A Fast and Accurate Matrix Completion Method based on QR Decomposition and $L_{2,1}$-Norm Minimization

Qing Liu, Franck Davoine, Jian Yang, *Member, IEEE,* Ying Cui, Zhong Jin, and Fei Han

***Abstract*—Low-rank matrix completion aims to recover matrices with missing entries and has attracted considerable attention from machine learning researchers. Most of the existing methods, such as weighted nuclear-norm-minimization-based methods and QR-decomposition-based methods, cannot provide both convergence accuracy and convergence speed. To investigate a fast and accurate completion method, an iterative QR-decomposition-based method is proposed for computing an approximate Singular Value Decomposition (CSVD-QR). This method can compute the largest $r(r > 0)$ singular values of a matrix by iterative QR decomposition. Then, under the framework of matrix tri-factorization, a CSVD-QR-based $L_{2,1}$-norm minimization method (LNM-QR) is proposed for fast matrix completion. Theoretical analysis shows that this QR-decomposition-based method can obtain the same optimal solution as a nuclear norm minimization method, i.e., the $L_{2,1}$-norm of a submatrix can converge to its nuclear norm. Consequently, an LNM-QR-based iteratively reweighted $L_{2,1}$-norm minimization method (IRLNM-QR) is proposed to improve the accuracy of LNM-QR. Theoretical analysis shows that IRLNM-QR is as accurate as an iteratively reweighted nuclear norm minimization method, which is much more accurate than the traditional QR-decomposition-based matrix completion methods. Experimental results obtained on both synthetic and real-world visual datasets show that our methods are much faster and more accurate than the state-of-the-art methods.**

***Index Terms*—Matrix Completion, QR Decomposition, Approximate SVD, Iteratively Reweighted $L_{2,1}$-Norm.**

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. U1713208, 61672287, 61602244, 91420201, 61472187, and 61572241 and in part by the Natural Science Foundation of Zhejiang Province (LQ18F030014) and National Basic Research Program of China under Grant No. 2014CB349303 and Innovation Foundation from Key Laboratory of Intelligent Perception and Systems for High-Dimensional Information of Ministry of Education (JYB201706). This work was also carried out in the framework of the Labex MS2T, program Investments for the future, French ANR (Ref. ANR-11-IDEX-0004-02). (*Corresponding author: Zhong Jin.*)

Q. Liu is with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China and is with the School of Software, Nanyang Institute of Technology, Nanyang, 473004, China (e-mail: clyqig2008@126.com).

F. Davoine is with Sorbonne Universités, Université de technologie de Compiègne, CNRS, Heudiasyc, UMR 7253, Compiègne, France (e-mail: franck.davoine@hds.utc.fr).

J. Yang and Z. Jin are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, 210094, China (e-mail: csjyang@njust.edu.cn; zhongjin@njust.edu.cn).

Y. Cui is with the College of Computer Science and Technology, Zhejiang University of Technology, Hang'zhou, 310023, China (e-mail: cuiying@zjut.edu.cn).

F. Han is with the School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, Jiangsu, 212013, China (e-mail: hanfei@ujs.edu.cn).

I. Introduction

THE problem of recovering an incomplete matrix with missing values has recently attracted considerable attention from researchers in the image processing [1-10], signal processing [11-13], and machine learning [14-20] fields. Conventional methods formulate this task as a low-rank matrix minimization problem. Suppose that $M(M \in \mathbb{R}^{m\times n},\ m \geq n > 0)$ is an incomplete matrix; then, the traditional low-rank minimization problem is formulated as follows:

$$\min_{X} \operatorname{rank}(X), \quad \text{s.t.} \quad X_{i,j} = M_{i,j}, \quad (i,j) \in \Omega, \tag{1}$$

where $X \in \mathbb{R}^{m\times n}$ is the considered low-rank matrix, $\operatorname{rank}(X)$ is the rank of X, and Ω is the set of locations corresponding to the observed entries. The problem in Eq. (1) is NP-hard and is difficult to optimize. Fortunately, the missing values in a matrix can be accurately recovered by a nuclear norm minimization under broad conditions [21, 22]. The most widely used methods based on the nuclear norm are singular value thresholding (SVT) [23] and accelerated proximal gradient [24]. These methods are not fast because of the high computational cost of singular value decomposition (SVD) iterations. Moreover, these methods are not very accurate when recovering matrices with complex structures. One of the reasons is that the nuclear norm may not be a good approximation of the rank function [28] in these cases.

To improve the accuracies of nuclear-norm-based methods, some improved methods based on the Schatten p-norm [25, 35, 36], weighted nuclear norm [27], γ-norm [33], and arctangent rank [34], have been proposed. In 2015, F. Nie et al. [25] proposed a joint Schatten p-norm and L_p-norm robust matrix completion method. This method can obtain a better convergence accuracy than that of SVT. However, it may become slow when addressing large-scale matrices because of using SVD in each iteration. C. Lu et al. [26] proposed an iteratively reweighted nuclear norm minimization (IRNN) method [26] in 2016. By using nonconvex functions to update the weights for the singular values, IRNN is much more accurate than SVT. However, it still relies on SVD to obtain the singular values for recovering incomplete matrices, which may cause it to be slow when applied to real-world datasets. Some other methods in references [33] and [34] also face the same difficulty.

To improve the speed of SVT, some methods based on matrix factorization [13, 17, 18, 31] have recently been proposed. In 2013, A fast tri-factorization (FTF) method [32] based on Qatar Riyal (QR) decomposition [29, 30] was proposed. FTF

relies on the cheaper QR decomposition as a substitute for SVD to extract the orthogonal bases of rows and columns of an incomplete matrix and applies SVD to a submatrix whose size can be set in advance. FTF is very fast when applied to low-rank data matrices. However, it will become slow if the test matrices are not of low rank. Moreover, the FTF method is not as accurate as a weighted nuclear-norm-based method, such as IRNN [26]. A more recent work, i.e., the robust bilinear factorization (RBF) method [18], is slightly more accurate than FTF. However, it is still not fast and not accurate enough for real applications. Some other methods based on matrix factorization proposed in references [13], [17], and [31] also have similar characteristics. Thus, the traditional methods based on the weighted nuclear norm and matrix factorization cannot provide both convergence speed and convergence accuracy.

Recently, the $L_{2,1}$-norm was successfully applied to feature selection [37, 38], optimal mean robust principle component analysis [53], and low-rank representation [39-42]. The feature selection methods and the method in [53] use a combination of the nuclear norm and $L_{2,1}$-norm as their loss function to extract the subspace structures of test datasets. Because they use the $L_{2,1}$-norm, they are more robust with respect to outliers. However, they are still not fast because they use SVD to search for the optimal solutions. In low-rank representation, the outliers among data points can be removed by solving an $L_{2,1}$-norm minimization problem, the optimal solution of which can be obtained without using SVD. However, an $L_{2,1}$-norm-based matrix completion method does not exist.

In general, developing a fast and accurate matrix completion method remains a significant open challenge.

In fact, the singular values and singular vectors can also be obtained by QR decomposition [51], which is much faster than SVD. Additionally, the $L_{2,1}$-norm can be applied to fast matrix completion methods under the framework of matrix tri-factorization. Thus, this paper aims to propose a fast and accurate matrix completion method based on $L_{2,1}$-norm minimization and QR decomposition to address the aforementioned open challenge. Our main contributions are as follows:

- A QR-decomposition-based method for computing an approximate SVD (CSVD-QR) is proposed. It can compute the largest $r(r > 0)$ singular values of a given matrix.
- A CSVD-QR-based $L_{2,1}$-norm minimization method (LNM-QR) is proposed for fast matrix completion. By using QR as a substitute for SVD, LNM-QR is much faster than the methods that utilize SVD.
- A CSVD-QR-based iteratively reweighted $L_{2,1}$-norm minimization method (IRLNM-QR) is proposed to improve the accuracy of LNM-QR. A theoretical analysis shows that IRLNM-QR has the same optimal solution as that of the IRNN method, which is much more accurate than the traditional QR-decomposition-based methods.
- The $L_{2,1}$-norm of a matrix is proven to be the upper bound of its nuclear norm (in Section III.C). Thus, the proposed methods can also be applied to improve the performances of the nuclear-norm-based low-rank representation [42, 44], multi-view data analysis [45-47], and matrix/tensor completion [54, 55] methods.

II. Related Work

In this section, the SVD of a matrix and some widely used matrix completion methods based on SVD and QR decomposition are respectively introduced.

A. Singular Value Decomposition

Suppose that $X \in \mathbb{R}^{m\times n}$ is an arbitrary real matrix; then, the SVD of X is as follows:

$$X = U\Lambda V^T, \tag{2}$$

$$U = (u_1, \cdots, u_m) \in \mathbb{R}^{m\times n}, \tag{3}$$

$$V = (v_1, \cdots, v_n) \in \mathbb{R}^{n\times n}, \tag{4}$$

where U and V are column orthogonal matrices, the columns of which are the left and right singular vectors of X, respectively. $\Lambda \in \mathbb{R}^{m\times n}$ is a diagonal matrix with diagonal entries, where $\Lambda_{ii} = \sigma_i(X)$, that are assumed to be in order of decreasing magnitude. $\sigma_i(X)$ is the i^{th} singular value of X.

Many papers on how to compute the singular values of X exist [48-50]. Here, we introduce a simple SVD method (SVD-SIM), which was proposed by Paul Godfrey [51] in 2006. In this method, the singular values and singular vectors can be computed by iterative QR decomposition. The QR decomposition [30] of X is as follows:

$$X = LR, \tag{5}$$

where $L \in \mathbb{R}^{m\times m}$ is an orthogonal matrix, the columns of which are the basis of the space spanned by the columns of X. $R \in \mathbb{R}^{m\times n}$ is a weakly upper-triangular matrix.

Let $\Lambda_1 = X^T$, $U_1 = eye(m, m)$, and $V_1 = eye(n, n)$. In the j^{th} iteration of SVD-SIM, the variables, i.e., Λ_j, U_j, and V_j, are alternately updated in two steps.

In step 1, U_{j+1} is updated by QR decomposition. Suppose that the QR decomposition of Λ_j^T is

$$\Lambda_j^T = Q_1 S_1, \tag{6}$$

where $Q_1 \in \mathbb{R}^{m\times m}$ and $S_1 \in \mathbb{R}^{m\times n}$ are intermediate variables. U_{j+1} is updated as follows:

$$U_{j+1} = U_j Q_1. \tag{7}$$

In step 2, the variable S_1^T in Eq. (6) is decomposed by QR decomposition as follows:

$$S_1^T = Q_2 S_2, \tag{8}$$

where $Q_2 \in \mathbb{R}^{m\times m}$ and $S_2 \in \mathbb{R}^{m\times n}$ are intermediate variables. Λ_{j+1} and V_{j+1} are updated as follows:

$$\Lambda_{j+1} = S_2, \tag{9}$$

$$V_{j+1} = V_j Q_2. \tag{10}$$

U_j, Λ_j, and V_j, produced by Eqs. (6-10) can converge to U, Λ, and V, respectively, where Λ is a diagonal matrix with entries satisfying

$$||\Lambda_{ii}||_1 = \sigma_i(X), \tag{11}$$

where $||\Lambda_{ii}\ ||_1$ is the l_1-norm of Λ_{ii}. The columns of U and V are the left and right singular vectors of X, respectively.

