Outlier-Robust Matrix Completion via ℓ_p-Minimization

Wen-Jun Zeng, *Member, IEEE*, and Hing Cheung So, *Fellow, IEEE*

***Abstract*—Matrix completion refers to recovering a low-rank matrix from only a subset of its possibly noisy entries, and has a variety of important applications because many real-world signals can be modeled by a $n_1 \times n_2$ matrix with rank $r \ll \min(n_1, n_2)$. Most existing techniques for matrix completion assume Gaussian noise and, thus, they are not robust to outliers. In this paper, we devise two algorithms for robust matrix completion based on low-rank matrix factorization and ℓ_p-norm minimization of the fitting error with $0 < p < 2$. The first method tackles the low-rank matrix factorization with missing data by iteratively solving $(n_1 + n_2)$ linear ℓ_p-regression problems, whereas the second applies the alternating direction method of multipliers (ADMM) in the ℓ_p-space. At each iteration of the ADMM, it requires performing a least squares (LS) matrix factorization and calculating the proximity operator of the pth power of the ℓ_p-norm. The LS factorization is efficiently solved using linear LS regression while the proximity operator has closed-form solution for $p = 1$ or can be obtained by root finding of a scalar nonlinear equation for other values of p. The two proposed algorithms have comparable recovery capability and computational complexity of $\mathcal{O}(K|\Omega|r^2)$, where $|\Omega|$ is the number of observed entries and K is a fixed constant of several hundreds to thousands and dimension independent. It is demonstrated that they are superior to the singular value thresholding, singular value projection, and alternating projection schemes in terms of computational simplicity, statistical accuracy, and outlier-robustness.**

***Index Terms*—Matrix completion, matrix factorization, outlier, robust recovery, ℓ_p-minimization, alternating direction method of multipliers (ADMM).**

I. Introduction

LOW-RANK matrices [1]–[3] frequently arise in many areas of science and engineering including recommender systems [4], [5], data mining and machine learning [1], [2], social network analysis [6], bioinformatics [7], image inpainting [8], computer vision and graphics [1], [8]. It is because many real-world signals can be approximated by a matrix whose rank r is much smaller than the row and column numbers, denoted by n_1 and n_2, respectively. Matrix completion aims at finding the missing entries of a low-rank matrix from incomplete and possibly noisy observations [1]–[11]. A representative example was the Netflix Prize [4], [5], whose goal was to accurately predict user preferences with the use of a database of over 100 million movie ratings made by 480,189 users in 17,770 films, which corresponds to the task of completing a matrix with around 99% missing entries.

Matrix completion can be formulated as a constrained rank minimization problem [1]. Unfortunately, this problem is NP-hard in general because the rank is discrete and nonconvex. Analogous to the strategy of employing the ℓ_1-norm instead of the ℓ_0-norm for sparse signal recovery [12]–[14], convex relaxation for rank minimization replaces the nonconvex rank by the convex nuclear norm, which is the sum of all singular values of the matrix [1], [11], and its theoretical guarantees have been provided in [9]. Typically, nuclear norm minimization is converted into a semi-definite programming (SDP) [1], [11] and hence can be solved by the interior-point methods [10], [15]. However, directly realizing the SDP leads to a high computational load. On the other hand, algorithms which are more computationally efficient than the SDP-based methods have been suggested, such as singular value thresholding (SVT) [16], fixed point continuation (FPC) [17], and proximal gradient descent [18]. Nevertheless, these faster schemes still require performing full singular value decomposition (SVD) of a $n_1 \times n_2$ matrix at each iteration, implying the high complexity to deal with a large matrix. Using the Schatten p-quasi-norm with $0 < p < 1$, namely, ℓ_p-norm of the singular values instead of the nuclear norm can further improve the recovery performance [19]–[23]. Note that the Schatten p-quasi-norm minimization also involves the time-consuming full SVD calculation, which constitutes its dominant computational cost. As a modification to the standard nuclear norm minimization treating each singular value equally, the weighted nuclear norm minimization (WNNM) method [24] adaptively assigns weights to different singular values to enhance the rank sparsity. However, the WNNM for matrix completion [24] is designed for the noiseless case and hence is not robust to outliers.

In sparse signal recovery, the ℓ_1-norm minimization can be solved by iterative soft thresholding (IST) [14]. The SVT [16] in fact iteratively applies thresholding and shrinkage to the singular values to achieve "rank sparsity". Different from the IST, the iterative hard thresholding (IHT) [25] for sparse recovery constrains that the number of nonzero elements does not exceed a specific value to obtain a sparse result. Borrowing the idea from

Manuscript received September 19, 2017; revised November 30, 2017; accepted December 6, 2017. Date of publication December 18, 2017; date of current version January 26, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Cédric Févotte. The work described in this paper was supported by a grant from CityU (Project No. 7004431). *(Corresponding author: Wen-Jun Zeng.)*

W.-J. Zeng is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, and also with the Institute of Telecommunications, Technische Universität Darmstadt, Darmstadt 64289, Germany (e-mail: wenjzeng@cityu.edu.hk).

H. C. So is with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong (e-mail: hcso@ee.cityu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2017.2784361

IHT, a class of approaches, including the singular value projection (SVP) [26], normalized IHT [27], and alternating projection (AP) [28], directly exploits a rank constraint to ensure a low-rank solution. In particular, the SVP and normalized IHT adopt gradient projection method [29] to solve the rank constrained problem. Compared with the nuclear norm or Schatten p-norm minimization that needs to calculate the full SVD, the IHT-type method only requires performing truncated SVD to obtain the r dominant singular values and singular vectors, assuming that the rank information is available. Hence, the computational cost can be greatly reduced especially when the rank is much smaller compared to the matrix dimensions [30].

The third approach for matrix completion utilizes low-rank matrix factorization, where the target matrix is represented by the product of two much smaller matrices so that the low-rank property is automatically fulfilled [31]–[35]. The gradient descent method can be applied as the solver [31], [32], but it suffers from slow convergence. To overcome this drawback, alternating minimization, where the objective function is minimized with respect to (w.r.t.) one factored matrix while the other factor is fixed, is suitable to tackle the resultant bi-convex problem [32], [34], [35]. It is worth pointing out that one main advantage of the matrix factorization based solutions is that they avoid the SVD calculation.

Conventional techniques for matrix completion often rely on the Gaussian noise assumption and their derivation is based on the ℓ_2-space. In spite of providing theoretical and computational convenience, it is generally understood that the validity of Gaussian distribution is at best approximate in reality. In fact, the occurrence of non-Gaussian impulsive noise has been reported in many fields [36]–[38]. For example, the salt-and-pepper noise is a common impulsive noise type in image processing and imaging. The conventional matrix completion algorithms may fail to work properly when the observations contain outliers. In this study, outliers refer to outlying entries whose values are abnormally large, and they are often sparse corruptions in the partially observed entries [39]. Several existing schemes have utilized the fact that the entry-wise ℓ_p-norm with $p < 2$ is less sensitive to outliers than the Frobenius norm for robust matrix factorization [40], [41]. However, [40] only considers the case where there are no missing data. That is, it solves robust low-rank matrix approximation with full observations, which is different from matrix completion. The ℓ_1-Wiberg algorithm [41] that is applicable for the case of incomplete observations exploits ℓ_1-norm to enhance the robustness to outliers, but it has a very high computational complexity. In [19], the ℓ_p-norm and Schatten p-norm are jointly used for robust matrix completion. The augmented Lagrange method (ALM) is employed to solve the resultant joint ℓ_p-norm and Schatten p-norm minimization, in which the full SVD is required. Thus, the computational cost of the ALM is also high. In [28], matrix completion is formulated as a feasibility problem, where the target matrix lies in the intersection of low-rank constraint set and fidelity constraint set. The AP algorithm is developed to find a common point of the two sets. By modeling the fidelity constraint set as an ℓ_p-ball with the center of the ball being the observed entries, the AP achieves robustness to outliers if $p < 2$ is adopted. However, the AP needs the prior knowledge on the ℓ_p-norm of the noise, which is difficult to obtain in practice. The proximal alternating robust subspace minimization (PARSuMi) algorithm is proposed in [42], which directly exploits rank constraint on the completed matrix and ℓ_0 pseudo-norm constraint to enhance the robustness to sparse outliers. However, the rank and an upper bound of the number of outliers are required in this method. Unlike most approaches based on standard basis, matrix completion with column-sparse outliers in general basis is addressed in [39].

The celebrated robust principal component analysis (RPCA) [43] is originally designed for the case with full observations. It can be extended to the case with missing entries. This representative approach [43], [44] models the observed matrix as the sum of a low-rank matrix and a sparse matrix. That is, the outliers are modeled by a sparse matrix. It minimizes the nuclear norm of the unknown low-rank matrix plus the ℓ_1-norm of the sparse component as regularization term to robustly recover the low-rank matrix. Other state-of-the-art robust matrix completion methods include the hierarchical system performing bootstrapping [45] and variational Bayesian matrix factorization based on L_1-norm (VBMFL_1) [46].

In this work, our main contribution is to devise two computationally attractive algorithms for outlier-robust matrix completion based on low-rank matrix factorization under ℓ_p-minimization. The two proposed schemes are iterative ℓ_p-regression algorithm and alternating direction method of multipliers (ADMM). It should be pointed out that the ADMM in our work is different from the ALM in [19] in the following three aspects. First, they solve different optimization problems. The ALM solves a Schatten p-norm regularized ℓ_p-norm minimization problem where the Schatten p-norm is used to obtain a low-rank solution. However, our formulation does not utilize the Schatten p-norm regularization and is based on low-rank matrix factorization, where the target matrix is decomposed as the product of two matrices of smaller size and thus the low-rank property is automatically satisfied. Second, the algorithmic parameters of the two formulations are different. The one used in [19] is the regularization parameter between the ℓ_p-norm and Schatten p-norm, while that of our formulation is the rank. Third, the ALM needs to compute the proximity operator of Schatten p-norm in each iteration, which involves the full SVD. Nonetheless, the proposed ADMM does not require the time-consuming SVD.

The remainder of this paper is organized as follows. In Section II, the matrix completion problem is formulated and representative solvers, including the matrix factorization approach, are briefly reviewed. The two algorithms for robust matrix completion, i.e., iterative ℓ_p-regression and ADMM are derived in Sections III and IV, respectively. Simulation results are provided in Section V to demonstrate the superiority of our solutions over several existing techniques. Finally, conclusions are drawn in Section VI.

II. Problem Formulation and Preliminaries

Let $\boldsymbol{X}_\Omega \in \mathbb{R}^{n_1 \times n_2}$ be a matrix with missing entries where Ω is a subset of the complete set of entries $[n_1] \times [n_2]$, with $[n]$

being the list $\{1, \cdots, n\}$. Throughout the paper, the subscript $(\cdot)_\Omega$ denotes the projection on the known entries. The (i,j) entry of $\boldsymbol{X}_\Omega$, denoted by $[\boldsymbol{X}_\Omega]_{ij}$, can be written as:

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

In addition, we use the lower-case bold letter $\boldsymbol{x}_\Omega \in \mathbb{R}^{|\Omega|}$ to represent the vector stacking all the observed entries of $\boldsymbol{X}_\Omega$ in a column-by-column manner, where $|\Omega|$ stands for the cardinality of Ω, that is, number of observed entries. As an illustration, suppose the original matrix is:

$$\boldsymbol{X} = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{bmatrix}. \tag{2}$$

If only four elements are observed over $\Omega = \{(1,1),(2,1),(3,2),(1,3)\}$, we then have:

$$\boldsymbol{X}_\Omega = \begin{bmatrix} 1 & 0 & 3 \\ 2 & 0 & 0 \\ 0 & 6 & 0 \end{bmatrix} \tag{3}$$

with $\boldsymbol{x}_\Omega = [1,2,6,3]^T$.

The task of matrix completion is to find a matrix $\boldsymbol{M} \in \mathbb{R}^{n_1 \times n_2}$ given incomplete observations $\boldsymbol{X}_\Omega$ by incorporating the low-rank information. Mathematically, it is formulated as a rank minimization problem:

$$\begin{aligned} &\min_{\boldsymbol{M}} \ \operatorname{rank}(\boldsymbol{M}) \\ &\text{s.t. } \boldsymbol{M}_\Omega = \boldsymbol{X}_\Omega. \end{aligned} \tag{4}$$

That is, among all matrices consistent with the observed entries, we look for the one with minimum rank. However, (4) is NP-hard. A popular and practical solution is to replace the nonconvex rank by convex nuclear norm [1]–[15], resulting in

$$\begin{aligned} &\min_{\boldsymbol{M}} \ \|\boldsymbol{M}\|_* \\ &\text{s.t. } \boldsymbol{M}_\Omega = \boldsymbol{X}_\Omega \end{aligned} \tag{5}$$

where the nuclear norm $\|\boldsymbol{M}\|_*$ equals the sum of singular values of $\boldsymbol{M}$. This convex relaxation is analogous to the relaxation of the intractable problem of ℓ_0-minimization to ℓ_1-minimization in sparse signal recovery [12]. In the presence of noise, (5) is modified as

$$\begin{aligned} &\min_{\boldsymbol{M}} \ \|\boldsymbol{M}\|_* \\ &\text{s.t. } \|\boldsymbol{M}_\Omega - \boldsymbol{X}_\Omega\|_F \le \epsilon_F \end{aligned} \tag{6}$$

where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix and $\epsilon_F > 0$ is a tolerance parameter that controls the fitting error. By converting (5) and (6) into SDP [1], [11], they can be solved by interior-point methods [11], [15]. The complexity of the interior-point method for solving SDP is high. But there exists faster alternatives such as SVT [16], FPC [17] and proximal gradient descent [18]. Note that full SVD is still required for these faster methods.

