Fast Robust Matrix Completion via Entry-Wise ℓ_0-norm Minimization

Xiao Peng Li, Zhang-Lei Shi, Qi Liu, Hing Cheung So, Fellow, IEEE

Abstract—Matrix completion (MC) aims at recovering missing entries given an incomplete matrix. Existing algorithms for MC are mainly designed for noiseless or Gaussian noise scenarios and thus they are not robust to impulsive noise. For outlier resistance, entry-wise ℓ_p-norm with $0 < p < 2$ and M-estimation are two popular approaches. Yet the optimum selection of p for the entry-wise ℓ_p-norm based methods is still an open problem. Besides, M-estimation is limited by a breakdown point, that is, the largest proportion of outliers. In this paper, we adopt entry-wise ℓ_0-norm, namely, the number of nonzero entries in a matrix, to separate anomalies from the observed matrix. Prior to separation, Laplacian kernel is exploited for outlier detection, which provides a strategy to automatically update the entry-wise ℓ_0-norm penalty parameter. The resultant multi-variable optimization problem is addressed by block coordinate descent (BCD), yielding ℓ_0-BCD and ℓ_0-BCD-F. The former detects and separates outliers, as well as its convergence is guaranteed. In contrast, the latter attempts to treat outlier-contaminated elements as missing entries, which leads to higher computational efficiency. Making use of majorization-minimization (MM), we further propose ℓ_0-BCD-MM and ℓ_0-BCD-MM-F for robust nonnegative MC where the nonnegativity constraint is handled by a closed-form update. Experimental results of image inpainting and hyperspectral image recovery demonstrate that the suggested algorithms outperform several state-of-the-art methods in terms of recovery accuracy and computational efficiency.

Index Terms—Matrix completion, nonnegative matrix completion, robust recovery, outlier detection, ℓ_0-norm optimization.

I. Introduction

MATRIX completion (MC) refers to restoring the missing entries of an incomplete matrix by making use of the low-rank property, and has various applications, including machine learning [1], system identification [2], computer vision [3], image inpainting [4], and target estimation [5]. This is because many real-world signals can be represented/approximated as low-rank matrices. For instance, image data can be modeled as low-rank matrices since the main information is dominated by the largest singular values, associating with their left and right singular vectors [6].

Mathematically, MC is formulated as a constrained rank minimization problem [7]. Unfortunately, it is NP-hard as the rank function is both nonconvex and discrete. The nuclear norm has been proved to be a convex envelop of the rank function [8] and hence can be used as its substitute. MC based on the nuclear norm has been handled by singular value thresholding (SVT) [9], fixed point continuation (FPC) [10], accelerated proximal gradient (APG) [11] and truncated nuclear norm regularization (TNNR) [12]. As singular value decomposition (SVD) is required, the aforementioned methods are computationally demanding, especially for large-size matrices.

On the other hand, the nuclear norm, which is equal to the sum of all singular values, may cause the solution to deviate seriously from the ground truth because of its slack relaxation. To handle this issue, weighted nuclear norm method (WNNM) [13] assigns different weights to the singular values. Besides, the capped trace norm and Schatten p-norm are suggested to approximate the rank function [14], [15], [16]. Although these two norms have a better rank function approximation, they also require SVD computation. Another way to avoid the slack relaxation is to convert the rank minimization into a rank constraint [17] and then use gradient projection to handle the resultant problem. Since only the truncated SVD is involved, its computational complexity is lower than those implementing full SVD.

To avoid SVD computation, matrix factorization scheme has been suggested for MC [18]. The basic idea is to exploit the product of two much smaller matrices to represent the objective matrix under the assumption that the rank of target matrix is known. Low-rank matrix fitting (LMaFit) [19] is first proposed using matrix factorization, but its global convergence cannot be ensured. Subsequently, MC based on low-rank factorization is proved to obtain the globally optimal solution under some mild conditions [20]. Moreover, locally linear approximation (LLA) [21] adopts the local structure of visual data for recovery performance improvement. Recently, the online robust matrix completion (ORMC) method [22] considers the incomplete matrix containing several Gaussian noise-contaminated columns, and adopts $\ell_{2,1}$-norm as the loss function.

The above-mentioned algorithms can work well when

The work described in this paper was supported in part by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China [Project No. CityU 11207922], in part by the Fundamental Research Funds for the Central Universities under Grant 22CX06039A, and in part by the National Natural Science Foundation of China under Grant 62202174. (Corresponding author: Zhang-Lei Shi.)

Xiao Peng Li and Hing Cheung So are with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong SAR, China (e-mail: x.p.li@my.cityu.edu.hk; hcso@ee.cityu.edu.hk).

Zhang-Lei Shi is with the College of Science, China University of Petroleum (East China), Qingdao 266580, China. (e-mail: zlshi@upc.edu.cn).

Q. Liu is with the School of Future Technology, South China University of Technology, Guangzhou 511442, and is also with Pazhou Lab, Guangzhou 510330, China. (e-mail: drliuqi@scut.edu.cn).

the observed matrix is noise-free or corrupted by white Gaussian noise. Although Gaussian distribution is the best approximation to common noise, non-Gaussian distributed noise has occurred in some fields [23]. For instance, images may be corrupted by salt-and-pepper noise, and communication channels may contain impulsive components. To resist outliers, the entry-wise ℓ_p-norm with $0 < p < 2$ has been adopted as the loss function, resulting in the alternating projection (AP) [24], ℓ_p regression (ℓ_p-reg) [25], and ℓ_p-norm based matching pursuit (ℓ_p-MP) [26]. These approaches demonstrate outstanding performance in the presence of gross errors. However, the entry-wise ℓ_p-norm has two open problems. The first is how to select the optimum p for the impulsive noise of a certain intensity since the choice of p affects the recovery performance. The second is that the entry-wise ℓ_p-norm with $0 < p < 1$ is nonsmooth and nonconvex, which poses a challenge for optimization. Besides, the works in [27], [28] suggest adopting the M-estimators to deal with robust matrix completion (RMC). Nevertheless, M-estimation has a breakdown point that depends on the largest proportion of anomalies [29]. Moreover, Bayesian approach has been exploited to solve RMC, which includes the Bayesian estimator for noisy MC (BENMC) [30] and variational Bayesian matrix factorization based on ℓ_1-norm (VBMFL1) [31]. Recently, the correntropy-induced loss function and ℓ_1-norm formulated as a regularization term are applied for RMC [32], [33].

In applications such as blind recommender systems [34], sensor calibration [35], and hyperspectral imaging [36], [37], data are nonnegative. However, MC methods may result in negative entries in the recovered matrix, which violates the inherent data property. To handle this issue, nonnegative matrix completion (NMC) approaches are proposed, including matrix factorization with alternating direction method of multipliers (MF-ADMM) [38], low-rank approximation with ADMM (LRA-ADMM) [39], and NMC using Nesterov iterations (NeNMC) [40]. Compared with MC, NMC ensures that all entries of the recovered matrix are nonnegative. However, most existing NMC algorithms are not robust to outliers.

In this paper, we exploit the entry-wise ℓ_0-norm, which is equal to the number of nonzero entries, to separate anomalies from the observed matrix. In addition, we combine the entry-wise ℓ_0-norm with the matrix factorization strategy to formulate the RMC problem. Then, block coordinate descent (BCD) [41] is adopted as the solver for the resultant multi-variable optimization problem. The developed algorithms, called ℓ_0-BCD and ℓ_0-BCD-F, update one block with fixing the remaining blocks at each iteration. The former detects and separates outliers from the noisy matrix, while the latter considers the outlier-contaminated elements as missing entries. Therefore, the ℓ_0-BCD-F has lower computational complexity than the ℓ_0-BCD. To detect outliers, Laplacian kernel is exploited, resulting in an adaptive method to update the penalty parameter of the ℓ_0-norm. On the other hand, the above scheme is applied to deal with robust NMC (RNMC), yielding two algorithms, namely, ℓ_0-BCD-MM and ℓ_0-BCD-MM-F. They use majorization-minimization (MM) to handle the least squares problem with nonnegativity constraint. Unlike the projected gradient descent whose convergence depends on an appropriate choice of step-size, the ℓ_0-BCD-MM and ℓ_0-BCD-MM-F provide a closed-form update without this parameter. Our main contributions are summarized as:

(i) The entry-wise ℓ_0-norm is formulated as a penalty term to separate anomalies. In addition, the Laplacian kernel function is exploited to devise an outlier detector for identifying anomalies, by which the penalty parameter of the ℓ_0-norm is automatically updated. The proposed algorithms achieve a higher recovery accuracy than existing approaches for synthetic and real-world data in the presence of outliers.

(ii) A novel perspective is suggested to address RMC, i.e., outlier-contaminated entries are considered as missing elements. This reduces the algorithm computational requirement, resulting in a faster variant, namely, ℓ_0-BCD-F. Its computational efficiency is much higher than the existing techniques.

(iii) To tackle the nonnegative least squares problem in RNMC, we adopt MM to develop a closed-form update which is proved to meet the Karush-Kuhn-Tucker (KKT) conditions. Compared with the projected gradient descent that requires an appropriate value of step-size [42], our algorithm is parameter-free.

The remainder of this paper is organized as follows. In Section II, we introduce notations and preliminaries, including problem formulation and representative solvers. Two efficient algorithms for RMC, i.e., ℓ_0-BCD and ℓ_0-BCD-F, are developed in Section III. In Section IV, we propose ℓ_0-BCD-MM and ℓ_0-BCD-MM-F to tackle RNMC. Numerical results based on synthetic and real-world data are presented in Section V. Finally, Section VI provides concluding remarks.

II. Overview of Related Work

A. Notations

Scalars, vectors, and matrices are represented by italic, bold lower-case, and bold upper-case letters, respectively. Besides, a matrix of ones is signified by $\mathbf{1}$. For matrices, $\|\cdot\|_*$ is the nuclear norm while $\|\boldsymbol{A}\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j}^2}$ is the Frobenius norm where $a_{i,j}$ is the (i,j) entry of $\boldsymbol{A} \in \mathbb{R}^{m\times n}$. The $\|\boldsymbol{A}\|_p^p$ with $0 < p < 2$ is calculated using the sum of p power of all elements. The element-wise absolute operation of $\boldsymbol{A}$ is represented by $|\boldsymbol{A}|$. Moreover, $\boldsymbol{A} \geq 0$ and $\boldsymbol{A} \succeq 0$ signify that $\boldsymbol{A}$ is a nonnegative matrix and positive semidefinite matrix, respectively. Furthermore, $\boldsymbol{a}_i^T$ is the ith row of $\boldsymbol{A}$, while $\boldsymbol{a}_j$ is the jth column of $\boldsymbol{A}$. For a vector $\boldsymbol{a} \in \mathbb{R}^m$, $\|\boldsymbol{a}\|_2 = \sqrt{\sum_{i=1}^{m} a_i^2}$ and $\|\boldsymbol{a}\|_1 = \sum_{i=1}^{m} |a_i|$ are ℓ_2-norm and ℓ_1-norm, respectively. In addition, $\dim(\boldsymbol{a})$ indicates the length of $\boldsymbol{a}$. The entry-wise ℓ_0-norm for vectors and matrices is denoted by $\|\cdot\|_0$

which is the total number of nonzero entries. The pseudo-inverse and transpose operators are signified by $(\cdot)^\dagger$ and $(\cdot)^T$, respectively.

B. Matrix Completion Formulation

Consider an incomplete matrix $\boldsymbol{X}_{\boldsymbol{\Omega}} \in \mathbb{R}^{m\times n}$ where $\boldsymbol{\Omega} \in \mathbb{R}^{m\times n}$ is a binary matrix, consisting of 0 and 1. Here, $\boldsymbol{X}_{\boldsymbol{\Omega}}$ denotes that $\boldsymbol{X}$ projects onto $\boldsymbol{\Omega}$, resulting in

$$(\boldsymbol{X}_{\boldsymbol{\Omega}})_{i,j} = \begin{cases} x_{i,j}, & \text{if } \Omega_{i,j} = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Ideally, the MC problem is formulated as a rank minimization problem [7]:

$$\min_{\boldsymbol{M}} \operatorname{rank}(\boldsymbol{M}) \text{ s.t. } \boldsymbol{X}_{\boldsymbol{\Omega}} = \boldsymbol{M}_{\boldsymbol{\Omega}}. \tag{2}$$

It means that MC aims at seeking $\boldsymbol{M} \in \mathbb{R}^{m\times n}$ with minimum rank under the constraint where the entries of the recovered and observed matrices in $\boldsymbol{\Omega}$ are equal. Yet rank minimization is an NP-hard problem. One popular and feasible method is to adopt matrix factorization [19], leading to

$$\min_{\boldsymbol{U},\boldsymbol{V}} \|(\boldsymbol{U}\boldsymbol{V})_{\boldsymbol{\Omega}} - \boldsymbol{X}_{\boldsymbol{\Omega}}\|_F^2, \tag{3}$$

where the recovered matrix is represented by the product of two low-dimensional matrices, namely, $\boldsymbol{U} \in \mathbb{R}^{m\times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r\times n}$. Herein, r is the rank of the recovered matrix. Since (3) does not require computing SVD, it has lower computational complexity compared to the nuclear norm based algorithms. However, its performance will be degraded when the observed entries involve gross errors. The reason is that the Frobenius norm amplifies the power of outliers which is much larger than that of the Gaussian noise.

One of the prevailing approaches to resist outliers is to use the entry-wise ℓ_p-norm with $0 < p < 2$ [25]:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \|(\boldsymbol{U}\boldsymbol{V})_{\boldsymbol{\Omega}} - \boldsymbol{X}_{\boldsymbol{\Omega}}\|_p^p. \tag{4}$$

Consider a residual $e = (\boldsymbol{U}\boldsymbol{V})_{i,j} - x_{i,j}$ corresponding to an outlier, we get $|e|^p < |e|^2$ with $0 < p < 2$. That is, the entry-wise ℓ_p-norm is able to weaken the impact of outliers and thus the entry-wise ℓ_p-norm has better performance than the Frobenius norm in impulsive noise environment.

