsymbol{x} - P_{\mathcal{U}}\boldsymbol{x}\|_2^2 \leq \left\|\boldsymbol{x}_\Omega - P_{\mathcal{U}_\Omega}\boldsymbol{x}_\Omega\right\|_2^2 \tag{28}$$

and

$$\left\|\boldsymbol{x}_\Omega - P_{\mathcal{U}_\Omega}\boldsymbol{x}_\Omega\right\|_2^2 \leq (1+\alpha)\frac{k}{m}\|\boldsymbol{x} - P_{\mathcal{U}}\boldsymbol{x}\|_2^2 \tag{29}$$

where $\alpha = \sqrt{([2\mu(\boldsymbol{x}_{\mathcal{U}^\perp})^2]/m)\log(1/\delta)}$, $\beta = \sqrt{2\mu(\boldsymbol{x}_{\mathcal{U}^\perp})\log(1/\delta)}$, and $\gamma = \sqrt{([8m\mu(\mathcal{U})]/3k)\log(2m/\delta)}$. This theorem has been proved in [50, Sec. 3.6.2]. Theorem 1 demonstrates that the magnitude of residual of $\boldsymbol{x}_\Omega$ is close to the residual of $\boldsymbol{x}$ when scaled by the fraction of known entries. That is, $e_\Omega = \|\boldsymbol{x}_\Omega - P_{\mathcal{U}_\Omega}\boldsymbol{x}_\Omega\|_2^2$ can be seen as a replacement of e to some extent.

Now, let us return to Algorithm 1. We update d_i by (10) at the $(k+1)$th iteration. According to (18), we have

$$d^{k+1} = \frac{1}{2\left\|\left(\boldsymbol{I} - \boldsymbol{U}^{k+1}\left(\boldsymbol{U}^{k+1}\right)^T\right)\boldsymbol{x}^{k+1}\right\|_2}. \tag{30}$$

Here, we remove the subscript w.r.t. the indices of columns for simplifying description. Observing (20) and (21), we can find that the entries of $\boldsymbol{E}_{\Omega^c}^{k+1}$ are 0, where

$$\boldsymbol{E}^{k+1} = \boldsymbol{X}^{k+1} - \boldsymbol{U}^{k+1}\left(\boldsymbol{V}^{k+1}\right)^T. \tag{31}$$

Naturally, the weight d can be rewritten as

$$d^{k+1} = \frac{1}{2\left\|\boldsymbol{x}_\Omega^{k+1} - P_{\mathcal{U}_\Omega}^{k+1}\boldsymbol{x}_\Omega^{k+1}\right\|_2}. \tag{32}$$

According to Theorem 1, we know that the value of d^{k+1} can be used to measure the distance between $\boldsymbol{x}$ and subspace $\mathcal{U}$. That is, we can regard the vector $\boldsymbol{x}_i$ corresponding to a small value of d_i as a column outlier. [Naturally, we want its loss has smaller effect in optimizing problem (9).]. Hence, the locations of potential outliers can be determined through the weighted matrix $\boldsymbol{D}$. Extensive experiments on generated data and real data will prove the validity of this assumption.

C. Vector Completion

In this section we will investigate the vector completion problem. Actually, as shown in [34], the claim that batch-based approaches cannot deal with the out-of-sample is not rigorous, because all vector completion problem is equivalent to a subspace learning problem. That is, determining the subspace where the out of sample lies is the core of vector completion, and it is the reason why the online method can deal with the out-of-sample problem directly. For the batch-based method, in order to obtain the subspace of the incomplete matrix lies, we conduct one time SVD to the recovered low rank matrix $\boldsymbol{L}$, $\boldsymbol{USV}^T = svd_r(\boldsymbol{L})$. The subspace spanned by the columns of $\boldsymbol{U}$ could be used to estimate the missing values of out of sample if the precision of $\boldsymbol{L}$ is high enough. Next, we will present the details of the proposed vector completion method, and it is a generalization of ORMC.

Proposed Vector Completion Model: The subspace $\boldsymbol{U}$ that the data matrix lies is known. For a new incomplete vector that lies in the same subspace, the missing values of $\boldsymbol{x}$ and its coordinate vector $\boldsymbol{v}$ can be estimated by solving the following problem:

$$\min_{\boldsymbol{v}, \boldsymbol{x}_{\Omega^c}} \|\boldsymbol{x} - \boldsymbol{U}\boldsymbol{v}\|_2^2. \tag{33}$$

Taking the derivative w.r.t $\boldsymbol{v}$ and setting it to 0, we have

$$\boldsymbol{v} = \boldsymbol{U}^T\boldsymbol{x}. \tag{34}$$

Substituting $\boldsymbol{v}$ into (33), the problem can be recast as

$$\min_{\boldsymbol{x}_{\Omega^c}} \left\|\left(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^T\right)\boldsymbol{x}\right\|_2^2. \tag{35}$$

Defining $\boldsymbol{W} = (\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}^T)$, where $\boldsymbol{W} \in R^{m\times m}$, this problem can be reformulated as

$$\min_{\boldsymbol{x}_{\Omega^c}} \|(\boldsymbol{W}_{\Omega^c}, \boldsymbol{W}_\Omega)(\boldsymbol{x}_{\Omega^c}; \boldsymbol{x}_\Omega)\|_2^2 \tag{36}$$

where $\boldsymbol{W}_\Omega$ is a submatrix stacked by the columns of $\boldsymbol{W}$ whose indices correspond to Ω. Furthermore, problem (36) can be rewritten as

$$\min_{\boldsymbol{x}_{\Omega^c}} \|\boldsymbol{W}_{\Omega^c}\boldsymbol{x}_{\Omega^c} + \boldsymbol{W}_\Omega\boldsymbol{x}_\Omega\|_2^2. \tag{37}$$

Obviously, this is a least squares problem (LSP). Then, we can achieve the optimal solution by

$$\boldsymbol{x}_{\Omega^c} = \left(\boldsymbol{W}_{\Omega^c}^T\boldsymbol{W}_{\Omega^c}\right)^{-1}\boldsymbol{W}_{\Omega^c}^T\boldsymbol{c} \tag{38}$$

where $\boldsymbol{c} = -\boldsymbol{W}_\Omega\boldsymbol{x}_\Omega$. In addition, we require that $k \geq r$, where k is the number of known entries. Then, the missing values of the out-of-sample vector $\boldsymbol{x}$ can be estimated by utilizing the low rank subspace $\boldsymbol{U}$ learned from training data.

IV. Theoretical Analysis

The reweighted method has been widely used in numerous algorithms [51], [52], and the authors have proved that the final solution that provided the reweighted method is a local optimal solution of the original problem. Nevertheless, this conclusion is not suitable for ORMC, because it contains multiple variables. To the best of our knowledge, there is no

work that has provided analogous conclusion for the nonconvex problem with multiple variables. Hence, in this article, we only prove that Algorithm 1 will make the objective value of (7) monotonically decrease. Before continuing our prove, we state the proof about the following lemma.