In order to avoid SVD, matrix factorization has been exploited, corresponding to the following optimization:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \ f_2(\boldsymbol{U},\boldsymbol{V}) := \|(\boldsymbol{U}\boldsymbol{V})_\Omega - \boldsymbol{X}_\Omega\|_F^2 \tag{7}$$

where $\boldsymbol{U} \in \mathbb{R}^{n_1 \times r}$ and $\boldsymbol{V} \in \mathbb{R}^{r \times n_2}$. After determining $\boldsymbol{U}$ and $\boldsymbol{V}$, the target matrix is obtained as $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{V}$. Apparently, the low-rank property of $\boldsymbol{M}$ is automatically fulfilled. To handle (7), it can be relaxed as a bi-convex problem [35], which is then solved via alternating minimization. To be more specific, in the $(k+1)$th $(k = 0, 1, \cdots)$ iteration, $\boldsymbol{U}$ and $\boldsymbol{V}$ are alternately minimized according to

$$\begin{aligned} \boldsymbol{V}^{k+1} &= \arg\min_{\boldsymbol{V}} \ \|(\boldsymbol{U}^k\boldsymbol{V})_\Omega - \boldsymbol{X}_\Omega\|_F^2 \\ \boldsymbol{U}^{k+1} &= \arg\min_{\boldsymbol{U}} \ \|(\boldsymbol{U}\boldsymbol{V}^{k+1})_\Omega - \boldsymbol{X}_\Omega\|_F^2 \end{aligned} \tag{8}$$

where the algorithm is initialized with $\boldsymbol{U}^0$ and $\boldsymbol{U}^k$ represents the estimate of $\boldsymbol{U}$ at the kth iteration.

Although the formulations (6) and (7) work well in the presence of additive Gaussian disturbance [1], its performance can significantly degrade when $\boldsymbol{X}_\Omega$ contains outliers.

III. Iterative ℓ_p-Regression

To achieve outlier resistance, we robustify (7) by replacing the Frobenius norm by the ℓ_p-norm where $0 < p < 2$, that is:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \ f_p(\boldsymbol{U},\boldsymbol{V}) := \|(\boldsymbol{U}\boldsymbol{V})_\Omega - \boldsymbol{X}_\Omega\|_p^p, \quad 0 < p < 2 \tag{9}$$

where $\|\cdot\|_p$ denotes the element-wise ℓ_p-norm of a matrix, which has the form of:

$$\|\boldsymbol{E}_\Omega\|_p = \left(\sum_{(i,j)\in\Omega} |\boldsymbol{E}_{ij}|^p \right)^{1/p}. \tag{10}$$

where $\boldsymbol{E}_\Omega = (\boldsymbol{U}\boldsymbol{V})_\Omega - \boldsymbol{X}_\Omega$ with $\boldsymbol{E} = \boldsymbol{U}\boldsymbol{V} - \boldsymbol{X}$ being the error matrix. Note that (9) can be considered as a generalization of (7) because substituting $p = 2$ into the former reduces to the latter. For the special case with $p = 1$, (9) corresponds to the least absolute deviations (LAD), which was first proposed by Laplace [47] and has been widely used in statistics for robust estimation and regression [48], [49]. Different from the least squares (LS) using ℓ_2-minimization, the LAD aims at minimizing the sum of the absolute errors, i.e., the ℓ_1-norm of the residual. Furthermore, the nonconvex ℓ_p-minimization of (9) is different from the robust low-rank matrix approximation in [40], which addresses

$$\min_{\boldsymbol{U},\boldsymbol{V}} \ \|\boldsymbol{U}\boldsymbol{V} - \boldsymbol{X}\|_p^p \tag{11}$$

where there are no missing entries. While in matrix completion, we only have incomplete observations over Ω. It is clear that (9) and (11) are different. In this work, we devise two algorithms for solving (9) and the first one adopts the alternating minimization strategy:

$$\boldsymbol{V}^{k+1} = \arg\min_{\boldsymbol{V}} \ \|(\boldsymbol{U}^k\boldsymbol{V})_\Omega - \boldsymbol{X}_\Omega\|_p^p \tag{12}$$

$$\boldsymbol{U}^{k+1} = \arg\min_{\boldsymbol{U}} \ \|(\boldsymbol{U}\boldsymbol{V}^{k+1})_\Omega - \boldsymbol{X}_\Omega\|_p^p \tag{13}$$

which generalizes (8). We now focus on solving (12) for a fixed $\boldsymbol{U}$:

$$\min_{\boldsymbol{V}} f_p(\boldsymbol{V}) := \|(\boldsymbol{UV})_\Omega - \boldsymbol{X}_\Omega\|_p^p \tag{14}$$

where the superscript $(\cdot)^k$ is dropped for notational simplicity. Denoting the ith row of $\boldsymbol{U}$ and the jth column of $\boldsymbol{V}$ as $\boldsymbol{u}_i^T$ and $\boldsymbol{v}_j$, respectively, where $\boldsymbol{u}_i, \boldsymbol{v}_j \in \mathbb{R}^r$, $i = 1, \cdots, n_1$, $j = 1, \cdots, n_2$, (14) can be rewritten as

$$\min_{\boldsymbol{V}} f_p(\boldsymbol{V}) := \sum_{(i,j)\in\Omega} |\boldsymbol{u}_i^T \boldsymbol{v}_j - \boldsymbol{X}_{ij}|^p. \tag{15}$$

Since $f_p(\boldsymbol{V})$ is decoupled w.r.t. $\boldsymbol{v}_j$, (15) is equivalent to solving the following n_2 independent subproblems:

$$\min_{\boldsymbol{v}_j} f_p(\boldsymbol{v}_j) := \sum_{i\in\mathcal{I}_j} |\boldsymbol{u}_i^T \boldsymbol{v}_j - \boldsymbol{X}_{ij}|^p,\ j = 1, \cdots, n_2 \tag{16}$$

where $\mathcal{I}_j = \{j_1, \cdots, j_{|\mathcal{I}_j|}\} \subseteq \{1, \cdots, n_1\}$ denotes the set containing the row indices for the jth column in Ω. Here $|\mathcal{I}_j|$ stands for the cardinality of $\mathcal{I}_j$ and in general $|\mathcal{I}_j| > r$. As an illustration, we provide a simple example for determining $\mathcal{I}_j$ as follows. Consider $\boldsymbol{X}_\Omega \in \mathbb{R}^{4\times 3}$:

$$\boldsymbol{X}_\Omega = \begin{bmatrix} 0 & \times & 0 \\ \times & 0 & \times \\ 0 & \times & \times \\ \times & 0 & \times \end{bmatrix} \tag{17}$$

where the observed and missing entries are represented by $\times$ and 0, respectively. For $j = 1$, the $(2,1)$ and $(4,1)$ entries are observed. Thus we have $\mathcal{I}_1 = \{2,4\}$. It is easy to see that $\mathcal{I}_2 = \{1,3\}$ and $\mathcal{I}_3 = \{2,3,4\}$. Apparently, $\sum_{j=1}^{n_2} |\mathcal{I}_j| = |\Omega|$. Defining a matrix $\boldsymbol{U}_{\mathcal{I}_j} \in \mathbb{R}^{|\mathcal{I}_j|\times r}$ containing the $|\mathcal{I}_j|$ rows indexed by $\mathcal{I}_j$:

$$\boldsymbol{U}_{\mathcal{I}_j} = \begin{bmatrix} \boldsymbol{u}_{j_1}^T \\ \vdots \\ \boldsymbol{u}_{j_{|\mathcal{I}_j|}}^T \end{bmatrix} \tag{18}$$

and a vector $\boldsymbol{b}_{\mathcal{I}_j} = [\boldsymbol{X}_{j_1 j}, \cdots, \boldsymbol{X}_{j_{|\mathcal{I}_j|} j}]^T \in \mathbb{R}^{|\mathcal{I}_j|}$, then (16) is compactly rewritten as

$$\min_{\boldsymbol{v}_j} f_p(\boldsymbol{v}_j) := \|\boldsymbol{U}_{\mathcal{I}_j}\boldsymbol{v}_j - \boldsymbol{b}_{\mathcal{I}_j}\|_p^p \tag{19}$$

which is a robust linear regression in ℓ_p-space. It is worth mentioning that for $p = 2$, (19) is an LS problem with solution being $\boldsymbol{v}_j = \boldsymbol{U}_{\mathcal{I}_j}^\dagger \boldsymbol{b}_{\mathcal{I}_j}$, and the corresponding computational complexity is $\mathcal{O}(|\mathcal{I}_j|r^2)$.

For $0 < p < 2$, the ℓ_p-regression of (19) can be efficiently solved by the iteratively reweighted least squares (IRLS) algorithm [50], [51] where global convergence can be achieved for the convex case of $p \geq 1$ while only a stationary point is obtained for the nonconvex case of $p < 1$. At the tth iteration[1], the IRLS solves the following weighted LS problem:

$$\boldsymbol{v}_j^{t+1} = \arg\min_{\boldsymbol{v}_j} \|\boldsymbol{W}^t(\boldsymbol{U}_{\mathcal{I}_j}\boldsymbol{v}_j - \boldsymbol{b}_{\mathcal{I}_j})\|_2^2 \tag{20}$$

where $\boldsymbol{W}^t = \mathrm{diag}\{w_1^t, \cdots, w_{n_1}^t\}$ is a diagonal weighting matrix with the ith diagonal element being

$$w_i^t = \frac{1}{(|\xi_i^t|^2 + \epsilon)^{\frac{1-p/2}{2}}}. \tag{21}$$

The ξ_i^t is the ith element of the residual vector $\boldsymbol{\xi}^t = \boldsymbol{U}_{\mathcal{I}_j}\boldsymbol{v}_j^t - \boldsymbol{b}_{\mathcal{I}_j}$ and $\epsilon > 0$ is a small positive parameter to avoid division by zero and ensure numerical stability, especially for $p \leq 1$. A typical value of ϵ is taken as $\epsilon = 100\epsilon_{\mathrm{machine}}$ with $\epsilon_{\mathrm{machine}}$ being the machine precision. Only one LS problem is required to solve in each IRLS iteration. Therefore, the complexity of ℓ_p-regression is $\mathcal{O}(|\mathcal{I}_j|r^2 N_{\mathrm{IRLS}})$ where N_{IRLS} is the iteration number required for the IRLS algorithm to converge. Due to its fast convergence rate [50], N_{IRLS} will not be large, with a typical value of several tens, and is independent of the problem dimension. The total complexity for handling the n_2 ℓ_p-regressions of (15) is $\mathcal{O}(|\Omega|r^2 N_{\mathrm{IRLS}})$ due to $\sum_{j=1}^{n_2} |\mathcal{I}_j| = |\Omega|$.

Since (12) and (13) have the same structure, we solve (13) in the same manner. The ith row of $\boldsymbol{U}$ is updated by

$$\min_{\boldsymbol{u}_i^T} \|\boldsymbol{u}_i^T \boldsymbol{V}_{\mathcal{J}_i}^{k+1} - \boldsymbol{b}_{\mathcal{J}_i}^T\|_p^p \tag{22}$$

where $\mathcal{J}_i = \{i_1, \cdots, i_{|\mathcal{J}_i|}\} \subseteq \{1, \cdots, n_2\}$ is the set containing the column indices for the ith row in Ω. Employing (17) again, only the $(1,2)$ entry is observed for $i = 1$, and thus $\mathcal{J}_1 = \{2\}$. We also easily obtain $\mathcal{J}_2 = \{1,3\}$, $\mathcal{J}_3 = \{2,3\}$, and $\mathcal{J}_4 = \{1,3\}$. Here, $\boldsymbol{V}_{\mathcal{J}_i}^{k+1} \in \mathbb{R}^{r\times|\mathcal{J}_i|}$ contains the $|\mathcal{J}_i|$ columns indexed by $\mathcal{J}_i$ and $\boldsymbol{b}_{\mathcal{J}_i}^T = [\boldsymbol{X}_{ii_1}, \cdots, \boldsymbol{X}_{ii_{|\mathcal{J}_i|}}]^T \in \mathbb{R}^{|\mathcal{J}_i|}$. The involved complexity in (22) is $\mathcal{O}(|\mathcal{J}_i|r^2 N_{\mathrm{IRLS}})$ and hence the total complexity for solving the n_1 ℓ_p-regressions of (22) is $\mathcal{O}(|\Omega|r^2 N_{\mathrm{IRLS}})$ because of $\sum_{i=1}^{n_1} |\mathcal{J}_i| = |\Omega|$.

The steps of the iterative ℓ_p-regression for matrix completion is summarized in Algorithm 1. Note that the complexity for a k-iteration is $\mathcal{O}(|\Omega|r^2 N_{\mathrm{IRLS}})$. For the special case when $p = 2$, Algorithm 1 reduces to solving the problem of (7). In this case, we have $N_{\mathrm{IRLS}} = 1$ and the complexity reduces to $\mathcal{O}(|\Omega|r^2)$ per k-iteration. In many practical applications, the number of observed entries is much smaller than the number of total entries, that is, $|\Omega| \ll n_1 n_2$. Thus, the proposed algorithm becomes more computationally efficient as the percentage of the observations decreases. Now it is clear that the total complexity of the iterative ℓ_p-regression is $\mathcal{O}(|\Omega|r^2 N_{\mathrm{IRLS}} K_{\mathrm{reg}})$ where K_{reg} is the number of outer iterations, namely, the k-iteration. Empirically, a value of several tens for K_{reg} is sufficient for convergence. Finally, it is worth pointing out that the n_2 problems of (19) and n_1 problems of (22) are independent and hence can be realized in a parallel or distributed manner. As the number of processors increases, the complexity reduces.