Another approach for RMC is based on the maximum correntropy criterion [32]:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \|(\boldsymbol{U}\boldsymbol{V})_{\boldsymbol{\Omega}} - \boldsymbol{X}_{\boldsymbol{\Omega}}\|_{G_\sigma}, \tag{5}$$

where $\|\cdot\|_{G_\sigma}$ is the correntropy-induced loss function, defined as

$$\|\boldsymbol{A}\|_{G_\sigma} = \sum_{i=1}^{m}\sum_{j=1}^{n} \sigma^2 \left(1 - \exp\left(-\frac{a_{i,j}^2}{2\sigma^2}\right)\right), \tag{6}$$

where $\sigma > 0$ is the kernel width. Since solving (5) requires a three-layer iterative procedure, the resultant algorithm has relatively high computational complexity.

In addition, the entry-wise ℓ_0-norm has been applied for RMC [43], resulting in

$$\begin{aligned} \min_{\boldsymbol{M},\boldsymbol{E}} &\|\boldsymbol{M}_{\boldsymbol{\Omega}} - \boldsymbol{X}_{\boldsymbol{\Omega}} + \boldsymbol{E}_{\boldsymbol{\Omega}}\|_F^2 + \frac{\alpha}{2}\|\boldsymbol{M}_{\widetilde{\boldsymbol{\Omega}}}\|_F^2, \\ \text{s.t. } &\|\boldsymbol{E}\|_0 \le N_0,\ \|\boldsymbol{E}\|_2 \le K_E,\ \operatorname{rank}(\boldsymbol{M}) \le r, \end{aligned} \tag{7}$$

where $\alpha > 0$ is a penalty parameter, $\widetilde{\boldsymbol{\Omega}}$ is a subset of $\boldsymbol{\Omega}$ and does not contain the indices of the outlier-contaminated entries, N_0 is a positive integer to limit the number of outliers, and K_E is a finite constant to facilitate the convergence. However, the method to solve (7) is computationally expensive since it requires performing SVD. Besides, the number of anomalies should be estimated in advance, which is a challenge for real-world data.

C. Nonnegative Matrix Completion Formulation

Analogous to MC, NMC can be formulated as [38], [39]:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \|(\boldsymbol{U}\boldsymbol{V})_{\boldsymbol{\Omega}} - \boldsymbol{X}_{\boldsymbol{\Omega}}\|_F^2, \text{ s.t. } \boldsymbol{U} \ge 0, \boldsymbol{V} \ge 0, \tag{8}$$

where $\boldsymbol{U} \ge 0$ and $\boldsymbol{V} \ge 0$ ensure the recovered matrix nonnegative. In this work, we develop a method to solve the nonnegativity constraint of RNMC. Prior to introducing our idea, two existing approaches are first reviewed. To facilitate presentation, we assume that $\boldsymbol{\Omega} = \boldsymbol{1}$ in (8). In this way, we can obtain a special case of RNMC, i.e., nonnegative low-rank matrix approximation.

The most famous method for nonnegative matrix factorization is the multiplicative update [44]:

$$u_{i,j}^{k+1} = u_{i,j}^{k} \frac{(\boldsymbol{X}_{\boldsymbol{\Omega}}(\boldsymbol{V}^k)^T)_{i,j}}{(\boldsymbol{U}^k\boldsymbol{V}^k(\boldsymbol{V}^k)^T)_{i,j}}, \tag{9a}$$

$$v_{i,j}^{k+1} = v_{i,j}^{k} \frac{((\boldsymbol{U}^{k+1})^T\boldsymbol{X}_{\boldsymbol{\Omega}})_{i,j}}{((\boldsymbol{U}^{k+1})^T\boldsymbol{U}^{k+1}\boldsymbol{V}^k)_{i,j}}. \tag{9b}$$

Note that the solution to (9) is $\boldsymbol{U} > 0$ and $\boldsymbol{V} > 0$ which is not strictly equivalent to $\boldsymbol{U} \ge 0$ and $\boldsymbol{V} \ge 0$. Moreover, the convergence of the multiplicative update cannot be ensured [45].

Another approach is to exploit projected gradient descent [42], resulting in

$$\boldsymbol{U}^{k+1} = \max\left(0, \boldsymbol{U}^k - \eta((\boldsymbol{U}^k\boldsymbol{V}^k)_{\boldsymbol{\Omega}} - \boldsymbol{X}_{\boldsymbol{\Omega}})(\boldsymbol{V}^k)^T\right), \tag{10a}$$

$$\boldsymbol{V}^{k+1} = \max\left(0, \boldsymbol{V}^k - \eta(\boldsymbol{U}^{k+1})^T((\boldsymbol{U}^{k+1}\boldsymbol{V}^k)_{\boldsymbol{\Omega}} - \boldsymbol{X}_{\boldsymbol{\Omega}})\right), \tag{10b}$$

where $\eta > 0$ is the step-size. Compared with multiplicative update, its convergence is guaranteed. But η should be appropriately chosen to compromise between the convergence speed and accuracy.

III. Proposed Algorithms for Robust Matrix Completion

In this section, two algorithms are proposed for RMC, including a basic version and its fast variant. Besides, a Laplacian kernel based anomaly detector is developed for adaptively updating the penalty parameter.

A. Algorithm Development

We consider that the impulsive noise comprises Gaussian noise with small power and sparse impulses with large power. Hence, $\boldsymbol{X}_{\boldsymbol{\Omega}}$ that is corrupted by impulsive noise is formulated as

$$\boldsymbol{X}_{\boldsymbol{\Omega}} = \widetilde{\boldsymbol{X}}_{\boldsymbol{\Omega}} + \boldsymbol{G}_{\boldsymbol{\Omega}} + \boldsymbol{S}_{\boldsymbol{\Omega}}, \tag{11}$$

where $\widetilde{\boldsymbol{X}}_{\boldsymbol{\Omega}}$ is the incomplete noise-free matrix, $\boldsymbol{G}_{\boldsymbol{\Omega}}$ represents the Gaussian noise, and $\boldsymbol{S}_{\boldsymbol{\Omega}}$ signifies the sparse impulses. Based on matrix factorization, we recast the RMC problem as

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{S}} \|\boldsymbol{X}_{\boldsymbol{\Omega}} - (\boldsymbol{U}\boldsymbol{V})_{\boldsymbol{\Omega}} - \boldsymbol{S}_{\boldsymbol{\Omega}}\|_F^2 + \mu\|\boldsymbol{S}_{\boldsymbol{\Omega}}\|_0, \tag{12}$$

where $\|\boldsymbol{S}_{\boldsymbol{\Omega}}\|_0$ is utilized to separate outliers from $\boldsymbol{X}_{\boldsymbol{\Omega}}$, $\|\boldsymbol{X}_{\boldsymbol{\Omega}} - (\boldsymbol{U}\boldsymbol{V})_{\boldsymbol{\Omega}} - \boldsymbol{S}_{\boldsymbol{\Omega}}\|_F^2$ is able to resist the Gaussian noise, and $\mu > 0$ is the penalty parameter which controls the sparsity of $\boldsymbol{S}_{\boldsymbol{\Omega}}$. In our algorithms, μ is automatically updated by a Laplacian kernel based method.

Before proceeding, we analyze the advantage of (12) over the entry-wise ℓ_p-norm based methods and Huber loss based approaches. It is known that the Frobenius norm is superior to the ℓ_p-norm under Gaussian noise. For impulsive noise, it is generally comprised of normal disturbances and outliers. In comparison with the Frobenius norm, the ℓ_p-norm weakens the impact of outliers, but cannot handle Gaussian noise as good as the Frobenius norm. In (12), the ℓ_0-norm and Frobenius norm are able to attain optimality to separate the anomalies and resist the Gaussian noise, respectively. Therefore, our formulation may attain better recovery performance than traditional robust models.

It is clear that (12) is an unconstrained and multivariable optimization problem with three variables, namely, $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{S}$. Therefore, BCD is adopted as the solver, leading to the following iterative procedure:

$$\boldsymbol{U}^{k+1} = \arg\min_{\boldsymbol{U}} \|\boldsymbol{X}_{\boldsymbol{\Omega}} - (\boldsymbol{U}\boldsymbol{V}^k)_{\boldsymbol{\Omega}} - \boldsymbol{S}^k_{\boldsymbol{\Omega}}\|_F^2, \tag{13a}$$

$$\boldsymbol{V}^{k+1} = \arg\min_{\boldsymbol{V}} \|\boldsymbol{X}_{\boldsymbol{\Omega}} - (\boldsymbol{U}^{k+1}\boldsymbol{V})_{\boldsymbol{\Omega}} - \boldsymbol{S}^k_{\boldsymbol{\Omega}}\|_F^2, \tag{13b}$$

$$\boldsymbol{S}^{k+1} = \arg\min_{\boldsymbol{S}} \|\boldsymbol{N}^{k+1}_{\boldsymbol{\Omega}} - \boldsymbol{S}_{\boldsymbol{\Omega}}\|_F^2 + \mu^{k+1}\|\boldsymbol{S}_{\boldsymbol{\Omega}}\|_0, \tag{13c}$$

where $\boldsymbol{N}^{k+1}_{\boldsymbol{\Omega}} = \boldsymbol{X}_{\boldsymbol{\Omega}} - (\boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1})_{\boldsymbol{\Omega}}$. Note that μ is updated by Algorithm 2 before computation of $\boldsymbol{S}^{k+1}$, hence μ^{k+1} replaces μ. It is seen that the BCD alternately optimizes one of the variables with fixing the two remaining variables at each iteration. We first focus on computing $\boldsymbol{U}^{k+1}$ given $\boldsymbol{V}^k$ and $\boldsymbol{S}^k$. As (13a) can be decoupled with respect to (w.r.t.) $\boldsymbol{u}_i^T$, it is equivalent to the following m independent subproblems

$$(\boldsymbol{u}_i^T)^{k+1} = \arg\min_{\boldsymbol{u}_i^T} \|(\boldsymbol{u}_i^T\boldsymbol{V}^k)_{\boldsymbol{\Omega}_i^T} - (\boldsymbol{y}_i^T)_{\boldsymbol{\Omega}_i^T}\|_2^2, \tag{14}$$

where $(\boldsymbol{y}_i^T)_{\boldsymbol{\Omega}_i^T}$ and $\boldsymbol{\Omega}_i^T$ are the ith row of $\boldsymbol{Y}_{\boldsymbol{\Omega}} = \boldsymbol{X}_{\boldsymbol{\Omega}} - \boldsymbol{S}^k_{\boldsymbol{\Omega}}$ and $\boldsymbol{\Omega}$, respectively. It is clear that the residual between $(\boldsymbol{u}_i^T\boldsymbol{V}^k)_{\boldsymbol{\Omega}_i}$ and $(\boldsymbol{y}_i^T)_{\boldsymbol{\Omega}_i}$ is only affected by the observed entries. Therefore, we remove the unobserved elements, leading to

$$(\boldsymbol{u}_i^T)^{k+1} = \arg\min_{\boldsymbol{u}_i^T} \|\boldsymbol{u}_i^T\boldsymbol{A} - \boldsymbol{b}^T\|_2^2, \tag{15}$$

where $\boldsymbol{b}^T \in \mathbb{R}^{\|\boldsymbol{\Omega}_i^T\|_1}$ and $\boldsymbol{A} \in \mathbb{R}^{r\times\|\boldsymbol{\Omega}_i^T\|_1}$ only contain the observed entries of $(\boldsymbol{y}_i^T)_{\boldsymbol{\Omega}_i^T}$ and the corresponding columns of $\boldsymbol{V}^k$, respectively. To be specific, we provide an example for determining $\boldsymbol{A}$ and $\boldsymbol{b}$. Consider $\boldsymbol{V}^k = [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3, \boldsymbol{v}_4, \boldsymbol{v}_5] \in \mathbb{R}^{r\times 5}$ and $(\boldsymbol{y}_i^T)_{\boldsymbol{\Omega}_i^T} = [0, y_2, 0, y_4, 0] \in \mathbb{R}^5$ with $\boldsymbol{\Omega}_i^T = [0, 1, 0, 1, 0]$. Then, we get $\boldsymbol{A} = [\boldsymbol{v}_2, \boldsymbol{v}_4]$ and $\boldsymbol{b}^T = [y_2, y_4]$.

Since (15) is a linear least squares problem, its closed-form solution is

$$(\boldsymbol{u}_i^T)^{k+1} = \boldsymbol{b}^T(\boldsymbol{A})^\dagger, \tag{16}$$

whose computational complexity is $\mathcal{O}(\|\boldsymbol{\Omega}_i^T\|_1 r^2)$.

Problem (13b) has the same structure as (13a) and hence (13b) can be addressed in a similar manner. Specifically, we decompose (13b) into n independent subproblems:

$$\boldsymbol{v}_j^{k+1} = \arg\min_{\boldsymbol{v}_j} \|(\boldsymbol{U}^{k+1}\boldsymbol{v}_j)_{\boldsymbol{\Omega}_j} - (\boldsymbol{y}_j)_{\boldsymbol{\Omega}_j}\|_2^2, \tag{17}$$

where $(\boldsymbol{y}_j)_{\boldsymbol{\Omega}_j}$ and $\boldsymbol{\Omega}_j$ are the jth column of $\boldsymbol{Y}_{\boldsymbol{\Omega}}$ and $\boldsymbol{\Omega}$, respectively. After removing the missing entries, (17) is rewritten as

$$\boldsymbol{v}_j^{k+1} = \arg\min_{\boldsymbol{v}_j} \|\boldsymbol{C}\boldsymbol{v}_j - \boldsymbol{d}\|_2^2, \tag{18}$$

where $\boldsymbol{d} \in \mathbb{R}^{\|\boldsymbol{\Omega}_j\|_1}$ and $\boldsymbol{C} \in \mathbb{R}^{\|\boldsymbol{\Omega}_j\|_1\times r}$ only involve the observed entries of $(\boldsymbol{y}_j)_{\boldsymbol{\Omega}_j}$ and corresponding rows of $\boldsymbol{U}^{k+1}$, respectively. Then, the solution to (18) is

$$\boldsymbol{v}_j^{k+1} = \boldsymbol{C}^\dagger\boldsymbol{d}, \tag{19}$$

with computational complexity of $\mathcal{O}(\|\boldsymbol{\Omega}_j\|_1 r^2)$.