Lemma 1: For any nonzero vectors $\boldsymbol{a}, \boldsymbol{b} \in R^n$, we know that the following inequality holds:

$$\|\boldsymbol{a}\|_2 - \frac{\|\boldsymbol{a}\|_2^2}{2\|\boldsymbol{b}\|_2} \leq \|\boldsymbol{b}\|_2 - \frac{\|\boldsymbol{b}\|_2^2}{2\|\boldsymbol{b}\|_2}. \tag{39}$$

Proof: For any values $a > 0$ and $b > 0$, we have $0 \leq (a-b)^2$. Furthermore, we can obtain

$$0 \leq a^2 - 2ab + b^2 \Rightarrow 2ab - a^2 \leq 2b^2 - b^2. \tag{40}$$

Since b is larger than 0, we have

$$a - \frac{a^2}{2b} \leq b - \frac{b^2}{2b}. \tag{41}$$

Obviously, when $\boldsymbol{a}$ and $\boldsymbol{b}$ are nonzero vectors, $\|\boldsymbol{a}||_2 > 0$ and $\|\boldsymbol{b}\|_2 > 0$. Furthermore, we can prove that Lemma 1 is consistent. ■

Theorem 2: In each iteration, the objective value of (7) is monotonically decreasing until the algorithm converges.

Proof: We denote the updated variables $\boldsymbol{V}$, $\boldsymbol{U}$, and $\boldsymbol{X}$ per iteration in Algorithm 1 by $\boldsymbol{U}_*$, $\boldsymbol{V}_*$, and $\boldsymbol{X}_*$, respectively. Here, we substitute $\boldsymbol{X}_{\Omega^c}$ by $\boldsymbol{X}$. Since $\boldsymbol{U}_*$, $\boldsymbol{V}_*$, and $\boldsymbol{X}_*$ are the optimal solution of problem (9), we have

$$\sum_{i=1}^{n} d_i \left\| \boldsymbol{x}_{i*} - \boldsymbol{U}_* \left(\boldsymbol{v}_*^i\right)^T \right\|_2^2 \leq \sum_{i=1}^{n} d_i \left\| \boldsymbol{x}_i - \boldsymbol{U} \left(\boldsymbol{v}^i\right)^T \right\|_2^2. \tag{42}$$

Substituting of d_i into (42), we can obtain

$$\sum_{i=1}^{n} \frac{\left\| \boldsymbol{x}_{i*} - \boldsymbol{U}_* \left(\boldsymbol{v}_*^i\right)^T \right\|_2^2}{2\left\| \boldsymbol{x}_i - \boldsymbol{U} \left(\boldsymbol{v}^i\right)^T \right\|_2} \leq \sum_{i=1}^{n} \frac{\left\| \boldsymbol{x}_i - \boldsymbol{U} \left(\boldsymbol{v}^i\right)^T \right\|_2^2}{2\left\| \boldsymbol{x}_i - \boldsymbol{U} \left(\boldsymbol{v}^i\right)^T \right\|_2}. \tag{43}$$

In addition, utilizing the conclusion of Lemma 1, we have

$$\sum_{i=1}^{n} \left(\left\| \boldsymbol{x}_{i*} - \boldsymbol{U}_* \left(\boldsymbol{v}_*^i\right)^T \right\|_2 - \frac{\left\| \boldsymbol{x}_{i*} - \boldsymbol{U}_* \left(\boldsymbol{v}_*^i\right)^T \right\|_2^2}{2\left\| \boldsymbol{x}_i - \boldsymbol{U} \left(\boldsymbol{v}^i\right)^T \right\|_2} \right) \leq \sum_{i=1}^{n} \left(\left\| \boldsymbol{x}_i - \boldsymbol{U} \left(\boldsymbol{v}^i\right)^T \right\|_2 - \frac{\left\| \boldsymbol{x}_i - \boldsymbol{U} \left(\boldsymbol{v}^i\right)^T \right\|_2^2}{2\left\| \boldsymbol{x}_i - \boldsymbol{U} \left(\boldsymbol{v}^i\right)^T \right\|_2} \right). \tag{44}$$

Summing (43) and (44), we obtain

$$\sum_{i=1}^{n} \left\| \boldsymbol{x}_{i*} - \boldsymbol{U}_* \left(\boldsymbol{v}_*^i\right)^T \right\|_2 \leq \sum_{i=1}^{n} \left\| \boldsymbol{x}_i - \boldsymbol{U} \left(\boldsymbol{v}^i\right)^T \right\|_2. \tag{45}$$

It can be rewritten as

$$\left\| \boldsymbol{X}_* - \boldsymbol{U}_* \boldsymbol{V}_*^T \right\|_{2,1} \leq \left\| \boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^T \right\|_{2,1}. \tag{46}$$

It is obvious that the objective of problem (7) has a lower bound 0, and (46) demonstrates that its objective value is monotonically decreasing. So, the objective function value of (7) is convergent. ■

TABLE I
DETAILS OF ALTERNATIVE METHODS

Algorithm	Model form	Noise form	Optimization
RegL1-ALM	Batch	Sparse noise	IALM
OPRMC	Batch	Column outliers	IALM
GRASTA	Online	Sparse noise	IALM
LMafit	Batch	Gausse noise	SOR
ScaledASD	Batch	Gausse noise	ASD
RBF	Batch	Sparse noise	ADMM
FaNCL	Batch	Sparse noise	GSVT
ORMC (ours)	Batch	Column outliers	RW

*Attention: 'IALM' represents the inexact Augmented Lagrange Multiplier; 'SOR' represents a nonlinear successive over-relaxation method; ASD represents the alternating steepest descent method; 'ADMM' represents the alternating direction method of multipliers; 'GSVT' represents generalized singular value thresholding; 'RW' represents the re-weighted method.

V. EXPERIMENTS

In this section, we provide extensive experimental results on several matrix completion methods including ORMC proposed in this article. In Section V-A, the comparison mainly involves the matrix completion problem. In Section V-B, we utilize ORMC and two generalized batch-based methods to cope with the vector completion problem. In Section V-C, the proposed method will be used to estimate outliers from incomplete information. The code of ORMC and implementation details can be found in https://github.com/sudalvxin/TKDE.git, and the codes of other methods are downloaded from the authors' homepage.

The rank r of the matrix we want to recover is the only parameter needed to be tuned in ORMC, and it is necessary for all BF-based algorithms. For alternative methods, the parameters are provided by original paper or authors' suggestion. All experiments were performed on a Lenovo 5060 Workstation with four Intel i5-4590 Processors and 16 GB of RAM.

A. Experiments on Matrix Completion

For matrix completion, we compare the performance of ORMC with six state-of-the-art matrix completion algorithms including the following.

1) *RegL1-ALM*[5] *[28]:* A robust batch-based algorithm for matrix completion, which considers the setting that input data are corrupted by sparse noise.
2) *GRASTA*[6] *[53]:* A robust online matrix completion algorithm that aims to learn a low-dimensional subspace and utilizes it to estimate the missing values of input data.
3) *LMafit*[7] *[22]:* A representative matrix completion method based on BF, which introduces a new matrix $\boldsymbol{L}$ to reduce the ovefitting risk of the algorithm.
4) *SASD*[8] *[54]:* A representative method based on BF. The constructed model is solved by the scaled alternating steepest descent (SASD) method.