The SVD-SIM method offers a simple way to directly obtain the singular values and singular vectors by QR decomposition. However, SVD-SIM computes all the singular values simultaneously, which may limit its suitability for fast matrix completion (see a deeper analysis in Section III).

B. Methods based on SVD and QR Decomposition for Matrix Completion

1) Singular Value Thresholding (SVT) Method: A classical nuclear-norm-based method is the SVT method proposed by Cai et al. [23]. The minimization problem of SVT is

$$\min_X ||X||_* + \alpha||X||_F^2, \quad \text{s.t. } P_\Omega(X) = P_\Omega(M), \tag{12}$$

where α>0 and the nuclear norm of X is defined as follows:

$$||X||_* = \Sigma_{i=1}^n \sigma_i(X), \tag{13}$$

where $\sigma_i(X)$ is the i^{th} singular value of X and $P_\Omega(X)$ is

$$(P_\Omega(X))_{i,j} = \begin{cases} X_{i,j}, & (i,j) \in \Omega, \\ 0, & (i,j) \notin \Omega. \end{cases} \tag{14}$$

The Lagrange function of the problem in Eq. (12) is

$$\text{Lag} = \mu \left\|X\right\|_* + \frac{1}{2}\left\|X\right\|_F^2 + \text{tr}(Y^T P_\Omega(X - M)), \tag{15}$$

where $Y \in \mathbb{R}^{m\times n}$ and $\mu > 0$. The variable X can be updated by solving the following problem:

$$X = \arg\min_X \mu \left\|X\right\|_* + \frac{1}{2}\left\|X - P_\Omega(Y)\right\|_F^2. \tag{16}$$

The problem in Eq. (16) can be solved by the singular value shrinking operator [23] shown in Lemma 1.

Lemma 1 [23] For each $\tau \geq 0$, $Y \in \mathbb{R}^{m\times n}$ is a given real matrix, where $Y = U\Lambda V^T$ is the SVD decomposition of Y. The global solution to

$$S_\tau(Y) = \arg\min_X \tau \left\|X\right\|_* + \frac{1}{2}\left\|X - Y\right\|_F^2 \tag{17}$$

is given by the singular value shrinking operator

$$S_\tau(Y) = U\text{diag}(P(\Lambda_{ii}), i = 1,, n)V^T, \tag{18}$$

where $P_i(\Lambda_{ii}) = \max\{\Lambda_{ii} - \tau, 0\}$ and $i = 1, \cdots, n$.

The SVT method converges efficiently on synthetic data matrices with strict low-rank structures. However, it is not always accurate and not always fast when recovering matrices with complex structures [28].

2) Iteratively Reweighted Nuclear Norm (IRNN) Method: An IRNN method [26] has been proposed for improving the convergence accuracy of SVT. This method solves the minimization problem in its k^{th} iteration as follows:

$$\min_X \Sigma_{i=1}^n \nabla g(\sigma_i(X)) \cdot \sigma_i(X) + \frac{\alpha}{2}\left\|F(X)\right\|_F^2, \tag{19}$$

where $F(X) = X - \frac{1}{\alpha}(X_k - \nabla f(X_k))$, $f(X) = P_\Omega(X - M)$, and $\alpha > 0$. $g(x)$ is a continuous, concave, and monotonically increasing function on $[0, \infty)$. $\nabla g(x_0)$ is the supergradient of $g(x)$ at x_0, and $\nabla g(x)$ obeys

$$\nabla g(\sigma_i(X)) \leq \nabla g(\sigma_j(X)), \tag{20}$$

where $\sigma_i(X) \geq \sigma_j(X)$ and $1 \leq i < j \leq n$.

Lemma 2 [26] For each $\tau \geq 0$, $Y \in \mathbb{R}^{m\times n}$ is a given real matrix, where $Y = U\Lambda V^T$ is the SVD decomposition of Y. The global solution to

$$S_\tau(Y) = \min_X \tau\Sigma_{i=1}^n w_i \cdot \sigma_i(X) + \frac{1}{2}\left\|X - Y\right\|_F^2 \tag{21}$$

can be given by

$$S_\tau(Y) = U\text{diag}(P_i(\Lambda_{ii}), i = 1, \cdots, n)V^T, \tag{22}$$

where $P_i(\Lambda_{ii}) = \max\{\Lambda_{ii} - \tau w_i, 0\}, 0 < w_i < w_j (i < j)$, and $i, j = 1, \cdots, n$.

The reason why the weights $w_i (i = 1, \cdots, n)$ should be in increasing order is that $P_i(\Lambda_{ii})$ must obey [26]

$$(P_i(\Lambda_{ii}) - P_j(\Lambda_{jj}))(\Lambda_{ii} - \Lambda_{jj}) \geq 0. \tag{23}$$

The IRNN method is much faster and more accurate than SVT. However, it is still not fast enough for real applications because of the high computational cost of SVD iterations.

3) Fast Tri-Factorization (FTF) Method: An FTF method based on QR decomposition was recently proposed by Liu et al. [32] for fast matrix completion. Suppose that $X \in \mathbb{R}^{m\times n}$ is a real matrix whose rank is r, which can be decomposed as

$$X = LDR, \tag{24}$$

where $L \in \mathbb{R}^{m\times r}$, $D \in \mathbb{R}^{r\times r}$, $R \in \mathbb{R}^{r\times n}$, and $r \in (0, n]$. If L is a column orthogonal matrix and R is a row orthogonal matrix, then the following conclusion will be obtained:

$$\left\|X\right\|_* = \left\|D\right\|_*. \tag{25}$$

Consequently, the nuclear norm minimization problem on X in SVT can be modified as follows:

$$\min_D \left\|D\right\|_*, \quad \text{s.t.} \begin{cases} L^T L = I, RR^T = I, \\ P_\Omega(LDR) = P_\Omega(M), \end{cases} \tag{26}$$

where r is a preset parameter that regulates the computational cost of FTF.

The variables L, R, and D can be alternately optimized by fixing both other variables. L and R can be updated by applying QR decomposition, the computational cost of which is much lower than that of SVD, to two matrices with sizes of $m \times r$ and $r \times n$, respectively [32]. Additionally, D is updated by applying the singular value shrinking operator to a matrix of size $r \times r$. Therefore, FTF is much faster than the traditional methods, such as SVT and IRNN, when it applies SVD to small-scale matrices. However, it may become slow when recovering matrices with complex structures, the ranks of which are full or near full. The reason is that the parameter r should be given a large value, which makes the computational cost large in that case. Another disadvantage of FTF is that it is still a nuclear-norm-minimization-based method, which is less accurate than a weighted nuclear-norm-based method, such as IRNN.

In general, we may conclude that the weighted nuclear-norm-based methods and QR-decomposition-based methods cannot achieve satisfactory levels of both convergence speed and convergence accuracy. Thus, a fast and accurate method for matrix completion should be investigated.

C. An $L_{2,1}$-Norm Minimization Solver for Low-Rank Representation

Recently, the $L_{2,1}$-norm was successfully used in low-rank representation [39] to optimize the noise data matrix $E \in \mathbb{R}^{m\times n}$. The optimal E can be updated by solving the minimization problem as follows:

$$\min_{E} \tau \|E\|_{2,1} + \frac{1}{2}\|E - C\|_F^2, \tag{27}$$

where $C \in \mathbb{R}^{m\times n}$ is a given real matrix and $\tau > 0$. The $L_{2,1}$-norm of E is defined as

$$\|E\|_{2,1} = \Sigma_{j=1}^{n}\sqrt{\Sigma_{i=1}^{m}E_{ij}^2}. \tag{28}$$

The optimal $E(:,j)$ (denoting the j^{th} column of E) of the problem in Eq. (27) obeys

$$E(:,j) = \frac{(\|C(:,j)\|_2 - \tau)_+}{\|C(:,j)\|_2}C(:,j), \tag{29}$$

where

$$\|C(:,j)\|_2 = \sqrt{\Sigma_{i=1}^{m}C_{ij}^2}, \tag{30}$$

$$(x)_+ = \max\{x, 0\}, \tag{31}$$

with $x \in (-\infty, +\infty)$ being a real number. The $L_{2,1}$-norm minimization solver in Eq. (29) is referred to as LNMS in this paper for convenience. The computational cost of the LNMS is much lower than that of the singular value shrinking operator. In this paper, a fast and accurate matrix completion method using the LNMS under the matrix tri-factorization framework is introduced. A deeper analysis is presented in the next section.

III. Our Proposed Methods

A. Motivation

This paper aims to investigate a fast and accurate matrix completion method based on $L_{2,1}$-norm minimization.

1) Application of $L_{2,1}$-Norm Minimization to Matrix Completion: The nuclear norm minimization problem in Lemma 1 is a special case of an $L_{2,1}$-norm minimization problem. Suppose that X is a variable whose SVD is $X = U\Lambda V^T$. The problem in Eq. (17) is equivalent to the following problem:

$$\min_{U,\Lambda,V} \tau\Sigma_{j=1}^{n}\Lambda_{jj} + \frac{1}{2}\left\|U\Lambda V^T - Y\right\|_F^2, \tag{32}$$

where $\tau > 0$ and Y is a given real matrix. Because Λ is a diagonal matrix, we have the following collusion:

$$\Sigma_{j=1}^{n}\Lambda_{jj} = \|\Lambda\|_{2,1}. \tag{33}$$

Thus, the problem in Eq. (32) can be reformulated as

$$\min_{U,\Lambda,V} \tau\|\Lambda\|_{2,1} + \frac{1}{2}\left\|U\Lambda V^T - Y\right\|_F^2. \tag{34}$$

Because the variables U and V are column orthogonal matrices, the optimal solution to the problem in Eq. (32) is equivalent to that of the following problem:

$$\min_{U,\Lambda,V} \tau\|\Lambda\|_{2,1} + \frac{1}{2}\left\|\Lambda - U^TYV\right\|_F^2. \tag{35}$$

The optimal Λ to the problem in Eq. (35) can be given by the LNMS as follows:

$$\Lambda(:,j) = \frac{(\|C(:,j)\|_2 - \tau)_+}{\|C(:,j)\|_2}C(:,j), \tag{36}$$

$$C = U^TYV, \tag{37}$$

where $j = 1, \cdots, n$. The LNMS in Eq. (36) can recover the columns of Λ one by one, which shows that the matrix C in Eq. (36) does not need to be diagonal. Therefore, it is suitable to decompose X into three matrices as follows:

$$X = LDR, \tag{38}$$

where $L \in \mathbb{R}^{m\times r}$, $D \in \mathbb{R}^{r\times r}$, $R \in \mathbb{R}^{r\times n}$, and $r \in (0, n]$. The variables L and R denote a column orthogonal matrix and row orthogonal matrix, respectively. Specifically, they are the orthogonal bases of the columns and rows of X, respectively. The matrix D does not need to be diagonal, which is different from the matrix Λ in SVD. Then, we formulate the following $L_{2,1}$-norm minimization problem:

$$\min_{L,D,R} \tau\|D\|_{2,1} + \frac{1}{2}\|LDR - Y\|_F^2, \quad \text{s.t. } X = LDR. \tag{39}$$

According to Eqs. (36-37), the variable D can be optimized very efficiently after obtaining the variables L and R. The problem in Eq. (32) is a special case of the problem in Eq. (39). Therefore, the $L_{2,1}$-norm minimization problem in Eq. (39) can also be applied to matrix completion. One key issue is how to extract the orthogonal bases L and R. Because D does not need to be diagonal, L and R can be obtained via a method more efficient than SVD.