For (13c), its solution is only determined by the observed entries of $\boldsymbol{N}^{k+1}_{\boldsymbol{\Omega}}$. In addition, $s_{i,j}$ only depends on $(n^{k+1})_{i,j}$. We then rewrite (13c) in vector form:

$$\boldsymbol{s}^{k+1} = \arg\min_{\boldsymbol{s}} \|\boldsymbol{n}^{k+1} - \boldsymbol{s}\|_2^2 + \mu^{k+1}\|\boldsymbol{s}\|_0, \tag{20}$$

where $\boldsymbol{s} \in \mathbb{R}^{\|\boldsymbol{\Omega}\|_1}$ and $\boldsymbol{n}^{k+1} \in \mathbb{R}^{\|\boldsymbol{\Omega}\|_1}$. Note that the method to determine μ^{k+1} is presented in the next subsection. The process of attaining $\boldsymbol{n}^{k+1}$ from $\boldsymbol{N}^{k+1}_{\boldsymbol{\Omega}}$ is illustrated as follows

$$\boldsymbol{N}^{k+1}_{\boldsymbol{\Omega}} = \begin{bmatrix} 0 & n_{12} & 0 & n_{14} \\ n_{21} & n_{22} & n_{23} & 0 \end{bmatrix} \tag{21}$$

and the corresponding $\boldsymbol{\Omega}$ is $\boldsymbol{\Omega} = [0, 1, 0, 1; 1, 1, 1, 0]$. Then, we get $\boldsymbol{n}^{k+1} = [n_{21}, n_{12}, n_{22}, n_{23}, n_{14}]^T$.

For (20), its solution is computed as

$$\boldsymbol{s}^{k+1} = \mathcal{T}_{\mu^{k+1}}(\boldsymbol{n}^{k+1}) = \begin{cases} (n^{k+1})_i, & \text{if } |(n^{k+1})_i| \geq \sqrt{\mu^{k+1}} \\ 0, & \text{otherwise} \end{cases}, \tag{22}$$

where $\mathcal{T}_{\mu^{k+1}}(\cdot)$ is a hard-thresholding operator that keeps the values above the threshold, and sets the variables below the threshold to 0 [46]. After obtaining $\boldsymbol{s}^{k+1}$, we update $\boldsymbol{S}^{k+1}$ based on $\boldsymbol{s}^{k+1}$ and $\boldsymbol{\Omega}$ via the inverse operation of getting $\boldsymbol{n}^{k+1}$ from $\boldsymbol{N}^{k+1}_{\boldsymbol{\Omega}}$. The proposed method is named ℓ_0-BCD whose steps are summarized in Algorithm 1. Furthermore, its convergence analysis is provided in Appendix A.

Algorithm 1 ℓ_0-BCD for RMC

Input: $\boldsymbol{X}_\Omega$, Ω, r and $K_{\max}$
  Initialize: Randomly initialize $\boldsymbol{V}^1 \in \mathbb{R}^{r\times n}$, and $\boldsymbol{S}^1 = \boldsymbol{0} \in \mathbb{R}^{m\times n}$
  for $k = 1, 2, \cdots, K_{\max}$ do
    for $i = 1, 2, \cdots, m$ do
      Compute $(\boldsymbol{u}_i^T)^{k+1}$ using (15)
    end for
    for $j = 1, 2, \cdots, n$ do
      Compute $\boldsymbol{v}_j^{k+1}$ using (18)
    end for
    Construct $\boldsymbol{N}_{\boldsymbol{\Omega}}^{k+1} = \boldsymbol{X}_{\boldsymbol{\Omega}} - (\boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1})_{\boldsymbol{\Omega}}$
    Compute $\widetilde{\mu}_{k+1}$ with input $|\boldsymbol{n}^{k+1}|$ via Algorithm 2
    $\mu^{k+1} = \min(\widetilde{\mu}^{k+1}, \mu^k)$
    $\boldsymbol{s}^{k+1} = \mathcal{T}_{\mu^{k+1}}(\boldsymbol{n}^{k+1})$
    Calculate $\boldsymbol{S}^{k+1}$ based on $\boldsymbol{s}^{k+1}$ and $\boldsymbol{\Omega}$
  end for
Output: $\boldsymbol{M} = \boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1}$

B. Adaptive Penalty

In traditional penalty methods, the penalty parameter μ is manually set and fixed during iterations. It is known that if μ is chosen too large or too small, the results may be considerably different. Therefore, a dynamical and adaptive μ may be preferred in real-world applications.

It can be known from (22) that μ in ℓ_0-BCD determines the sparsity of $\boldsymbol{S}_{\boldsymbol{\Omega}}$, that is, the number of outliers. Hence, the core of estimating μ is how to identify anomalies accurately from the residual. Laplacian kernel [47] is one popular method to detect outliers, defined as

$$k_\sigma(x-y) = \exp\left(-\frac{|x-y|}{\sigma}\right). \tag{23}$$

Note that σ is termed as kernel size or bandwidth. It is determined using the concept of kernel density estimation, namely, Silverman's rule [48]. The value of $k_\sigma(x-y)$ decreases as the value of $|x-y|$ increases. Especially, the very large value of $|x-y|$ results in $k_\sigma(x-y) = 0$. Hence, the Laplacian kernel has outlier detection capability [49].

To estimate μ given an input $\boldsymbol{n}$, we first adopt the Silverman's rule to compute the kernel size:

$$\sigma = 1.06 \times \min(\sigma_E, IQR/1.34) \times \dim(\boldsymbol{n})^{-0.2}, \tag{24}$$

where σ_E and IQR are the standard deviation and interquartile range of $\boldsymbol{n}$, respectively. After obtaining σ, we compute $\boldsymbol{w}$ as $\boldsymbol{w} = k_\sigma(\boldsymbol{n})$ where $w_i \le \epsilon$ means that n_i is an outlier. In our method, ϵ is set to 10^{-20}. Based on $\boldsymbol{w}$, we get a coordinate set Ψ of anomalies for $w_i \le \epsilon$ and then μ is calculated as

$$\mu = \min(n_1^2, n_2^2, \cdots, n_i^2) \text{ s.t. } i \in \Psi. \tag{25}$$

Estimation of μ based on the Laplacian kernel is summarized in Algorithm 2. Give $\boldsymbol{n} \in \mathbb{R}^{\|\boldsymbol{\Omega}\|_1}$, the computational complexity for calculating μ is $\mathcal{O}(\|\boldsymbol{\Omega}\|_1)$.

Algorithm 2 Outlier detector based on Laplacian kernel

Input: $\boldsymbol{n}$ and ϵ
  $\sigma^2 = 1.06 \times \min(\sigma_E, IQR/1.34) \times \dim(\boldsymbol{n})^{-0.2}$
  $\boldsymbol{w} = k_\sigma(\boldsymbol{n})$
  $\Psi = \{i\}$ based on $w_i \le \epsilon$
  $\mu = \min(n_1^2, n_2^2, \cdots, n_i^2)$ s.t. $i \in \Psi$
Output: μ and Ψ

C. Fast Variant

In ℓ_0-BCD, we retain the indices of the anomalies in $\boldsymbol{\Omega}$. A novel perspective on solving RMC is that outlier-contaminated elements are treated as missing entries. In accordance with this scheme, we divide $\boldsymbol{\Omega}$ into two binary matrices such that $\boldsymbol{\Omega} = \boldsymbol{\Omega}^g + \boldsymbol{\Omega}^o$. That is, 1 in $\boldsymbol{\Omega}^o$ and $\boldsymbol{\Omega}^g$ denotes an observed entry with and without outlier, respectively. In other words, $\boldsymbol{\Omega}^g$ implies that the observed entries are noise-free or corrupted by Gaussian noise. The computational complexities of calculating $(\boldsymbol{u}_i^T)^{k+1}$ and $\boldsymbol{v}_j^{k+1}$ are $\mathcal{O}(\|\boldsymbol{\Omega}_i^T\|_1 r^2)$ and $\mathcal{O}(\|\boldsymbol{\Omega}_j\|_1 r^2)$, respectively. Simply speaking, the computational cost depends on the number of nonzero entries in $\boldsymbol{\Omega}$. If $\boldsymbol{\Omega}^g$ replaces $\boldsymbol{\Omega}$ to handle MC problem, the computational complexity will reduce because $\|\boldsymbol{\Omega}^g\|_1 \le \|\boldsymbol{\Omega}\|_1$.

Based on the above-mentioned idea, the iterative procedure to solve (12) is rewritten as

$$\boldsymbol{U}^{k+1} = \arg\min_{\boldsymbol{U}} \|\boldsymbol{X}_{\boldsymbol{\Omega}^{g,k}} - (\boldsymbol{U}\boldsymbol{V}^k)_{\boldsymbol{\Omega}^{g,k}} - \boldsymbol{S}^k_{\boldsymbol{\Omega}^{g,k}}\|_F^2, \tag{26a}$$

$$\boldsymbol{V}^{k+1} = \arg\min_{\boldsymbol{V}} \|\boldsymbol{X}_{\boldsymbol{\Omega}^{g,k}} - (\boldsymbol{U}^{k+1}\boldsymbol{V})_{\boldsymbol{\Omega}^{g,k}} - \boldsymbol{S}^k_{\boldsymbol{\Omega}^{g,k}}\|_F^2, \tag{26b}$$

$$\boldsymbol{S}^{k+1} = \arg\min_{\boldsymbol{S}} \|\boldsymbol{N}_{\boldsymbol{\Omega}}^{k+1} - \boldsymbol{S}_{\boldsymbol{\Omega}}\|_F^2 + \mu^{k+1}\|\boldsymbol{S}_{\boldsymbol{\Omega}}\|_0. \tag{26c}$$

Note that the observation set in updating $\boldsymbol{U}^{k+1}$ and $\boldsymbol{V}^{k+1}$ becomes $\boldsymbol{\Omega}^{g,k}$. However, updating $\boldsymbol{S}^{k+1}$ is still based on $\boldsymbol{\Omega}$ since one-off anomaly detection may not be accurate. Computation of $\boldsymbol{S}^{k+1}$ founded on $\boldsymbol{\Omega}$ is able to release the mistaken entries to $\boldsymbol{\Omega}$. The process of solving (26a), (26b), and (26c) is similar to that of ℓ_0-BCD. Accordingly, $\boldsymbol{U}^{k+1}$, $\boldsymbol{V}^{k+1}$ and $\boldsymbol{S}^{k+1}$ are determined analogous to the ℓ_0-BCD. The difference is that the fast method requires computing $\boldsymbol{\Omega}^{g,k}$ before updating $\boldsymbol{U}^{k+1}$.

To determine $\boldsymbol{\Omega}^{g,k}$, we first compute a binary $\boldsymbol{\Omega}^{o,k} \in \mathbb{R}^{m\times n}$ based on $\boldsymbol{S}^k$:

$$(\boldsymbol{\Omega}^{o,k})_{i,j} = \begin{cases} 1, & \text{if } (\boldsymbol{S}^k)_{i,j} \neq 1, \\ 0, & \text{otherwise.} \end{cases} \tag{27}$$

Then, $\boldsymbol{\Omega}^{g,k} = \boldsymbol{\Omega} - \boldsymbol{\Omega}^{o,k}$.

Since $\boldsymbol{S}^k$ is updated iteratively by ℓ_0-BCD, the anomaly impact on computation of $\boldsymbol{U}^k$ and $\boldsymbol{V}^k$ slowly diminishes as iteration increases. In contrast, this fast variant assumes that the entries containing outliers are unobserved. In other words, updating $\boldsymbol{U}^k$ and $\boldsymbol{V}^k$ is not affected by anomalies at the beginning. Compared with the solution which is affected by outliers, the result without the impact of outliers has a smaller estimation error at the same number of iterations. In addition, the fast variant has lower computational complexity than the ℓ_0-BCD and thus

Algorithm 3 ℓ_0-BCD-F for RMC

Input: $\boldsymbol{X}_{\boldsymbol{\Omega}}$, $\boldsymbol{\Omega}$, r and $K_{\max}$
  Initialize: Randomly initialize $\boldsymbol{V}^1 \in \mathbb{R}^{r\times n}$, and $\boldsymbol{S}^1 = \boldsymbol{0} \in \mathbb{R}^{m\times n}$
  for $k = 1, 2, \cdots, K_{\max}$ do
    Calculate $\boldsymbol{\Omega}^{o,k}$ via (27)
    Update $\boldsymbol{\Omega}^{g,k} = \boldsymbol{\Omega} - \boldsymbol{\Omega}^{o,k}$
    Compute $\boldsymbol{U}^{k+1}$ and $\boldsymbol{V}^{k+1}$ based on $\boldsymbol{\Omega}^{g,k}$ using Algorithm 1
    Compute $\boldsymbol{S}^{k+1}$ based on $\boldsymbol{\Omega}$ using Algorithm 1
  end for
Output: $\boldsymbol{M} = \boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1}$

it is termed as ℓ_0-BCD-F. The steps of the ℓ_0-BCD-F are summarized in Algorithm 3.

Note that (26a)–(26c) cannot be considered to address the original problem (12) since the solutions of $\boldsymbol{U}$ and $\boldsymbol{V}$ are computed based on a subset of $\boldsymbol{\Omega}$. Thereby, it is difficult to prove its convergence. Its convergence is verified using empirical results in Section V.

D. Computational Complexity Analysis

The computational complexities of calculating $(\boldsymbol{u}_i^T)^k$ and $\boldsymbol{v}_j^k$ are $\mathcal{O}(\|\boldsymbol{\Omega}_i^T\|_1 r^2)$ and $\mathcal{O}(\|\boldsymbol{\Omega}_j\|_1 r^2)$, respectively. Besides, calculating μ^k requires the complexity of $\mathcal{O}(\|\boldsymbol{\Omega}\|_1)$. Since $\sum_{i=1}^m \|\boldsymbol{\Omega}_i^T\|_1 = \sum_{j=1}^n \|\boldsymbol{\Omega}_j\|_1 = \|\boldsymbol{\Omega}\|_1$, ℓ_0-BCD has the computational complexity of $\mathcal{O}(\|\boldsymbol{\Omega}\|_1 r^2)$ for one iteration.

For ℓ_0-BCD-F, the outlier-contaminated elements are considered as missing entries and hence we have $\|\boldsymbol{\Omega}^g\|_1 < \|\boldsymbol{\Omega}\|_1$. Besides, the ℓ_0-BCD-F's complexity is dominated by the computations of $\boldsymbol{U}^k$ and $\boldsymbol{V}^k$. Accordingly, its complexity is $\mathcal{O}(\|\boldsymbol{\Omega}^g\|_1 r^2)$.