[5]https://sites.google.com/site/yinqiangzheng/
[6]https://sites.google.com/site/hejunzz/grasta
[7]http://lmafit.blogs.rice.edu/
[8]http://www.sdspeople.fudan.edu.cn/weike/publications.html

TABLE II
EXTENSIVE COMPARISONS ON MATRIX COMPLETION. "RE" REPRESENTS THE RELATIVE ERROR. THIS EXPERIMENT FOLLOWS CASE 1 WITH $n = 500$ AND $r = 5$, WHERE c DENOTES THE NUMBER OF OUTLIERS AND p DENOTES THE PERCENTAGE OF KNOWN ENTRIES

Algorithms		Lmafit	SASD	GRASTA	FaNCL	RegL1-ALM	RBF-L1	OPRMC	ORMC
c=0	p=0.5	4.9417E-07	1.0110E-05	2.7035E-10	3.9281E-05	1.1815E-08	1.8397E-05	5.3034E-05	**5.1787E-16**
	p=0.7	8.3282E-07	8.4807E-06	1.2008E-10	1.2819E-05	9.7206E-09	1.0177E-05	6.5096E-05	**4.5303E-16**
c=50	p=0.5	1.2293E-01	1.2257E-01	1.0713E-03	1.2309E-01	7.5466E-07	1.4404E-05	2.9336E-05	**1.9373E-07**
	p=0.7	1.2198E-01	1.2129E-01	8.3201E-04	1.2161E-01	1.3458E-07	1.0020E-05	2.4328E-05	**3.8477E-08**
c=100	p=0.5	1.2985E-01	1.2977E-01	1.6119E-03	1.1928E-01	7.5249E-07	1.6928E-05	3.8091E-05	**3.9196E-07**
	p=0.7	1.2361E-01	1.2329E-01	1.2185E-03	1.2366E-01	1.5817E-07	1.3108E-05	2.7342E-05	**7.2035E-08**

TABLE III
EXTENSIVE COMPARISONS ON MATRIX COMPLETION. "RE" REPRESENTS THE RELATIVE ERROR. THIS EXPERIMENT FOLLOWS CASE 2 WITH $c = 0.2n$ AND $r = 5$, WHERE n DENOTES THE NUMBER OF SAMPLES AND p DENOTES THE PERCENTAGE OF KNOWN ENTRIES

Algorithms		Lmafit	SASD	GRASTA	FaNCL	RegL1-ALM	RBF-L1	OPRMC	ORMC
500X500	p=0.5	1.2311E-01	1.2308E-01	1.5419E-03	1.2278E-01	8.2586E-07	1.5583E-05	3.7851E-05	**3.9792E-07**
R5	p=0.7	1.2577E-01	1.2554E-01	1.3087E-03	1.2676E-01	1.5314E-07	1.4235E-05	2.7361E-05	**1.0844E-07**
1000X1000	p=0.5	1.2575E-01	1.2512E-01	9.6667E-04	1.2568E-01	**1.5612E-07**	9.9329E-06	3.5823E-05	2.3892E-07
R5	p=0.7	1.1655E-01	1.1197E-01	8.4891E-04	1.1706E-01	1.0735E-07	9.1415E-06	2.6786E-05	**7.5085E-08**
1000X1000	p=0.5	9.4549E-02	9.4263E-02	1.6495E-03	9.4237E-02	3.3541E-07	1.8426E-05	5.2463E-05	**2.1310E-07**
R10	p=0.7	8.5588E-02	8.4603E-02	1.3747E-03	8.6035E-02	1.3768E-07	1.1807E-05	3.8599E-05	**6.5674E-08**
1000X1000	p=0.2	9.6904E-02	9.6902E-02	9.4783E-02	9.6905E-02	**1.5075E-06**	1.1378E-04	2.1189E-04	1.8036E-06
R10	p=0.3	9.5731E-02	9.6140E-02	7.9855E-02	9.5923E-02	1.2016E-06	9.4747E-05	8.9064E-05	**7.4012E-07**

5) *RBF [55]:* A robust matrix completion algorithm based on the BF method, which considers the setting that the input data are corrupted by sparse noise.
6) *FaNCL [56]:* A matrix completion method based on an inexact proximal algorithm, where the algorithm is robust to sparse noise.
7) *OPRMC*[9] *[33]:* A robust matrix completion algorithm based on NNM, which considers the setting that the input data is corrupted by column outliers.

Here, RPCA-GD, RegL1-ALM, GRASTA, RBF, and FaNCL are algorithms for coping with sparse noise, LMafit and SASD are two algorithms for coping with Gaussian noise, and OPRMC is an algorithm for coping with column outliers. The details of all alternative methods are summarized in Table I.

1) Experiments on Synthetic Data: In this section, we compare ORMC with alternative methods on synthetic data. Each low rank matrix $\boldsymbol{L} \in R^{m\times n}$ with rank r is a product of $\boldsymbol{U}\boldsymbol{V}^T$, where $\boldsymbol{U} \in R^{m\times r}$ and $\boldsymbol{V} \in R^{n\times r}$ are two full rank matrices. Suppose I and O are two sets recording the indices of inliers and outliers respectively. The matrix $\boldsymbol{L}$ will be divided into two submatrices, $\boldsymbol{L}_I$ and $\boldsymbol{L}_O$, and the columns of $\boldsymbol{L}_O$ will be replaced by outliers. Suppose c is the number of outliers. We first generate a random matrix $\boldsymbol{C} \in R^{m\times c}$ whose elements are sampled from $\mathcal{N}(0, 2)$, and then replace the columns of $\boldsymbol{L}_O$ by $\boldsymbol{C}$. The final noisy matrix is denoted by $\boldsymbol{X}$. For a given sampling parameter p, each entry of $\boldsymbol{X}$ is sampled with probability p as known entries.

There are three significant parameters in the process of generating an input matrix: 1) n—the number of samples; 2) c—the number of potential column outliers; and 3) p—the percentage of known entries. To verify the robustness of ORMC and alternative methods, we test them in different parameter settings including the following.

1) *Case 1:* We fix $n = 500$ and $r = 5$, and vary c in the set S_1, where $S_1 = \{0, 50, 100\}$.
2) *Case 2:* We fix $r = \{5, 10\}$ and $c = 0.2n$, and vary n in the set S_2, where $S_2 = \{500, 1000\}$.
3) *Case 3:* We fix $n = 500$, $r = 5$, and $c = 0.2n$, and vary *presumed input rank* in the set $\{1, 3, 5, 7, 9, 11, 13, 15, 17, 19\}$.

Note that in each case, we fix $m = 500$, and vary p in the set $S_3 = \{0.4, 0.6\}$. We evaluate the performance of all methods by relative error (RE) w.r.t. *inliers*, that is, $\boldsymbol{L}_I$. The RE is defined by

$$\text{RE} = \frac{\left\|\boldsymbol{X}_I^* - \boldsymbol{L}_I\right\|_F}{\|\boldsymbol{L}_I\|_F}. \tag{47}$$

Following the three directions mentioned above, we implement the experiments and report the final results in Tables II and III and Fig. 1, respectively. Observing the results, we can find that all methods perform well on noiseless data. For the synthetic data matrices disturbed by column outliers, RegL1, RBF, OPRMC, and our algorithm ORMC perform

[9]http://guppy.mpe.nus.edu.sg/ mpexuh/publication.html

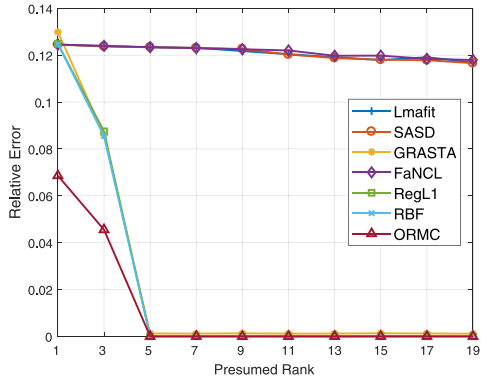


Fig. 1. RE of all methods on synthetic data with different presumed rank. This experiment follows Case 3 with $n = 500$, $r = 5$, and $c = 0.2n$.

