



Low rank matrix completion using truncated nuclear norm and sparse regularizer



Jing Dong^{a,*}, Zhichao Xue^a, Jian Guan^b, Zi-Fa Han^c, Wenwu Wang^d

^a College of Electrical Engineering and Control Science, Nanjing Tech University, Nanjing, Jiangsu, China

^b Computer Application Research Center, Harbin Institute of Technology, Shenzhen, China

^c Central Research Institute, 2012Labs, Huawei Technologies, Co., Ltd, China

^d Centre for Vision, Speech and Signal Processing, University of Surrey, Guildford GU2 7XH, UK

ARTICLE INFO

Keywords:

Matrix completion

Low rank

Truncated nuclear norm

Sparse representation

ABSTRACT

Matrix completion is a challenging problem with a range of real applications. Many existing methods are based on low-rank prior of the underlying matrix. However, this prior may not be sufficient to recover the original matrix from its incomplete observations. In this paper, we propose a novel matrix completion algorithm by employing the low-rank prior and a sparse prior simultaneously. Specifically, the matrix completion task is formulated as a rank minimization problem with a sparse regularizer. The low-rank property is modeled by the truncated nuclear norm to approximate the rank of the matrix, and the sparse regularizer is formulated as an ℓ_1 -norm term based on a given transform operator. To address the raised optimization problem, a method alternating between two steps is developed, and the problem involved in the second step is converted to several subproblems with closed-form solutions. Experimental results show the effectiveness of the proposed algorithm and its better performance as compared with the state-of-the-art matrix completion algorithms.

1. Introduction

Matrix completion arising widely in many fields has attracted a great deal of attention in recent years. Many problems in signal processing, computer vision, and machine learning can be formulated as matrix completion, for instance, image inpainting [1,2], video denoising [3], classification [4,5], recommender systems [6,7], and so on. Given a matrix with some of its entries missing, the goal of matrix completion is to recover the missing entries so that the reconstructed matrix approximates the original complete matrix. Obviously, this is inherently an ill-posed problem as there are infinite possible completions and a unique optimal solution cannot be determined. Prior information related to the complete matrix data needs to be exploited to make this problem well-defined.

In many real applications, the underlying matrix has low rank or approximately low rank property. For instance, natural image data has the low rank structure [8]. As a result, the low rank assumption of the expected complete matrix is commonly used in matrix completion [8–11]. Given a partially observed matrix $M \in \mathbb{R}^{m \times n}$, the general matrix

completion problem can be formulated as a constrained rank minimization problem, that is

$$\begin{aligned} \min_X \text{rank}(X) \\ \text{s.t. } X_{ij} = M_{ij}, (i, j) \in \Omega \end{aligned} \quad (1)$$

where $X \in \mathbb{R}^{m \times n}$, $\text{rank}(\cdot)$ denotes the rank of its operand, and $\Omega \subset \{1, \dots, m\} \times \{1, \dots, n\}$ is the set of indices corresponding to the observed entries in M .

However, the above problem is NP-hard in general due to the non-convex and discontinuous nature of the rank function. It has been proven theoretically that, under some general conditions, low rank matrices can be recovered exactly from most sets of sampled entries by minimizing the nuclear norm of the matrix [6]. Therefore, most existing methods for matrix completion use the nuclear norm, i.e., the sum of singular values of a matrix, as a convex surrogate of the rank function. Typical examples are singular value thresholding (SVT) [11], robust principal components analysis [12,13], and nuclear norm regularized least squares [14]. Unfortunately, these nuclear norm based methods may lead to suboptimal results, since the nuclear norm may not approximate the rank function well in practice. In particular, all of the nonzero

* Corresponding author.

E-mail addresses: jingdong@njtech.edu.cn (J. Dong), xuezhichao@njtech.edu.cn (Z. Xue), j.guan@cs.hitsz.edu.cn (J. Guan), zifahan@gmail.com (Z.-F. Han), w.wang@surrey.ac.uk (W. Wang).

<https://doi.org/10.1016/j.image.2018.06.007>

Received 31 October 2017; Received in revised form 15 June 2018; Accepted 18 June 2018

Available online 11 July 2018

0923-5965/© 2018 Elsevier B.V. All rights reserved.

singular values have equal contributions in the rank function while they are treated differently in the nuclear norm when added together and minimized simultaneously. Recently, the truncated nuclear norm regularization (TNNR) method [15,8] was proposed by only minimizing the sum of the $\min(m, n) - r$ minimum singular values, i.e., the truncated nuclear norm, rather than the summation of all singular values as in the nuclear norm based methods. A two-step optimization scheme was proposed to address the truncated nuclear norm minimization problem. The TNNR method outperforms the nuclear norm based methods as it gives better approximation of the rank function.