TABLE I: Complexity comparison of different algorithms

Method	Computational complexity
ℓ_0-BCD-F	$\mathcal{O}(\|\boldsymbol{\Omega}^g\|_1 r^2)$
ℓ_0-BCD	$\mathcal{O}(\|\boldsymbol{\Omega}\|_1 r^2)$
ℓ_p-reg	$\mathcal{O}(T\|\boldsymbol{\Omega}\|_1 r^2)$
ℓ_p-ADMM	$\mathcal{O}(T\|\boldsymbol{\Omega}\|_1 r^2)$
M-estimation	$\mathcal{O}(T\|\boldsymbol{\Omega}\|_1 r^2)$
VBMFL1	$\mathcal{O}((m+n)r^3 + mnr^2)$

Table I tabulates the computational requirement of six algorithms where $T > 0$ is the number of inner iterations in the ℓ_p-reg, ℓ_p-ADMM and M-estimation. For VBMFL1, $mn > \|\boldsymbol{\Omega}\|_1$ holds. It is clear that the proposed methods have lower computational complexity than the existing algorithms.

IV. Extension to Nonnegative Matrix Completion

In this section, we extend the ideas in RMC to NRMC, yielding a basic method and its fast variant.

A. Algorithm Development

By modifying RMC, the RNMC formulation is:

$$\min_{\boldsymbol{U}\geq 0,\boldsymbol{V}\geq 0,\boldsymbol{S}} \|\boldsymbol{X}_{\boldsymbol{\Omega}} - (\boldsymbol{U}\boldsymbol{V})_{\boldsymbol{\Omega}} - \boldsymbol{S}_{\boldsymbol{\Omega}}\|_F^2 + \mu\|\boldsymbol{S}_{\boldsymbol{\Omega}}\|_0, \tag{28}$$

where $\boldsymbol{S}$ is not constrained to be positive. Although (28) is a constrained and multi-variable problem, the constraints can be affiliated with each block. Thereby, BCD can also be adopted as the solver, leading to

$$\boldsymbol{U}^{k+1} = \arg\min_{\boldsymbol{U}\geq 0} \|\boldsymbol{X}_{\boldsymbol{\Omega}} - (\boldsymbol{U}\boldsymbol{V}^k)_{\boldsymbol{\Omega}} - \boldsymbol{S}_{\boldsymbol{\Omega}}^k\|_F^2, \tag{29a}$$

$$\boldsymbol{V}^{k+1} = \arg\min_{\boldsymbol{V}\geq 0} \|\boldsymbol{X}_{\boldsymbol{\Omega}} - (\boldsymbol{U}^{k+1}\boldsymbol{V})_{\boldsymbol{\Omega}} - \boldsymbol{S}_{\boldsymbol{\Omega}}^k\|_F^2, \tag{29b}$$

$$\boldsymbol{S}^{k+1} = \arg\min_{\boldsymbol{S}} \|\boldsymbol{N}_{\boldsymbol{\Omega}}^{k+1} - \boldsymbol{S}_{\boldsymbol{\Omega}}\|_F^2 + \mu^{k+1}\|\boldsymbol{S}_{\boldsymbol{\Omega}}\|_0. \tag{29c}$$

It is worth mentioning that (13c) and (29c) are the same. To handle (29a) and (29b), the MM algorithm is exploited. We briefly introduce MM for completeness.

B. Majorization-Minimization

Consider a general optimization problem

$$\min_x h(x) \text{ s.t. } x \in \mathcal{X}, \tag{30}$$

where $h : \mathcal{X} \to \mathbb{R}$ is a continuous function and $\mathcal{X}$ is a nonempty closed set. MM [50], [51] employs a surrogate function $g(x|x^t)$ of $h(x)$ to solve the original problem iteratively. Note that $g(x|x^t)$ should satisfy the following properties:

(i) $h(x^t) = g(x^t|x^t)$.
(ii) $h(x) \leq g(x|x^t)$, s.t. $x \in \mathcal{X}$.
(iii) $\nabla h(x^t) = \nabla g(x^t|x^t)$.

Then, x is iteratively updated as

$$x^{t+1} = \arg\min_x g(x|x^t). \tag{31}$$

We see that MM uses $g(x|x^t)$ to approximate the original function at x^t and then searches for the solution to $g(x|x^t)$. In addition, the convergence of MM has been proved [51].

C. ℓ_0-BCD-MM

We first focus on tackling (29a). From ℓ_0-BCD, it is known that (29a) can be decomposed into m independent subproblems

$$\begin{aligned}(\boldsymbol{u}_i^T)^{k+1} &= \arg\min_{\boldsymbol{u}_i^T\geq 0} \|\boldsymbol{u}_i^T\boldsymbol{A} - \boldsymbol{b}^T\|_2^2 = \arg\min_{\boldsymbol{u}_i\geq 0} \|\boldsymbol{A}^T\boldsymbol{u}_i - \boldsymbol{b}\|_2^2 \\ &= \arg\min_{\boldsymbol{u}_i\geq 0} \boldsymbol{u}_i^T\boldsymbol{A}\boldsymbol{A}^T\boldsymbol{u}_i - 2(\boldsymbol{A}\boldsymbol{b})^T\boldsymbol{u}_i + \boldsymbol{b}^T\boldsymbol{b} \\ &= \arg\min_{\boldsymbol{u}_i\geq 0} \boldsymbol{u}_i^T\boldsymbol{L}\boldsymbol{u}_i - 2(\boldsymbol{A}\boldsymbol{b})^T\boldsymbol{u}_i,\end{aligned} \tag{32}$$

where $\boldsymbol{b}^T\boldsymbol{b}$ is dropped since it is a constant w.r.t. $\boldsymbol{u}_i$, and $\boldsymbol{L} = \boldsymbol{A}\boldsymbol{A}^T \in \mathbb{R}^{r\times r}$. In order to derive a closed-form update to solve (32), we introduce a surrogate function to majorize the quadratic term of $\boldsymbol{u}_i^T\boldsymbol{L}\boldsymbol{u}_i$.

Lemma 1 [52]: Let $\boldsymbol{L} \in \mathbb{R}^{r\times r}$ and $\boldsymbol{Q} \in \mathbb{R}^{r\times r}$ be real symmetric matrices such that $\boldsymbol{Q} \succeq \boldsymbol{L}$. Then, for any vector $\boldsymbol{u} \in \mathbb{R}^r$, the quadratic function $\boldsymbol{u}^T\boldsymbol{L}\boldsymbol{u}$ is majorized at $\boldsymbol{u}^t$ by $\boldsymbol{u}^T\boldsymbol{Q}\boldsymbol{u} + 2\boldsymbol{u}^T(\boldsymbol{L}-\boldsymbol{Q})\boldsymbol{u}^t + (\boldsymbol{u}^t)^T(\boldsymbol{Q}-\boldsymbol{L})\boldsymbol{u}^t$.

Based on Lemma 1, the surrogate function of $\boldsymbol{u}_i^T\boldsymbol{L}\boldsymbol{u}_i$ is formulated as

$$\boldsymbol{u}_i^T\boldsymbol{Q}\boldsymbol{u}_i + 2\boldsymbol{u}_i^T(\boldsymbol{L}-\boldsymbol{Q})\boldsymbol{u}_i^t + (\boldsymbol{u}_i^t)^T(\boldsymbol{Q}-\boldsymbol{L})\boldsymbol{u}_i^t. \tag{33}$$

To meet $\boldsymbol{Q} \succeq \boldsymbol{L}$, we set $\boldsymbol{Q} = \lambda_{\max}\boldsymbol{I}$ where $\lambda_{\max}$ is the largest eigenvalue of $\boldsymbol{L}$. Then, we have

$$\begin{aligned}\boldsymbol{u}_i^T\boldsymbol{L}\boldsymbol{u}_i \leq& \boldsymbol{u}_i^T\boldsymbol{Q}\boldsymbol{u}_i + 2\boldsymbol{u}_i^T(\boldsymbol{L}-\boldsymbol{Q})\boldsymbol{u}_i^t + (\boldsymbol{u}_i^t)^T(\boldsymbol{Q}-\boldsymbol{L})\boldsymbol{u}_i^t\\ =& \lambda_{\max}\boldsymbol{u}_i^T\boldsymbol{u}_i + 2\boldsymbol{u}_i^T(\boldsymbol{L}-\lambda_{\max}\boldsymbol{I})\boldsymbol{u}_i^t + (\boldsymbol{u}_i^t)^T(\boldsymbol{Q}-\boldsymbol{L})\boldsymbol{u}_i^t\\ =& \lambda_{\max}\boldsymbol{u}_i^T\boldsymbol{u}_i + 2\left((\boldsymbol{L}-\lambda_{\max}\boldsymbol{I})\boldsymbol{u}_i^t\right)^T\boldsymbol{u}_i + (\boldsymbol{u}_i^t)^T(\boldsymbol{Q}-\boldsymbol{L})\boldsymbol{u}_i^t,\end{aligned} \tag{34}$$

where the superscript t of $\boldsymbol{u}_i^t$ is the iteration number of the MM method. Hence, the surrogate function $g(\boldsymbol{u}_i|\boldsymbol{u}_i^t)$ of $\boldsymbol{u}_i^T\boldsymbol{L}\boldsymbol{u}_i - 2(\boldsymbol{A}\boldsymbol{b})^T\boldsymbol{u}_i$ in (32) becomes

$$\begin{aligned}g(\boldsymbol{u}_i|\boldsymbol{u}_i^t) =& \lambda_{\max}\boldsymbol{u}_i^T\boldsymbol{u}_i + 2\left((\boldsymbol{L}-\lambda_{\max}\boldsymbol{I})\boldsymbol{u}_i^t - \boldsymbol{A}\boldsymbol{b}\right)^T\boldsymbol{u}_i\\ &+ (\boldsymbol{u}_i^t)^T(\boldsymbol{Q}-\boldsymbol{L})\boldsymbol{u}_i^t.\end{aligned} \tag{35}$$

Then, according to MM, we iteratively update $g(\boldsymbol{u}|\boldsymbol{u}^t)$ to solve (32) by

$$\begin{aligned}\boldsymbol{u}_i^{t+1} &= \arg\min_{\boldsymbol{u}_i\geq 0} g(\boldsymbol{u}_i|\boldsymbol{u}_i^t)\\ &= \arg\min_{\boldsymbol{u}_i\geq 0} \boldsymbol{u}_i^T\boldsymbol{u}_i + (\boldsymbol{q}^t)^T\boldsymbol{u}_i,\end{aligned} \tag{36}$$

where the term $(\boldsymbol{u}_i^t)^T(\boldsymbol{Q}-\boldsymbol{L})\boldsymbol{u}_i^t$ is dropped as it is a constant w.r.t. $\boldsymbol{u}_i$, and $(\boldsymbol{q}^t)^T = \frac{2}{\lambda_{\max}}\left((\boldsymbol{L}-\lambda_{\max}\boldsymbol{I})\boldsymbol{u}_i^t - \boldsymbol{A}\boldsymbol{b}\right)^T$. The closed-form solution to (36) is

$$\boldsymbol{u}_i^{t+1} = \mathcal{I}(\boldsymbol{q}^t) = \begin{cases} -\frac{q_i^t}{2}, & \text{if } q_i^t < 0,\\ 0, & \text{otherwise.}\end{cases} \tag{37}$$

The proof of the closed-form solution is provided in Appendix B. When $\boldsymbol{u}_i^{t+1}$ converges, it becomes the optimal solution to (32).

For (29b), it can be handled in a similar way, leading to the following n independent subproblems:

$$\begin{aligned}\boldsymbol{v}_j^{k+1} &= \arg\min_{\boldsymbol{v}_j\geq 0} \|\boldsymbol{C}\boldsymbol{v}_j - \boldsymbol{d}\|_2^2\\ &= \arg\min_{\boldsymbol{v}_j\geq 0} \boldsymbol{v}_j^T\boldsymbol{C}^T\boldsymbol{C}\boldsymbol{v}_j - 2(\boldsymbol{C}^T\boldsymbol{d})^T\boldsymbol{v}_j + \boldsymbol{d}^T\boldsymbol{d}\\ &= \arg\min_{\boldsymbol{v}_j\geq 0} \boldsymbol{v}_j^T\boldsymbol{L}\boldsymbol{v}_j - 2(\boldsymbol{C}^T\boldsymbol{d})^T\boldsymbol{v}_j,\end{aligned} \tag{38}$$

where $\boldsymbol{L} = \boldsymbol{C}^T\boldsymbol{C}$. It is clear that (38) is equivalent to (32). Thus, $\boldsymbol{v}_j^{k+1}$ can be sought via the same way

$$\boldsymbol{v}_j^{t+1} = \arg\min_{\boldsymbol{v}_j\geq 0} \boldsymbol{v}_j^T\boldsymbol{v}_j + (\boldsymbol{q}^t)^T\boldsymbol{v}_j, \tag{39}$$

where $(\boldsymbol{q}^t)^T = \frac{2}{\lambda_{\max}}((\boldsymbol{L}-\lambda_{\max}\boldsymbol{I})\boldsymbol{v}_j^t - \boldsymbol{C}^T\boldsymbol{d})^T$. When $\boldsymbol{v}_j^{t+1}$ converges, $\boldsymbol{v}_j^{k+1}$ is determined, that is, $\boldsymbol{v}_j^{k+1} = \boldsymbol{v}_j^{t+1}$.

Finally, (29c) is the same as (13c) in the ℓ_0-BCD and hence $\boldsymbol{S}^{k+1}$ can be computed as in Algorithm 1.

Algorithm 4 summarizes the steps of ℓ_0-BCD-MM. Note that there are three layers of iterative updates. The outer updates on $\boldsymbol{U}^{k+1}$, $\boldsymbol{V}^{k+1}$, and $\boldsymbol{S}^{k+1}$ correspond to the BCD method. The middle layer calculates $\boldsymbol{u}_i^{k+1}$ and $\boldsymbol{v}_j^{k+1}$ of $\boldsymbol{U}^{k+1}$ and $\boldsymbol{V}^{k+1}$, respectively. The inner refers to the iteration of the MM algorithm. Parallel and distributed computing can be used to calculate $\boldsymbol{u}_i^{k+1}$ and $\boldsymbol{v}_j^{k+1}$ and thus the computational efficiency can be greatly increased.