We give a short remark on the convergence of the iterative ℓ_p-regression. Since Algorithm 1 monotonically non-increases a below-bounded objective function for all $p \leq 2$, the sequence $\{f_p(\boldsymbol{U}^k, \boldsymbol{V}^k)\}$ converges to a limit point. However, it does not imply that $\{(\boldsymbol{U}^k, \boldsymbol{V}^k)\}$ converges. If we further assume that

[1]It should be pointed out that the iteration number t refers to IRLS iteration and should not be mixed up with the iteration number k. That is, k is the index of outer iteration while t is the index of inner iteration.

Algorithm 1: Iterative ℓ_p-Regression for Robust Matrix Completion.

Input: $\boldsymbol{X}_\Omega$, Ω, and rank r
  Initialize: Randomly initialize $\boldsymbol{U}^0 \in \mathbb{R}^{n_1 \times r}$
  Determine $\{\mathcal{I}_j\}_{j=1}^{n_2}$ and $\{\mathcal{J}_i\}_{i=1}^{n_1}$ according to Ω.
  for $k = 0, 1, \cdots$ **do**
    // Fix $\boldsymbol{U}^k$, optimize $\boldsymbol{V}$
    for $j = 1, 2, \cdots, n_2$ **do**
      $\boldsymbol{v}_j^{k+1} \leftarrow \arg\min_{\boldsymbol{v}_j} \|\boldsymbol{U}_{\mathcal{I}_j}^k \boldsymbol{v}_j - \boldsymbol{b}_{\mathcal{I}_j}\|_p^p$
    end for
    // Fix $\boldsymbol{V}^{k+1}$, optimize $\boldsymbol{U}$
    for $i = 1, 2, \cdots, n_1$ **do**
      $(\boldsymbol{u}_i^T)^{k+1} \leftarrow \arg\min_{\boldsymbol{u}_i^T} \|\boldsymbol{u}_i^T \boldsymbol{V}_{\mathcal{J}_i}^{k+1} - \boldsymbol{b}_{\mathcal{J}_i}^T\|_p^p$
    end for
    Stop if a termination condition is satisfied.
  end for
Output: $\boldsymbol{M} = \boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1}$

either (12) or (13) has a unique minimizer, then Algorithm 1 converges to a stationary point based on the convergence result of a block coordinate descent method using the cyclic rule in [52], which contains Algorithm 1 as a special case.

IV. ADMM

In this section, we apply the ADMM to solve (9).

A. Framework of ADMM

With the use of $\boldsymbol{E}_\Omega = (\boldsymbol{UV})_\Omega - \boldsymbol{X}_\Omega$, (9) is equivalent to a linearly constrained optimization problem:

$$\min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}_\Omega} \|\boldsymbol{E}_\Omega\|_p^p \quad \text{s.t. } \boldsymbol{E}_\Omega = (\boldsymbol{UV})_\Omega - \boldsymbol{X}_\Omega. \tag{23}$$

where $\boldsymbol{E}_\Omega$ is treated as decision variables that are independent of $\boldsymbol{U}$ and $\boldsymbol{V}$. Note that $[\boldsymbol{E}_\Omega]_{ij} = 0$ if $(i,j) \notin \Omega$. The augmented Lagrangian of (23) is

$$\mathcal{L}_\mu(\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}_\Omega,\boldsymbol{\Lambda}_\Omega) = \|\boldsymbol{E}_\Omega\|_p^p + \langle \boldsymbol{\Lambda}_\Omega, (\boldsymbol{UV})_\Omega - \boldsymbol{E}_\Omega - \boldsymbol{X}_\Omega\rangle + \frac{\mu}{2}\|(\boldsymbol{UV})_\Omega - \boldsymbol{E}_\Omega - \boldsymbol{X}_\Omega\|_F^2 \tag{24}$$

where $\boldsymbol{\Lambda}_\Omega \in \mathbb{R}^{n_1 \times n_2}$ with $[\boldsymbol{\Lambda}_\Omega]_{ij} = 0$ for $(i,j) \notin \Omega$ contains $|\Omega|$ Lagrange multipliers (dual variables), $\langle \boldsymbol{A}, \boldsymbol{B}\rangle = \sum_{(i,j)} \boldsymbol{A}_{ij}\boldsymbol{B}_{ij}$ represents the inner product of two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, and $\mu > 0$ is the penalty parameter. The augmented Lagrangian reduces to the unaugmented one if $\mu = 0$. If the objective function is closed, proper and convex, and the unaugmented Lagrangian $\mathcal{L}_0$ has a saddle point, then the iterates approach feasibility and the objective function of the iterates approaches the optimal value [53]. However, the objective function of our problem is nonconvex. The theoretical proof of the convergence of the nonconvex ADMM is very challenging and remains an open problem. We give a brief discussion on this issue at the end of Section IV-C. Empirically, numerical examples [53] demonstrate that the selection of μ is flexible. We can use a fixed appropriate positive constant for μ or properly adapt the penalty parameter at each iteration for convergence speedup [54], [55]. In the simulations, we simply use $\mu = 5$ and it is observed that this value always makes the ADMM converge. The Lagrange multiplier method solves (23) by finding a saddle point of the augmented Lagrangian

$$\max_{\boldsymbol{\Lambda}_\Omega} \min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}_\Omega} \mathcal{L}_\mu(\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}_\Omega,\boldsymbol{\Lambda}_\Omega). \tag{25}$$

The ADMM uses the following iterative steps:

$$(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1}) = \arg\min_{\boldsymbol{U},\boldsymbol{V}} \mathcal{L}_\mu(\boldsymbol{U},\boldsymbol{V},\boldsymbol{E}_\Omega^k,\boldsymbol{\Lambda}_\Omega^k) \tag{26}$$

$$\boldsymbol{E}_\Omega^{k+1} = \arg\min_{\boldsymbol{E}_\Omega} \mathcal{L}_\mu(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{E}_\Omega,\boldsymbol{\Lambda}_\Omega^k) \tag{27}$$

$$\boldsymbol{\Lambda}_\Omega^{k+1} = \boldsymbol{\Lambda}_\Omega^k + \mu\left((\boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1})_\Omega - \boldsymbol{E}_\Omega^{k+1} - \boldsymbol{X}_\Omega\right) \tag{28}$$

to calculate the saddle point in (25), where $(\boldsymbol{U}^k,\boldsymbol{V}^k,\boldsymbol{E}_\Omega^k,\boldsymbol{\Lambda}_\Omega^k)$ denotes the result at the kth iteration. Several remarks and explanations on the three subproblems (26), (27), and (28) are given as follows.

Since the gradient of $\mathcal{L}_\mu(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{E}_\Omega^{k+1},\boldsymbol{\Lambda}_\Omega)$ w.r.t. $\boldsymbol{\Lambda}_\Omega$ is

$$\frac{\partial \mathcal{L}_\mu(\boldsymbol{U}^{k+1},\boldsymbol{V}^{k+1},\boldsymbol{E}_\Omega^{k+1},\boldsymbol{\Lambda}_\Omega)}{\partial \boldsymbol{\Lambda}_\Omega} = (\boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1})_\Omega - \boldsymbol{E}_\Omega^{k+1} - \boldsymbol{X}_\Omega \tag{29}$$

we can see that (28) adopts a gradient ascent with a step size μ to update the dual variable $\boldsymbol{\Lambda}_\Omega$. ADMM updates $(\boldsymbol{U},\boldsymbol{V})$ and $\boldsymbol{E}_\Omega$ in an alternating or sequential fashion to circumvent the difficulty in jointly minimizing w.r.t. the two primal blocks. Noting that (26) minimizes $(\boldsymbol{U},\boldsymbol{V})$ simultaneously, (26)–(28) correspond to a two-block ADMM where the blocks refer to $(\boldsymbol{U},\boldsymbol{V})$ and $\boldsymbol{E}_\Omega$, and are not of three blocks. It has been observed that updating more than two blocks may result in divergence of the ADMM [56]. Nevertheless, the divergence caused by multi-block update will not happen to the proposed ADMM since it is a two-block one.

By ignoring the constant term independent of $(\boldsymbol{U},\boldsymbol{V})$, we derive that the subproblem (26) is equivalent to the following Frobenius norm minimization problem:

$$\min_{\boldsymbol{U},\boldsymbol{V}} \left\|(\boldsymbol{UV})_\Omega - \left(\boldsymbol{E}_\Omega^k - \frac{\boldsymbol{\Lambda}_\Omega^k}{\mu} + \boldsymbol{X}_\Omega\right)\right\|_F^2 \tag{30}$$

which can be solved by the iterative ℓ_2-regression, namely, Algorithm 1 with $p = 2$, with a complexity bound of $\mathcal{O}(K_{\ell_2}|\Omega|r^2)$. Here, K_{ℓ_2} is the iteration number for Algorithm 1 to converge at $p = 2$.

On the other hand, the subproblem (27) is concisely simplified as

$$\min_{\boldsymbol{E}_\Omega} \frac{1}{2}\left\|\boldsymbol{E}_\Omega - \boldsymbol{Y}_\Omega^k\right\|_F^2 + \frac{1}{\mu}\|\boldsymbol{E}_\Omega\|_p^p \tag{31}$$

where

$$\boldsymbol{Y}_\Omega^k = (\boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1})_\Omega + \frac{\boldsymbol{\Lambda}_\Omega^k}{\mu} - \boldsymbol{X}_\Omega. \tag{32}$$

We only need to consider the entries indexed by Ω because other entries of $\boldsymbol{E}_\Omega$ and $\boldsymbol{Y}_\Omega^k$ which are not in Ω are zero. Define $\boldsymbol{e}_\Omega$,

$\boldsymbol{y}_\Omega^k$, $\boldsymbol{\lambda}_\Omega^k$, and $\boldsymbol{t}_\Omega^k \in \mathbb{R}^{|\Omega|}$ as the vectors that contain the observed entries in $\boldsymbol{E}_\Omega, \boldsymbol{Y}_\Omega^k, \boldsymbol{\Lambda}_\Omega^k$, and $(\boldsymbol{U}^k\boldsymbol{V}^k)_\Omega$, respectively, in a column-by-column manner. Apparently, (31) is equivalent to the vector optimization problem:

$$\min_{\boldsymbol{e}_\Omega} \frac{1}{2}\left\|\boldsymbol{e}_\Omega - \boldsymbol{y}_\Omega^k\right\|_2^2 + \frac{1}{\mu}\|\boldsymbol{e}_\Omega\|_p^p \tag{33}$$

whose solution defines the proximity operator [57] of the pth power of ℓ_p-norm, which is written as

$$\boldsymbol{e}_\Omega^{k+1} = \mathrm{prox}_{1/\mu}(\boldsymbol{y}_\Omega^k). \tag{34}$$

After obtaining $\boldsymbol{e}_\Omega^{k+1}$, $\boldsymbol{E}_\Omega^{k+1}$ is then determined. We will address computing this proximity operator shortly. For (28), its equivalent form in terms of vectors is:

$$\boldsymbol{\lambda}_\Omega^{k+1} = \boldsymbol{\lambda}_\Omega^k + \mu\left(\boldsymbol{t}_\Omega^{k+1} - \boldsymbol{e}_\Omega^{k+1} - \boldsymbol{x}_\Omega\right) \tag{35}$$

That is, the operations are now in terms of vectors but not matrices, and its complexity is $\mathcal{O}(|\Omega|)$. Also, at each iteration, we just need to compute $(\boldsymbol{UV})_\Omega$ instead of $\boldsymbol{UV}$, whose complexity is $\mathcal{O}(|\Omega|r)$ because only $|\Omega|$ inner products $\{\boldsymbol{u}_i^T\boldsymbol{v}_j\}_{(i,j)\in\Omega}$ are calculated.

B. Proximity Operator of pth Power of ℓ_p-Norm

In this section, the proximity operator is determined. First, we rewrite (33) as:

$$\min_{\boldsymbol{e}\in\mathbb{R}^{|\Omega|}} \frac{1}{2}\|\boldsymbol{e}-\boldsymbol{y}\|_2^2 + \frac{1}{\mu}\|\boldsymbol{e}\|_p^p \tag{36}$$

where the subscripts and superscripts are ignored for notational simplicity. Denote e_i and $y_i, i = 1, \cdots, |\Omega|$, as the ith entry of $\boldsymbol{e}$ and $\boldsymbol{y}$, respectively. As (36) is separable, it can be decomposed into $|\Omega|$ independent scalar problems:

$$\min_{e_i\in\mathbb{R}} g(e_i) := \frac{1}{2}(e_i - y_i)^2 + \frac{1}{\mu}|e_i|^p,\ i = 1, \cdots, |\Omega|. \tag{37}$$

The closed-form solution of (37) for $p = 1$ is

$$e_i^\star = \mathrm{sgn}(y_i)\max(|y_i| - 1/\mu, 0) \tag{38}$$

which is known as the soft-thresholding operator [14], [58] and is easily computed with a marginal complexity of $\mathcal{O}(|\Omega|)$.