2) Using QR Decomposition to Extract Orthogonal Bases L and R: The orthogonal bases L and R can also be computed by QR decomposition [31, 32], the computational cost of which is approximately ten percent that of SVD. According to the SVD-SIM [51] method, the left and right singular vectors can be directly obtained by QR decomposition. One may think that the variables L and R in Eq. (39) can be obtained by SVD-SIM with only a few iterations. However, SVD-SIM computes all the singular values and the corresponding singular vectors simultaneously, which allows us to forego this idea. Obviously, computing all the singular values may reduce the recovery speed of a matrix completion method [43], which motivates us to propose a method for computing the largest $r(r \in (0, n])$ singular values and the corresponding singular vectors. Some methods have already been proposed, such as the power method [30, 52]. However, the computational cost of the power method increases sharply with increasing r as a result of applying SVD to a submatrix.

In this paper, a method for computing the SVD of a matrix by iterative QR decomposition (CSVD-QR) is proposed. This method can compute the largest $r(r \in (0, n])$ singular values and the corresponding singular vectors of a matrix, which is different from SVD-SIM. Consequently, the orthogonal bases L and R in Eq. (39) can be computed by CSVD-QR with only a few iterations. Then, using the results obtained by CSVD-QR, two fast matrix completion methods based on the $L_{2,1}$-norm are proposed:

- A CSVD-QR-based $L_{2,1}$-norm minimization method (LNM-QR) is proposed for matrix completion. By using QR decomposition as a substitute for SVD, LNM-QR is much faster than the compared methods using SVD.
- A CSVD-QR-based iteratively reweighted $L_{2,1}$-norm minimization method (IRLNM-QR) is proposed to improve the accuracy of LNM-QR. IRLNM-QR has advantages in terms of both convergence speed and convergence accuracy over the traditional methods.

We can now introduce the proposed method for computing an approximate SVD based on QR decomposition and the two matrix completion methods based on the $L_{2,1}$-norm.

B. Method for Computing an Approximate SVD based on QR Decomposition (CSVD-QR)

Suppose that $X \in \mathbb{R}^{m\times n}$ is a given real matrix. In this section, we propose a method that can compute the largest $r(r \in (0, n])$ singular values and the corresponding singular vectors of X by QR decompositions directly. Specifically, we aim to find three matrices, i.e., L, D, and R, such that

$$\|X - LDR\|_F^2 \leq \varepsilon_0, \tag{40}$$

where ε_0 is a positive tolerance. Please see Eq. (38) for the definitions of L, D, and R. Consequently, a minimization problem is formulated:

$$\min_{L,D,R} \|X - LDR\|_F^2, \quad \text{s.t.} \quad L^T L = I, RR^T = I. \tag{41}$$

The minimization function in Eq. (41) is convex to each one of the variables, i.e., L, D, and R, when the remaining two are fixed. Thus, the variables can be alternately updated one by one. Suppose that L_j, D_j, and R_j denote the results of the j^{th} iteration in the alternating method. Let $L_1 = eye(m, r)$, $D_1 = eye(r, r)$, and $R_1 = eye(r, n)$. In the j^{th} iteration, L_{j+1} is updated with fixed D_j and R_j as follows:

$$L_{j+1} = \arg\min_{L} \|X - LD_jR_j\|_F^2. \tag{42}$$

Since R_j is a row orthogonal matrix, the optimal solution to Eq. (42) is as follows:

$$L_{j+1} = XR_j^T D_j^+, \tag{43}$$

where D_j^+ is the Moore–Penrose pseudo-inverse of D_j. Because the optimal L should be a column orthogonal matrix, L_{j+1} can be set to the orthogonal basis of the range space spanned by the columns of $XR_j^T D_j^+$ as follows:

$$L_{j+1} = \text{orth}(XR_j^T D_j^+), \tag{44}$$

where $\text{orth}(X)$ is an operator that extracts the orthogonal basis of the columns of X. In view of Eq. (40), L_{j+1} is the orthogonal basis of the columns of X, which can be set to the orthogonal basis of XA, where $A \in \mathbb{R}^{n\times r}$ is a random matrix [30]. Therefore, the solution in Eq. (44) can also be given as

$$L_{j+1} = \text{orth}(XR_j^T). \tag{45}$$

In this paper, we use QR decomposition to compute the orthogonal basis of XR_j^T in Eq. (45) as follows:

$$[Q, T] = \text{qr}(XR_j^T), \tag{46}$$

$$L_{j+1} = Q(q_1, \cdots, q_r), \tag{47}$$

where $Q \in \mathbb{R}^{m\times m}$ and $T \in \mathbb{R}^{m\times r}$ are intermediate variables. Eq. (46) indicates that the QR decomposition of XR_j^T is $XR_j^T = QT$. Similarly, R_{j+1} can be updated as follows:

$$[Q, T] = \text{qr}(X^T L_{j+1}), \tag{48}$$

$$R_{j+1} = Q(q_1, \cdots, q_r), \tag{49}$$

where $Q \in \mathbb{R}^{n\times n}$ and $T \in \mathbb{R}^{n\times r}$ are intermediate variables. Since the optimal R is a row orthogonal matrix, we set

$$R_{j+1} = R_{j+1}^T. \tag{50}$$

Finally, D_{j+1} is updated as follows:

$$D_{j+1} = \arg\min_{D} \|X - L_{j+1}DR_{j+1}\|_F^2 \tag{51}$$

$$= L_{j+1}^T X R_{j+1}^T. \tag{52}$$

According to Eq. (48), we have

$$T^T = L_{j+1}^T XQ. \tag{53}$$

Because R_{j+1} is generated by Eqs. (49-50), we have

$$D_{j+1} = T^T(1 \cdots r, 1 \cdots r). \tag{54}$$

The sequences of $\{L_j\}$, $\{R_j\}$, and $\{D_j\}(j = 1, \cdots, n, \cdots)$ generated by Eqs. (46-47), Eqs. (48-50), and Eq. (54) can converge to matrices L, R, and D, respectively, with matrix D satisfying

$$\|D_{ii}\|_1 = \|\Lambda_{ii}\|_1, \tag{55}$$

where $i = 1, \cdots, r$ and Λ_{ii} is the i^{th} singular value of X. The columns of L and R^T are left and right singular vectors corresponding to the largest r singular values, respectively. This method of computing an approximate SVD based on QR decomposition is called CSVD-QR, the main steps of which are shown in Table I.

TABLE I: MAIN STEPS OF CSVD-QR.

Input:	X, a real matrix;
Output:	L, D, R $(X = LDR)$;
Initialization:	$r > 0$, $q > 0$, $j = 1$; $Itmax > 0$;
	ε_0 is a positive tolerance;
	$C = eye(n, r)$; $L_1 = eye(m, r)$;
	$D_1 = eye(r, r)$; $R_1 = eye(r, n)$.
Repeat:	
	$L_{j+1} : Eqs.(46 - 47)$;
	$R_{j+1} : Eqs.(48 - 50)$;
	$D_{j+1} : Eqs.(48, 54)$; $\quad j = j + 1$;
Until:	$\|L_jD_jR_j - X\|_F^2 \leq \varepsilon_0$ or $j > Itmax$.
Return:	$L = L_j$, $D = D_j$, $R = R_j$.

Theorem 1 When $r = n$, the diagonal entries of D_j at the $j^{th}(j > 1)$ iteration of CSVD-QR are equal to those of Λ_j in SVD-SIM. For the proof, please see Appendix A.

Theorem 1 shows that CSVD-QR can converge as fast as SVD-SIM. In the proposed methods for matrix completion, we use only the output of CSVD-QR with one iteration (see more details in Section III.C).

C. Proposed Methods for Fast and Accurate Matrix Completion

According to Section III.A.1, the $L_{2,1}$-norm minimization problem in Eq. (39) can be applied to matrix completion. The variable D in Eq. (39) does not need to be diagonal. Consequently, the variables L and R can be given by CSVD-QR with only a few iterations. Moreover, the orthogonal subspace of the recovery result may not considerably change after two conservative iterations [43]. Thus, we use the orthogonal bases of the rows and columns of the recovered matrix in the previous iteration in the proposed matrix completion methods to initialize CSVD-QR. By using this smart warm initialization, L and R computed from CSVD-QR with one iteration can be used to recover the original incomplete matrix.

1) An $L_{2,1}$-Norm Minimization based on QR Decomposition for Matrix Completion (LNM-QR): According to the analysis in Section III.A, the considered matrix X can be decomposed as in Eq. (38). Moreover, the original incomplete matrix can be recovered efficiently by solving the $L_{2,1}$-norm minimization problem in Eq. (39). The relationship between $\|D\|_*$ and $\|D\|_{2,1}$ can confirm this conclusion. The matrix D can be decomposed as follows:

$$D = \Sigma_{j=1}^{r} D^j, \tag{56}$$

$$D_{k,i}^j = \begin{cases} D_{k,j}, & (i=j), \\ 0, & (i \neq j), \end{cases} \tag{57}$$

where $i, j, k = 1, \cdots, r$, and $D^j \in \mathbb{R}^{r \times r}$. From Eq. (56), we have

$$\|D\|_* = \left\|\Sigma_{j=1}^{r} D^j\right\|_*. \tag{58}$$

Because the nuclear norm is a convex function, we have

$$\|D\|_* \leq \Sigma_{j=1}^{r} \left\|D^j\right\|_*. \tag{59}$$

Because the $\Sigma_{j=1}^{r} \left\|D^j\right\|_*$ term is equal to $\|D\|_{2,1}$, i.e.,

$$\Sigma_{j=1}^{r} \left\|D^j\right\|_* = \|D\|_{2,1}, \tag{60}$$

we obtain the following conclusion:

$$\|D\|_* \leq \|D\|_{2,1}. \tag{61}$$

From Eq. (61), the $L_{2,1}$-norm of a matrix is clearly the upper bound of its nuclear norm. This conclusion motivates us to apply the $L_{2,1}$-norm minimization problem in Eq. (39) to matrix completion as follows:

$$\min_D \|D\|_{2,1}, \text{ s.t.} \begin{cases} L^T L = I, X = LDR, \\ RR^T = I, P_\Omega(LDR) = P_\Omega(M). \end{cases} \tag{62}$$

Please see Eq. (38) for the definitions of L, D, and R. From the analysis in Section III.A, the variable D does not need to be diagonal. Because the optimization function in Eq. (62) is convex, the corresponding problem can be solved by the alternating direction method of multipliers (ADMM). The augmented Lagrange function of the problem in Eq. (62) is

$$\begin{aligned} \text{Lag} = &\|D\|_{2,1} \\ &+ \text{tr}(Y^T(X - LDR)) + \frac{\mu}{2}\|X - LDR\|_F^2, \end{aligned} \tag{63}$$

where $\mu > 0$ and $Y \in \mathbb{R}^{m \times n}$. Suppose that X_k denotes the result of the k^{th} iteration in the ADMM. The Lagrange function is optimized in two steps. In step 1, L_{k+1} and R_{k+1} are updated by solving the following minimization problem:

$$\min_{L,R} \left\|(X_k + \frac{Y_k}{\mu_k}) - LD_kR\right\|_F^2. \tag{64}$$

According to the analyses corresponding to Eq. (41), L_{k+1} and R_{k+1} can be given by CSVD-QR. If CSVD-QR is initialized by L_i and R_i, it will converge within a few iterations because the matrices L and R will not considerably change in two consecutive iterations [43]. In our method, $X_k + \frac{Y_k}{\mu_k}$ is decomposed in one iteration of CSVD-QR as follows:

$$X_k + \frac{Y_k}{\mu_k} = L_{k+1} D_T R_{k+1}, \tag{65}$$

where $D_T \in \mathbb{R}^{r \times r}$. By using this smart warm initialization, LNM-QR can converge very quickly.