Algorithm 4 ℓ_0-BCD-MM for RNMC

Input: $\boldsymbol{X}_{\boldsymbol{\Omega}}$, $\boldsymbol{\Omega}$, r and $K_{\max}$
  Initialize: Randomly initialize $\boldsymbol{V}^1 \in \mathbb{R}^{r\times n}$, and $\boldsymbol{S}^1 = \boldsymbol{0} \in \mathbb{R}^{m\times n}$
  for $k = 1, 2, \cdots, K_{\max}$ do
    for $i = 1, 2, \cdots, m$ do
      Randomly initialize $\boldsymbol{u}_i^0 \in \mathbb{R}^r$
      Compute $\lambda_{\max}$ based on $\boldsymbol{L} = \boldsymbol{A}\boldsymbol{A}^T$
      for $t = 1, 2, \cdots$ do
        $\boldsymbol{q}^t = \frac{2}{\lambda_{\max}}((\boldsymbol{L}-\lambda_{\max}\boldsymbol{I})\boldsymbol{u}_i^t - \boldsymbol{A}\boldsymbol{b})$
        $\boldsymbol{u}_i^{t+1} = \mathcal{I}(\boldsymbol{q}^t)$
        Stop if stopping criterion is met.
      end for
      $\boldsymbol{u}_i^{k+1} = \boldsymbol{u}_i^{t+1}$
    end for
    for $j = 1, 2, \cdots, n$ do
      Randomly initialize $\boldsymbol{v}_j^0 \in \mathbb{R}^r$
      Compute $\lambda_{\max}$ based on $\boldsymbol{L} = \boldsymbol{C}^T\boldsymbol{C}$
      for $t = 1, 2, \cdots$ do
        $\boldsymbol{q}^t = \frac{2}{\lambda_{\max}}((\boldsymbol{L}-\lambda_{\max}\boldsymbol{I})\boldsymbol{v}_j^t - \boldsymbol{C}^T\boldsymbol{d})$
        $\boldsymbol{v}_j^{t+1} = \mathcal{I}(\boldsymbol{q}^t)$
        Stop if stopping criterion is met.
      end for
      $\boldsymbol{v}_j^{k+1} = \boldsymbol{v}_j^{t+1}$
    end for
    Construct $\boldsymbol{N}_{\boldsymbol{\Omega}}^{k+1} = \boldsymbol{X}_{\boldsymbol{\Omega}} - (\boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1})_{\boldsymbol{\Omega}}$
    Compute $\widetilde{\mu}_{k+1}$ with input $|\boldsymbol{n}^{k+1}|$ via Algorithm 2
    Update $\mu^{k+1} = \min(\widetilde{\mu}_{k+1}, \mu^k)$
    Compute $\boldsymbol{s}^{k+1} = \mathcal{T}_{\mu^{k+1}}(\boldsymbol{n}^{k+1})$
    Update $\boldsymbol{S}^{k+1}$ based on $\boldsymbol{s}^{k+1}$ and $\boldsymbol{\Omega}$
  end for
Output: $\boldsymbol{M} = \boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1}$

Moreover, the termination condition for $\boldsymbol{u}^{t+1}$ is suggested as $\|\boldsymbol{u}^{t+1} - \boldsymbol{u}^t\|_2^2/\dim(\boldsymbol{u}) < 10^{-5}$ which means that the average power difference between adjacent iterations of each entry is less than 10^{-5}. Similarly, $\boldsymbol{v}^{t+1}$ also adopts this stopping criterion.

Regarding its objective value convergence, the proof of ℓ_0-BCD is applicable. Since the convergence of MM algorithm is ensured [51], $\boldsymbol{u}_i^{k+1}$ and $\boldsymbol{v}_j^{k+1}$ are the optimal solutions to the convex problems (32) and (38), respectively. That is, the updates of $\boldsymbol{U}^{k+1}$ and $\boldsymbol{V}^{k+1}$ by Algorithm 4 keep the objective value nonincreasing. Thus, in accordance to the convergence analysis of the ℓ_0-BCD, the ℓ_0-BCD-MM's convergence is guaranteed.

D. Fast Variant

The strategy in ℓ_0-BCD-F can be used to speed up the ℓ_0-BCD-MM. When we treat the outlier-contaminated elements as unobserved entries, the ℓ_0-BCD-MM evolves into ℓ_0-BCD-MM-F. The steps of the ℓ_0-BCD-MM-F are summarized in Algorithm 5.

Algorithm 5 ℓ_0-BCD-MM-F for RNMC

Input: $\boldsymbol{X}_{\boldsymbol{\Omega}}$, $\boldsymbol{\Omega}$, r and $K_{\max}$
  Initialize: Randomly initialize $\boldsymbol{V}^1 \in \mathbb{R}^{r\times n}$, and $\boldsymbol{S}^1 = \boldsymbol{0} \in \mathbb{R}^{m\times n}$.
  for $k = 1, 2, \cdots, K_{\max}$ do
    Calculate $\boldsymbol{\Omega}^{o,k}$ via (27)
    Update $\boldsymbol{\Omega}^{g,k} = \boldsymbol{\Omega} - \boldsymbol{\Omega}^{o,k}$
    Compute $\boldsymbol{U}^{k+1}$ and $\boldsymbol{V}^{k+1}$ based on $\boldsymbol{\Omega}^{g,k}$ using Algorithm 4
    Compute $\boldsymbol{S}^{k+1}$ based on $\boldsymbol{\Omega}$ using Algorithm 4
  end for
Output: $\boldsymbol{M} = \boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1}$

E. Computational Complexity Analysis

To facilitate presentation, we assume that $\|\boldsymbol{\Omega}_i^T\|_1 > \|\boldsymbol{\Omega}_j\|_1$. For ℓ_0-BCD-MM, the computational complexity for $\boldsymbol{L}$ is $\mathcal{O}(r^2\|\boldsymbol{\Omega}_i^T\|_1)$. To compute $\lambda_{\max}$, the complexity is $\mathcal{O}(r^3)$. The MM computational complexity is $\mathcal{O}(rT\|\boldsymbol{\Omega}_i^T\|_1)$ where T is the required iteration number. Thus, the total computational complexity is $\mathcal{O}(rT\|\boldsymbol{\Omega}\|_1)$ because $T \gg r$ and $\sum_{i=1}^m \|\boldsymbol{\Omega}_i^T\|_1 = \|\boldsymbol{\Omega}\|_1$. On the other hand, the computational complexity of ℓ_0-BCD-MM-F is $\mathcal{O}(rT\|\boldsymbol{\Omega}^g\|_1)$.

V. Simulation Results

In this section, the proposed methods are evaluated using synthetic and real-world data. In detail, subsections I and II examine our RMC algorithms, while the remaining two subsections test the suggested NRMC approaches.

A. Synthetic data

The noise-free and complete matrix $\boldsymbol{X} \in \mathbb{R}^{400\times 500}$ with $r = 10$ is generated by the product of $\boldsymbol{X}_1 \in \mathbb{R}^{400\times 10}$ and $\boldsymbol{X}_2 \in \mathbb{R}^{10\times 500}$ whose entries obey the standard Gaussian distribution. Then, the incomplete matrix without noise $\widetilde{\boldsymbol{X}}_{\boldsymbol{\Omega}}$ consists of randomly selected 50% entries from $\boldsymbol{X}$. In other words, 50% entries of $\boldsymbol{\Omega}$ are equal to 1, and the rest are 0. Moreover, $\widetilde{\boldsymbol{X}}_{\boldsymbol{\Omega}}$ is added with independent impulsive noise which is modeled by Gaussian mixture model (GMM). The probability density function (PDF) of GMM is given by

$$p_v(v) = \frac{c_1}{\sqrt{2\pi}\sigma_1}\exp\left(-\frac{v^2}{2\sigma_1^2}\right) + \frac{c_2}{\sqrt{2\pi}\sigma_2}\exp\left(-\frac{v^2}{2\sigma_2^2}\right), \tag{40}$$

where $c_1 + c_2 = 1$ with $0 < c_i < 1$, σ_1^2 and σ_2^2 are variances. To simulate the impulsive noise, it requires $\sigma_2^2 \gg \sigma_1^2$ and $c_2 < c_1$. It means that sparse and high power noise samples with σ_2^2 and c_2 are mixed in Gaussian background noise with small variance σ_1^2. In our simulations, we set $\sigma_2^2 = 100\sigma_1^2$ and $c_2 = 0.1$. The signal-to-noise ratio (SNR) is defined as

$$\text{SNR} = \frac{\|\widetilde{\boldsymbol{X}}_{\boldsymbol{\Omega}}\|_F^2}{\|\boldsymbol{\Omega}\|_1\sigma_v^2}, \tag{41}$$

where $\sigma_v^2 = \sum_{i=1}^2 c_i\sigma_i^2$ is the total noise variance.

The recovery performance is measured by mean square error (MSE), defined as

$$\text{MSE} = \frac{\|\boldsymbol{M} - \boldsymbol{X}\|_F^2}{mn}, \tag{42}$$

where $\boldsymbol{M}$ is the recovered matrix.

1) Convergence Behavior: We first present the convergence behaviors of the ℓ_0-BCD and ℓ_0-BCD-F. Fig. (S.1) in the supplementary material plots the objective value convergence where "Loss" denotes the value of $\mathcal{L}_{\mu^k}(\boldsymbol{U}^k, \boldsymbol{V}^k, \boldsymbol{S}^k) = \|\boldsymbol{X}_{\boldsymbol{\Omega}} - (\boldsymbol{U}^k\boldsymbol{V}^k)_{\boldsymbol{\Omega}} - \boldsymbol{S}_{\boldsymbol{\Omega}}^k\|_F^2 + \mu^k\|\boldsymbol{S}_{\boldsymbol{\Omega}}^k\|_0$. It is seen that the objective value of the proposed algorithms is nonincreasing, and converges within 10 iterations.

Besides, Figs. (S.2) and (S.3) show the sequence convergence of the ℓ_0-BCD and ℓ_0-BCD-F, respectively. Since there are numerous entries of $\boldsymbol{U}^k$, $\boldsymbol{V}^k$, and $\boldsymbol{S}^k$, we only plot partial entries, that is, $u_{i,i}$, $v_{i,i}$ and $s_{i,i}$. As the rank is 10, there are ten curves for $\boldsymbol{U}^k$ and $\boldsymbol{V}^k$. Besides, the number of curves for $\boldsymbol{S}^k$ is much less than 400 because of its sparsity. It is observed that the sequence $\{\boldsymbol{U}^k, \boldsymbol{V}^k, \boldsymbol{S}^k\}$ is able to converge within 10 iterations.

Moreover, we investigate the impact of initialization, namely, $\boldsymbol{V}$, in 50% randomly missing data and 3dB GMM noise. Fig. (S.4) plots 100 convergence curves for each algorithm where $\boldsymbol{V}$ are randomly initialized. Specifically, the average steady-state MSEs of the ℓ_0-BCD-F and ℓ_0-BCD are 5.4582×10^{-2} and 5.4592×10^{-2}, respectively. Hence, their performance seems insensitive to the parameter initialization.

2) Performance Comparison: We compare the ℓ_0-BCD-F and ℓ_0-BCD with popular methods, including ℓ_p-reg [25], ℓ_p-ADMM [25], M-estimation [27], VBMFL1 [31], and ORMC [22]. Fig. 1 depicts the MSE convergence performance of different methods under 50% randomly missing data and additive GMM noise of SNR = 3dB. It is observed that the ℓ_0-BCD-F and ℓ_0-BCD have a similar steady-state MSE which is lower than that of the competing algorithms. Hence, the proposed methods have superior recovery performance over the existing approaches. As the ORMC is designed for the situation that the observed matrix has a few Gaussian noise-contaminated columns, it cannot achieve satisfactory performance in impulsive noise.

The runtimes are tabulated in Table II where the stopping condition is MSE < 0.1. Since the ORMC cannot attain this criterion, its result is not presented. We see that the elapsed time of our algorithms is much less than that of the competitors. In addition, the ℓ_0-BCD-F is faster than ℓ_0-BCD.

TABLE II: Elapsed time of different methods.

Method	ℓ_0-BCD-F	ℓ_0-BCD	ℓ_p-reg	ℓ_p-ADMM	M-estimation	VBMFL1
Time (s)	0.12	0.26	25.99	9.33	2.01	1.02

The impact of the percentage of missing data with 3dB GMM noise is plotted in Fig. 2. We see that the MSEs of ℓ_0-BCD-F and ℓ_0-BCD are the smallest among the seven algorithms for all percentages. Thus, our methods achieve

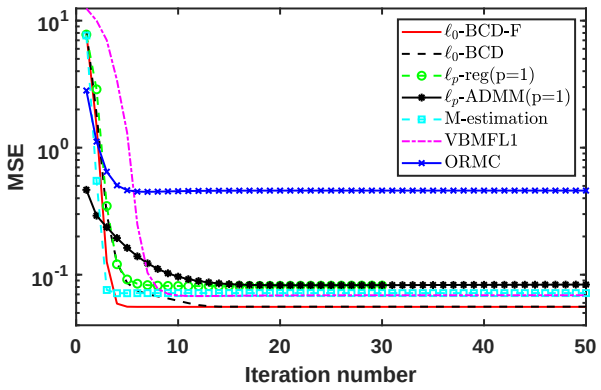

Fig. 1: MSE versus iteration number with 50% randomly missing data and SNR=3dB GMM noise.

superior recovery performance over the ℓ_p-reg, ℓ_p-ADMM, M-estimation, VBMFL1, and ORMC. It is worth noting that the ℓ_0-BCD-F has comparable performance to ℓ_0-BCD, indicating that the scheme of discarding the outlier-contaminated entries is feasible.

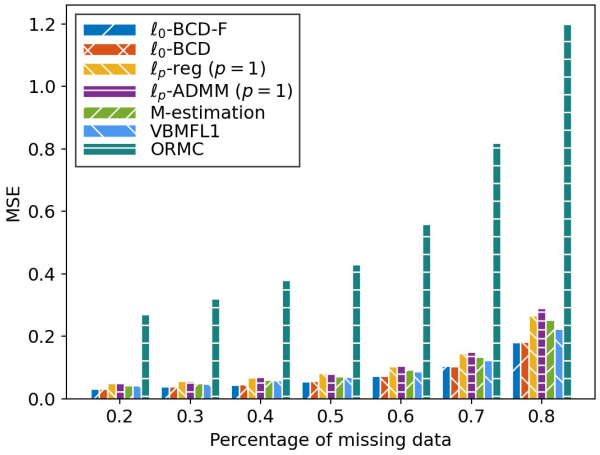

Fig. 2: MSE versus percentage of randomly missing data by different methods under 3dB GMM noise.