When the noise is very impulsive, the value of $p < 1$ may be required. The scalar minimization problem of (37) with $p < 1$ has already been solved recently in [19], [23], [59], whose solution is:

$$e_i^\star = \begin{cases} 0, & \text{if } |y_i| \le \tau \\ \arg\min\limits_{e_i\in\{0,t_i\}} g(e_i), & \text{if } |y_i| > \tau \end{cases} \tag{39}$$

where

$$\tau = \left(\frac{p(1-p)}{\mu}\right)^{\frac{1}{2-p}} + \frac{p}{\mu}\left(\frac{p(1-p)}{\mu}\right)^{\frac{p-1}{2-p}} \tag{40}$$

is the threshold and $t_i = \mathrm{sgn}(y_i)r_i$ with r_i being the unique root of the nonlinear equation:

$$h(\theta) := \theta + \frac{p}{\mu}\theta^{p-1} - |y_i| = 0 \tag{41}$$

in the interval $[(p(1-p)/\mu)^{\frac{1}{2-p}}, |y_i|]$ where the bisection method [60] can be applied. Although computing the proximity operator for $p < 1$ still has a complexity of $\mathcal{O}(|\Omega|)$, it is more complicated than $p = 1$ because there is no closed-form solution. On the other hand, the case of $p \in (1, 2)$ is not difficult to solve since (37) is a scalar convex problem but it also requires an iterative procedure for numerical calculation. For the purpose of completeness, we present the solver of (37) for $p \in (1, 2)$ since this has not been addressed. Obviously, if $y_i \ge 0$, the minimizer $e_i^\star \ge 0$. Otherwise, $e_i^\star < 0$. That is to say, we only need to consider minimizing $g(e_i)$ in $[0, \infty)$ if $y_i \ge 0$. The minimizer is either the stationary point satisfying the nonlinear equation

$$g'(e_i) = e_i - y_i + \frac{p}{\mu}e_i^{p-1} = 0 \tag{42}$$

or the boundary point 0. Due to $g'(0) = -y_i \le 0$ and $g'(y_i) = py_i^{p-1}/\mu \ge 0$, i.e., $g'(0)g'(y_i) \le 0$, there exists a root in $[0, y_i]$ for the equation $g'(e_i) = 0$. Moreover, $g''(e_i) = 1 + \frac{p(p-1)}{\mu}e_i^{p-2} > 0$ holds for all $e_i \ge 0$, implying that $g'(e_i)$ monotonically increases in $[0, +\infty)$. Thus, the positive root of $g'(e_i) = 0$ in $[0, y_i]$ is unique, which is denoted as r_i^+. This root can be quickly found using the bisection or secant method with a complexity of $\mathcal{O}(1)$ [60]. After obtaining r_i^+, the minimizer in $[0, -\infty)$ is $e_i^\star = \arg\min\{g(0), g(r_i^+)\}$.

Similarly, we only need to minimize $g(e_i)$ in $(-\infty, 0]$ if $y_i < 0$. The minimizer is either the stationary point fulfilling

$$g'(e_i) = e_i - y_i - \frac{p}{\mu}(-e_i)^{p-1} = 0 \tag{43}$$

or the boundary point 0. Since $g'(y_i) = -p(-y_i)^{p-1}/\mu \le 0$ and $g'(0) = -y_i \ge 0$, namely., $g'(0)g'(y_i) \le 0$, $g'(e_i) = 0$ has a root in $[y_i, 0]$. Noting that $g''(e_i) = 1 + \frac{p(p-1)}{\mu}(-e_i)^{p-2} > 0$ holds for all $e_i \le 0$, $g'(e_i)$ monotonically increases in $(-\infty, 0]$. Then, the negative root of $g'(e_i) = 0$ in $[y_i, 0]$, which is denoted as r_i^-, is unique and can be solved easily. Once r_i^- is obtained, the minimizer in $(-\infty, 0]$ is $e_i^\star = \arg\min\{g(0), g(r_i^-)\}$. The solution of (37) for $p \in (1, 2)$ is compactly written as

$$e_i^\star = \begin{cases} \arg\min\left\{g(0), g(r_i^+)\right\}, & \text{if } y_i \ge 0 \\ \arg\min\left\{g(0), g(r_i^-)\right\}, & \text{if } y_i < 0. \end{cases} \tag{44}$$

Again, calculating the proximity operator for $1 < p < 2$ has a complexity of $\mathcal{O}(|\Omega|)$ although an iterative procedure for root finding is required. Nevertheless, the choice of $p = 1$ is more robust than employing $p \in (1, 2)$ and is computationally simpler. In the case of very impulsive noise, $p < 1$ will be adopted.

C. Summary of ADMM

The steps of ADMM for robust matrix completion are summarized in Algorithm 2. The ℓ_2-norm of the residual, that is, $\|\boldsymbol{t}_\Omega^k - \boldsymbol{e}_\Omega^k - \boldsymbol{x}_\Omega\|_2$ is used to check for convergence. Specifically, the iteration is terminated when

$$\|\boldsymbol{t}_\Omega^k - \boldsymbol{e}_\Omega^k - \boldsymbol{x}_\Omega\|_2 < \delta \tag{45}$$

where $\delta > 0$ is a small tolerance parameter.

The dominant complexity of the ADMM is $\mathcal{O}(|\Omega|r^2 K_{\ell_2} K_{\mathrm{ADMM}})$ where K_{ADMM} is the number of outer iterations

Algorithm 2: ADMM for Robust Matrix Completion.

Input: $\boldsymbol{X}_\Omega$, Ω, and rank r

Initialize: $\boldsymbol{e}^0 = \boldsymbol{0}$ and $\boldsymbol{\lambda}^0 = \boldsymbol{0}$

for $k = 0, 1, \cdots$ **do**

1) Solve LS matrix factorization

$$(\boldsymbol{U}^{k+1}, \boldsymbol{V}^{k+1}) = \arg\min_{\boldsymbol{U},\boldsymbol{V}} \left\|(\boldsymbol{UV})_\Omega - \left(\boldsymbol{E}_\Omega^k - \boldsymbol{\Lambda}_\Omega^k/\mu + \boldsymbol{X}_\Omega\right)\right\|_F^2$$

using Algorithm 1 with $p = 2$.

2) Compute $\boldsymbol{Y}_\Omega^k = (\boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1})_\Omega + \boldsymbol{\Lambda}_\Omega^k/\mu - \boldsymbol{X}_\Omega$ and form $\boldsymbol{y}_\Omega^k$ and $\boldsymbol{t}_\Omega^{k+1} \leftarrow (\boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1})_\Omega$.

3) $\boldsymbol{e}_\Omega^{k+1} \leftarrow \text{prox}_{1/\mu}(\boldsymbol{y}_\Omega^k)$

4) $\boldsymbol{\lambda}_\Omega^{k+1} \leftarrow \boldsymbol{\lambda}_\Omega^k + \mu\left(\boldsymbol{t}_\Omega^{k+1} - \boldsymbol{e}_\Omega^{k+1} - \boldsymbol{x}_\Omega\right)$

Stop if a termination condition is satisfied.

end for

Output: $\boldsymbol{M} = \boldsymbol{U}^{k+1}\boldsymbol{V}^{k+1}$

of the ADMM, namely, the k-iteration. Empirically, a value of several tens for K_{ADMM} will result in an accurate estimation.

It should be pointed out that our ADMM is different from the ALM of [19] for matrix completion that solves

$$\min_{\boldsymbol{M}} \|\boldsymbol{M}_\Omega - \boldsymbol{X}_\Omega\|_p^p + \gamma\|\boldsymbol{M}\|_{S_p}^p \tag{46}$$

where $\gamma > 0$ is the regularization parameter and $\|\boldsymbol{M}\|_{S_p}$ is the Schatten p-norm, which equals the ℓ_p-norm of the vector containing all singular values of $\boldsymbol{M}$. As $p \to 0$, $\|\boldsymbol{M}\|_{S_p}$ approaches the rank of $\boldsymbol{M}$. Therefore, the Schatten p-norm regularization with $p \le 1$ can be employed to find a low-rank solution. Especially, when $p = 1$, (46) is a convex program because $\|\boldsymbol{M}\|_{S_p}$ is the nuclear norm. In [19], the ALM is applied to solve (46), in which the full SVD of a $n_1 \times n_2$ matrix is computed. Thus, the complexity of the ALM [19] is $\mathcal{O}(n_1^2 n_2)$ per iteration, assuming that $n_1 \ge n_2$ without loss of generality.

The proposed method is also different from the RPCA that models the observed matrix as the sum of a low-rank matrix $\boldsymbol{L}$ and a sparse outlier matrix $\boldsymbol{S}$. When partial observations are available, the RPCA can be applied for matrix completion by solving the minimization problem

$$\min_{\boldsymbol{M},\boldsymbol{S}} \|\boldsymbol{L}\|_* + \alpha\|\boldsymbol{S}\|_1 \quad \text{s.t. } [\boldsymbol{L} + \boldsymbol{S}]_\Omega = \boldsymbol{X}_\Omega \tag{47}$$

where $\alpha > 0$ is the regularization parameter that needs to estimate in practice. Although (47) is a convex optimization and the global minimum is guaranteed, it has a high computational cost even fast algorithms are employed because the full SVD is required [43], [44].

In a unified manner, the total complexity of the iterative ℓ_p-regression and ADMM can be written as $\mathcal{O}(K|\Omega|r^2)$ where

$$K = \begin{cases} N_{\text{IRLS}} K_{\text{reg}}, & \text{for } \ell_p\text{−regression} \\ K_{\ell_2} K_{\text{ADMM}}, & \text{for ADMM.} \end{cases} \tag{48}$$

The magnitude-of-order of K corresponds to several hundreds to thousands because N_{IRLS}, K_{reg}, K_{ℓ_2}, and K_{ADMM}, are of several tens.

The convergence of the two-block ADMM has only been proved for convex optimization [53]. Although the convergence of the ADMM for a class of nonconvex and nonsmooth optimization problems, including the ℓ_p-regularization with $p < 1$, has been established very recently in [61], the corresponding results are not applicable to our problem. The first reason is that the ℓ_p-norm appears as a regularization term to promote sparsity in [61] while our problem minimizes the ℓ_p-norm of the fitting error. This results in that the mathematical formulations of [61] and our problem are different. The second reason is that the nonconvexity of our problem is not only due to the ℓ_p-norm with $p < 1$ but also induced by the matrix product $\boldsymbol{UV}$. These two reasons make the theoretical proof of the convergence of the proposed ADMM challenging. It remains an open problem for future research. Although the convergence is not proved theoretically, we observe that the proposed ADMM always converges in the simulations. Thus, it is deemed that the proposed ADMM is empirically convergent in practice.

D. Algorithmic Parameter Selection

There are two parameters of the proposed algorithms, namely, the rank r and p. We discuss how to appropriately select them.

If the true rank is unknown, it needs to be estimated. Determining the rank is a model selection problem [62]. However, conventional model selection methods such as Akaike information criterion and minimum description length [62] are not applicable because there are missing data and outliers in our problem. Denoting the estimate for a given r as $\widehat{\boldsymbol{M}}(r)$, the optimal r aims at minimizing the estimation error

$$\min_{r \in \mathbb{Z}^+} \|\widehat{\boldsymbol{M}}(r) - \boldsymbol{X}\|_F^2 \tag{49}$$

where $\mathbb{Z}^+$ is the set of positive integers. However, we cannot obtain the optimal r from (49) because $\boldsymbol{X}$ is not available.

In this study, we estimate the rank by cross-validation [7], [63]. Specifically, the observation set Ω is divided into two disjoint subsets Ω_1 and Ω_2 such that $\Omega_1 \cup \Omega_2 = \Omega$. In cross-validation, we just randomly select a portion of the observed entries, i.e., $\boldsymbol{X}_{\Omega_1}$, as the training data for matrix completion. The portion of training data $|\Omega_1|/|\Omega|$ can be set to 95%. For a given rank, matrix completion is performed based on $\boldsymbol{X}_{\Omega_1}$. We then compute the mean prediction error on the testing data $\boldsymbol{X}_{\Omega_2}$ based on multiple random divisions of Ω_1 and Ω_2. The rank is chosen as the one which corresponds to the smallest prediction error. Suppose that L random trials are carried out for calculating the prediction error. In the lth trial, the two sets are randomly generated, which are denoted as Ω_1^l and Ω_2^l. A matrix completion algorithm using partial noisy observations $\boldsymbol{X}_{\Omega_1^l}$ and rank r gives the result $\widehat{\boldsymbol{M}}(r)$. Since $\boldsymbol{X}$ is unknown, we cannot calculate the estimation error of (49). Instead, the prediction error of the testing data $\boldsymbol{X}_{\Omega_2}$ is evaluated. That is, the rank is estimated by minimizing the following root mean square

prediction error (RMSPE)

$$\widehat{r} = \arg\min_{r\in\mathbb{Z}^+} \sum_{l=1}^{L} \frac{\left\| [\widehat{\boldsymbol{M}}(r)]_{\Omega_2^l} - \boldsymbol{X}_{\Omega_2^l} \right\|_F^2}{\left\| \boldsymbol{X}_{\Omega_2^l} \right\|_F^2} \tag{50}$$

or mean absolute prediction error (MAPE)

$$\widehat{r} = \arg\min_{r\in\mathbb{Z}^+} \sum_{l=1}^{L} \frac{\left\| [\widehat{\boldsymbol{M}}(r)]_{\Omega_2^l} - \boldsymbol{X}_{\Omega_2^l} \right\|_1}{\left\| \boldsymbol{X}_{\Omega_2^l} \right\|_1}. \tag{51}$$

The reason why we also adopt the MAPE is that $\boldsymbol{X}_{\Omega_2}$ can contain outliers and the ℓ_1-norm is a more outlier-robust distance measure. Simulation results on the choice of r with outliers are provided in Section V.