In step 2, X_{k+1} is updated by solving an $L_{2,1}$-norm minimization. First, the variable D can be optimized, with X_k, Y_k, L_{k+1}, and R_{k+1} held fixed, by solving the following problem:

$$\begin{aligned} D_{k+1} = &\arg\min_D \frac{1}{\mu_k}\|D\|_{2,1} \\ &+ \frac{1}{2}\left\|D - L_{k+1}^T(X_k + \frac{Y_k}{\mu_k})R_{k+1}\right\|. \end{aligned} \tag{66}$$

From Eqs. (65-66), we have the following conclusion:

$$L_{k+1}^T(X_k + \frac{Y_k}{\mu_k})R_{k+1}^T = D_T, \tag{67}$$

where D_T is as shown in Eq. (65). Therefore, Eq. (66) can be reformulated as follows:

$$D_{k+1} = \arg\min_D \frac{1}{\mu_k}\|D\|_{2,1} + \frac{1}{2}\|D - D_T\|_F^2. \tag{68}$$

According to Eqs. (29-31), the minimization problem in Eq. (68) can be solved by the LNMS as follows:

$$D_{k+1} = D_T K, \tag{69}$$

where K is a diagonal matrix, i.e.,

$$K = \text{diag}(k_1, \cdots, k_r), \tag{70}$$

where the j^{th} entry k_j can be given as follows:

$$k_j = \frac{(\|D_T(:,j)\|_F - \frac{1}{\mu_k})_+}{\|D_T(:,j)\|_F}. \tag{71}$$

Second, by fixing the variables L_{k+1}, D_{k+1}, R_{k+1}, and Y_k, X_{k+1} is updated as follows:

$$\begin{aligned} X_{k+1} = &L_{k+1}D_{k+1}R_{k+1} + P_\Omega(M) \\ &- P_\Omega(L_{k+1}D_{k+1}R_{k+1}). \end{aligned} \tag{72}$$

Finally, by fixing the variables, i.e., L_{k+1}, D_{k+1}, R_{k+1}, and X_{k+1}, Y_{k+1} and μ_k are updated as follows:

$$Y_{k+1} = Y_k + \mu_k(X_{k+1} - L_{k+1}D_{k+1}R_{k+1}), \tag{73}$$

$$\mu_{k+1} = \rho\mu_k, \tag{74}$$

where $\rho \geq 1$. The proposed $L_{2,1}$-norm minimization method based on CSVD-QR is called LNM-QR, the main steps of which are summarized in Table II.

TABLE II: MAIN STEPS OF LNM-QR.

Input:	M, a real matrix with missing values; Ω, the set of locations corresponding to the observed entries.
Output:	X_{opt}, the recovery result.
Initialization:	$r > 0$, $q > 0$, $k = 0$; $Itmax > 0$; $C = eye(n, r)$; $L_1 = eye(m, r)$; $D_1 = eye(r, r)$; $R_1 = eye(r, n)$; $X_0 = M$, ε_0 is a positive tolerance.
Repeat:	
	Step 1: L_{k+1}, R_{k+1}: Eq. (65);
	Step 2: D_{k+1}: Eq. (69);
	X_{k+1}: Eq. (72); $k = k + 1$.
Until:	$\|X_k - X_{k-1}\|_F^2 \leq \varepsilon_0$ or $k > Itmax$.
Return:	$L = L_j$, $D = D_j$, $R = R_j$.

Because the convex optimization function in Eq. (62) is minimized by the ADMM, which is a gradient-search-based method, LNM-QR can converge to its optimal solution. Suppose that N iterations are required for LNM-QR to converge. If the updating steps in LNM-QR are continued, then $X_k (k > N)$ will be equal to X_N. Because CSVD-QR is initialized by matrices L and R in the previous iteration, LNM-QR can fall back to CSVD-QR. Thus, the sequence of $\{D_k\}$ produced by LNM-QR (see Eqs. (69-71)) can converge to a diagonal matrix D with entries D_{jj} that obey

$$\|D_{jj}\|_1 = \sigma_j(X_N). \tag{75}$$

Therefore, the $L_{2,1}$-norm minimization function of the LNM-QR model (in Eq. (62)) can converge to the nuclear norm of D, which motivates us to improve LNM-QR as an iteratively reweighted $L_{2,1}$-norm minimization method.

2) Extension of LNM-QR: Since the $L_{2,1}$-norm of D can converge to its nuclear norm and the IRNN [26] method performs much better than the nuclear-norm-based methods, it is suitable to use an iteratively reweighted $L_{2,1}$-norm minimization to replace the $L_{2,1}$-norm minimization in step 2 of LNM-QR. According to Eq. (60), the weighted $L_{2,1}$-norm of X is denoted as

$$\|X\|_{w\cdot(2,1)} = \Sigma_{j=1}^{n} w_j \left\|X^j\right\|_*, \tag{76}$$

where $w_j > 0$ ($j \in [1, n]$). The definition of X^j is the same as that of D^j in Eq. (56). The minimization problem of LNM-QR in Eq. (62) can be modified as follows:

$$\begin{aligned} &\min_D \Sigma_{j=1}^{r} \nabla g(\left\|D^j\right\|_*) \left\|D^j\right\|_*, \\ &\text{s.t.} \quad X = LDR, P_\Omega(LDR) = P_\Omega(M), \end{aligned} \tag{77}$$

Please see Eq. (38) for the definitions of L, D, and R. $\nabla g(x_0)$ is the supergradient of $g(x)$ at x_0. $g(x)$ is a continuous and monotonically increasing function on $[0, +\infty)$. The problem in Eq. (77) can be solved by the ADMM. The augmented Lagrange function of Eq. (77) is

$$\begin{aligned} \text{Lag} =& \Sigma_{j=1}^{r} \bigtriangledown g(\left\|D^j\right\|_*) \left\|D^j\right\|_* \\ &+ \text{tr}(Y^T(X - LDR)) + \frac{\mu}{2} \|X - LDR\|_F^2. \end{aligned} \tag{78}$$

Suppose that X_k denotes the result of the k^{th} iteration in the ADMM. The variables L_{k+1} and R_{k+1} can be given by CSVD-QR (see Eq. (65)). The variables, i.e., X_{k+1}, Y_{k+1}, and μ_k, are updated according to Eq. (72), Eq. (73), and Eq. (74), respectively. The value of D_{k+1} can be determined by solving the following problem:

$$\min_D \frac{1}{\mu_k} \Sigma_{j=1}^{r} \nabla g(\left\|D^j\right\|_*) \left\|D^j\right\|_* + \frac{1}{2} \|D - D_T\|_F^2, \tag{79}$$

where D_T is as shown in Eq. (65). In each iteration of IRNN, $g(x)$ is a concave function[26]. In this paper, we design a novel function for $g(x)$. In each iteration of this extension model, the $\nabla g(\left\|D^j\right\|_*)$ term obeys

$$\nabla g(\left\|D^j\right\|_*) = \mu_j (1 - \bar{k}_j) \left\|D_T^j\right\|_*, \tag{80}$$

where $1 \geq \bar{k}_1 \geq \bar{k}_2 \geq \cdots \bar{k}_r > 0$, $\mu > 0$, and $j \in [1, \cdots, r]$.

Theorem 2 For any given real matrix $C \in \mathbb{R}^{r \times r}$and $\mu > 0$, the optimal solution to the following problem

$$\min_{X \in \mathbb{R}^{r \times r}} \frac{1}{\mu} \|X\|_{w\cdot(2,1)} + \frac{1}{2} \|X - C\|_F^2, \tag{81}$$

can be given as follows:

$$X_{opt} = CK, \tag{82}$$

where

$$K = \text{diag}(k_1, \cdots, k_r), \tag{83}$$

$$k_j = \frac{(\Lambda_{jj} - \frac{w_j}{\mu})_+}{\Lambda_{jj}}, \tag{84}$$

with $j \in [1, \cdots, r]$ and Λ_{jj} being the singular value of C^j. The definition of C^j is the same as that of D^j (see Eq. (56)). For the proof, please see Appendix B.

According to Theorem 2 and Eq. (80), the variable D_{k+1} can be updated as follows:

$$D_{k+1} = D_T \bar{K}, \tag{85}$$

$$\bar{K} = \text{diag}(\bar{k}_1, \cdots, \bar{k}_r). \tag{86}$$

The proposed iteratively reweighted $L_{2,1}$-norm minimization method based on CSVD-QR for matrix completion is called IRLNM-QR.

Theorem 3 If the weights $\nabla g(||D^j||_*)(j = 1, \cdots, r)$ in Eq. (77) are given by Eq. (80), and if $(\bar{k}_1, \cdots, \bar{k}_r)$ in Eq. (80) are arranged in decreasing order, then IRLNM-QR will converge to the optimal solution of an iteratively reweighted nuclear norm minimization method. For the proof, please see Appendix C.

According to Theorem 3, the weights in Eqs. (85-86) should be arranged in decreasing order. In the experiment, $\bar{k}_j (j = 1, \cdots, r)$ are given as follows:

$$w_j = \begin{cases} 1, & 1 \leq j \leq S, 1 < S < r, \\ \frac{\theta - 1}{r - S} + w_{j-1}, & S < j \leq r, \end{cases} \tag{87}$$

$$\bar{k}_j = \frac{1}{w_j}, \tag{88}$$

where $\theta > 1$, r is the row number of $\bar{K}$, and $S (S < r)$ is a positive integer. The effects of S and θ were studied by Liu et al. [4].

D. Complexity Analysis

In this section, the computational complexities of SVT, IRNN, FTF, LNM-QR, and IRLNM-QR are analyzed. Suppose that $X \in \mathbb{R}^{m\times n}$ is a real matrix. The computational cost of SVD on X is $O(mn^2)$. The main CPU times of SVT and IRNN are consumed by performing SVD on X. Thus, their computational complexities are $O(mn^2)$. The main CPU time of FTF is consumed by performing QR decomposition twice to update L and R and SVD once on submatrix D. Thus, the computational cost of FTF is $O(r^2(m+n)+r^3)$, where $r \ll \min(m,n)$. The main CPU times of LNM-QR and IRLNM-QR are consumed by performing QR decomposition twice to update L and R (please see Eqs. (46-50)). Thus, the computational complexities of LNM-QR and IRLNM-QR are $O(r^2(m+n))$. Clearly, the computational complexities of LNM-QR and IRLNM-QR are much smaller than those of FTF, SVT, and IRNN. Hence, LNM-QR and IRLNM-QR are much faster than the traditional methods based on SVD.