The performance under different SNRs is investigated in Fig. 3. As the MSEs of the ORMC are too large such that it is difficult to differentiate the remaining algorithms, its results are not included. We clarify that, under 9dB GMM noise, the MSEs of the ℓ_0-BCD-F, ℓ_0-BCD, ℓ_p-reg, ℓ_p-ADMM, M-estimation and VBMFL1 are 0.01427, 0.01434, 0.01996, 0.02074, 0.01791, and 0.01737, respectively. It is seen that our developed methods outperform the competing algorithms at different levels of the impulsive noise.

B. Image Inpainting

A popular application of MC is gray-scale image inpainting [26]. Images, in practice, may not be fully captured due to damage to the photosensitive device or the shadow from other objects. In addition, the image data may be mixed with impulsive noise during wireless transmission or bit errors in the signal acquisition stage. Although the impulse

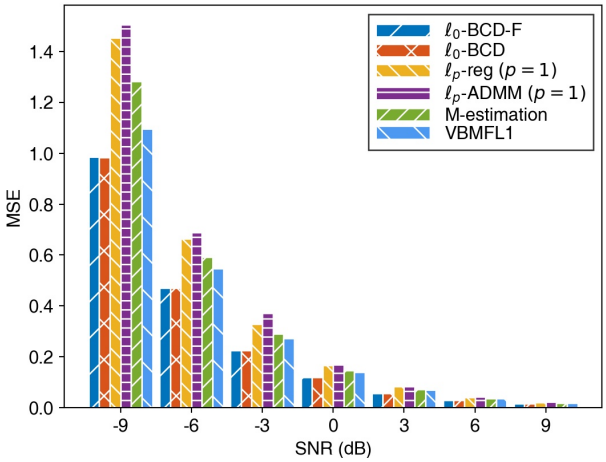

Fig. 3: MSE versus SNR by different algorithms with 50% randomly missing data.

amplitude may be very large, they are set to the largest or smallest admissible values after quantization, giving the image a "salt and pepper" like appearance. For 8-bit images, the maximum and minimum values are 0 (black) and 255 (white), respectively. The salt-and-pepper noise has a noise density coefficient, denoted as τ. Note that the relationship between τ and SNR is $\tau = 1/\text{SNR}$ [25]. For a given value of τ, approximately 0.5τ of the pixels become 255 or 0. Fig. 4 shows the noiseless and noisy pictures at $\tau = 0.2$.

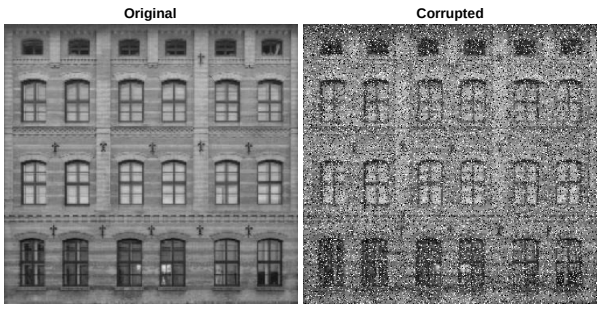

Fig. 4: Illustration of original image and noisy image corrupted by salt-and-pepper noise with SNR = 5dB.

It is worth mentioning that the PDF of the salt-and-pepper noise does not comply with GMM and hence the proposed algorithms cannot be directly adopted to resist the salt-and-pepper noise. Since the pixels which are corrupted by salt-and-pepper noise are either 255 or 0, it is easy to detect the location of outliers. In general, dispersed black and white pixels are not the integers of 0 and 255, respectively, but have values close to 0 and 255. If there exists an area of white or black, it is not difficult to differentiate it and the salt-and-pepper noise. This is because that the noise is discrete, while the area is continuous. Therefore, we directly utilize 255 and 0 to identify the salt and pepper pixels, which is also adopted

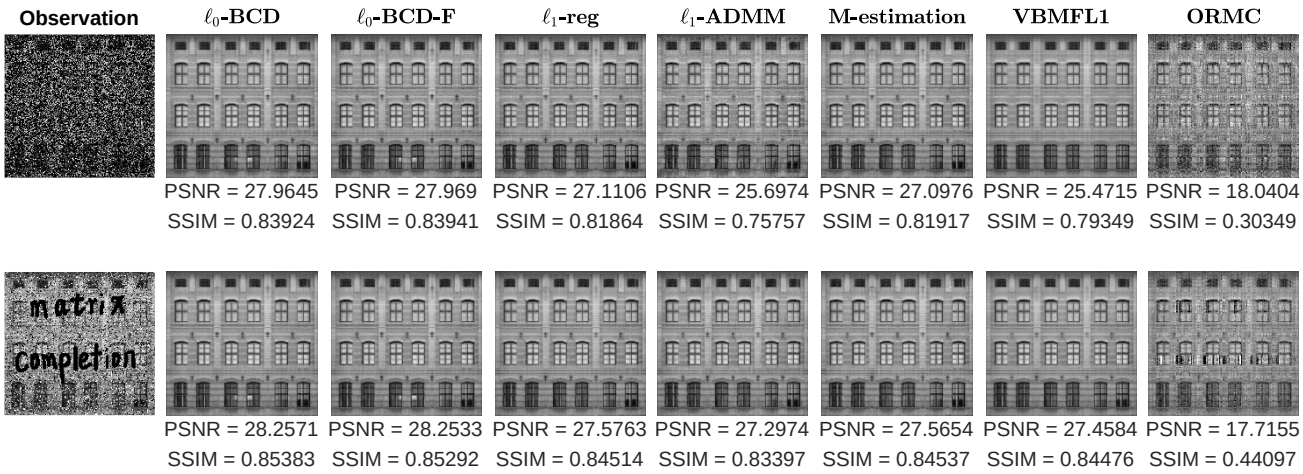

Fig. 5: Performance of different approaches in two types of masks with 5dB salt-and-pepper noise. The left column shows observed pictures with different masks, while the recovered images are shown from the second column to the end.

TABLE III: PSNR and SSIM versus percentage of randomly observed data with 5dB salt-and-pepper noise by different methods.

Percentage	Metric	ℓ_0-BCD	ℓ_0-BCD-F	ℓ_p-reg	ℓ_p-ADMM	M-estimation	VBMFL1	ORMC
0.2	PSNR	23.0734	24.2082	15.8689	13.8750	16.2060	22.6711	13.5667
	SSIM	0.6704	0.7096	0.3017	0.2402	0.3082	0.6673	0.1277
0.3	PSNR	26.9514	26.9993	20.3070	17.7409	20.1598	23.8800	15.2891
	SSIM	0.8025	0.8046	0.5684	0.4853	0.5683	0.7296	0.1955
0.4	PSNR	27.6047	27.6657	25.6055	19.4561	25.5207	24.6114	17.0142
	SSIM	0.8263	0.8276	0.7690	0.6312	0.7711	0.7586	0.2731
0.5	PSNR	27.9645	27.9690	27.1106	25.6974	27.0976	25.4715	18.0404
	SSIM	0.8392	0.8394	0.8186	0.7576	0.8192	0.7935	0.3035
0.6	PSNR	28.1555	28.1714	27.4636	26.7988	27.3794	25.5083	19.0859
	SSIM	0.8448	0.8459	0.8272	0.8103	0.8272	0.7958	0.3761
0.7	PSNR	28.2917	28.2884	27.4390	27.3343	27.6063	27.4049	19.7565
	SSIM	0.8503	0.8497	0.8335	0.8234	0.8392	0.8382	3.9845
0.8	PSNR	28.3515	28.3514	27.6399	27.2980	27.8382	27.5732	20.0245
	SSIM	0.8530	0.8524	0.8442	0.8311	0.8423	0.8440	0.4345
0.9	PSNR	28.4173	28.4182	27.7497	27.3501	27.7473	27.6420	20.7896
	SSIM	0.8547	0.8546	0.8474	0.8342	0.8477	0.8462	0.4675

by [53]. Based on the above strategy, (13a)–(13c) are simplified as

$$\boldsymbol{U}^{k+1} = \arg\min_{\boldsymbol{U}} \|\boldsymbol{X}_{\boldsymbol{\Omega}} - (\boldsymbol{U}\boldsymbol{V}^k)_{\boldsymbol{\Omega}} - \boldsymbol{S}^k_{\boldsymbol{\Omega}}\|_F^2, \tag{43a}$$

$$\boldsymbol{V}^{k+1} = \arg\min_{\boldsymbol{V}} \|\boldsymbol{X}_{\boldsymbol{\Omega}} - (\boldsymbol{U}^{k+1}\boldsymbol{V})_{\boldsymbol{\Omega}} - \boldsymbol{S}^k_{\boldsymbol{\Omega}}\|_F^2, \tag{43b}$$

$$\boldsymbol{S}^{k+1} = \boldsymbol{X}_{\boldsymbol{\Omega}^o} - (\boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1})_{\boldsymbol{\Omega}^o}, \tag{43c}$$

where $\boldsymbol{\Omega}^o \in \mathbb{R}^{m\times n}$ is the subset of $\boldsymbol{\Omega}$, and contains the indices of outliers. Similarly, (26a)–(26c) can be re-expressed as

$$\boldsymbol{U}^{k+1} = \arg\min_{\boldsymbol{U}} \|(\boldsymbol{U}\boldsymbol{V}^k)_{\boldsymbol{\Omega}-\boldsymbol{\Omega}^o} - \boldsymbol{X}_{\boldsymbol{\Omega}-\boldsymbol{\Omega}^o}\|_F^2, \tag{44a}$$

$$\boldsymbol{V}^{k+1} = \arg\min_{\boldsymbol{V}} \|(\boldsymbol{U}^{k+1}\boldsymbol{V})_{\boldsymbol{\Omega}-\boldsymbol{\Omega}^o} - \boldsymbol{X}_{\boldsymbol{\Omega}-\boldsymbol{\Omega}^o}\|_F^2. \tag{44b}$$

It is seen that RMC is converted into a tractable problem with the salt-and-pepper noise.

One image, called windows, is first used to compare the proposed algorithms with existing approaches, including ℓ_p-reg, ℓ_p-ADMM, M-estimation, VBMFL1 and ORMC. All algorithms use the same rank of 10. Fig. 4 shows the original image and the noisy image with salt-and-pepper noise. The recovery performance is measured by two metrics, namely, peak SNR (PSNR) and structural similarity (SSIM). They can be computed by the built-in MATLAB commands, i.e., 'psnr(recovered, original)' and 'ssim(recovered, original)'. Larger values of PSNR and SSIM mean that the recovery performance is better.

Fig. 5 depicts the observed pictures and recovered images by seven methods where two types of masks, namely, random and fixed masks are investigated. The random mask implies that the image has 50% randomly distributed missing entries, while the fixed mask contains the distorted text of "matrix completion". It is observed that the PSNRs and SSIMs of ℓ_0-BCD and ℓ_0-BCD-F are larger than those of the competing algorithms. Hence, the devised methods outperform the existing approaches in these two missing pattern types.

Moreover, the impact of the percentage of randomly

TABLE IV: PSNR and SSIM of different algorithms on eight images with two types of masks and 5dB salt-and-pepper noise.

Image	Mask	Metric	ℓ_0-BCD	ℓ_0-BCD-F	ℓ_p-reg	ℓ_p-ADMM	M-estimation	VBMFL1	ORMC
Image-1	random	PSNR	29.7972	29.7951	25.9674	21.6455	26.3926	27.4465	19.3620
		SSIM	0.8652	0.8662	0.7878	0.6129	0.8026	0.8173	0.3426
	fixed	PSNR	30.1949	30.1836	29.5219	28.7051	29.3843	28.9289	19.8721
		SSIM	0.8785	0.8783	0.8697	0.8528	0.8703	0.8579	0.4654
Image-2	random	PSNR	29.2469	29.2203	27.6974	25.3052	27.7281	24.2193	19.2240
		SSIM	0.8354	0.8361	0.8191	0.7161	0.8278	0.8060	0.2020
	fixed	PSNR	27.8409	26.3372	25.7804	24.8006	27.8379	25.7197	18.4366
		SSIM	0.8873	0.8880	0.8768	0.7798	0.8871	0.8542	0.3291
Image-3	random	PSNR	22.3954	22.4208	21.3991	20.7240	21.2911	21.3308	17.2389
		SSIM	0.5354	0.5304	0.5115	0.4491	0.5087	0.5115	0.1866
	fixed	PSNR	21.1611	19.9556	20.117	17.223	20.575	20.2249	17.3795
		SSIM	0.6585	0.6588	0.6309	0.5543	0.6355	0.6086	0.2019
Image-4	random	PSNR	31.1593	31.8156	27.4336	22.2496	28.7826	28.0039	19.5554
		SSIM	0.8304	0.8409	0.7636	0.6178	0.7748	0.7579	0.1788
	fixed	PSNR	30.8210	31.4570	30.0173	28.3492	30.7888	30.2283	19.6228
		SSIM	0.8820	0.8847	0.8601	0.7533	0.8659	0.8279	0.2966
Image-5	random	PSNR	24.8283	24.8349	23.6927	23.4951	23.5805	23.8226	18.2287
		SSIM	0.6753	0.6756	0.6717	0.6442	0.6665	0.6604	0.2174
	fixed	PSNR	25.5551	24.6314	22.7487	21.0082	23.4980	24.5877	17.6406
		SSIM	0.8264	0.8280	0.7619	0.7287	0.7739	0.7784	0.3077
Image-6	random	PSNR	24.4083	24.4225	23.2758	22.5301	23.1543	23.2736	18.2118
		SSIM	0.6581	0.6540	0.6451	0.5493	0.6539	0.6515	0.2019
	fixed	PSNR	22.7007	22.3292	19.1339	19.3220	20.5518	22.3266	17.8605
		SSIM	0.7282	0.7286	0.7104	0.6939	0.7192	0.7269	0.3081
Image-7	random	PSNR	25.8709	25.8744	24.6104	23.9876	24.5643	23.5304	18.1187
		SSIM	0.8902	0.8912	0.8764	0.8574	0.8756	0.8640	0.5959
	fixed	PSNR	25.1458	25.3920	24.2575	23.8491	23.9278	24.8200	18.2791
		SSIM	0.8937	0.8976	0.8902	0.8865	0.8896	0.8963	0.6706
Image-8	random	PSNR	22.8408	22.8355	21.6169	20.5329	21.4327	20.8915	17.0360
		SSIM	0.5840	0.5837	0.5813	0.4894	0.5795	0.5832	0.1966
	fixed	PSNR	21.5860	21.6376	20.8151	19.4291	19.8861	21.5761	17.5104
		SSIM	0.6261	0.6283	0.6243	0.6198	0.6164	0.6263	0.2028

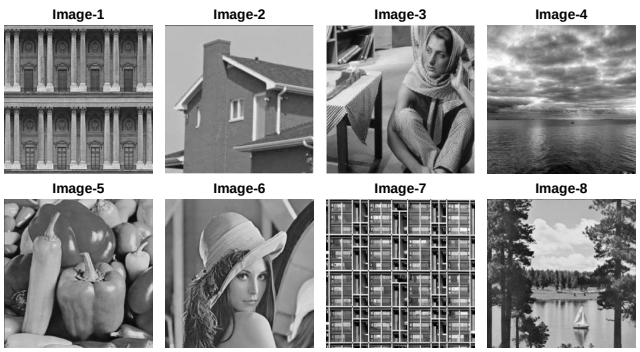

Fig. 6: Original images for image inpainting.

missing data on recovery performance is tabulated in Table III under 5dB salt-and-pepper noise. We see that the PSNRs and SSIMs of the proposed methods are higher than those of ℓ_p-reg, ℓ_p-ADMM, M-estimation, VBMFL1 and ORMC in different ratios.