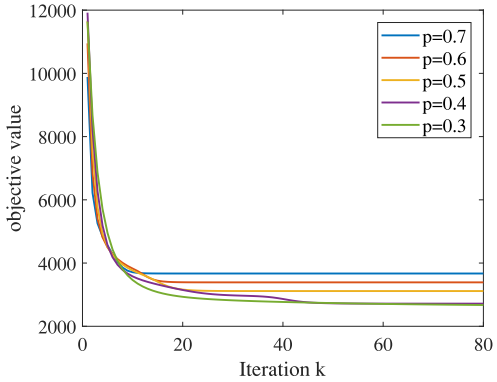


Fig. 2. Objective values of ORMC with different percentages of known entries.

better than the other algorithms, where OPRMC and our ORMC demonstrate the robustness of ℓ-21 norm to outliers. Meanwhile, although RBF and regL1 show the robustness to the noises, their performance depends on the selection of the regularization parameters. GRASTA performs better than the remaining three algorithms, but it was still clear that column outliers had a significant influence on its performance. For Lmafit, SASD, and FaNCL, the outliers generate an imprecise estimation for noise and further reduce the precision of low rank approximation. That is, the outliers make it fall into a vicious circle. Particularly, the performance of our method achieves the best result in most of the cases. Similar conclusions can be found in Fig. 1, from which we can see that GRASTA, RegL1, RBF, and ORMC present better performance than the other algorithms on synthetic data with outliers. Besides, the performance of all the algorithms tends to stabilize after the presumed rank is bigger than 5. (Since OPRMC is an NNM-based method, and the parameter to be tuned in OPRMC is the regularization parameter λ, not the rank r, so the comparison of OPRMC is not included here.) This is because that the synthetic data matrix is strictly low-rank and the rank of the matrix obtained by the algorithm is smaller than the presumed rank. In fact, for matrix completion with an approximate low-rank data matrix, an unreasonable rank may degrade the performance or even lead to failure.

In addition, we report the convergence results of ORMC in Fig. 2. In this test, we fix $n = 500$, $c = 100$, and $r = 10$, and vary p in the set $\mathcal{S}_p = \{0.3, 0.4, 0.5, 0.6, 0.7\}$. Fig. 2 shows that the convergence speed of RW is very fast when p tasks a larger value.

TABLE IV
DETAILS OF THE REAL DATA

datasets	#users	#items	#rating	max/min
Jester1	24983	100	1×10^6	$10/-10$
Jester2	23500	100	1×10^6	$10/-10$
Jester3	24983	100	6×10^5	$10/-10$
MovieLins100k	943	1682	1×10^5	5/1
MovieLins1M	6040	3706	1×10^6	5/1
MovieLins10M	69878	10677	1×10^7	5/1

2) Experiments on Real Data: To test the practicability of the proposed method, we conduct it and other matrix completion methods to cope with the collaborative filtering problem, which is widely used in recommender systems. In this article, two real datasets: 1) Jester joke[10] [57] and 2) MovieLens[11] [58], are selected. The Jester is a joke rating dataset consisting of "Jester 1," "Jester 2," "Jester 3," and "Jester-all" four rating matrices, where "Jester-all," generated by uniting the "Jester-1, 2, 3," was abandoned in this test. The "movie-100K' and "movie-1M" are two movie rating datasets. The information of each rating matrix are reported in Table IV.

Since only partial rating values are available in each real data, RE is inapplicable for evaluating the quality of the recovered matrix. Hence, normalized mean absolute error (NMAE) and root mean-square error (RMSE) will be used. The former is defined as

$$\text{NMAE} = \frac{1}{(r_{\max} - r_{\min})|\Omega_t|} \sum_{(i,j)\in\Omega_t} \left| X^*_{i,j} - X_{i,j} \right|$$

where Ω_t denotes a set recording the indices of all test ratings, and $|\Omega_t|$ denotes the total number of test entries. $r_{\max}$ and $r_{\min}$ denote the maximum value and minimum value of rating, respectively. Besides, the RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{\sum_{(i,j)\in\Omega_t} \left(X^*_{ij} - X_{ij} \right)^2}{|\Omega_t|}}.$$

In each experiment without special instructions, we select 10% known entries randomly as test ratings, and the residual as training data.

Since the rank r of real data is indeterminate, we conduct all methods with different rank settings. The different values of r and corresponding results of all algorithms are reported in Table V. Note that we remove the results of OPRMC, because it is time consuming on large-scale data.

Experiments on Noiseless Real Data: The experimental results on noiseless real data are presented in Table V, which demonstrates that ORMC obtains comparable quality on majority data. It is particularly interesting to compare LMafit and SASD with three robust methods, including RegL1-ALM,

[10] http://www.ieor.berkeley.edu/ Egoldberg/jester-data
[11] http://www.grouplens. org

TABLE V
RMSE AND NMAE OF ALL METHODS ON *Noiseless* REAL DATA WITH DIFFERENT *Rank Settings*. THE BEST RESULTS HAVE BEEN HIGHLIGHTED IN BOLD

	Method	Jester-1			Jester-2			Jester-3			Movie-100k			Movie-1M		
		r=3	r=5	r=7	r=3	r=5	r=7	r=3	r=5	r=7	r=3	r=5	r=7	r=3	r=5	r=7
NMAE	Lmafit	0.1635	0.1594	0.1584	0.1624	0.1583	**0.1573**	**0.1730**	**0.1816**	0.1968	0.1823	0.1865	0.1910	0.1738	0.1710	0.1713
	ScaledASD	0.1635	**0.1592**	**0.1583**	0.1625	0.1583	**0.1573**	**0.1730**	0.1820	0.1933	0.1828	0.1848	0.1886	0.1736	0.1713	0.1707
	GRASTA	0.1651	0.1597	0.1586	0.1613	0.1592	0.1577	0.2349	0.2807	0.2831	0.1836	0.1875	0.1945	0.1740	0.1752	0.1754
	FaNCL	0.1635	**0.1592**	0.1584	0.1625	0.1583	**0.1573**	**0.1730**	**0.1816**	0.1978	0.1823	0.1871	0.1977	0.1730	0.1705	0.1709
	RegL1-ALM	0.1620	0.1594	0.1593	0.1603	0.1586	0.1577	0.1769	0.1826	**0.1923**	0.1828	0.1856	**0.1872**	0.1779	0.1750	0.1729
	RBF-L1	**0.1618**	0.1593	0.1586	**0.1600**	0.1584	0.1577	0.1758	0.1834	0.1962	0.1823	0.1843	0.1891	**0.1717**	**0.1695**	**0.1692**
	ORMC	0.1635	**0.1592**	**0.1583**	0.1625	**0.1582**	**0.1573**	**0.1730**	**0.1816**	0.1977	**0.1822**	**0.1841**	0.1890	0.1757	0.1735	0.1732
RMSE	Lmafit	**4.2036**	4.1440	4.1474	4.2152	4.1527	4.1567	4.5157	4.8834	5.4664	0.9347	0.9736	1.0080	0.8869	0.8773	0.8843
	ScaledASD	4.2042	4.1426	**4.1429**	4.2172	4.1520	**4.1555**	4.5146	4.8965	5.2876	0.9358	0.9574	0.9863	0.8834	0.8771	0.8781
	GRASTA	4.6950	4.3201	4.3358	4.8269	4.3469	4.3671	5.2189	5.0801	5.8909	0.9593	0.9816	1.0434	0.9037	0.9101	0.9248
	FaNCL	4.2038	4.1493	4.1510	4.2165	4.1530	4.1579	4.5161	4.8902	5.5362	0.9350	0.9810	1.0796	**0.8827**	**0.8751**	0.8817
	RegL1-ALM	4.4019	4.3394	4.3419	4.4186	4.3600	4.3739	4.8747	4.9708	**5.2413**	0.9481	0.9654	**0.9748**	0.9286	0.9150	0.9043
	RBF-L1	4.3886	4.3389	4.3368	4.4161	4.3646	4.3777	4.8919	5.1017	5.4472	0.9431	0.9564	0.9813	0.8907	0.8813	0.8797
	ORMC	4.2059	**4.1419**	4.1492	**4.2147**	**4.1515**	4.1628	**4.5106**	**4.8825**	5.5197	**0.9318**	**0.9556**	0.9889	0.8953	0.8804	**0.8734**