IV. Experimental Results

To demonstrate the effectiveness of the proposed methods, several comparative experiments are performed. First, the convergence of CSVD-QR is tested. Then, the proposed LNM-QR and IRLNM-QR methods for matrix completion are evaluated using synthetic and real-world datasets.

The experiments are performed on a MATLAB 2012a platform equipped with an i5-6300U CPU and 4 GB of RAM.

A. Convergence of CSVD-QR

In this section, CSVD-QR is tested on a synthetic matrix X that is generated as follows:

$$X = M_L^{m\times r_1} M_R^{r_1\times n}, \tag{89}$$

$$M_L^{m\times r_1} = \text{randn}(m, r_1), \tag{90}$$

$$M_R^{r_1\times n} = \text{randn}(r_1, n), \tag{91}$$

where $r_1 \in [1, n]$ is the rank of X. Suppose that $H^k = (h_1, h_2, \cdots, h_r)$ is a vector, the i^{th} entry of which h_i is equal to $D_{k(i,i)}$, where D_k (in Eq. (52)) is the submatrix in the k^{th} iteration of CSVD-QR. The CSVD-QR method is stopped when the relative error $\frac{\Sigma_{i=1}^{t}\left|\|_i\|_1-\sigma_i(X)\right|}{\Sigma_{i=1}^{t}\sigma_i(X)} < \varepsilon_0$, $(t = \min(r, r_1))$. The experiments for CSVD-QR are conducted as follows.

First, let $m = n = r = 300$, $r_1 = 250$, and $\varepsilon_0 = 0.001$. All the singular values of matrix X are computed via the CSVD-QR and SVD-SIM methods. Because r is larger than r_1, CSVD-QR can compute all the singular values of X. The CPU times of CSVD-QR and SVD-SIM are 0.830 s and 0.900 s, respectively. Their CPU times are almost equal to each other because the diagonal entries of D_k in the k^{th} iteration of CSVD-QR $(r = n)$ are equal to those of Λ_k in SVD-SIM (see Theorem 1). More details are shown by the relative error curve of the CSVD-QR method in Fig. 1.

Fig. 1 shows that the relative errors of CSVD-QR and SVD-SIM are equal to each other in every iteration, which is consistent with the conclusion in Theorem 1. The relative error of CSVD-QR is still shown to sharply decrease during the first 10 iterations and converge gradually after approximately

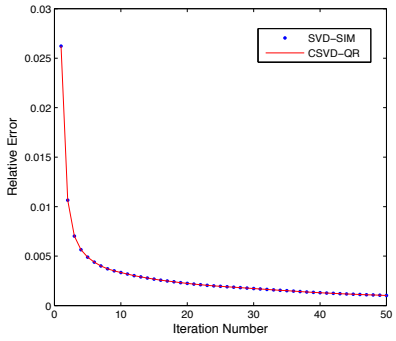
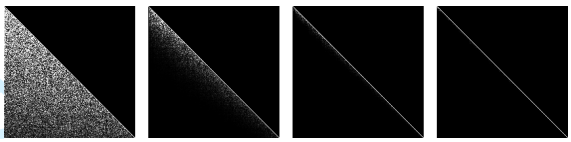


Fig. 1 The Relative Error curve of CSVD-QR.

30 iterations. Thus, the proposed CSVD-QR can accurately compute all the singular values of X.

Second, let $m = n = 300$, $r_1 = 250$, $r = 50$, and $\varepsilon_0 = 0.0005$, with the largest r singular values computed by CSVD-QR. The variable D_k can converge to a diagonal matrix when CSVD-QR converges. Suppose that $T_k \in \mathbb{R}^{r\times r}$ is a square matrix whose entries $T_{k(i,j)} = \left|D_{k(i,j)}\right| (i, j = 1, \cdots, r)$. The convergence of D_k can be shown by plotting $10T_k(k = 2, 10, 30, 60)$, as in Fig. 2.



(A) $k = 2$, (B) $k = 10$, (C) $k = 20$, (D) $k = 60$.

Fig. 2 The convergence procedure of D_k in CSVD-QR.

Fig. 2 shows that the sequence of $\{D_k\}$ can converge to a diagonal matrix. The relative error of the singular values computed by CSVD-QR reaches 0.0005 in the 60^{th} iteration, which means that CSVD-QR can accurately compute the largest r singular values. In the proposed LNM-QR and IRLNM-QR methods, the matrices L and R computed by CSVD-QR with one iteration, are used to recover the original incomplete matrix. Therefore, the proposed methods are much faster than the traditional methods based on SVD.

B. Experimental Results of the Proposed Methods for Matrix Completion

In this section, LNM-QR and IRLNM-QR are tested using synthetic datasets and real-world images. The convergence accuracies and speeds are compared with those of the SVT [23], IRNN-SCAD (IRNN with SCAD function) [26], FTF [32], and RBF (a nuclear-norm-based method) [18] methods. The maximum numbers of iterations for LNM-QR, IRLNM-QR, FTF, RBF, SVT, and IRNN-SCAD are 50, 50, 200, 200, 200, and 200, respectively. The parameters of FTF, RBF, SVT, and IRNN-SCAD are set to the optimal values. The total reconstruction error (ERR) and peak signal-to-noise ratio (PSNR), which are two measures commonly used for

evaluation purposes, are defined as follows:

$$\text{ERR} = \|X_{REC} - X\|_F, \tag{92}$$

$$\text{PSNR} = 10 \times \log_{10} \frac{255^2}{\text{MSE}}, \tag{93}$$

$$\text{MSE} = \frac{1}{3T \cdot \text{SE}}, \tag{94}$$

$$\text{SE} = \text{ERR}_r^2 + \text{ERR}_g^2 + \text{ERR}_b^2, \tag{95}$$

where T is the total number of missing entries, X is the original matrix, and X_{REC} is the recovered matrix.

1) Synthetic Data: LNM-QR and IRLNM-QR are tested on a synthetic low-rank data matrix M, generated as follows:

$$M = M_L^{m\times r_1} M_R^{r_1 \times n} + P_\Omega(\sigma \cdot \text{randn}(m, n)), \tag{96}$$

where M_L and M_R are generated as in Eqs. (90-91), respectively. $r_1 > 0$ is the rank of M, and σ regulates the noise level for M. Clearly, a larger $\sigma(\sigma > 0)$ can make M more difficult to recover. In this section, $m = 1000$, $n = 1000$, $r_1 = 50$, and 50% of the entries of M are randomly missing. Let $\mu_0 = 10^{-2}$ and $\rho = 1.4$ (see Eq. (74)) for LNM-QR and IRLNM-QR, and let $\mu_0 = 10^{-4}$ and $\rho = 1.4$ for FTF and MBF. The parameter S in Eqs. (87-88) was tested from 2 to 40 to determine the best value for IRLNM-QR. The parameter θ in Eqs. (87-88) was set to 20 for IRLNM-QR.

First, the effects of r (the rank of D) on LNM-QR, IRLNM-QR and FTF are tested. Let $\sigma = 0.5$ and r increase from 10 to 300, with a step size of 10. The effects of r on the three methods are shown in Fig. 3.

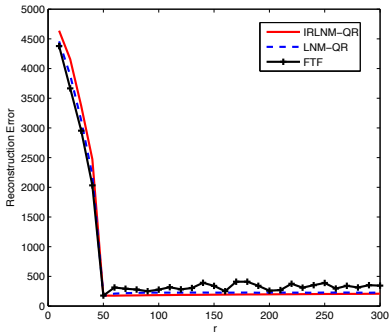


Fig. 3 The effects of parameter r (rank of D) on the LNM-QR, IRLNM-QR, and FTF methods using synthetic data matrices.

Fig. 3 shows that the reconstruction error of FTF is much larger than those of LNM-QR and IRLNM-QR when $r > 50$. Thus, LNM-QR and IRLNM-QR are much more robust with respect to r than FTF. This result also shows that the three methods obtain the best reconstruction error at $r = 50$, i.e., the rank of M. Because estimating the rank of an incomplete matrix is difficult, we must set r to a larger value in FTF. Consequently, the accuracy and speed of FTF will be reduced.

Second, the effects of the noise level (σ) on LNM-QR and IRLNM-QR are compared with those on the FTF, RBF, SVT, and IRNN methods. Let $r = 1.5r_1$ for FTF, LNM-QR, and IRLNM-QR. Then, we repeat the experiments 10 times, with σ increasing from 0.1 to 0.9 for each method. The mean reconstruction errors with standard errors of the six methods are shown in Table III (the standard errors are shown in parentheses) and the corresponding CPU times are shown in Fig. 4.

TABLE III: MEAN RECONSTRUCTION ERRORS AND STANDARD ERRORS OF THE SIX METHODS USING SYNTHETIC DATA, 50% OF WHICH IS RANDOMLY MISSING.

Noise level	LNM -QR	IRLNM -QR	IRNN -SCAD	SVT	FTF	RBF
0.1	41.463	**35.493**	**36.266**	55.325	59.825	59.751
	(0.153)	(0.065)	(0.084)	(0.027)	(16.770)	(5.931)
0.2	81.741	**71.254**	**72.423**	108.333	115.333	103.768
	(0.274)	(0.261)	(0.119)	(0.145)	(16.082)	(2.871)
0.3	123.654	**107.247**	**106.708**	176.110	162.110	150.452
	(0.357)	(0.263)	(0.175)	(0.140)	(41.325)	(1.668)
0.4	169.648	**142.676**	**140.824**	200.791	209.791	199.547
	(0.702)	(0.405)	(0.341)	(0.143)	(26.347)	(1.770)
0.5	218.630	**177.684**	**175.013**	302.106	310.106	247.318
	(0.765)	(0.495)	(0.504)	(0.479)	(27.893)	(1.891)
0.6	261.513	**213.589**	**211.327**	323.247	333.247	295.879
	(1.191)	(0.452)	(0.612)	(0.453)	(25.894)	(1.918)
0.7	308.720	**249.447**	**253.629**	343.896	353.896	343.460
	(1.213)	(0.571)	(0.637)	(0.726)	(28.369)	(1.752)
0.8	354.001	**284.824**	**281.419**	384.264	387.264	390.843
	(1.348)	(0.689)	(0.792)	(0.657)	(26.232)	(1.984)
0.9	398.841	**320.044**	**315.756**	412.214	412.214	412.175
	(1.419)	(0.976)	(1.174)	(1.123)	(26.367)	(1.998)

Table III shows that the IRLNM-QR method, which is as accurate as IRNN-SCAD, is more accurate than LNM-QR, SVT, RBF, and FTF. The reason is that IRLNM-QR can obtain the same optimal solution as an iteratively reweighted nuclear norm minimization method. Table III still shows that IRLNM-QR and LNM-QR are as stable as SVT and IRNN-SCAD, which are considerably more stable than FTF. The standard errors of IRLNM-QR and LNM-QR are much smaller than those of FTF and RBF. Unlike other tested methods, the standard error of RBF is relatively large when the noise level is equal to 0.1 and becomes small when the noise level varies from 0.2 to 0.9. This may result from the fact that optimizing the l_1-norm of a sparse matrix [18] may make RBF be more stable when recovering matrices with high noise levels.