On the other hand, the optimal choice of p is case-dependent. It relies on the statistical properties of the noise. As mentioned in Section II, $p = 2$ is optimal for Gaussian noise. In the presence of impulsive noise or outliers, $p < 2$ will bring a better performance. Consider a special case where the noise satisfies a zero-mean generalized Gaussian distribution (GGD) [40], whose probability density function (PDF) with variance σ_v^2 is

$$p_v(v) = \frac{\beta\Gamma(4/\beta)}{2\pi\sigma_v^2\Gamma^2(2/\beta)} \exp\left(-\frac{|v|^\beta}{\kappa\sigma_v^\beta}\right) \tag{52}$$

where $\beta > 0$ is the shape parameter, $\Gamma(\cdot)$ is the Gamma function, and $\kappa = (\Gamma(2/\beta)/\Gamma(4/\beta))^{\beta/2}$ [40]. When $\beta = 2$, GGD reduces to the Gaussian distribution. The case of $\beta < 2$ models super-Gaussian distributions. Especially, $\beta = 1$ corresponds to the Laplacian distribution [40]. The smaller the value of β is, the more impulsive the noise is. If the shape parameter β is known, then we can select $p = \beta$ which gives the maximum likelihood (ML) estimate. Since the ML estimate asymptotically approaches the minimum variance, $p = \beta$ is statistically optimal for GGD noise. In the general case with possibly unknown noise statistics, the optimal p aims at minimizing the estimation error

$$\min_{p>0} \|\widehat{\boldsymbol{M}}(p) - \boldsymbol{X}\|_F^2. \tag{53}$$

where $\widehat{\boldsymbol{M}}(p)$ denotes the solution of (9) for a given p. Again, it is impractical to obtain the optimal p because $\boldsymbol{X}$ is not available in practice. Roughly speaking, to select a proper p from $(0, 2)$, we need to consider the following two aspects.

1) Statistical perspective. The statistical property of the noise needs to be taken into account. The more impulsive the noise is, the smaller value of p is preferred. If the noise is not so impulsive, the choice of $1 < p \leq 2$ is suitable. If the noise is more impulsive, it has a more spike-like property, which is somewhat analogous to sparsity. As analyzed in the literature on sparse signal recovery and compressed sensing [12], [64], sparsity can be well measured using the ℓ_1-norm or even better using ℓ_p-norm with $p < 1$.
2) Computational perspective. As p decreases to zero, the nonconvexity and nonsmoothness of the ℓ_p-norm becomes stronger, which brings more difficulties in minimization. The computational challenges induced by a very small p includes increased probability of being trapped into local minima far away from the global minimum and slow convergence rate. Therefore, it is not recommended to choose p close to 0.

To summarize, choosing an appropriate p is a trade-off between the statistical and computational aspects. For ADMM, the proximity operator of the ℓ_1-norm is computationally simplest since it has closed-form expression. Thus, it is preferred to choose $p = 1$ for the ADMM. If there is no prior information for the noise, we can resort to cross-validation, which has been discussed above for rank selection, to determine p. The reader is referred to the simulation results on the choice of p with outliers in Section V.

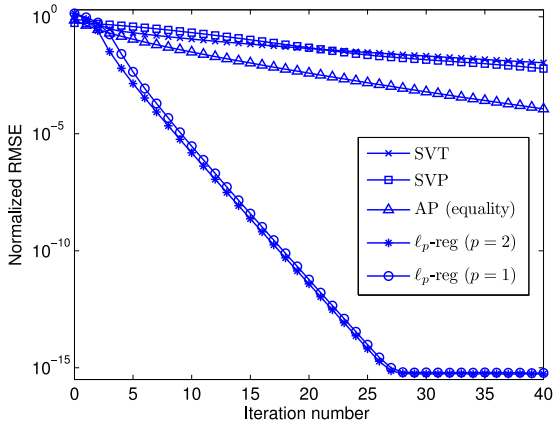

Fig. 1. Normalized RMSE versus iteration number for noise-free case.

V. Simulation Results

All the simulations are conducted using a computer with a 3.2 GHz CPU and 4 GB memory.

A. Results of Synthetic Random Data

Under stated otherwise, a typical experimental setting in [16] is considered where $n_1 = 150$, $n_2 = 300$, and the rank is $r = 10$. The proposed algorithms are compared with SVT [16], SVP [26], and AP [28], WNNM [24], RPCA for matrix completion (RPCA-MC) [43], PARSuMi [42], VBMFL_1 [46]. A noise-free matrix $\boldsymbol{X} \in \mathbb{R}^{n_1\times n_2}$ of rank r is generated by the product of $\boldsymbol{X}_1 \in \mathbb{R}^{n_1\times r}$ and $\boldsymbol{X}_2 \in \mathbb{R}^{r\times n_2}$ whose entries satisfy the standard Gaussian distribution. We randomly select 45% entries of $\boldsymbol{X}$ as the available observations. The normalized root mean square error (RMSE) is employed as the performance measure, which is defined as:

$$\mathrm{RMSE}(\widehat{\boldsymbol{M}}) = \sqrt{\mathrm{E}\left\{\frac{\|\widehat{\boldsymbol{M}} - \boldsymbol{X}\|_F^2}{\|\boldsymbol{X}\|_F^2}\right\}} \tag{54}$$

where $\widehat{\boldsymbol{M}}$ is the result obtained by a matrix completion method, and is computed based on 100 independent trials.

Fig. 1 plots the RMSE versus iteration number in the noise-free case where ℓ_p-reg represents the iterative ℓ_p-regression method in Algorithm 1. Note that we do not show the result

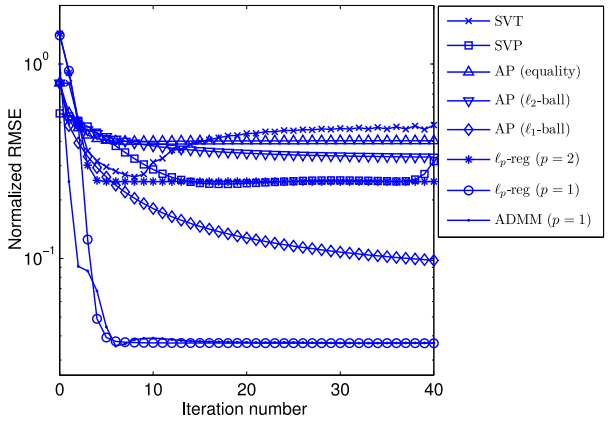

Fig. 2. Normalized RMSE versus iteration number in GMM noise at SNR = 6 dB.

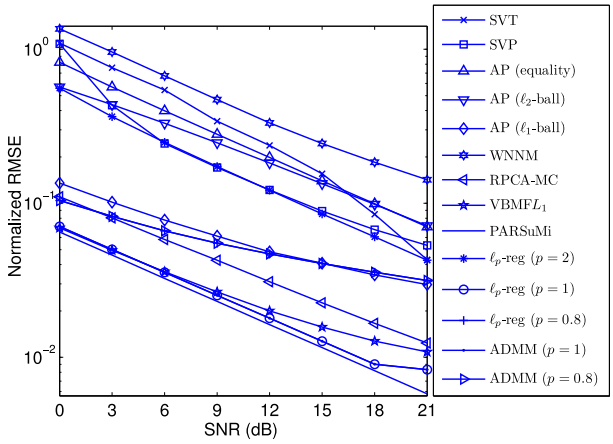

Fig. 3. Normalized RMSE versus SNR in GMM noise.

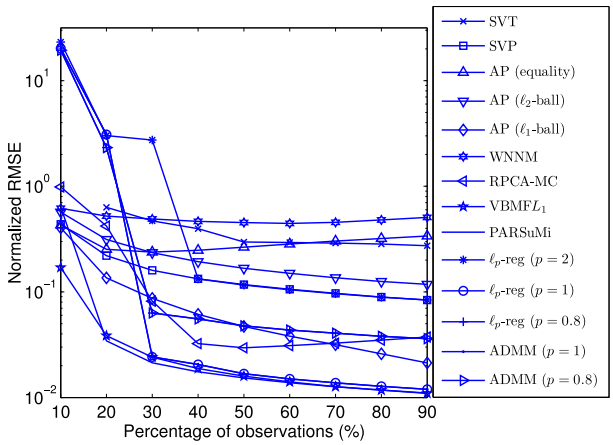

Fig. 4. Normalized RMSE versus percentage of observations in GMM noise at SNR = 9 dB.

of the ADMM because for any p, Algorithm 2 converges to the true solution in one iteration. It is observed that the SVT, SVP, AP with equality projection, and ℓ_p-regression schemes converge to the true matrix with a linear rate. However, our proposed method converges much faster and only about ten iterations are needed to obtain an accurate solution. The CPU times for attaining $\mathrm{RMSE} \leq 10^{-5}$ of the SVT, SVP, AP with equality projection, ℓ_p-reg with $p = 2$ and $p = 1$ are 10.7 s, 8.0 s, 6.7 s, 0.28 s, and 4.5 s, respectively.

We then consider the noisy scenario where impulsive components are added to the availableentries in $\boldsymbol{X}$. They are modeled by the two-term zero-mean Gaussian mixture model (GMM) whose PDF is given by

$$p_v(v) = \sum_{i=1}^{2} \frac{c_i}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{v^2}{2\sigma_i^2}\right) \tag{55}$$

where $0 \leq c_i \leq 1$ and σ_i^2 are the probability and variance of the ith term, respectively, with $c_1 + c_2 = 1$. If $\sigma_2^2 \gg \sigma_1^2$ and $c_2 < c_1$ are selected, large noise samples of variance σ_2^2 occurring with a smaller probability c_2 can be viewed as outliers embedded in Gaussian background noise of variance σ_1^2. Thus, the GMM can well model the phenomenon with both Gaussian noise and outliers. The total noise variance is $\sigma_v^2 = \sum_i c_i \sigma_i^2$ and the signal-to-noise ratio (SNR) is defined as

$$\mathrm{SNR} = \frac{\|\boldsymbol{X}_\Omega\|_F^2}{|\Omega|\sigma_v^2}. \tag{56}$$

Fig. 2 plots the RMSE versus iteration number in additive GMM noise at SNR = 6 dB with $\sigma_2^2 = 100\sigma_1^2$ and $c_2 = 0.1$. We see that the SVT and SVP cannot stably converge to a reasonable solution. The iterative ℓ_p-regression and ADMM with $p = 1$ converge fast to a solution with a higher accuracy while those with $p = 2$ and the AP with projections onto equality and ℓ_2-ball cannot achieve a reliable estimation in impulsive noise. The AP with projection onto ℓ_1-ball is somewhat robust to outliers. Still, its performance is worse than the proposed schemes. Importantly, we see that about ten iterations are enough for our two algorithms to converge. That is, a value of several tens for K_{reg} and K_{ADMM} is enough for convergence. Employing the stopping criteria of relative change of the current and previous iterations is less than 10^{-4} and (45) with $\delta = 10^{-3}$ in the ℓ_p-regression and ADMM algorithms, respectively, the CPU times of the SVT, SVP, AP with projections onto equality, ℓ_2-ball, and ℓ_1-ball, ℓ_p-reg with $p = 2$ and $p = 1$, and ADMM with $p = 1$ are 197.3 s, 10.6 s, 7.5 s, 7.9 s, 8.4 s, 0.25 s, 5.2 s, and 3.1 s, respectively.

Fig. 3 plots the RMSE versus SNR for different methods. It is seen that the ℓ_1-regression, ADMM with $p = 1$, PARSuMi and VBMFL_1 have comparable performance. The four schemes have the minimum RMSE for all SNRs and thus they are superior to the remaining schemes in terms of robustness. Though it is slightly inferior to the four methods above, the RPCA-MC performs better than the SVT, SVP, AP, and WNNM. Fig. 4 plots the RMSE versus percentage of observations, i.e., $|\Omega|/(n_1 n_2)$ at SNR = 9 dB in GMM noise and $r = 5$. Again, the two proposed methods with $p = 1$, PARSuMi and VBMFL_1 have the best performance. Note that the SVT reports divergence for percentage of 10% and thus the result at this point is not included.

Fig. 5 plots the average running time versus percentage of observations of the ℓ_p-reg with $p = 1, 2$ and ADMM with $p = 1$.

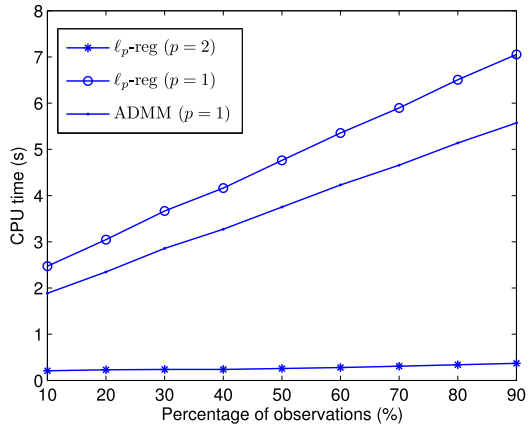

Fig. 5. CPU time versus percentage of observations.

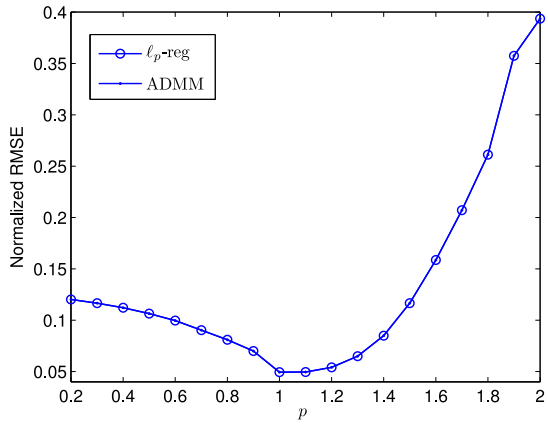

Fig. 6. Normalized RMSE versus p in strongly impulsive GMM noise.