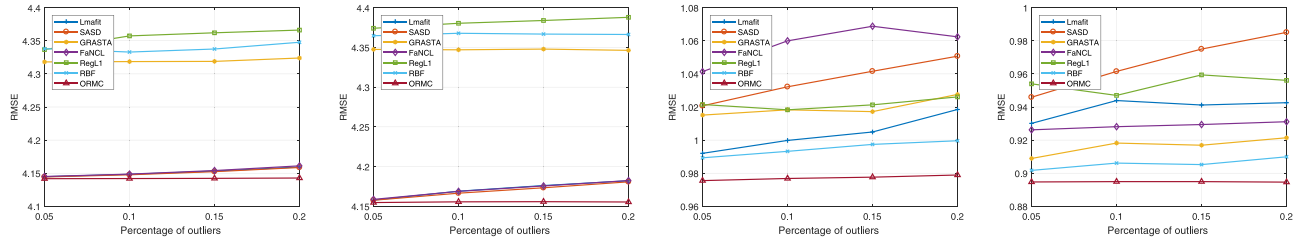


Fig. 3. RMSE of all methods on noisy real world data. Here, ϕ denotes the percentage of outliers in input data. (a) Jester-1. (b) Jester-2. (c) Movie-100K. (d) Movie-1M.

GRASTA, RBF, and ORMC. Even the latter are robust methods, and perform well on synthetic data, the performance of them is worse than LMafit on some real-world data. There are two reasons about this phenomenon. First, the optimization method used by them are different. Second, the noise distribution of real data is unknown.

Note that ORMC did not achieve the best result on all test data, However, it does not need any additional parameters except rank r. It is more attractive for solving practical problems.

Experiments on Noisy Real Data: Observing Table V, we can find that our method ORMC does not perform significantly better than residuals as we expect, because the noise distribution of real data is uncertain in selected real data. Finding a real rating data with numerous column outliers is unrealistic due to the indistinguishability between outliers and inliers. Therefore, to verify the robustness of the proposed method to column outliers, we compare it with alternative methods on artificial noisy real rating data, which are generated as follows.

Supposing ϕ is the proportion of outliers in real data matrix $\boldsymbol{X} \in R^{m\times n}$, we generate a random matrix $\boldsymbol{O} \in R^{m\times c}$ that serve as outliers, where $c = \text{round}(\phi \times n)$ denotes the number of outliers. The range of known entries and percentage of missing values of $\boldsymbol{O}$ are same as the original matrix $\boldsymbol{X}$. The columns of $\boldsymbol{O}$ will be randomly embed into the matrix $\boldsymbol{X}$, and the new data matrix with outliers is denoted by $\boldsymbol{M}$. For all real data, we fix $r = 5$. The comparisons are conducted on the original data matrix $\boldsymbol{X}$.

The results of all test algorithms on noisy data with different values of ϕ are presented in Fig. 3, which shows that ORMC provides a large advantage over others methods. The performance of ORMC with different ϕ validates the effectiveness of ORMC for processing data with column outliers. In addition, we can find that the performance of robust methods, including RegL1-ALM, GRASTA, and RBF, demonstrates that ℓ_1 norm did not show the robustness to column outliers on noisy real datasets with different ϕ.

B. Experiments on Vector Completion

In this section, we utilize ORMC to cope with the vector completion problem. The experimental results of our method will be compared with GRASTA [53], an online robust matrix completion method. In addition, we try to generalize OPRMC, LMafit, and RegL1-ALM three batch-based methods to deal with this issue. The generalized methods are called GOPRMC, GLMafit, and GRegL1-ALM, respectively. For both of them, the subspace is provided by the recovered low rank matrix $\boldsymbol{L}$.

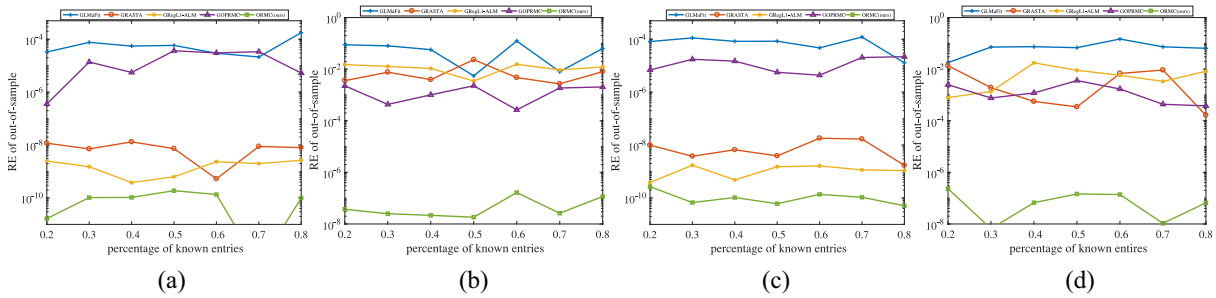


Fig. 4. RE of out of samples on synthetic data with different sampling ratio p. c denotes the number of column outliers in training data. In each case, 100 columns are selected randomly from $\boldsymbol{L}$ serve as test data. In (a) and (b), the out-of-samples are selected from $\boldsymbol{L}_I$. In (b) and (d), the out of samples are selected from $\boldsymbol{L}_O$. In (a) and (c), the number of outliers in training data is 0. In (b) and (d), the number of outliers in training data is 100. (a) $c = 0$. (b) $c = 100$. (c) $c = 0$. (d) $c = 100$.

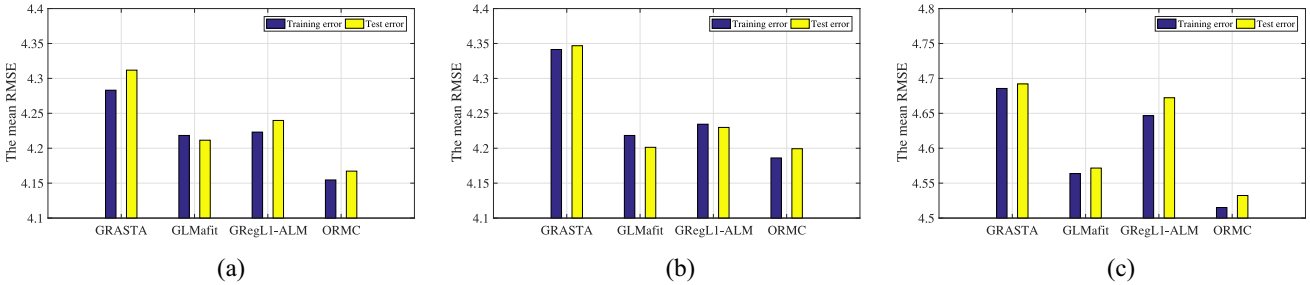


Fig. 5. RMSE of training data and test data (out-of-sample). The training error is the RMSE of the estimated missing entries in the training sets, while the test error is the RMSE of the estimated missing entries in the out-of-sample vectors. (a) Jester 1. (b) Jester 2. (c) Jester 3.