Furthermore, eight well-known images, depicted in Fig. 6, are selected to evaluate the inpainting performance. Each picture is evaluated with the random and fixed masks under the previous settings. The comparison results are tabulated in Table IV. It is seen that, in most cases, the ℓ_0-BCD and ℓ_0-BCD-F show remarkable superiority over their competitors. In a few cases, the metric values of the suggested algorithms are slightly smaller than those of the existing methods, such as the PSNR of the ℓ_0-BCD-F in Image-3, as well as SSIM of ℓ_0-BCD in Image-7 and Image-8 in the text mask scenarios.

C. Nonnegative Synthetic data

In this subsection, ℓ_0-BCD-MM and ℓ_0-BCD-MM-F are examined using synthetic data. A noise-free matrix $\boldsymbol{X} \in \mathbb{R}^{400\times 500}$ of $r = 10$ is generated by the product of $|\boldsymbol{X}_1| \in \mathbb{R}^{400\times 10}$ and $|\boldsymbol{X}_2| \in \mathbb{R}^{10\times 500}$ where entries of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ satisfy the standard Gaussian distribution. We randomly select 50% entries of $\boldsymbol{X}$ as the incomplete observations and then add GMM noise of SNR = 6dB. Two methods, namely, MF-ADMM [38] and LRA-ADMM [39], are compared with our approaches.

Fig. 7 illustrates the MSE convergence performance. It is observed that the recovery errors of the two proposed methods are lower than those of the competing algorithms. This is because the MF-ADMM and LRA-ADMM do not take impulsive noise into account in their formulation.

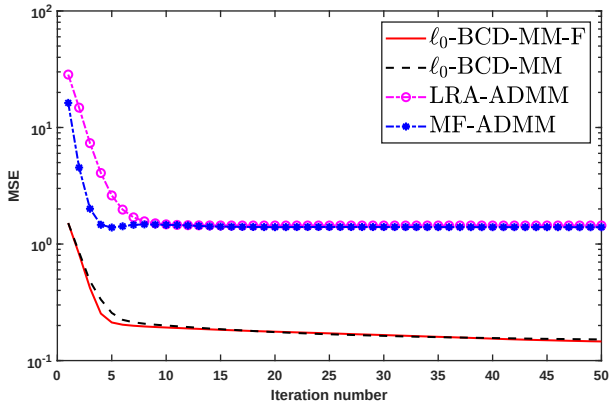

Fig. 7: MSE versus iteration number with 50% missing data and 6dB GMM noise.

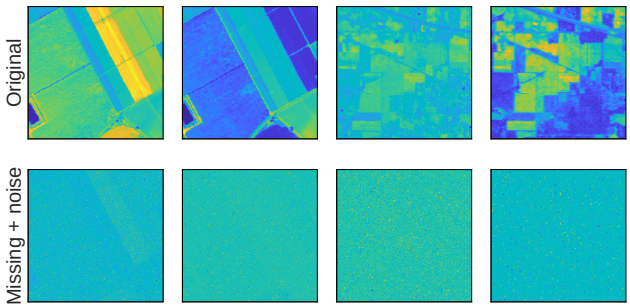

Fig. 8: Original images of Salinas and Indian pines scenes, and corresponding incomplete images with 30% randomly missing data and 6dB GMM noise.

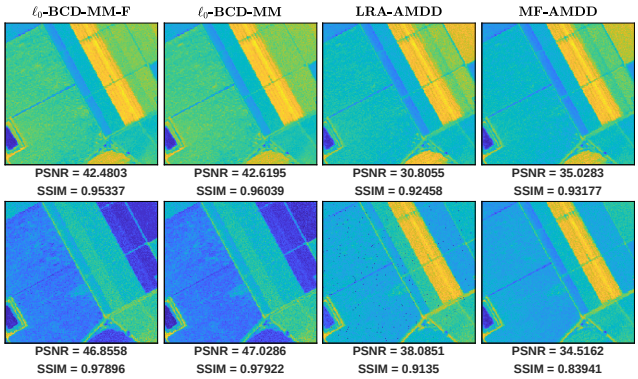

Fig. 9: Recovered images by ℓ_0-BCD-MM-F, ℓ_0-BCD-MM, LRA-ADMM and MF-ADMM using Salinas dataset.

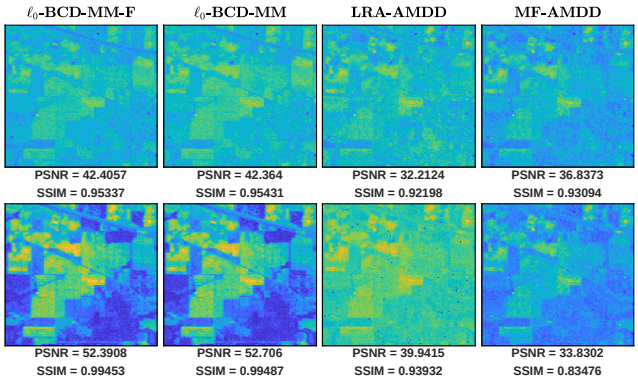

Fig. 10: Recovered images by ℓ_0-BCD-MM-F, ℓ_0-BCD-MM, LRA-ADMM and MF-ADMM using Indian pines dataset.

D. Hyperspectral Imaging

There are two approaches to solve hyperspectral imaging, namely, matrix-based and tensor-based methods. Comparison of these two strategies is studied in [54]. The results show that matrix-based methods outperform tensor-based approaches when the spectral bands are sufficient. This is because restoring unobserved entries are difficult at a certain frequency based on the same location in other frequencies when spectral bands are limited.

In this subsection, we apply the proposed matrix-based algorithms to restore hyperspectral data. Two open-source datasets are utilized, i.e., Indian pines and Salinas scene[1] whose dimensions are $145 \times 145 \times 200$ and $200 \times 200 \times 200$, respectively. The dimensions of $145 \times 145 \times 200$ imply that the hyperspectral dataset has 200 spectral bands or slices, and each slice has the dimensions of 145×145. Prior to handling hyperspectral image inpainting, it requires reshaping each slice into a vector and then combining all vectors into a matrix. Furthermore, the incomplete matrix and observed matrix with GMM noise can be generated.

Two slices of each dataset are selected to evaluate the recovery performance, including the 90th and 175th slices. Fig. 8 shows the original images of Salinas and Indian pines scenes as well as the corresponding observed images with 30% randomly missing data and 6dB GMM noise. The visual presentation of Salinas is shown in Fig. 9. Moreover, the results of the Indian pines are depicted in Fig. 10. It is seen that the PSNRs and SSIMs of the proposed methods are much higher than those of MF-ADMM and LRF-ADMM. Therefore, our algorithms attain better recovery performance than the existing approaches.

VI. Conclusion

In this paper, we exploit the entry-wise ℓ_0-norm and matrix factorization for RMC. The principle is that outliers are separated from the observed matrix by a ℓ_0-norm regularization term. Besides, the penalty parameter of the ℓ_0-norm is automatically updated during the iterative procedure. The resultant multi-variable optimization problem is solved by BCD, yielding two algorithms, namely, ℓ_0-BCD and ℓ_0-BCD-F. In the ℓ_0-BCD, anomalies are discerned and separated from the observed matrix, while the ℓ_0-BCD-F treats the outlier-contaminated data as unobserved entries. Our RMC approach is extended to solve RNMC where the nonnegativity constraint is handled by MM. Then, ℓ_0-BCD-MM and ℓ_0-BCD-MM-F are proposed for RNMC. Simulation results on synthetic and real-world data demonstrate the superiority of our algorithms over the state-of-the-art methods in terms of

[1]URL: http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes

recovery accuracy and computational efficiency. Although the convergence of the objective value $\{\mathcal{L}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k)\}$ has been proved, we will establish sequence convergence of BCD for the nonconvex and noncontinuous objective function in the future work.

Appendix A
Convergence Analysis of Algorithm 1

Before the analysis of the objective value convergence of Algorithm 1, we define a loss function $\mathcal{L}_{\mu^k}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k) = \|\boldsymbol{X}_{\boldsymbol{\Omega}} - (\boldsymbol{U}^k\boldsymbol{V}^k)_{\boldsymbol{\Omega}} - \boldsymbol{S}^k_{\boldsymbol{\Omega}}\|_F^2 + \mu^k\|\boldsymbol{S}^k_{\boldsymbol{\Omega}}\|_0$. It is clear that $\mathcal{L}_{\mu^k}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k)$ is lower bounded by zero. To prove the objective value convergence, we need to show that the update of each variable by Algorithm 1 keeps its value nonincreasing.

$$\begin{aligned}
&\mathcal{L}_{\mu^{k+1}}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^{k+1}) - \mathcal{L}_{\mu^k}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k)\\
=&\mathcal{L}_{\mu^k}(\boldsymbol{U}^{k+1},\boldsymbol{V}^k,\boldsymbol{S}^k) - \mathcal{L}_{\mu^k}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k)\\
&+\mathcal{L}_{\mu^k}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^k) - \mathcal{L}_{\mu^k}(\boldsymbol{U}^{k+1},\boldsymbol{V}^k,\boldsymbol{S}^k)\\
&+\mathcal{L}_{\mu^{k+1}}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^k) - \mathcal{L}_{\mu^k}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^k)\\
&+\mathcal{L}_{\mu^{k+1}}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^{k+1}) - \mathcal{L}_{\mu^{k+1}}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^k). \quad (45)
\end{aligned}$$

Since $\boldsymbol{U}^{k+1}$ is determined by (15), it minimizes $\mathcal{L}_{\mu^k}(\boldsymbol{U},\boldsymbol{V}^k,\boldsymbol{S}^k)$, resulting in $\mathcal{L}_{\mu^k}(\boldsymbol{U}^{k+1},\boldsymbol{V}^k,\boldsymbol{S}^k) - \mathcal{L}_{\mu^k}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k) \le 0$. In addition, $\boldsymbol{V}^{k+1}$ and $\boldsymbol{S}^{k+1}$ are the optimal solutions to $\min_{\boldsymbol{V}} \mathcal{L}_{\mu^k}(\boldsymbol{U}^{k+1},\boldsymbol{V},\boldsymbol{S}^k)$ and $\min_{\boldsymbol{S}} \mathcal{L}_{\mu^{k+1}}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S})$, respectively. Therefore, $\mathcal{L}_{\mu^k}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^k) - \mathcal{L}_{\mu^k}(\boldsymbol{U}^{k+1},\boldsymbol{V}^k,\boldsymbol{S}^k) \le 0$ and $\mathcal{L}_{\mu^{k+1}}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^{k+1}) - \mathcal{L}_{\mu^{k+1}}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^k) \le 0$ hold. For update of μ^k, it is nonincreasing as described in Algorithm 1 and hence it makes $\mathcal{L}_{\mu^{k+1}}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^k) - \mathcal{L}_{\mu^k}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^k) \le 0$ Thereby, $\mathcal{L}_{\mu^{k+1}}(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{S}^{k+1}) - \mathcal{L}_{\mu^k}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k) \le 0$ holds, implying that $\{\mathcal{L}_{\mu^k}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k)\}$ is nonincreasing. Furthermore, the objective function is upper bounded by $\mathcal{L}_{\mu^0}(\boldsymbol{U}^0,\boldsymbol{V}^0,\boldsymbol{S}^0)$, and lower bounded by 0. As a result, $\{\mathcal{L}_{\mu^k}(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{S}^k)\}$ generated by ℓ_0-BCD is convergent. ■

Appendix B
Proof of Closed-Form Solution to Problem (36)

Since (36) is a quadratic programming problem with convex constraint, the KKT conditions are sufficient and necessary for its optimal solution. Thereby, we derive the solution using the KKT conditions. For the sake of presentation simplicity, we omit the superscripts of $\boldsymbol{u}$ and $\boldsymbol{q}$. Then, the Lagrangian of (36) is

$$\gamma(\boldsymbol{u},\boldsymbol{\nu}) = \boldsymbol{u}^T\boldsymbol{u} + \boldsymbol{q}^T\boldsymbol{u} - \boldsymbol{\nu}^T\boldsymbol{u}, \quad (46)$$

where $\boldsymbol{\nu}$ contains the Lagrange multipliers. The derivative of $\gamma(\boldsymbol{u},\boldsymbol{\nu})$ w.r.t. $\boldsymbol{u}$ is $2\boldsymbol{u}+\boldsymbol{q}-\boldsymbol{\nu}$. Thus, the solution is

$$u_i = \frac{1}{2}(\nu_i - q_i). \quad (47)$$

Based on the KKT conditions, we discuss three cases:

(i) $q_i > 0$: It must hold that $\nu_i \ge q_i > 0$ due to the primal feasibility: $u_i \ge 0$. Hence, $\nu_i > 0$ and then necessarily $u_i = 0$ according to the complementary slackness condition.