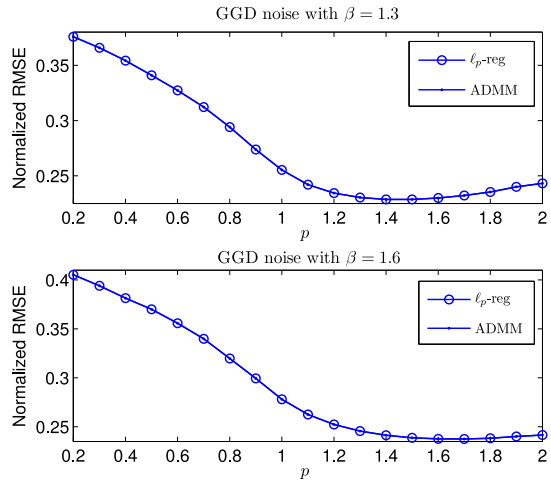

Fig. 7. Normalized RMSE versus p in moderately impulsive GGD noise.

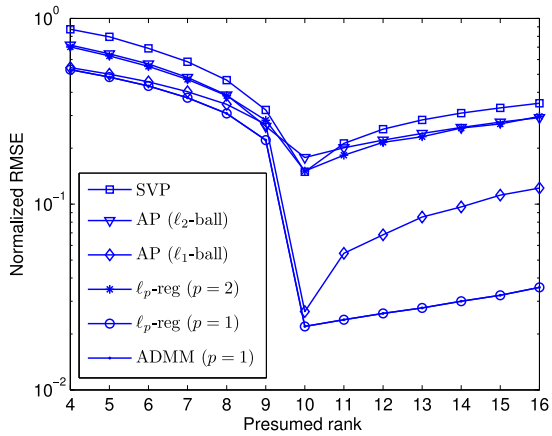

Fig. 8. Normalized RMSE versus presumed rank.

We observe that the running time linearly increases with the percentage of observations. Note that the computational complexity of the two proposed methods is $\mathcal{O}(K|\Omega|r^2)$ where the number of observations $|\Omega|$ is the product of the observation percentage and the number of total entries of the matrix. Therefore, the complexity is linearly proportional to the percentage of observations, which aligns the results of Fig. 5.

Impact of Rank and p: The impact of p on the performance is investigated. First a strongly impulsive GMM noise with SNR $= 6$ dB is used. Fig. 6 plots the RMSE versus $p \in [0.2, 2]$. It is seen that using $p < 1$ is worse than $p = 1$. This may be explained from the computational perspective. As p decreases to zero, the nonconvexity and nonsmoothness of the ℓ_p-norm makes its minimization more difficult. Therefore, it is not recommended to choose p close to 0. Also, as p increases in $[1, 2]$, the robustness degrades. For computational simplicity and performance improvement, the value of $p = 1$ is the best choice for strongly impulsive noise. Then the moderately impulsive GGD noises at SNR $= 6$ dB with $\beta = 1.3$ and $\beta = 1.6$ are used. Fig. 7 shows the RMSE versus $p \in [0.2, 2]$ in GGD noise. As we see, the optimal p is close to the shape parameter β of the GGD noise. If the noise is not so impulsive, it is preferred to employ $p \in (1, 2)$ instead of $p \leq 1$.

We study how the presumed rank affects the performance of the proposed approaches as well as SVP and AP, which also require rank information. The experimental setting is the same as above except SNR $= 9$ dB. Fig. 8 shows the normalized RMSE versus the presumed rank varying from 4 to 16 while the true value is 10. All the methods degrade when the rank is not accurately estimated. In addition, the performance degradation when the rank is underestimated ($r < 10$) is much severer than the case when the rank is overestimated ($r > 10$). This result implies that it is not preferred to underestimate the rank. The ℓ_p-reg and ADMM with $p = 1$ exhibit the best robustness to the rank estimation error.

Results of Cross-Validation: Rank estimation using cross-validation is investigated. The experimental setting is the same as above except that only 95% of the observed entries are randomly drawn out as training data while the remaining 5% observed entries are taken as testing data, i.e., $|\Omega_1|/|\Omega| = 0.95$. For each rank $r \in [4, 16]$, 100 random trials are conducted to compute the average RMSPE of (50) and MAPE of (51). Figs. 9 and 10 plot the RMSPE and MAPE versus the presumed rank. It is clearly observed that all methods, including the proposed algorithms, RMSPE and MAPE are minimized at $r = 10$, which

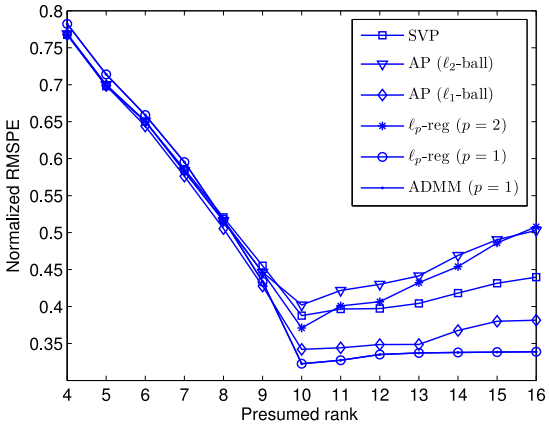

Fig. 9. Normalized RMSPE versus presumed rank in cross-validation.

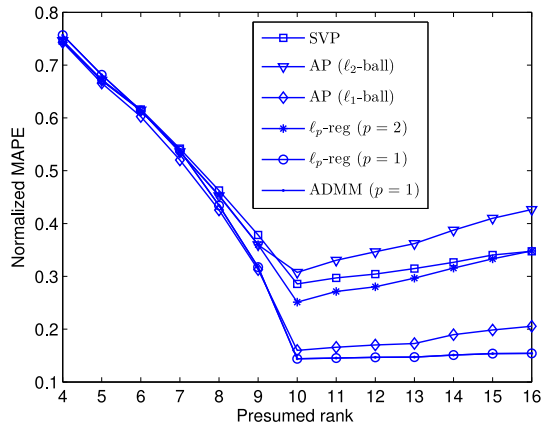

Fig. 10. Normalized MAPE versus presumed rank in cross-validation.

is exactly the true rank. The effectiveness of cross validation for rank estimation is thus verified. Nevertheless, the MAPE gives more stable result than the RMSPE, indicating that it is more suitable in the presence of outliers.

Phase Transition: Phase transition figures, i.e., the probability of recovery and normalized RMSE, versus rank and percentage of missing entries, are shown in Figs. 11 and 12, respectively. For each pair of rank and missing percentage, 100 independent trials are carried out. Since the observations are noisy, we declare a trial to be successful if the normalized RMSE is less than 0.2. The SNR is fixed as 9 dB while the rank and missing percentage vary. In addition to the two proposed methods with $p = 1$ and 2, the result of SVP is included for comparison. From Fig. 11, it is observed that the "white region" of the ℓ_p-reg and ADMM with $p = 1$ is larger than those of $p = 2$ and SVP. This means that the proposed methods with $p = 1$ perform better when the rank or missing percentage is large. The smaller RMSEs of the ℓ_p-reg/ADMM with $p = 1$ in Fig. 12 also validate their superior performance in the presence of outliers.

Scalability: In the era of big data, it is of great interest to know whether a matrix completion algorithm is scalable to the dimension of the problem. Theoretically, the computational complexity of the two proposed methods is $\mathcal{O}(Kp_{\text{obs}}n_1n_2r^2)$ where $p_{\text{obs}} \in (0, 1]$ is the percentage of observations. Herein, simulations are conducted to check this computational complexity. First we fix $p_{\text{obs}} = 0.45$ and $r = 10$ while the matrix dimension n_1n_2 varies from 10^3 to 10^7. In this simulation, we assign $n_1 = n_2$, meaning that n_1 varies from 32 to 3162. Fig. 13 shows the CPU time and RMSE versus n_1n_2. It is seen that the CPU time is linearly proportional to n_1n_2. This result verifies the linear time-complexity and hence the scalability of the proposed algorithms. Also, it is observed that the RMSE decreases as the matrix dimension increases provided that the rank and observation percentage remain unchanged. We then fix $n_1 = n_2 = 200$ and $p_{\text{obs}} = 0.45$ while the rank varies from 1 to 29. Fig. 14 shows the CPU time and RMSE versus the rank. We observe that the CPU time quadratically increases with the rank, which aligns the complexity of $\mathcal{O}(Kp_{\text{obs}}n_1n_2r^2)$. In this sense, the proposed schemes are not scalable to the rank. Fortunately, the rank is often much smaller than the size of the matrix in practical applications. The low-rank property is helpful to reduce the computational cost and improve the recovery performance.

B. Image Inpainting in Salt-and-Pepper Noise

Now matrix completion is applied to image inpainting in salt-and-pepper noise. A color image in [65], [66] is adopted where we first convert it to gray-scale so that it can be represented by a matrix. As shown in Fig. 15, the missing data of the original image correspond to "ICCV", "2009", and "LRTC". The available entries are contaminated by adding salt-and-pepper noise. We use the function "`imnoise(I, 'salt & pepper', ρ)` " in MATLAB, where the normalized noise intensity is ρ corresponding to SNR $= 1/\rho$, to generate the salt-and-pepper noise. The widely-used peak signal-to-noise ratio (PSNR)

$$\text{PSNR} = 255^2/\text{MSE} \tag{57}$$

where 255 is the peak value of a gray-scale image and

$$\text{MSE} = \frac{1}{n_1n_2}\|\widehat{\boldsymbol{M}} - \boldsymbol{X}\|_F^2. \tag{58}$$

Obviously, the smaller MSE, the larger PSNR. That is, a larger PSNR implies a better image reconstruction. The PSNR of the noisy image with missing values without any processing can be considered as the baseline. Generally, the PSNR will be increased after processing by an image inpainting algorithm.

We first set the rank as $r = 6$. The SVT shows divergence and we cannot include its result while it is observed that the SVP, AP with equality projection, and WNNM fail in recovering the image. The AP with projection onto ℓ_1-ball gives a satisfactory result but is still inferior to our two methods. Most important, the ℓ_p-regression and ADMM with $p = 1$ are quite robust to the salt-and-pepper noise and they provide accurate estimates of the original image. We also see that the ℓ_1-regression greatly improves the performance compared with the ℓ_2-regression in impulsive noise environment. The PARSuMi and VBMFL_1 also exhibit robustness to salt-and-pepper noise. The CPU times for the SVP, AP with projections onto equality and ℓ_1-ball, WNNM, PARSuMi, VBMFL_1, ℓ_p-regression with $p = 2$ and $p = 1$, and ADMM with $p = 1$ are 20.3 s, 15.6 s, 17.4 s, 140.3 s, 9.0 s, 7.4 s, 0.4 s, 7.8 s and 4.9 s, respectively.

The effect of rank selection to the performance of image inpainting is investigated. Fig. 16 shows the PSNR versus rank in salt-and-pepper noise at SNR $= 7$ dB. The baseline is also

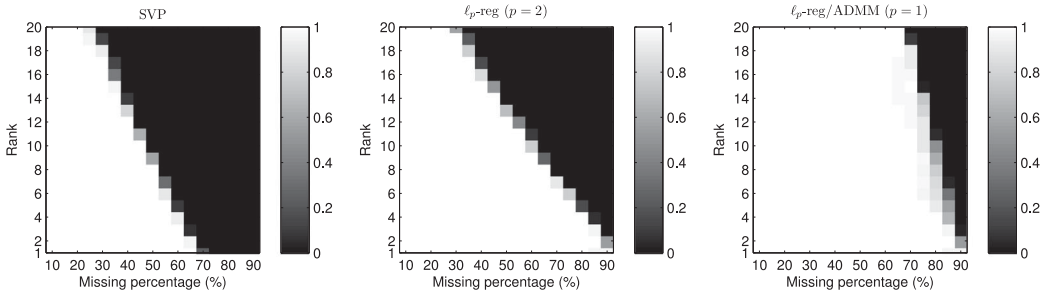

Fig. 11. Phase transition of probability of recovery versus missing percentage and rank.

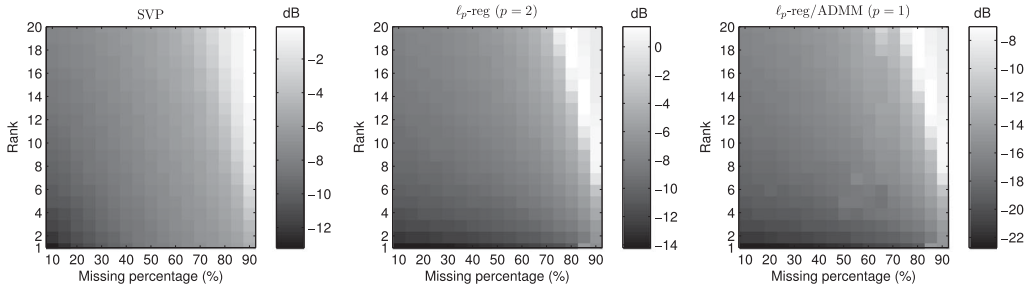

Fig. 12. Phase transition of normalized RMSE versus missing percentage and rank.