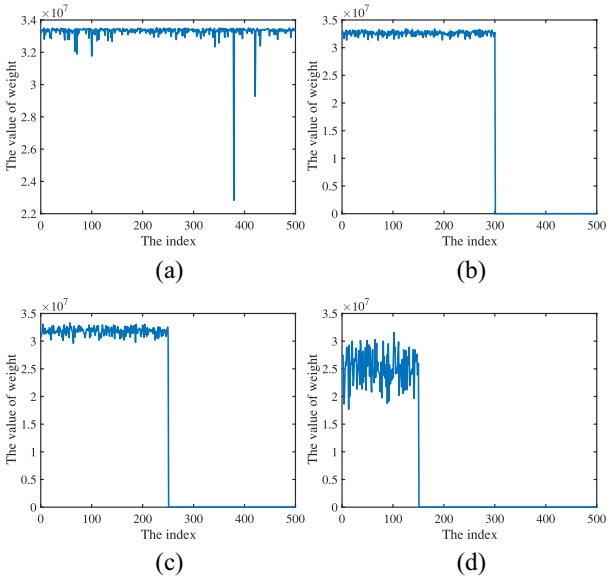


Fig. 6. Weight values corresponding to all columns on synthetic data. The last c columns are outliers. Obviously, the difference between inliers and outliers is distinct. (a) $c = 0$. (b) $c = 0.4$. (c) $c = 0.5$. (d) $c = 0.7$.

1) Experiments on Synthetic Data: In this experiment, two different types of out-of-sample will be considered.

1) In Type 1, the out of sample essentially belongs to the training data, but contains different known entries. For instance, in recommender systems, a user who is an associator but changes his ratings on items.
2) In Type 2, the out of sample is a completely fresh vector, but lies in the subspace learned from training data. For instance, the user that first visits the Web site.

For the low rank matrix $\boldsymbol{L}$, we generate the noisy matrix $\boldsymbol{X}$ with c column outliers as mentioned above. Specifically, the relevant parameters are fixed as: $m = n = 500$, $r = 10$, and $p = 0.4$. In this experiment, the matrix $\boldsymbol{X}$ plays the role of training data. We adopt two different ways to generate the test data. First, considering *Type* 1 where the out of sample belongs to $\boldsymbol{X}$, we select ω columns randomly from $\boldsymbol{L}_I$ as test data, that is, the out of sample. Second, considering *Type* 2 where the out of sample is completely fresh, we select ω columns randomly from $\boldsymbol{L}_O$ that serve as test data. We substitute the columns of $\boldsymbol{L}_O$ by outliers in the procedure of generating noisy data $\boldsymbol{X}$. Obviously, the columns of $\boldsymbol{L}_O$ are not a part of training data but lie in the subspace we aim to recover. Note that we sample ω columns randomly from $\boldsymbol{L}$ when $c = 0$.

Actually, *the quality of recovered subspace* provided by training data and *the percentage of known entries in test data* are two significant factors influencing the performance of vector completion. Hence, we test each type of out of sample in the following two cases.

1) *Case 1 (Noiseless Training Data):* We fix $c = 0$ for test data, and vary p in the set $S_3 = \{0.2, 0.3, 0.4, 0.5, 0.6\}$. Here p denotes the sampling rate of out of sample.
2) *Case 2 (Noisy Training Data):* We fix $c = 100$ for test data, and vary p in the set S_3.

The performance is also measured by RE. The final results are reported in Fig. 4. Fig. 4(a) and (c) demonstrates that each method can provide an exact estimation for out of sample when subspace is learned from noiseless data. While observing Fig. 4(b) and (d), we can find that only our method can predict the values of missing entries exactly when training data are contaminated by column outliers, which demonstrates that

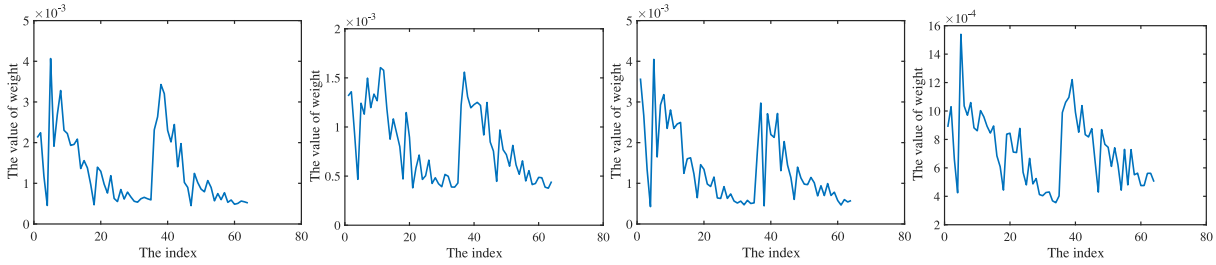


Fig. 7. Values of diagonal elements of weighted matrix $\boldsymbol{D}$. From left to right, the subjects are: 1, 10, 17, and 28.

Fig. 8. Face images corresponding to the 20 least diagonal elements of weighted matrix $\boldsymbol{D}$. From top to bottom, the subjects are: 1, 10, 17, and 28. Obviously, all selected images are contaminated by massive shadows and specularities.

Fig. 9. Face images corresponding to the 20 largest diagonal elements of weighted matrix $\boldsymbol{D}$. From top to bottom, the subjects are: 1, 10, 17, and 28. Obviously, all selected images are with high quality.

only our method can estimate the intrinsic low-dimensional subspace where out of sample lies. We can conclude that all batch methods with high performance on training data can be implemented to deal with the out-of-sample problem.

2) Numerical Vector Completion Results on Real Data: Next, we utilize all algorithms to deal with real-world data. Only Jester data will be used to test the performance of proposed method and generalized algorithms. All columns of a rating matrix will be divided into training data (80%) and test data (20%). We utilize the training data to learn the subspace, and evaluate the quality of them on test data. We sample 80% ratings randomly serve as known entries both in training data and test data. The comparisons between training error and test error are reported in Fig. 5.

Observing Fig. 5, we can find that ORMC provides a more accurate estimation both in training data and out of sample. The superiority on training data leads to the advantage on out of sample. However, for each algorithm, the gap between training error and test error is very small, which demonstrates that the subspace provided by training data can be used to estimate the missing values of out of sample. That is, all batch-based methods can be implemented as online methods.

C. Experiments on Outlier Detection

In this section, we address the problem of outliers detection from incomplete information. An assumption is that inliers lie close to a low-dimensional subspace. Although various approaches have been developed for outliers detection, most of them are unsuitable for data with missing values. Hence, in this experiment, we report only the results returned by our method.