(ii) $q_i < 0$: From the dual feasibility condition, we get $\nu_i \ge 0$, which results in $u_i > 0$. Moreover, the complementary slackness condition requires $u_i\nu_i = 0$. Thus, we have $\nu_i = 0$ and thereby $u_i = -q_i/2$.

(iii) $q_i = 0$: According to the complementary slackness condition, we conclude $u_i = \nu_i = 0$.

The compact expression of the solution is thus:

$$\boldsymbol{u}_i = \begin{cases} -\frac{q_i}{2}, & \text{if } q_i < 0 \\ 0, & \text{otherwise.} \end{cases} \quad (48)$$

■

References

[1] Y. Li, R. Wang, Y. Fang, M. Sun, and Z. Luo, "Alternating direction method of multipliers for convolutive non-negative matrix factorization," IEEE Trans. Cybern., Sep. 2022, Early Access.

[2] Z. Liu, A. Hansson, and L. Vandenberghe, "Nuclear norm system identification with missing inputs and outputs," Syst. Control. Lett., vol. 62, no. 8, pp. 605–612, Aug. 2013.

[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] Q. Liu, X. P. Li, and J. Yang, "Optimum co-design for image denoising between type-2 fuzzy identifier and matrix completion denoiser," IEEE Trans. Fuzzy Syst., vol. 30, no. 1, pp. 287–292, Jan. 2022.

[5] Q. Liu, X. Li, and H. Cao, "Two-dimensional localization: Low-rank matrix completion with random sampling in massive MIMO system," IEEE Syst. J., vol. 15, no. 3, pp. 3628–3631, Aug. 2020.

[6] S. Ahmadi and M. Rezghi, "Generalized low-rank approximation of matrices based on multiple transformation pairs," Pattern Recognit., vol. 108, p. 107545, Dec. 2020.

[7] E. J. Candès and Y. Plan, "Matrix completion with noise," Proc. IEEE, vol. 98, no. 6, pp. 925–936, Jun. 2010.

[8] M. Fazel, "Matrix rank minimization with applications," Ph.D. thesis, Dept. Elect. Eng., Stanford Univ., California, USA, 2002.

[9] 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, Mar. 2010.

[10] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and Bregman iterative methods for matrix rank minimization," Math. Program., vol. 128, no. 1, pp. 321–353, Sep. 2011.

[11] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems," Pacific J. Optim., vol. 6, no. 15, pp. 615–640, Jun. 2010.

[12] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 9, pp. 2117–2130, Dec. 2012.

[13] 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, Jan. 2017.

[14] F. Nie, Z. Huo, and H. Huang, "Joint capped norms minimization for robust matrix recovery," in Proc. Int. Joint Conf. Artif. Intell., Melbourne, vic. Australia, Aug. 2017, pp. 2557–2563.

[15] F. Nie, H. Huang, and C. Ding, "Low-rank matrix recovery via efficient schatten p-norm minimization," in Proc. 26th AAAI Conf. Artif. Intell., Toronto, ON, Canada, Jul. 2012, pp. 655–661.

[16] F. Nie, H. Wang, H. Huang, and C. Ding, "Joint Schatten p-norm and ℓ_p-norm robust matrix completion for missing value recovery," Knowl. Inf. Syst., vol. 42, no. 3, pp. 525–544, Mar. 2015.

[17] K. Lee and Y. Bresler, "ADMiRA: Atomic decomposition for minimum rank approximation," IEEE Trans. Inf. Theory, vol. 56, no. 9, pp. 4402–4416, Sep. 2010.

[18] 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.

[19] 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, Dec. 2012.

[20] Z. Zhu, Q. Li, G. Tang, and M. B. Wakin, "The global optimization geometry of low-rank matrix optimization," IEEE Trans. Inf. Theory, vol. 67, no. 2, pp. 1308–1331, Jan. 2021.

[21] J. Dai, H. Hu, Q. Hu, W. Huang, N. Zheng, and L. Liu, "Locally linear approximation approach for incomplete data," IEEE Trans. Cybern., vol. 48, no. 6, pp. 1720–1732, 2018.

[22] F. Nie, Z. Li, Z. Hu, R. Wang, and X. Li, "Robust matrix completion with column outliers," IEEE Trans. Cybern., vol. 52, no. 11, pp. 12 042–12 055, Nov. 2022.

[23] A. M. Zoubir, V. Koivunen, Y. Chakhchoukh, and M. Muma, "Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts," IEEE Signal Process. Mag., vol. 29, no. 4, pp. 61–80, Jul. 2012.

[24] X. Jiang, Z. Zhong, X. Liu, and H. C. So, "Robust matrix completion via alternating projection," IEEE Signal Process. Lett., vol. 24, no. 5, pp. 579–583, May 2017.

[25] W. Zeng and H. C. So, "Outlier-robust matrix completion via ℓ_p -minimization," IEEE Trans. Signal Process., vol. 66, no. 5, pp. 1125–1140, Mar. 2018.

[26] X. P. Li, Q. Liu, and H. C. So, "Rank-one matrix approximation with ℓ_p-norm for image inpainting," IEEE Signal Process. Lett., vol. 27, pp. 680–684, Apr. 2020.

[27] M. Muma, W. Zeng, and A. M. Zoubir, "Robust M-estimation based matrix completion," in Proc. IEEE Int. Conf. on Acoust., Speech and Signal Process., Brighton, UK, May 2019, pp. 5476–5480.

[28] F. Ruppel, M. Muma, and A. M. Zoubir, "Globally optimal robust matrix completion based on M-estimation," in Proc. IEEE Int. Workshop Mach. Learn. Signal Process., Espoo, Finland, Sep. 2020, pp. 1–6.

[29] A. M. Zoubir, V. Koivunen, E. Ollila, and M. Muma, Robust Statistics for Signal Processing. Cambridge Univ. Press., 2018.

[30] T. T. Mai and P. Alquier, "A Bayesian approach for noisy matrix completion: Optimal rate under general sampling distribution," Electron. J. Statist., vol. 9, no. 1, pp. 823–841, Apr. 2015.

[31] Q. Zhao, D. Meng, Z. Xu, W. Zuo, and Y. Yan, "ℓ_1-norm low-rank matrix factorization by variational Bayesian method," IEEE Trans. Neural Netw. Learn. Syst., vol. 26, no. 4, pp. 825–839, Jan. 2015.

[32] Y. He, F. Wang, Y. Li, J. Qin, and B. Chen, "Robust matrix completion via maximum correntropy criterion and half-quadratic optimization," IEEE Trans. Signal Process., vol. 68, pp. 181–195, Nov. 2019.

[33] Z. Li, Z. Hu, F. Nie, R. Wang, and X. Li, "Matrix completion with column outliers and sparse noise," Inf. Sci., vol. 573, pp. 125–140, Sep. 2021.

[34] X. Luo, M. Zhou, S. Li, L. Hu, and M. Shang, "Non-negativity constrained missing data estimation for high-dimensional and sparse matrices from industrial applications," IEEE Trans. Cybern., vol. 50, no. 5, pp. 1844–1855, May 2020.

[35] C. Dorffer, M. Puigt, G. Delmaire, and G. Roussel, "Blind mobile sensor calibration using an informed nonnegative matrix factorization with a relaxed rendezvous model," in Proc. IEEE Int. Conf. on Acoust., Speech and Signal Process., Shanghai, China, Apr. 2016, pp. 2941–2945.

[36] D. Landgrebe, "Hyperspectral image data analysis," IEEE Signal Process. Mag., vol. 19, no. 1, pp. 17–28, Aug. 2002.

[37] T. Xie, S. Li, and J. Lai, "Adaptive rank and structured sparsity corrections for hyperspectral image restoration," IEEE Trans. Cybern., vol. 52, no. 9, pp. 8729–8740, Sep. 2022.

[38] Y. Xu, W. Yin, Z. Wen, and Y. Zhang, "An alternating direction algorithm for matrix completion with nonnegative factors," Frontiers Math. China, vol. 7, no. 2, pp. 365–384, Apr. 2012.

[39] D. L. Sun and R. Mazumder, "Non-negative matrix completion for bandwidth extension: A convex optimization approach," in Proc. IEEE Int. Workshop Mach. Learn. Signal Process., Southampton, UK, Sep. 2013, pp. 1–6.

[40] C. Dorffer, M. Puigt, G. Delmaire, and G. Roussel, "Fast nonnegative matrix factorization and completion using Nesterov iterations," in Proc. Int. Conf. Latent Variable Anal., Grenoble, France, Oct. 2017, pp. 26–35.

[41] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," SIAM J. Imag. Sci., vol. 6, no. 3, pp. 1758–1789, Sep. 2013.

[42] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," Neural Comput., vol. 19, no. 10, pp. 2756–2779, Oct. 2007.

[43] L.-F. C. Y.-X. Wang, C. M. Lee and K.-C. Toh, "Practical matrix completion and corruption recovery using proximal alternating robust subspace minimization," Int. J. Comput. Vis., vol. 111, no. 3, pp. 315–344, Feb. 2015.

[44] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," Nature, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[45] E. F. Gonzalez and Y. Zhang, "Accelerating the Lee-Seung algorithm for nonnegative matrix factorization," Dept. Comput. Appl. Math., Rice Univ., Houston, TX, USA, Tech. Rep., 2005.

[46] A. Beck, First-Order Methods in Optimization. SIAM, 2017.

[47] O. Chapelle, P. Haffner, and V. N. Vapnik, "Support vector machines for histogram-based image classification," IEEE Trans. Neural Netw., vol. 10, no. 5, pp. 1055–1064, Sep. 1999.

[48] B. W. Silverman, Density Estimation for Statistics and Data Analysis. CRC press, 1986.

[49] C. Hu, G. Wang, K. Ho, and J. Liang, "Robust ellipse fitting with Laplacian kernel based maximum correntropy criterion," IEEE Trans. Image Process., vol. 30, pp. 3127–3141, Feb. 2021.

[50] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," Amer. Statist., vol. 58, no. 1, pp. 30–37, Jan. 2004.

[51] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," IEEE Trans. Signal Process., vol. 65, no. 3, pp. 794–816, Aug. 2016.

[52] J. Song, P. Babu, and D. P. Palomar, "Optimization methods for designing sequences with low autocorrelation sidelobes," IEEE Trans. Signal Process., vol. 63, no. 15, pp. 3998–4009, Apr. 2015.

[53] V. Singh, R. Dev, N. K. Dhar, P. Agrawal, and N. K. Verma, "Adaptive type-2 fuzzy approach for filtering salt and pepper noise in grayscale images," IEEE Trans. Fuzzy Syst., vol. 26, no. 5, pp. 3170–3176, Oct. 2018.

[54] M. Signoretto, R. Van de Plas, B. De Moor, and J. A. Suykens, "Tensor versus matrix completion: A comparison with application to spectral data," IEEE Signal Process. Lett., vol. 18, no. 7, pp. 403–406, May 2011.

Xiao Peng Li received the B.Eng. degree in electronic science & technology from Yanshan University, Qinhuangdao, China, in 2015, M.Sc. degree with distinction in electronic information engineering, and Ph.D. degree in electrical engineering from City University of Hong Kong, Hong Kong SAR, China, in 2018 and 2022, respectively. From 2018 to 2019, he was a Research Assistant with the Department of Information Engineering, Shenzhen University, Shenzhen, China. He is currently a Postdoctoral Fellow in the Department of Electrical Engineering, City University of Hong Kong, Hong Kong SAR, China.

His research interests include optimization methods, machine learning, sparse recovery, matrix processing, tensor processing, and their applications on target estimation, image recovery, video restoration, hyperspectral unmixing as well as stock market analysis.

Zhang-Lei Shi received the Ph.D. degree from the Department of Electrical Engineering, City University of Hong Kong, Hong Kong SAR, China, in 2021. He is currently a Lecturer with the College of Science, China University of Petroleum (East China), Qingdao, China. His current research interests include neural networks and machine learning.

Qi Liu is currently a Professor with the School of Future Technology at South China University of Technology. Dr. Liu received the Bachelor and Master degrees from Harbin Engineering University, Harbin, China, in 2013 and 2016, respectively, and the Ph.D degree in Electrical Engineering from City University of Hong Kong, Hong Kong, China, in 2019. During 2018 - 2019, he was a Visiting Scholar at University of California Davis, CA, USA. From 2019 to 2022, he worked as a Research Fellow in the Department of Electrical and Computer Engineering, National University of Singapore, Singapore. His research interests include machine learning, optimization methods, and neuromorphic computing with applications to image/video/speech signal processing.

Dr. Liu has been an Associate Editor of the IEEE Systems Journal (2022-), and Digital Signal Processing (2022-). He was also Guest Editor for the IET Signal Processing, International Journal of Antennas and Propagation, and Wireless Communications and Mobile Computing. He was the recipient of the Best Paper Award of IEEE International Conference on Signal, Information and Data Processing (ICSIDP) in 2019.

Hing Cheung So (S'90–M'95–SM'07–F'15) was born in Hong Kong. He received the B.Eng. degree from the City University of Hong Kong and the Ph.D. degree from The Chinese University of Hong Kong, both in electronic engineering, in 1990 and 1995, respectively. From 1990 to 1991, he was an Electronic Engineer with the Research and Development Division, Everex Systems Engineering Ltd., Hong Kong. During 1995–1996, he was a Postdoctoral Fellow with The Chinese University of Hong Kong. From 1996 to 1999, he was a Research Assistant Professor with the Department of Electronic Engineering, City University of Hong Kong, where he is currently a Professor. His research interests include detection and estimation, fast and adaptive algorithms, multidimensional harmonic retrieval, robust signal processing, source localization, and sparse approximation.

He has been on the editorial boards of IEEE Signal Processing Magazine (2014–2017), IEEE Transactions on Signal Processing (2010–2014), Signal Processing (2010–), and Digital Signal Processing (2011–). He was also Lead Guest Editor for IEEE Journal of Selected Topics in Signal Processing, special issue on "Advances in Time/Frequency Modulated Array Signal Processing" in 2017. In addition, he was an elected member in Signal Processing Theory and Methods Technical Committee (2011–2016) of the IEEE Signal Processing Society where he was chair in the awards subcommittee (2015–2016).