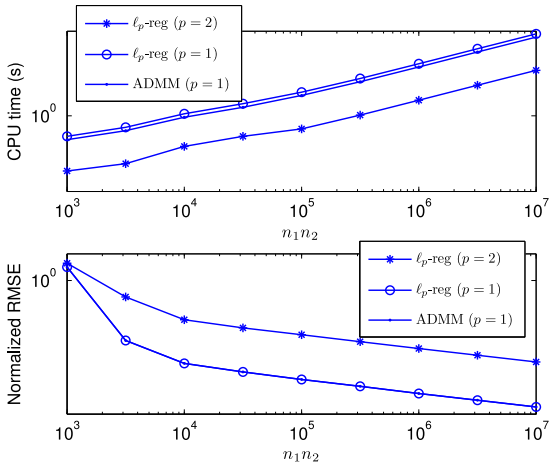

Fig. 13. Running time and normalized RMSE versus matrix dimension $n_1 n_2$.

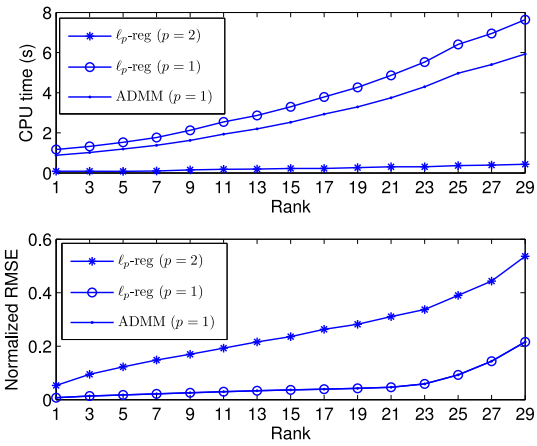

Fig. 14. Running time and normalized RMSE versus rank.

plotted. We see that the two proposed algorithms and the AP with $p = 1$ have the highest PSNR around $r = 6$ or 7. The PARSuMi is not sensitive to rank in this experiment example. The PSNR of VBMFL_1 quickly increases as the rank increases when $r \leq 6$ and it slowly decreases when $r \geq 14$. Therefore, the rank of VBMFL_1 can take values in $r \in [6, 14]$. Because the computational load becomes heavier as the rank increases, it is preferred to select a smaller rank when the performance is similar. Fig. 16 shows the PSNR versus SNR at $r = 6$. From Fig. 17, the VBMFL_1, AP with ℓ_1-ball projection, and the two proposed approaches with $p = 1$ have the best performance. Since the SVP and WNNM are not robust to the salt-and-pepper noise and their PSNRs are low, we do not show the corresponding results of them in Figs. 16 and 17.

We then investigate inpainting of another two images whose original versions are taken from [65], [66]. The color images are converted to gray-scale for a matrix representation. The first image is a building and the second is a texture. Both of them are structured and approximately have a low-rank property. The rank is set to $r = 6$. Figs. 18 and 19 show the original and incomplete images corrupted by salt-and-pepper noise, and the recovered results of SVP, AP, WNNM, PARSuMi, VBMFL_1, iterative ℓ_p-regression and ADMM. Again, we see that the VBMFL_1, ℓ_p-regression with $p = 1$ and ADMM are quite robust to impulsive noise and have the best recovery performance. The PARSuMi and AP with projection onto ℓ_1-ball are inferior to the three methods although they yield satisfactory results. The SVP, AP with equality projection, WNNM are not robust to salt-and-pepper noise.

C. Results of Recommender Systems

The application of our matrix completion methods to recommender systems is considered. The MovieLens 100 K Data

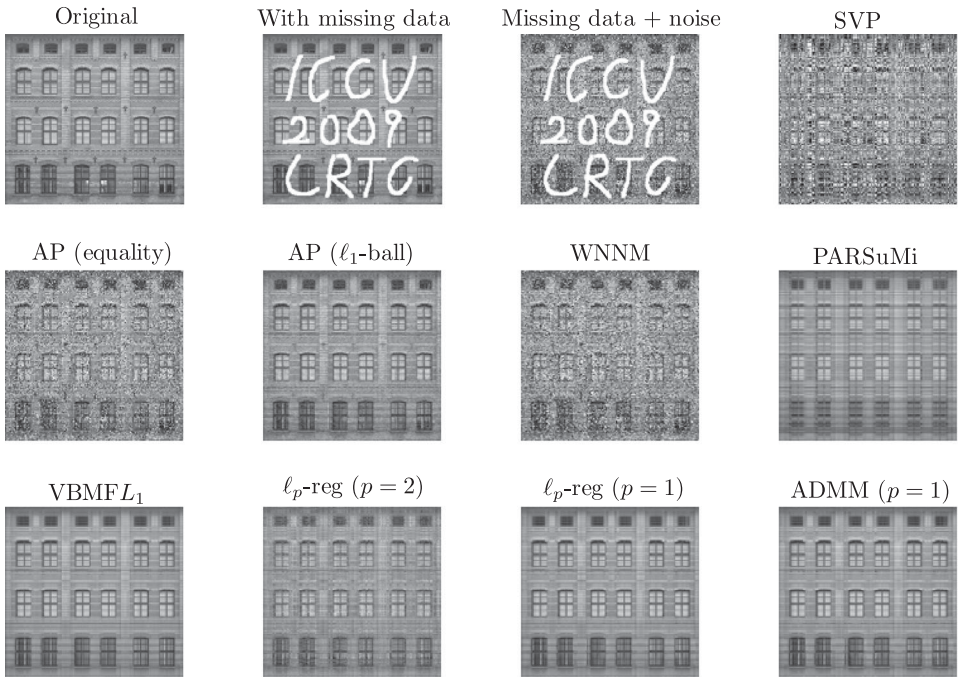

Fig. 15. Noisy image with missing data and recovered results of SVP, AP, WNNM, PARSuMi, VBMFL_1, iterative ℓ_p-regression and ADMM.

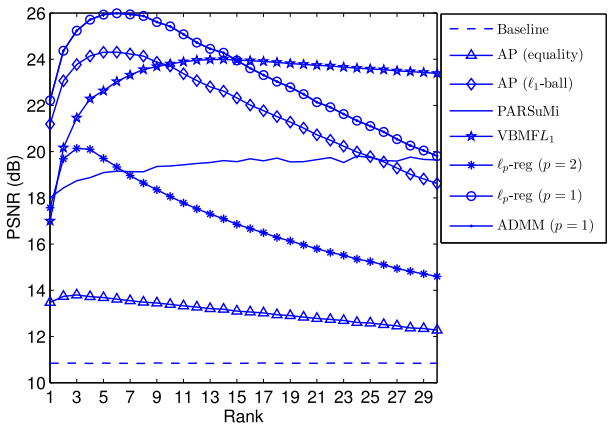

Fig. 16. PSNR versus rank in salt-and-pepper noise at SNR = 7 dB.

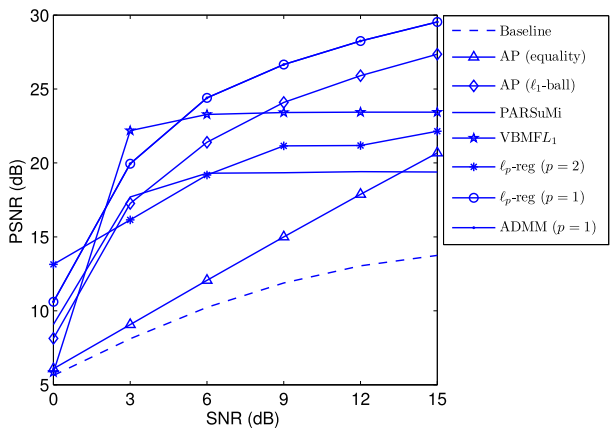

Fig. 17. PSNR versus SNR in salt-and-pepper noise.

set, which is available at [67], is used. This data set consists of 100,000 ratings from 943 users on 1,682 movies. The rating varies from 1 to 5. Each user has rated at least 20 movies. The rating of the ith user to the jth movie is stored as the (i, j)th entry of the matrix $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$. We have $n_1 = 943$, $n_2 = 1682$, and the number of known entries $|\Omega| = 10^5$, which is much smaller than the number of all entries $n_1 n_2 = 1.586 \times 10^6$. That is, the percentage of observations is only 6.3%. Low-rank matrix completion is applied to infer other unknown entries for a recommender system. Since the remaining 93.7% entries are unknown, we cannot judge whether the inferred entries are correct. Like the strategy in cross-validation, Ω is divided into Ω_1 and Ω_2 such that $\Omega_1 \cup \Omega_2 = \Omega$. In this experiment, $\boldsymbol{X}_{\Omega_1}$ and $\boldsymbol{X}_{\Omega_2}$ are used for matrix completion and prediction error computation, respectively. Define the result of matrix completion using partial observations $\boldsymbol{X}_{\Omega_1}$ as $\widehat{\boldsymbol{M}}$, we evaluate the mean absolute error (MAE)

$$\text{MAE} = \frac{1}{4\,|\Omega_2|} \left\| [\widehat{\boldsymbol{M}}]_{\Omega_2} - \boldsymbol{X}_{\Omega_2} \right\|_1 \tag{59}$$

using $\boldsymbol{X}_{\Omega_2}$, where the factor 4 is the difference of the maximum and minimum scores, namely, 5 and 1. Note that the MAE has been widely used as the performance measure of recommender systems [21], [37].

We first use cross-validation to estimate an appropriate rank, where the portion of training data $|\Omega_1|/|\Omega|$ is set to 95%. For a given rank, matrix completion is performed using $\boldsymbol{X}_{\Omega_1}$. For each rank $r \in [1, 15]$, 100 random divisions of Ω_1 and Ω_2 are conducted to compute the average MAE. Fig. 20 plots the MAE versus the estimated rank. For the AP and two proposed methods, $r = 3$ is the best rank estimate while for the SVP the rank estimate is $r = 7$. These rank estimates are adopted in the following tests.

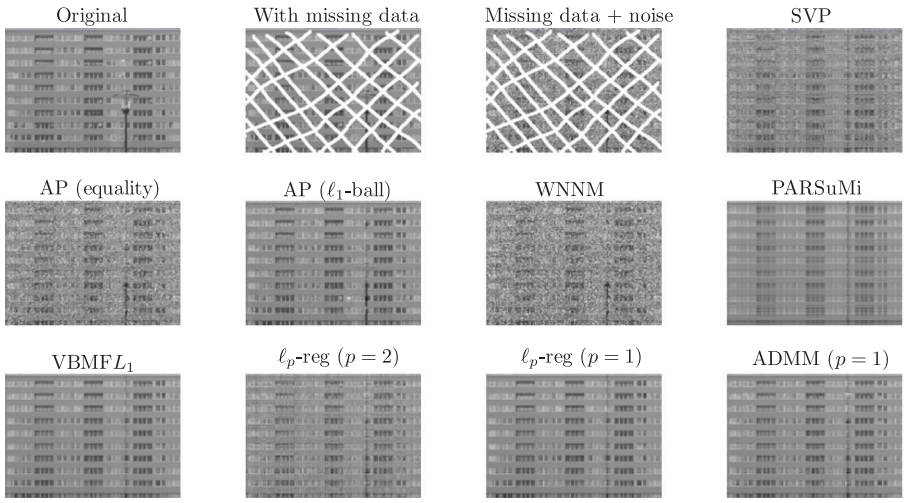

Fig. 18. Noisy image of a building with missing data and recovered results of SVP, AP, WNNM, PARSuMi, VBMFL_1, iterative ℓ_p-regression and ADMM.

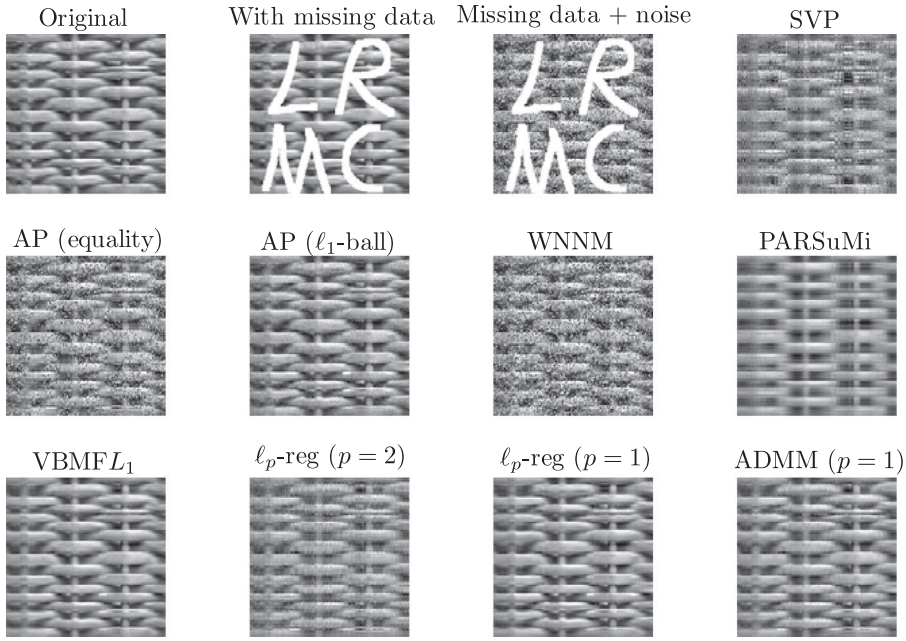

Fig. 19. Noisy image of a texture with missing values and recovered results of SVP, AP, WNNM, PARSuMi, VBMFL_1, iterative ℓ_p-regression and ADMM.