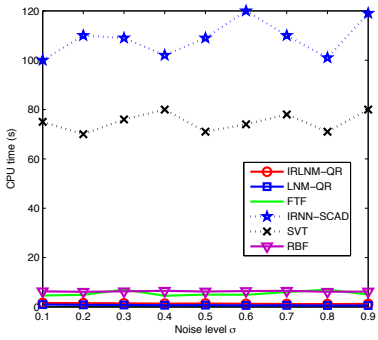


Fig. 4 The CPU times of the six methods on random data.

Fig. 4 shows that LNM-QR and IRLNM-QR are much faster than SVT and IRNN-SCAD. The speed of IRLNM-QR is approximately 80∼100 times those of SVT and IRNN-SCAD. The reason is that the computational cost of a full SVD in each iteration of SVT or IRNN-SCAD is much larger than that of the QR decomposition in LNM-QR and IRLNM-QR when $r = 75(1.5r_1)$. The speed of LNM-QR or IRLNM-QR is only 2∼3 times that of FTF or RBF. When $r = 75$,

the computational cost of the SVD in FTF is not very large. However, FTF will become slow if r is given a large value.

2) Real-World Data: Recovering useful information covered by the texts in an image is much more challenging than recovering matrices with missing entries randomly distributed. We first check the positions of the texts and then initialize the corresponding entries to be zero to generate incomplete images. The original images and incomplete images in Fig. 5 (on the next page) are 1024×1024 in size. Because the color images have three channels, we treat each channel separately and then combine the results to form the final recovered images. The incomplete images in Fig. 5 are recovered by LNM-QR and IRLNM-QR. Then, their results, i.e., convergence accuracies, CPU times, and numbers of iterations, are compared with those of RBF, FTF, SVT, and IRNN-SCAD.

First, the effects of the parameter r on FTF, LNM-QR, and IRLNM-QR are tested. Let $\mu_0 = 10^{-5}$ and $\rho = 1$ for FTF and RBF, and let $\mu_0 = 10^{-3}$ and $\rho = 1$ for LNM-QR and IRLNM-QR. Let r increase from 10 to 300, with a step size of 10. By using these values, the 5^{th} incomplete image in Fig. 5 is recovered by the three methods. The PSNR values and CPU times of the three methods for different r are shown in Fig. 6 and Fig. 7, respectively.

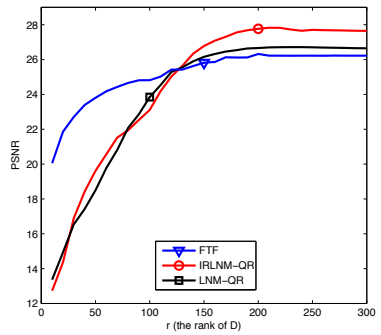


Fig. 6 The effects of r on the three methods, i.e., FTF, LNM-QR, and IRLNM-QR.

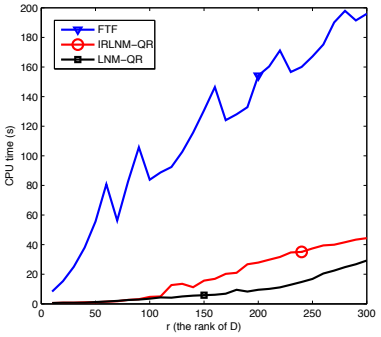


Fig. 7 The CPU times of the three methods with different r.

Fig. 6 shows that the convergence accuracies of the three methods increase with the parameter r. The PSNR curves of LNM-QR and IRLNM-QR converge when $r > 180$ and $r > 200$, respectively. Similarly, the PSNR curve of FTF increases sharply when $r < 120$ and increases gradually when $120 < r < 200$. However, FTF slows down when r becomes large because the computational cost of the SVD in FTF is dominated by the parameter r. The CPU times of the three methods for different values of r (Fig. 7) confirm this conclusion.

Fig. 7 shows that the CPU time of FTF increases very quickly compared to those of LNM-QR and IRLNM-QR. Approximately 0.5 s to 42 s and 9 s to 190s are required for LNM-QR and IRLNM-QR and for FTF, respectively, to recover the 5^{th} incomplete image in Fig. 5 when r ranges from 10 to 300. Thus, LNM-QR and IRLNM-QR are approximately 5$\sim$18 times faster than FTF. Furthermore, the accuracy of IRLNM-QR is much better than those of FTF and LNM-QR.

Second, the convergence accuracies of LNM-QR and IRLNM-QR are compared with those of RBF, FTF, SVT, and IRNN-SCAD. Let $\mu_0 = 10^{-5}$ and $\rho = 1.1$ for RBF, $\mu_0 = 10^{-5}$ and $\rho = 1$ for FTF, and $\mu_0 = 10^{-3}$ and $\rho = 1$ for LNM-QR and IRLNM-QR. From the analyses of Fig. 6, it is suitable to let $r = 200$ for RBF, FTF, LNM-QR, and IRLNM-QR when recovering the eight incomplete images in the second row of Fig. 5. The parameter S (see Eqs. (87-88)) is tested from 2 to 20 to choose the best value for IRLNM-QR, and θ (in Eqs. (87-88)) is set to 3. With these values, the incomplete images are recovered by the six methods. The convergence accuracies, recovered images, PSNR curves and CPU times of the six methods are shown in Table IV, Fig. 8, Fig. 9, and Fig. 10, respectively.

TABLE IV: PSNR OF RECOVERY RESULTS OF THE SIX METHODS ON THE INCOMPLETE IMAGES IN FIG. 5.

Images (1-8)	SVT	FTF	RBF	IRLNM -QR	LNM -QR	IRNN -SCAD
1	39.244	39.471	40.191	**41.202**	40.534	41.186
2	33.741	33.994	37.171	38.143	37.187	**38.695**
3	31.576	32.664	33.966	35.625	33.789	**35.739**
4	30.092	29.765	31.517	**35.173**	32.143	34.379
5	25.510	26.247	26.346	**27.769**	26.606	27.732
6	20.225	20.299	21.273	**22.217**	21.329	22.184
7	22.478	22.787	24.373	**25.314**	24.264	25.200
8	25.529	25.635	27.172	27.844	27.129	**28.045**

As shown in Table IV, the accuracy of LNM-QR is much better than that of SVT and is slightly better than those of RBF and FTF. However, the PSNR of IRLNM-QR on the eight images, which is much better than those of RBF and FTF, is approximately equal to that of IRNN-SCAD. The reason is that by using matrices L and R in the previous iteration as an initialization, IRLNM-QR can converge to the optimal solution of an iteratively reweighted nuclear norm method. Some of the recovery results are plotted in Fig. 8 due to space limitations.

As shown in Fig. 8, the recovery result of IRLNM-QR is as clear as that of IRNN-SCAD but much clearer than that of SVT. The recovery results of RBF, FTF, and LNM-QR are very similar to each other, with some abnormal points being present (please see the points between the bird and the tree in Fig. 8 (D), (E), and (F)). Note that by using the outputs of CSVD-QR with one iteration, LNM-QR and IRLNM-QR can converge very efficiently. The PSNR curves of the six methods on the 5^{th} incomplete image in Fig. 5 are shown in Fig. 9.

Fig. 9 shows that IRLNM-QR is much more accurate than FTF, SVT and LNM-QR. Moreover, IRLNM-QR and

(A) The original images used in Section IV.B.2).

(B) The incomplete images generated.

Fig. 5 The real-world images (1-8) used in Section IV.B.2), which are 1024× 1024 in size.

(A) (B) (C) (D)

(E) (F) (G) (H)

Fig. 8 The recovery results of the six methods on image 3 in Fig. 5 with text noise. (A) The original image. (B) The incomplete image. (C) IRLNM-QR, PSNR=35.625. (D) LNM-QR, PSNR=33.789. (E) RBF, PSNR=33.966. (F) FTF, PSNR=32.664. (G) SVT, PSNR=31.576. (H) IRNN-SCAD, PSNR=35.739.

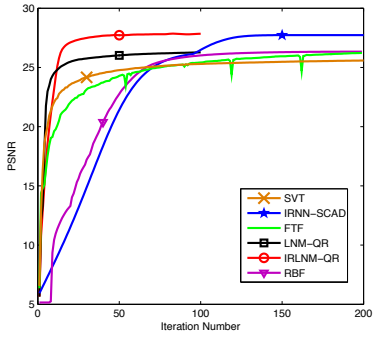


Fig. 9 The PSNR curve of the six methods on the 5^{th} incomplete image with texts in Fig. 5.

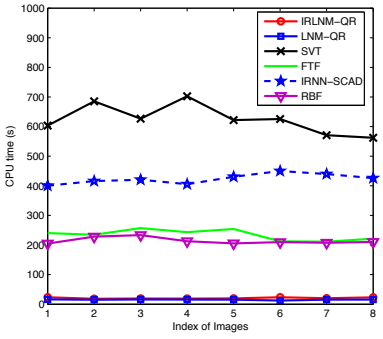


Fig. 10 The CPU times of the six methods.

LNM-QR can converge with fewer iterations than SVT, RBF, FTF, and IRNN-SCAD. IRLNM-QR converges to the optimal solution after approximately 50 iterations, whereas IRNN-SCAD, RBF, FTF, and SVT require at least 130, 120, 180, and 150 iterations, respectively.

Fig. 10 shows that LNM-QR is a bit faster than IRLNM-QR. The reasons is that IRLNM-QR may require a few more iterations to search for a better solution. IRNN-SCAD and SVT require approximately 400 s∼450 s and 580 s∼680 s, respectively. The FTF method, which is almost as fast as RBF, is not fast when recovering real-world images, as the parameter r in FTF should be set at a large value to improve its

convergence accuracy. In general, LNM-QR and IRLNM-QR are approximately 15 times, 15 times, 35 times, and 20 times faster than the FTF, RBF, SVT, and IRNN-SCAD methods.

V. Conclusions

To investigate a fast and accurate completion method, a QR-decomposition-based method for computing an approximate SVD (CSVD-QR) is proposed. This method can be used to compute the largest $r(r > 0)$ singular values of a matrix by QR decomposition iteratively. Then, under the framework of matrix tri-factorization, a CSVD-QR-based $L_{2,1}$-norm minimization method (LNM-QR) is proposed for fast matrix completion. Theoretical analysis shows that the $L_{2,1}$-norm of a submatrix in LNM-QR can converge to its nuclear norm. Consequently, an LNM-QR-based iteratively reweighted $L_{2,1}$-norm minimization method (IRLNM-QR) for improving the accuracy of LNM-QR is proposed. Theoretical analysis shows that IRLNM-QR is as accurate as an iteratively reweighted nuclear norm minimization method, which is much more accurate than the traditional QR-decomposition-based matrix completion methods. The experimental results obtained using both synthetic and real-world visual datasets show that LNM-QR and IRLNM-QR are much faster than the FTF, RBF, SVT, and IRNN-SCAD methods. The experimental results still show that IRLNM-QR is almost as accurate as the IRNN method.

VI. Appendix

In this appendix, some mathematical details regarding CSVD-QR, LNM-QR and IRLNM-QR are provided. In addition, Theorems 1, 2 and 3 are proven.