We have pointed in Section III-B that the location of outliers can be determined by observing the weighted matrix $\boldsymbol{D}$. Concretely, for matrix $\boldsymbol{X}$, if the value of $\boldsymbol{D}_{i,i}$ is very small, we could regard the column vector $\boldsymbol{x}_i$ as an outlier. Next, we validate this point on synthetic and real-world data.

1) Outliers Detection on Synthetic Data: Similarly, we first generate a low rank matrix $\boldsymbol{L}$ with $m = n = 500$, $r = 10$, as well as $p = 0.4$. The value of c varies in the set $S_4 = \{0, 200, 250, 350\}$. For visualization, we replace the last c columns of $\boldsymbol{L}$ by outliers. The final values of diagonal elements of $\boldsymbol{D}$ are reported in Fig. 6. It is obvious that the

weights of outliers are much less than inliers even when the percentage of outliers is larger than 50%. So, we can determine the location of all outliers exactly.

2) Outliers Detection on Real Data: We validate the performance of the proposed method on Extended YaleB dataset, which contains 2432 face images of 38 human subjects under 64 different illumination conditions. The task of this experiment is to detect the face images with massive shadows and specularities, which generally effect the precision of face restoration as well as recognition, and can be regarded as outliers.

For one subject, we generate the input matrix $\boldsymbol{X}$ by concatenating 64 vectored images together. Then, we sample 50% elements randomly from $\boldsymbol{X}$ serve as known entries. For ORMC, we fix rank $r = 5$. The weighted matrices of Subject 1, Subject 10, Subject 17, and Subject 28 are reported in Fig 7. The face images corresponding to the least 20 values of $\boldsymbol{D}$ are presented in Fig. 8, and the face images corresponding to the largest 20 values of $\boldsymbol{D}$ are presented in Fig. 9. We can find that all selected images corresponding to smaller weights are contaminated by massive shadows and specularities, while all selected images corresponding to larger weights are "good" samples with slight noise. Therefore, our method can serve as a preprocessing tool for removing the samples with poor quality or selecting the samples with high quality from collected data.

VI. Conclusion

We presented a novel model for robust matrix completion, namely, ORMC, which is based on matrix BF. As the proposed model is nonconvex and nonsmooth, we developed an effective method to solve it. The optimization process does not need any additional parameters and makes ORMC applicable in different practical applications. Moreover, a byproduct, the weighted matrix $\boldsymbol{D}$, can help users detect outliers from input data. Numerous experiments on both synthetic datasets and real-world datasets have validated the performance of ORMC. From the experimental results, we can find that ORMC can eliminate the influence of outliers with adaptive weights effectively, and make it robust to column outliers.

Considering that some real data are not static, we propose a vector completion model, which can help us estimate the missing values of out of sample. We show that the similar idea can be adopted by other batch-based methods. Note that the success of the proposed vector completion method is based on the following two assumptions. First, the inliers of input data are noiseless. Second, the subspace of streaming data is static. These assumptions may not be applicable to some practical applications. Actually, the former can be avoided by introducing a new noise matrix $\boldsymbol{E}$ and the latter can be solved by utilizing the subspace updating just like GRASTA or MD-ISVD. Our further work will focus on solving these two significant issues.

References

[1] N. Srebro, J. D. M. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1329–1336.

[2] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. Zahzah, "Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset," *Comput. Sci. Rev.*, vol. 23, pp. 1–71, Feb. 2017.

[3] K. Li, J. Yang, and J. Jiang, "Nonrigid structure from motion via sparse representation," *IEEE Trans. Cybern.*, vol. 45, no. 8, pp. 1401–1413, Aug. 2015.

[4] G. Ye *et al.*, "Free-viewpoint video of human actors using multiple handheld kinects," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1370–1382, Oct. 2013.

[5] L. Zhang, L. Song, B. Du, and Y. Zhang, "Nonlocal low-rank tensor completion for visual data," *IEEE Trans. Cybern.*, vol. 51, no. 2, pp. 673–685, Feb. 2021.

[6] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, p. 717, 2009.

[7] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.

[8] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and bregman iterative methods for matrix rank minimization," *Math. Program.*, vol. 128, nos. 1–2, pp. 321–353, 2011.

[9] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," 2010. [Online]. Available: arXiv:1009.5055.

[10] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems," *Pac. J. Optim.*, vol. 6, no. 3, pp. 615–640, 2010.

[11] Y. Liu, F. Shang, L. Jiao, J. Cheng, and H. Cheng, "Trace norm regularized CANDECOMP/PARAFAC decomposition with missing data," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2437–2448, Nov. 2015.

[12] F. Shang, Y. Liu, and J. Cheng, "Unified scalable equivalent formulations for schatten quasi-norms," 2016. [Online]. Available: arXiv:1606.00668.

[13] G. Marjanovic and V. Solo, "On ℓ_q optimization and matrix completion," *IEEE Trans. Signal Process.*, vol. 60, no. 11, pp. 5714–5724, Nov. 2012.

[14] F. Nie, H. Huang, and C. Ding, "Low-rank matrix recovery via efficient schatten p-norm minimization." *Proc. AAAI Conf. Artif. Intell.*, vol. 26, no. 1, pp. 655–661, 2012.

[15] F. Nie, Z. Huo, and H. Huang, "Joint capped norms minimization for robust matrix recovery," in *Proc. 26th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2017, pp. 2557–2563.

[16] D. Zhang, Y. Hu, J. Ye, X. Li, and X. He, "Matrix completion by truncated nuclear norm regularization," in *Proc. Comput. Vis. Pattern Recognit. Conf. (CVPR)*, 2012, pp. 2192–2199.

[17] Z. Fang, X. Yang, L. Han, and X. Liu, "A sequentially truncated higher order singular value decomposition-based algorithm for tensor completion," *IEEE Trans. Cybern.*, vol. 49, no. 5, pp. 1956–1967, May 2019.

[18] S. Gu, Q. Xie, D. Meng, W. Zuo, X. Feng, and L. Zhang, "Weighted nuclear norm minimization and its applications to low level vision," *Int. J. Comput. Vis.*, vol. 121, no. 2, pp. 183–208, 2017.

[19] Q. Gao, P. Zhang, W. Xia, D. Xie, X. Gao, and D. Tao, "Enhanced tensor RPCA and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Aug. 18, 2020, doi: 10.1109/TPAMI.2020.3017672.

[20] T. Oh, Y. Matsushita, Y.-W. Tai, and I. S. Kweon, "Fast randomized singular value thresholding for low-rank optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 376–391, Feb. 2018.

[21] J. P. Haldar and D. Hernando, "Rank-constrained solutions to linear matrix equations using powerfactorization," *IEEE Signal Process. Lett.*, vol. 16, no. 7, pp. 584–587, Jul. 2009.

[22] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Math. Program. Comput.*, vol. 4, no. 4, pp. 333–361, 2012.

[23] T. Ngo and Y. Saad, "Scaled gradients on grassmann manifolds for matrix completion," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1412–1420.

[24] R. H. Keshavan and S. Oh, "A gradient descent algorithm on the Grassman manifold for matrix completion," 2009. [Online]. Available: arXiv:0910.5260.

[25] Y. Chi, Y. M. Lu, and Y. Chen, "Nonconvex optimization meets low-rank matrix factorization: An overview," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5239–5269, Oct. 2019.