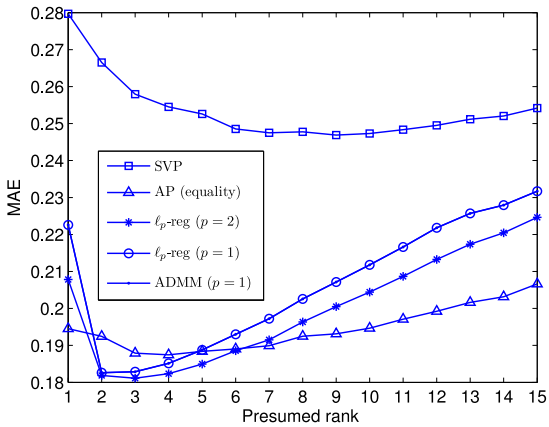

Fig. 20. MAE versus rank in cross-validation with MovieLens data set.

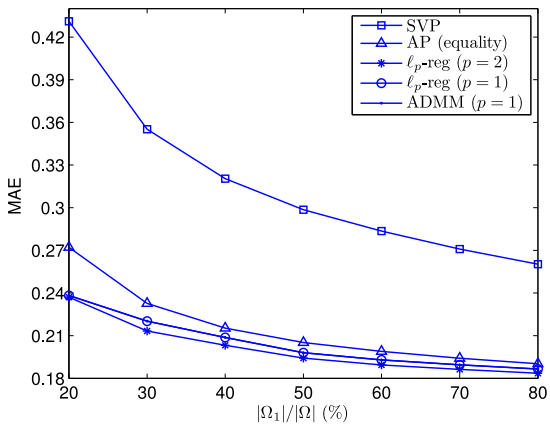

Fig. 21. MAE versus $|\Omega_1|/|\Omega|$ with MovieLens data set.

Fig. 21 plots the MAE versus $|\Omega_1|/|\Omega|$ varying from 20% to 80% of the SVP, AP with equality projection, and ℓ_p-regression and ADMM. The prediction accuracy of two proposed methods is higher than AP and SVP. The performances using $p = 1$ and $p = 2$ are quite similar. This is because the ratings are integers whose values are taken from $\{1, \cdots, 5\}$ and there are no random noises or outliers. In the absence of noise/outlier, the method with $p = 2$ is good enough for matrix completion. Still, the proposed scheme with $p = 1$ also works well and useful for the case where there is no noise or outlier.

VI. Conclusion

Many existing techniques for matrix completion are not robust to outliers. To overcome this drawback, we have devised two algorithms for robust matrix completion using low-rank factorization via the ℓ_p-minimization criterion with $0 < p < 2$. The first method tackles the nonconvex factorization with missing data by iteratively solving multiple independent linear ℓ_p-regressions. On the other hand, the second solution exploits the ADMM for incomplete factorization in ℓ_p-space. Each iteration of the ADMM requires solving an LS factorization problem and calculating the proximity operator of the ℓ_p-norm. The two algorithms have comparable recovery performance as well as computational efficiency and allow parallel or distributed realization. Their total complexity is $\mathcal{O}(K|\Omega|r^2)$, where K is a fixed constant of several hundreds to thousands, which is lower than the popular schemes employing the nuclear/Schatten p-norm minimization that require SVD. Furthermore, our solutions generalize the conventional matrix factorization based on Frobenius norm minimization. The superiority of the developed algorithms over the SVT, SVP, and AP in terms of implementation complexity, recovery capability and outlier-robustness is demonstrated using synthetic and real-world data.

References

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

[2] M. A. Davenport and J. Romberg, "An overview of low-rank matrix recovery from incomplete observations," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 608–622, Jun. 2016.

[3] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, pp. 211–218, 1936.

[4] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-scale parallel collaborative filtering for the Netflix prize," in *Proc. 4th Int Conf. Algorithmic Appl. Manag.*, 2008, pp. 337–348.

[5] S. Funk, "Netflix update: Try this at home," 2006. [Online]. Available: http://sifter.org/ simon/journal/20061211.html

[6] J. Huang, F. Nie, H. Huang, Y. Lei, and C. Ding, "Social trust prediction using rank-k matrix recovery," in *Proc. 23th Int. Joint Conf. Artif. Intell.*, Beijing, China, Aug. 2013, pp. 2647–2653.

[7] H. Kim, G. H. Golub, and H. Park, "Missing value estimation for DNA microarray gene expression data: Local least squares imputation," *Bioinformatics*, vol. 21, no. 2, pp. 187–198, 2005.

[8] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. ACM SIGGRAPH*, New Orleans, LA, USA, Jul. 2000, pp. 414–424.

[9] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2053–2080, May 2010.

[10] M. Fazel, H. Hindi, and S. Boyd, "A rank minimization heuristic with application to minimum order system approximation," in *Proc. Amer. Control Conf.*, Arlington, VA, USA, Jun. 2001, pp. 4734–4739.

[11] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, 2010.

[12] E. J. Candès and M. B. Wakin, "An introduction to compressed sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[13] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Statist. Soc., Ser. B*, vol. 58, no. 1, pp. 267–288, 1996.

[14] S. J. Wright, R. D. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Trans. Signal Process.*, vol. 57, no. 7, pp. 2479–2493, Jul. 2009.

[15] Z. Liu and L. Vandenberghe, "Interior-point method for nuclear norm approximation with application to system identification," *SIAM J. Matrix Anal. Appl.*, vol. 31, no. 3, pp. 1235–1256, Aug. 2009.

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

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

[18] K. C. Toh and S. W. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems," *Pac. J. Opt.*, vol. 6, pp. 615–640, 2010.

[19] 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.

[20] 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.

[21] F. Nie, H. Wang, X. Cai, H. Huang, and C. Ding, "Robust matrix completion via joint Schatten p-norm and ℓ_p-norm minimization," in *Proc. 12th Int. Conf. Data Mining*, Brussels, Belgium, Dec. 2012, pp. 566–574.

[22] K. Mohan and M. Fazel, "Iterative reweighted algorithms for matrix rank minimization," *J. Mach. Learn. Res.*, vol. 13, pp. 3441–3473, Nov. 2012.

[23] 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.

[24] 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.

[25] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comput. Harmonic Anal.*, vol. 27, no. 3, pp. 265–274, 2009.

[26] P. Jain, R. Meka, and I. S. Dhillon, "Guaranteed rank minimization via singular value projection," in *Proc. 23rd Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 937–945.

[27] J. Tanner and K. Wei, "Normalized iterative hard thresholding for matrix completion," *SIAM J. Sci. Comput.*, vol. 35, no. 5, pp. S104–S125, Oct. 2013.

[28] 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.

[29] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming*, 4th Ed. New York, NY, USA: Springer-Verlag, 2016.

[30] L. N. Trefethen and D. Bau, III, *Numerical Linear Algebra.* Philadelphia, PA, USA: SIAM, 1997.

[31] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2980–2998, Jun. 2010.

[32] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[33] R. Sun and Z.-Q. Luo, "Guaranteed matrix completion via nonconvex factorization," in *Proc. 56th IEEE Annu. Symp. Found. Comput. Sci.*, Berkeley, CA, USA, Oct. 2015, pp. 270–289.

[34] M. Hardt, "Understanding alternating minimization for matrix completion," in *Proc. 55th IEEE Annu. Symp. Found. Comput. Sci.*, Philadelphia, PA, USA, Oct. 2014, pp. 651–660.

[35] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proc. 45th Annu. ACM Symp. Theory Comput.*, Palo Alto, CA, USA, Jun. 2013, pp. 665–674.

[36] 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.

[37] J. Huang, F. Nie, and H. Huang, "Robust discrete matrix completion," in *Proc. 26th AAAI Conf. Artif. Intell.*, Bellevue, WA, USA, Jul. 2013, pp. 424–430.

[38] F. Nie, J. Yuan, and H. Huang, "Optimal mean robust principal component analysis," in *Proc. 31st Int. Conf. Mach. Learn.*, Beijing, China, Jun. 2014, pp. 1062–1070.

[39] H. Zhang, Z. Lin, and C. Zhang, "Completing low-rank matrices with corrupted samples from few coefficients in general basis," *IEEE Inf. Theory*, vol. 62, no. 8, pp. 4748–4768, Aug. 2016.
[40] W.-J. Zeng, H. C. So, and L. Huang, "ℓ_p-MUSIC: Robust direction-of-arrival estimator for impulsive noise environments," *IEEE Trans. Signal Process.*, vol. 61, no. 17, pp. 4296–4308, Sep. 2013.
[41] A. Eriksson and A. van den Hengel, "Efficient computation of robust low-rank matrix approximations in the presence of missing data using the L_1 norm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 771–778.
[42] Y.-X. Wang, C. M. Lee, L.-F. Cheong, 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.
[43] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?," *J. ACM*, vol. 58, no. 3, May 2011, Art. no. 11.
[44] J. Wright, A. Ganesh, K. Min, and Yi Ma, "Compressive prinicpal component pursuit," *Inf. Inference*, vol. 2, no. 1, pp. 32–68, Jun. 2013.
[45] M. Oskarsson, K. Batstone, and K. Åström, "Trust no one: Low rank matrix factorization using hierarchical RANSAC," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 5820–5825.
[46] 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, Apr. 2015.
[47] P.-S. Laplace, "Sur quelques points du système du monde," *Mémoires de l'Académie des Sciences de Paris*, 1789. Reprinted in *Evres Complétes*, vol. 11, pp. 475–558.
[48] P. Bloomfield and W. L. Steiger. *Least Absolute Deviations: Theory, Applications, and Algorithms.* Boston, MA, USA: Birkhäuser, 1983.
[49] S. Portnoy and R. Koenker, "The Gaussian hare and the Laplacian tortoise: Computability of squared-error versus absolute-error estimators," *Statist. Sci.*, vol. 12, no. 4, pp. 279–300, 1997.
[50] D. P. O'Leary, "Robust regression computation using iteratively reweighted least squares," *SIAM. J. Matrix Anal. Appl.*, vol. 11, no. 3, pp. 466–480, 1990.
[51] R. A. Maronna, R. D. Martin, and V. J. Yohai, *Robust Statistics: Theory and Methods.* New York, NY, USA: Wiley, 2006.
[52] P. Tseng, "Convergence of a block coordinate descent method for non-differentiable minimization," *J. Optim. Theory Appl.*, vol. 109, no. 3, pp. 475–494, Jun. 2001.
[53] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
[54] J. Eckstein, "Augmented Lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results," RUTCOR, Res. Rep. RRR 32–2012, Dec. 2012.
[55] R. Glowinski, "On alternating direction methods of multipliers: A historical perspective," in *Modeling, Simulation and Optimization for Science and Technology.* New York, NY, USA: Springer-Verlag, 2014, pp. 59–82.
[56] C. Chen, B. He, Y. Ye, and X. Yuan, "The direct extension of ADMM for multi-block convex minimization problems is not necessarily convergent," *Math. Program. Ser. A*, vol. 155, no. 1, pp. 57–79, Jan. 2016.
[57] P. L. Combettes and J.-C. Pesquet, "Proximal splitting methods in signal processing," in: *Fixed-Point Algorithms for Inverse Problems in Science and Engineering.* New York, NY, USA: Springer-Verlag, 2011.
[58] D. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 613–627, 1995.
[59] W. Zuo, D. Meng, L. Zhang, X. Feng, and D. Zhang, "A generalized iterated shrinkage algorithm for non-convex sparse coding," in *Proc. IEEE Int. Conf. Compu. Vis.*, Sydney, Australia, Dec. 2013, pp. 217–224.
[60] U. M. Ascher and C. Greif. *A First Course in Numerical Methods.* Philadelphia, PA, USA: SIAM, 2011.
[61] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," 2016. [Online]. Available: https://arxiv.org/abs/1511.06324.
[62] G. Claeskens, "Statistical model choice," *Annu. Rev. Stat. Appl.*, vol. 3, pp. 233–256, Jun. 2016.
[63] C.-G. Li and R. Vidal, "A structured sparse plus structured low-rank framework for subspace clustering and completion," *IEEE Trans. Signal Process.*, vol. 64, no. 24, pp. 6557–6570, Dec. 2016.
[64] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Proc. Int. Conf. Acoust. Speech Signal Process.*, Las Vegas, NV, USA, Mar./Apr., 2008, pp. 3869–3872.
[65] J Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.
[66] 2012. [Online]. Available: http://www.cs.rochester.edu/jliu/publications.html
[67] 2017. [Online]. Available: https://grouplens.org/datasets/movielens/

Wen-Jun Zeng (S'10–M'11) received the M.S. degree in electrical engineering from Tsinghua University, Beijing, China, in 2008.

He is a Senior Research Associate with the Department of Electronic Engineering, City University of Hong Kong, Hong Kong. His research interests broadly lie in signal processing, mathematical optimization, machine learning and their applications to data science, including estimation theory, approximation theory, inverse problem, robust statistics, sparse recovery, matrix completion, and phase retrieval. He holds three U.S. patents.

Hing Cheung So (S'90–M'95–SM'07–F'15) received the B.Eng. degree from the City University of Hong Kong, Hong Kong and the Ph.D. degree from The Chinese University of Hong Kong, Hong Kong, both in electronic engineering, in 1990 and 1995, respectively.

He is a Professor with the Department of Electronic Engineering, City University of Hong Kong. His research interests include robust signal processing, source localization, spectral analysis, and sparse approximation. He has been on the editorial boards for the IEEE SIGNAL PROCESSING MAGAZINE, IEEE TRANSACTIONS ON SIGNAL PROCESSING, SIGNAL PROCESSING, AND DIGITAL SIGNAL PROCESSING. He was also the 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 of the IEEE Signal Processing Society where he was chair in the awards subcommittee.