A. Proof of Theorem 1

Suppose that $X \in \mathbb{R}^{m\times n}$ $(m \geq n)$ is a real matrix, the QR decomposition of which [30] is as follows:

$$X = LR. \tag{97}$$

Let $T = QX$, where $Q \in \mathbb{R}^{m\times n}$ is an orthogonal matrix. Then, the QR decomposition of T is as follows:

$$T = \mathcal{L}\mathcal{R}, \tag{98}$$

where $\mathcal{L} \in \mathbb{R}^{m\times m}$ and $\mathcal{R} \in \mathbb{R}^{m\times n}$ satisfy

$$\mathcal{L} = QL, \tag{99}$$
$$\mathcal{R} = R, \tag{100}$$

where $\mathcal{L}$ is an orthogonal matrix.

Let $r = n$, $L_1 = eye(m, r)$, and $R_1 = eye(r, n)$. In the j^{th} iteration of CSVD-QR, L_{j+1} is updated as follows:

$$[Q, T_j] = \text{qr}(XR_j^T), \tag{101}$$
$$L_{j+1} = Q(q_1, \cdots, q_r), \tag{102}$$

R_{j+1} is updated as follows:

$$[Q, \mathcal{T}_j] = \text{qr}(X^T L_{j+1}), \tag{103}$$
$$R_{j+1} = Q(q_1, \cdots, q_r), \tag{104}$$
$$R_{j+1} = R_{j+1}^T, \tag{105}$$

and D_{j+1} is updated as follows:

$$D_{j+1} = \mathcal{T}_j^T(1\cdots r, 1\cdots r). \tag{106}$$

If $m > n$, we let $Q \in \mathbb{R}^{m\times n}$ and $T_j \in \mathbb{R}^{n\times n}$ for Eq. (101).

When $j = 1$, we have

$$[L_2, T_1] = \text{qr}(X), \tag{107}$$
$$[R_2, D_2^T] = \text{qr}(T_1^T L_2^T L_2). \tag{108}$$

Since $L_2^T L_2 = I$, D_2^T is equal to Λ_2 in SVD-SIM.

When $j = 2$, we have

$$[L_3, T_2] = \text{qr}(XR_2^T), \tag{109}$$
$$XR_2^T = L_2 D_2 R_2 R_2^T \tag{110}$$
$$= L_2 D_2. \tag{111}$$

Consequently,

$$[L_3, T_2] = \text{qr}(L_2 D_2). \tag{112}$$

According to Eqs. (99-100),

$$[L_2^T L_3, T_2] = \text{qr}(D_2). \tag{113}$$

Because D_2^T is equal to Λ_2, T_2 in Eq. (113) is equal to the S_1 term in Eq. (3). D_3 is updated as follows:

$$[R_3, D_3^T] = \text{qr}(X^T L_3), \tag{114}$$
$$X^T L_3 = R_2^T D_2^T L_2^T L_3. \tag{115}$$

According to Eq. (113), Eq. (115) is equal to

$$X^T L_3 = R_2^T T_2^T. \tag{116}$$

According to Eq. (114) and (116), we have

$$[R_2 R_3, D_3^T] = \text{qr}(T_2^T). \tag{117}$$

According to Eq. (6) in Section II.A, D_3^T is equal to Λ_3. Thus, we can conclude that the diagonal entries of D_j are equal to those of Λ_j, where $j = 1, \cdots, N$ $(N > 1)$.

B. Proof of Theorem 2

We rewrite the problem in Eq. (81) as follows:

$$\min_{X\in\mathbb{R}^{r\times r}} \frac{1}{\mu}\|X\|_{w\cdot(2,1)} + \frac{1}{2}\|X - C\|_F^2, \tag{118}$$

where $\mu > 0$. According to Eq. (76), the optimal solution X_{opt} to the problem in Eq. (118) is as follows:

$$X_{opt}^j = \arg\min_{X_{opt}^j} \frac{w_j}{\mu}\left\|X^j\right\|_* + \frac{1}{2}\left\|X^j - C^j\right\|_F^2, \tag{119}$$

where $j = 1, \cdots, r$ and $X_{opt} = \Sigma_{j=1}^r X_{opt}^j$. Suppose $C^j = U\Lambda V^T$ is the SVD of C^j. C^j has only one singular value that can be denoted as $\sigma(C^j)$. According to Lemma 1, the optimal solution to Eq. (119) is

$$X_{opt}^j = (\left\|C^j\right\|_F - \frac{w_j}{\mu})_+ UV^T, \tag{120}$$

where $\left\|C^j\right\|_F = \sigma(C^j)$. Then, we have

$$X_{opt}^j = \frac{(\left\|C^j\right\|_F - \frac{w_j}{\mu})_+}{\left\|C^j\right\|_F} C_j. \tag{121}$$

Finally, we form the final optimal X_{opt} as follows:

$$X_{opt} = CK, \tag{122}$$

$$K = \text{diag}(k_1, \cdots, k_r), \tag{123}$$

$$k_j = \frac{(\left\|C^j\right\|_F - \frac{w_j}{\mu})_+}{\left\|C^j\right\|_F}. \tag{124}$$

C. Proof of Theorem 3

Because the sequence of $\{D_k\}$ produced by IRLNM-QR can converge to a diagonal matrix D that obeys Eq. (75) in Section III, the D_T term in Eq. (85) in Section III can also converge to a diagonal matrix $T \in \mathbb{R}^{r\times r}$, the entries of which obey $\|T_{ii}\|_1 \geq \|T_{jj}\|_1 \, (i < j)$. Thus, the problem in Eq. (79) can be reformulated as follows:

$$\min_D \frac{1}{\mu_k}\Sigma_{j=1}^r \nabla g(\left\|D^j\right\|_*) \left\|D^j\right\|_* + \frac{1}{2}\left\|D - T\right\|_F^2. \tag{125}$$

Since $\left\|D^j\right\|_* = \sigma_j(D)$, Eq. (125) can be rewritten as follows:

$$\min_D \frac{1}{\mu_k}\Sigma_{j=1}^r \nabla g(\sigma_j(D))\sigma_j(D) + \frac{1}{2}\left\|D - T\right\|_F^2. \tag{126}$$

According to Lemma 2, the problem in Eq. (126) can be solved as follows:

$$D = U\text{diag}(P_j(\Lambda_{jj}), j = 1, \cdots, r)V^T, \tag{127}$$

$$P_j(\Lambda_{jj}) = (\|T_{jj}\|_1 - \frac{1}{\mu_k}\nabla g(\sigma_j(D)))_+, \tag{128}$$

$$\nabla g(\sigma_j(D)) = \mu_k(1 - \bar{k}_j)\left\|D_T^j\right\|_*, \tag{129}$$

where $T = U\Lambda V^T$, $\Lambda_{jj} = \|T_{jj}\|_1$, $j = 1, \cdots, r$. $1 \geq k_1 \geq k_2 \geq, \cdots, k_r > 0$, and $j \in [1, \cdots, r]$. According to Eq. (128) and Eq. (129) (Eq. (80) in Section III), we have

$$P_j(\Lambda_{jj}) = \|T_{jj}\|_1 - \frac{1}{\mu_k}(1 - \bar{k}_j)\|T_{jj}\|_1 \tag{130}$$

$$= \bar{k}_j \|T_{jj}\|_1. \tag{131}$$

When $\bar{k}_i \geq \bar{k}_j$, $P_i(\Lambda_{ii})$ obeys Eq. (23) in Lemma 2, i.e.,

$$(\bar{k}_i \|T_{ii}\|_1 - \bar{k}_j \|T_{jj}\|_1)(\|T_{ii}\|_1 - \|T_{jj}\|_1) \geq 0. \tag{132}$$

Thus, IRLNM-QR can converge to the optimal solution of an iteratively reweighted nuclear norm minimization method and can converge with an accuracy equal to that of IRNN.

REFERENCES

[1] C. Huang, X. Ding, C. Fang, and D. Wen, Robust Image Restoration via Adaptive Low-Rank Approximation and Joint Kernel Regression, IEEE Trans. on Image Processing, Vol. 23, No. 12, pp. 5284 – 5297, Dec. 2014.

[2] Y. Luo, T. Liu, D. Tao, and C. Xu, Multiview Matrix Completion for Multilabel Image Classification, IEEE Trans. on Image Processing, Vol. 24, No. 8, Aug. 2015.

[3] R. Cabral, F. Torre, J. Costeira, and A. Bernardino, Matrix Completion for Weakly-Supervised Multi-Label Image Classification, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 37, No. 1, Jan. 2015.

[4] Q. Liu, Z. Lai, Z. Zhou, F. Kuang, and Z. Jin , A Truncated Nuclear Norm Regularization Method based on Weighted Residual Error for Matrix Completion, IEEE Trans. on Image Processing, Vol. 25, No. 1, pp. 316 – 330, Jan. 2016.

[5] K. H. Jin and J. C. Ye, Annihilating Filter-Based Low-Rank Hankel Matrix Approach for Image Inpainting, IEEE Trans. on Image Processing, Vol. 24, No. 11, pp. 3498 – 3511, Nov. 2015.

[6] C. Lu, C. Zhu, C. Xu, S. Yan, and Z. Lin. Generalized Singular Value Thresholding, in Proc. of the AAAI Conference on Artificial Intelligence, 2015.

[7] Y. Kang, Robust and Scalable Matrix Completion, in Proc. of the International Conference on Big Data and Smart Computing, 2016.

[8] Y. Chen, Incoherence-Optimal Matrix Completion, IEEE Transactions on Information Theory, Vol. 61, pp. 2909 – 2923, March, 2015.

[9] H. Bi, B. Zhang, and W. Hong, Matrix-Completion-Based Airborne Tomographic SAR Inversion Under Missing Data, IEEE Geoscience and Remote Sensing Letters,Vol. 12, No. 11, pp. 2346 – 2350, Nov. 2015.

[10] J. Huang, F. Nie, and H. Huang, Robust Discrete Matrix Completion, in Proc. of the 27^{th} AAAI Conference on Artificial Intelligence, pp. 424 – 430, 2012.

[11] R. Ma, N. Barzigar, A. Roozgard, and S. Cheng, Decomposition Approach for Low-Rank Matrix Completion and Its Applications, IEEE Trans. on Signal Processing, Vol. 62, No. 7, Apr. 2014.

[12] C. Tzagkarakis, S. Becker, and A. Mouchtaris, Joint low-rank representation and matrix completion under a singular value thresholding framework, in Proc. of the 22^{nd} European Signal Processing Conference, 2014.

[13] J. P. Haldar and D. Hernando, Rank-Constrained Solutions to Linear Matrix Equations Using Power Factorization, IEEE Signal Processing Letters, Vol. 16, No. 7, pp. 584 – 587, Jul. 2009.

[14] N. Srebro and T. Jaakkola, Weighted low-rank approximations, in Proc. of International Conference on Machine Learning, 2003.

[15] Z. Kang, C. Peng, and Q. Cheng, Kernel-driven similarity learning. Neurocomputing, Vol. 267, pp. 210-219, 2017.