[26] S. Mao, L. Xiong, L. Jiao, T. Feng, and S.-K. Yeung, "A novel riemannian metric based on Riemannian structure and scaling information for fixed low-rank matrix completion," *IEEE Trans. Cybern.*, vol. 47, no. 5, pp. 1299–1312, May 2017.

[27] X. Yi, D. Park, Y. Chen, and C. Caramanis, "Fast algorithms for robust PCA via gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4152–4160.
[28] Y. Zheng, G. Liu, S. Sugimoto, S. Yan, and M. Okutomi, "Practical low-rank matrix approximation under robust l 1-norm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2012, pp. 1410–1417.
[29] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
[30] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 2080–2088.
[31] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 2496–2504.
[32] C. You, D. P. Robinson, and R. Vidal, "Provable self-representation based outlier detection in a union of subspaces," 2017. [Online]. Available: arXiv:1704.03925.
[33] Y. Chen, H. Xu, C. Caramanis, and S. Sanghavi, "Matrix completion with column manipulation: Near-optimal sample-robustness-rank tradeoffs," *IEEE Trans. Inf. Theory*, vol. 62, no. 1, pp. 503–526, Jan. 2016.
[34] Z. Liu, Z. Hu, and F. Nie, "Matrix completion and vector completion via robust subspace learning," *Neurocomputing*, vol. 306, pp. 171–181, Sep. 2018.
[35] J. R. Bunch and C. P. Nielsen, "Updating the singular value decomposition," *Numerische Mathematik*, vol. 31, no. 2, pp. 111–129, 1978.
[36] M. Brand, "Incremental singular value decomposition of uncertain data with missing values," in *Proc. Eur. Conf. Comput. Vis.*, vol. 2350, 2002, pp. 707–720.
[37] M. Brand, "Fast online SVD revisions for lightweight recommender systems," in *Proc. SIAM Int. Conf. Data Mining*, 2003, pp. 37–46.
[38] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Proc. 48th Annu. Allerton Conf. Commun. Control Comput.*, 2010, pp. 704–711.
[39] L. Balzano and S. J. Wright, "On GROUSE and incremental SVD," in *Proc. IEEE 5th Int. Workshop Comput. Adv. Multi-Sensor Adapt. Process. (CAMSAP)*, 2013, pp. 1–4.
[40] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2012, pp. 1568–1575.
[41] J. Feng, H. Xu, and S. Yan, "Online robust PCA via stochastic optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2013, pp. 404–412.
[42] J. Shen, H. Xu, and P. Li, "Online optimization for max-norm regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 1718–1726.
[43] Y. Chi, Y. C. Eldar, and R. Calderbank, "PETRELS: Parallel subspace estimation and tracking by recursive least squares from partial observations," *IEEE Trans. Signal Process.*, vol. 61, no. 23, pp. 5947–5959, Dec. 2013.
[44] M. Mardani, G. Mateos, and G. B. Giannakis, "Subspace learning and imputation for streaming big data matrices and tensors," *IEEE Trans. Signal Process.*, vol. 63, no. 10, pp. 2663–2677, May 2015.
[45] C. Jin, S. M. Kakade, and P. Netrapalli, "Provable efficient online matrix completion via non-convex stochastic gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 4520–4528.
[46] D. Zhang and L. Balzano, "Convergence of a Grassmannian gradient descent algorithm for subspace estimation from undersampled data," 2016. [Online]. Available: arXiv:1610.00199.
[47] A. Ilin and T. Raiko, "Practical approaches to principal component analysis in the presence of missing values," *J. Mach. Learn. Res.*, vol. 11, no. 66, pp. 1957–2000, 2010.
[48] F. Nie, J. Yuan, and H. Huang, "Optimal mean robust principal component analysis," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1062–1070.
[49] F. Nie, D. Wu, R. Wang, and X. Li, "Truncated robust principle component analysis with a general optimization framework," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Sep. 20, 2020, doi: 10.1109/TPAMI.2020.3027968.
[50] L. K. Balzano, "Handling missing data in high-dimensional subspace modeling," Ph.D. dissertation, Elect. Comput. Eng., Univ. Wisconsin-Madison, Madison, WI, USA, 2012.
[51] F. Nie, H. Huang, X. Cai, and C. H. Q. Ding, "Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization." in *Proc. Adv. Neural Inf. Process. Syst.*, 2010, pp. 1813–1821.
[52] C. Lu, Z. Lin, and S. Yan, "Smoothed low rank and sparse matrix recovery by iteratively reweighted least squares minimization," *IEEE Trans. Image Process.*, vol. 24, no. 2, pp. 646–654, Feb. 2015.
[53] J. He, L. Balzano, and J. Lui, "Online robust subspace tracking from partial information," 2011. [Online]. Available: arXiv:1109.3827.
[54] J. Tanner and K. Wei, "Low rank matrix completion by alternating steepest descent methods," *Appl. Comput. Harmonic Anal.*, vol. 40, no. 2, pp. 417–429, 2016.
[55] F. Shang, Y. Liu, H. Tong, J. Cheng, and H. Cheng, "Robust bilinear factorization with missing and grossly corrupted observations," *Inf. Sci.*, vol. 307, pp. 53–72, Jun. 2015.
[56] Q. Yao, J. Kwok, T. Wang, and T. Liu, "Large-scale low-rank matrix learning with nonconvex regularizers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 11, pp. 2628–2643, Nov. 2019.
[57] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Inf. Retrieval J.*, vol. 4, no. 2, pp. 133–151, 2001.
[58] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Berkeley, CA, USA, 1999, pp. 230–237.

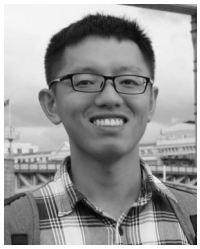
Feiping Nie (Member, IEEE) received the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2009.

He is currently a Full Professor with Northwestern Polytechnical University, Xi'an, China. He has published more than 100 papers in the following journals and conferences: IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, INTERNATIONAL JOURNAL OF COMPUTER VISION, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, ICML, NIPS, KDD, IJCAI, AAAI, ICCV, CVPR, and ACM MM. His papers have been cited more than 20 000 times and the H-index is 78. His research interests are machine learning and its applications, such as pattern recognition, data mining, computer vision, image processing, and information retrieval.

Prof. Nie is currently serving as an associate editor or PC member for several prestigious journals and conferences in the related fields.

Ziheng Li is currently pursuing the Ph.D. degree with Northwestern Polytechnical University, Xi'an, China, under the supervision of Prof. F. Nie.

His research interests are machine learning and its applications, including clustering, matrix completion, and information retrieval.



Zhanxuan Hu is currently pursuing the Ph.D. degree with Northwestern Polytechnical University, Xian, China.

His research interests include topics in data mining, computer vision, and background substraction.

Rong Wang received the B.S. degree in information engineering, the M.S. degree in signal and information processing, and the Ph.D. degree in computer science from Xi'an Research Institute of Hi-Tech, Xi'an, China, in 2004, 2007, and 2013, respectively.

From 2007 to 2013, he also studied with the Department of Automation, Tsinghua University, Beijing, China, for his Ph.D. degree. He is currently an Associate Professor with the School of Cybersecurity and School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an. His research interests focus on machine learning and its applications.

Xuelong Li (Fellow, IEEE) is a Full Professor with the School of Computer Science and School of Artificial Intelligence, Optics and Electronics (iOPEN), Northwestern Polytechnical University, Xi'an, China.