[16] Z. Kang, C. Peng, and Q. Cheng, Top-N Recommender System via Matrix Completion, in Proc. of the 30^{th} AAAI Conference on Artificial Intelligence, pp. 179 – 185, 2016.

[17] R. Cabral, F. De la Torre, J. P. Costeira, and A. Bernardino, Unifying nuclear norm and bilinear factorization approaches for low-rank matrix decomposition, in Proc. of the IEEE International Conference on Computer Vision, pp. 2488 – 2495, 2013.

[18] F. Shang, Y. Liu, H. Tong, J. Cheng, and H. Cheng, Robust Bilinear Factorization with Missing and Grossly Corrupted Observations, Information Sciences, Vol. 307, pp. 53 – 72, 2015.

[19] S. Zhang, W. Wang, J. Ford, and F. Makedon, Learning from incomplete ratings using non-negative matrix factorization, Siam International Conference on Data Mining, chap. 58, pp. 549 – 553, 2006.

[20] C. Dorffer, M. Puigt, G. Delmaire, and G. Roussel, Fast nonnegative matrix factorization and completion using Nesterov iterations, in Proc. of International Conference on Latent Variable Analysis and Signal Separation, Springer, pp. 26 – 35, 2017.

[21] M. Fazel, Matrix Rank Minimization with Applications, PhD thesis, Stanford Univ., 2002.

[22] E. Candès and B. Recht, Exact Matrix Completion via Convex Optimization, Foundations on Computational Math, Vol. 9, pp. 717 – 772, 2009.

[23] J. Cai, E. Candès, and Z. Shen, A Singular Value Thresholding Method for Matrix Completion, SIAM J. Optimization, Vol. 20, pp. 1956 – 1982, 2010.

[24] K. C. Toh and S. Yun, An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems, Pacific Journal of Optimization, Vol. 6, No. 3, pp. 615 – 640, 2010.

[25] F. Nie, H. Wang, and C. Ding, Joint Schatten-p norm and l_p norm robust matrix completion for missing value recovery, Knowledge and Information Systems , Vol. 42, No. 3, pp. 525 – 544, 2015.

[26] C. Lu, J. Tang, S. Yan, and Z. Lin, Non-convex Non-smooth Low Rank Minimization via Iteratively Reweighted Nuclear Norm, IEEE Trans. on Image Processing, Vol. 25, No. 2, pp. 829 – 839, 2016.

[27] S. Gu, L. Zhang, W. Zuo, and X. Feng, Weighted nuclear norm minimization with application to image denoising, IEEE Conference on Computer Vision and Pattern Recognition, pp. 2862 – 2869, 2014.

[28] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, Fast and accurate matrix completion via truncated nuclear norm regularization. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 35, No. 9, pp. 2117 – 2130, 2013.

[29] B. De Schutter and B. De Moor The QR decomposition and the singular value decomposition in the symmetrized maxplus algebra, SIAM Journal on Matrix Analysis and Applications, Vol.19, pp. 378 – 406, 1998.

[30] N. Halko. P. Martinsson, and J. Tropp, Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions, SIAM Review, Vol. 53, No. 2, pp. 217 – 288. 2011.

[31] Z. Wen, W. Yin, and Y. Zhang, Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm, Math. Prog. Comp., Vol. 4, pp. 333 – 361. Dec. 2012.

[32] Y. Liu, L. Jiao, and F. Shang, A fast tri-factorization method for low-rank matrix recovery and completion, Pattern Recognition Vol. 46, No. 1, pp. 163 – 173, 2013.
[33] Z. Kang, C. Peng, and Q. Cheng, Robust PCA via Nonconvex Rank Approximation, in Proc. of the IEEE International Conference on Data Mining, pp. 211 – 220, 2015.
[34] Z. Kang, C. Peng, and Q. Cheng, Robust Subspace Clustering via Tighter Rank Approximation, in Proc. of the 24^{th} ACM International Conference on Information and Knowledge Management, pp. 393 – 401, 2015.
[35] F. Nie, H. Wang, X. Cai, H. Huang, and C. Ding, Robust Matrix Completion via Joint Schatten p-Norm and lp-Norm Minimization, in Proc. of IEEE the 12^{th} International Conference on Data Mining, pp. 566 – 574, 2012.
[36] F. Nie, H. Wang, and C. Ding, Low rank matrix recovery via efficient Schatten-p norm minimization, in Proc. of the 26^{th} AAAI Conference on Artificial Intelligence, pp. 655 – 661, 2012.
[37] F. Nie, H. Huang, X. Cai, and C. Ding, Efficient and Robust Feature Selection via Joint $L_{2,1}$-Norms Minimization, in Proc. of Advances in Neural Information Processing Systems, pp. 1813 – 1821, 2010.
[38] C. Hou, F. Nie, X. Li, D. Yi, and Y. Wu, Joint Embedding Learning and Sparse Regression: A Framework for Unsupervised Feature Selection, IEEE Transactions on Cybernetics, Vol. 44, No. 6, pp.793 – 804, 2014.
[39] G. Liu, Z. Lin, and Y. Yu, Robust Subspace Segmentation by Low-Rank Representation, in Proc. of the 27^{th} International Conference on Machine Learning, pp. 663 – 670, 2010.
[40] K. Tang, R. Liu, Z. Su, and J. Zhang, Structure-Constrained Low-Rank Representation, IEEE Trans. on Neural Networks and Learning Systems, Vol. 25, No. 12, Dec. 2014.
[41] X. Fang, Y. Xu, X. Li, Z. Lai, and W. Wong, Robust Semi-Supervised Subspace Clustering via Non-Negative Low-Rank Representation, IEEE Trans. on Cybernetics, Vol. 46, No. 8, pp. 1828 – 1838, 2017.
[42] S. Xiao, M. Tan, D. Xu, and Z. Dong, Robust Kernel Low-Rank Representation, IEEE Trans. on Neural Networks and Learning Systems, Vol. 27, No. 11, pp: 2268 – 2281, 2016.
[43] C. J. Hsieh and P. A. Olsen, Nuclear Norm Minimization via Active Subspace Selection, in Proc. of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014.
[44] Y. Fu, J. Gao, D. Tien, Z. Lin, and X. Hong, Tensor LRR and Sparse Coding-Based Subspace Clustering, IEEE Trans. on Neural Networks and Learning Systems, Vol. 27, No. 10, pp: 2120 – 2133, 2016.
[45] Z. Ding and Y. Fu, Robust Multiview Data Analysis Through Collective Low-Rank Subspace, IEEE Trans. on Neural Networks and Learning Systems, Vol. 29, No. 5, pp. 1986 – 1997, 2018.
[46] Y. Wang, L. Wu, X. Lin, and J. Gao, Multiview Spectral Clustering via Structured Low-Rank Matrix Factorization, IEEE Trans. on Neural Networks and Learning Systems, 2018 (Early Access).
[47] W. Yang, Y. Shi, Y. Gao, L. Wang, and M. Yang, Incomplete-Data Oriented Multiview Dimension Reduction via Sparse Low-Rank Representation, IEEE Trans. on Neural Networks and Learning Systems, 2018 (Early Access).
[48] G. H. Golub and W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, Journal of the Society for Industrial and Applied Mathematics, Vol. 2, No. 2, pp. 205 – 224, 1965.
[49] J. Demmel and W. Kahan, Accurate Singular Values of Bidiagonal Matrices, SIAM J. Sci. and Stat. Comput., 11(5), 873 – 912, 1990.
[50] J. Demmel and K. Veselic, Jacobis Method is More Accurate than QR, SIAM. J. Matrix Anal. and Appl., 13(4), 1204 – 1245, 1992.
[51] Paul Godfrey, Simple SVD, version 1.0, https://cn.mathworks.com/matlabcentral/fileexchange/12674-simple-svd.
[52] M. Li, W. Bi, J. Kwok, and B. Lu, Large-Scale Nystrom Kernel Matrix Approximation Using Randomized SVD, IEEE Trans. on Neural Networks and Learning Systems, Vol. 26, No. 1, pp. 152 – 164, 2015.
[53] F. Nie, J. Yuan, and H. Huang, Optimal mean robust principal component analysis,International Conference on Machine Learning, in Proc. of the 31^{st} International Conference on Machine Learning, Beijing, China, pp. 1062 – 1070, 2014.
[54] Q. Shi, H. Lu, and Y. Cheung, Rank-One Matrix Completion With Automatic Rank Estimation via L1-Norm Regularization, IEEE Transactions on Neural Networks and Learning Systems, 2017 (Early Access).
[55] W. Hu, D. Tao, W. Zhang, Y. Xie, and Y. Yang, The Twist Tensor Nuclear Norm for Video Completion, IEEE Transactions on Neural Networks and Learning Systems, Vol. 28, No. 12, pp. 2961 – 2973, 2017.

Q. Liu received a B.S. in information and computer science from Inner Mongolia University, China, in 2009, an M.S. in pattern recognition and intelligent systems from Jiangsu University, China, in 2013, and a Ph.D. in control science and engineering from Nanjing University of Science and Technology, China, in 2017. Since 2018, he has been a faculty member in the School of Software, Nanyang Institute of Technology, China. His current research interests include image processing and machine learning.

F. Davoine received his Ph.D. in 1995 from Grenoble, France. He was appointed at Université de technologie de Compiègne, Heudiasyc Lab., France in 1997 as an associate professor and in 2002 as a researcher at CNRS. From 2007 to 2014, he was on leave at LIAMA Sino-European Lab. in Beijing, P.R. China, as PI of a project with CNRS and Peking University on Multi-sensor based perception and reasoning for intelligent vehicles. Since 2015, he is back in Compiègne, PI of a challenge-team within the Laboratory of Excellence MS2T, focusing on Collaborative vehicle perception and urban scene understanding for autonomous driving.

J. Yang received a Ph.D. in pattern recognition and intelligent systems from Nanjing University of Science and Technology (NUST), Nanjing, China, in 2002. He was a Post-Doctoral Researcher with the University of Zaragoza, Zaragoza, Spain, in 2003. He was a Post-Doctoral Fellow with the Biometrics Centre, Hong Kong Polytechnic University, Hong Kong, from 2004 to 2006 and with the Department of Computer Science, New Jersey Institute of Technology, Newark, NJ, USA, from 2006 to 2007. He is currently a Professor at the School of Computer Science and Technology, NUST. His current research interests include pattern recognition and machine learning. He is also an Associate Editor of Pattern Recognition Letters and the IEEE Trans. on Neural Networks and Learning Systems.

Y. Cui received a B.S. in computer science and technology and a Ph.D. in pattern recognition and intelligent systems from Nanjing University of Science and Technology, China, in 2008 and 2015, respectively. From 2013 to 2014, she was a visiting student at the University of Technology, Sydney, Australia. Since 2015, she has been a faculty member in the College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China. Her research interests include pattern recognition and image processing.

Z. Jin received a B.S. in mathematics, an M.S. in applied mathematics and a Ph.D. in pattern recognition and intelligent systems from Nanjing University of Science and Technology, Nanjing, China in 1982, 1984 and 1999, respectively. His current interests are in the areas of pattern recognition and face recognition.

F. Han received an M.A. from Hefei University of Technology in 2003 and a Ph.D. from the University of Science and Technology of China in 2006. He is currently a Professor of computer science at Jiangsu University. His research interests include neural networks and bioinformatics.