Although these low-rank based approaches have obtained good results, additional information should be considered for more accurate reconstructions. A promising choice is to exploit the sparse property of the complete matrix data in a certain domain, such as transform domains where many signals have inherently sparse structures [16,17]. The sparse low-rank texture inpainting (SLRTI) method proposed in [18] uses sparse structure obtained in transform domain to achieve better results for matrix completion. However, the sparse prior employed in this method is modeled using explicit bases in the form of matrix, which requires the transform to be separable. The SLRTI method employs a linearized approximation of the original objective function and thus only obtains an approximate solution. In addition, the nuclear norm is used in SLRTI to approximate the rank function, rather than the more accurate truncated nuclear norm.

This paper focuses on the matrix completion problem and proposes a novel method which simultaneously considers the low-rank and sparse priors. In particular, the truncated nuclear norm is used as the surrogate of the rank function, leading to a better approximation. The sparse prior is formulated as an ℓ_1 -norm regularizer in a more general way, as compared with the SLRTI method. Instead of using explicit bases to sparsify the underlying matrix, the sparse regularizer used in the proposed method is formulated in a more general way by applying the transform operator as an implicit function. As the proposed formulation cannot be addressed by traditional optimization methods directly, a two-step optimization method is proposed, which alternates between the singular value decomposition of the estimated matrix and the update of the matrix by solving a constrained optimization problem. To solve the problem involved in the second step, a variable splitting technique is used and a method following the alternating direction method of multipliers (ADMM) framework [19] is developed.

The remainder of the paper is organized as follows. In the next section, a brief review of the related work is provided. Our proposed method is presented in Section 3. Section 4 provides experimental results. Conclusions are drawn in Section 5.

2. Related work

As mentioned in the previous section, the matrix completion problem is usually addressed by considering the low-rank prior and minimizing the rank of the underlying matrix. Since the rank minimization problem (1) cannot be solved directly, the rank function in the objective function is relaxed to other forms that can be addressed more easily. The most common way is to use the nuclear norm to approximate the rank function [20,11], and thus the matrix completion problem (1) can be recast as

$$\begin{aligned} \min_X \|X\|_* \\ \text{s.t. } P_\Omega(X) = P_\Omega(M) \end{aligned} \quad (2)$$

where $\|X\|_* := \sum_{i=1}^{\min(m,n)} \sigma_i(X)$ is the nuclear norm of the matrix X and σ_i is the i th largest singular value of X . P_Ω is the orthogonal projection operator onto the span of matrices vanishing outside of Ω . This formulation is first used for matrix completion in [20] where the nuclear norm is proved to be the convex hull of the rank function. The optimization problem (2) can be reformulated as a semi-definite programming (SDP) problem which can be addressed by an interior-point method. However, the usage of this method is limited for large

scale matrices, as the SDP problem cannot be solved efficiently for matrices of very high dimensions.

To deal with the nuclear norm minimization problem more efficiently, especially for matrices with high dimensions, the SVT method was proposed in [11]. This method minimizes the approximation of the nuclear norm, that is

$$\begin{aligned} \min_X \|X\|_* + \tau \|X\|_F^2 \\ \text{s.t. } P_\Omega(X) = P_\Omega(M) \end{aligned} \quad (3)$$

with the parameter $\tau > 0$. The above optimization problem can be solved iteratively using a singular value shrinkage operator [11].

Different from the nuclear norm based approaches, the TNNR method [8] minimizes the truncated nuclear norm to get a better approximation of the rank function, i.e.,

$$\begin{aligned} \min_X \|X\|_r \\ \text{s.t. } P_\Omega(X) = P_\Omega(M) \end{aligned} \quad (4)$$

where $\|X\|_r := \sum_{i=r+1}^{\min(m,n)} \sigma_i(X)$ denotes the truncated nuclear norm of X . The TNNR method solves this problem using a two-step iterative scheme.

As the low-rank prior may not be sufficient to recover the original matrix accurately, the SLRTI method [18] introduces a sparse prior into the matrix completion task. Specifically, the underlying complete matrix is assumed to have sparse representation using bases B_1 and B_2 , i.e., $X = B_1 W B_2$ with W being a sparse matrix. Incorporating this sparse prior, the SLRTI method aims to solve the following optimization problem

$$\begin{aligned} \min_{X,W} \lambda \|W\|_1 + \|X\|_* \\ \text{s.t. } P_\Omega(X) = P_\Omega(M) \\ X = B_1 W B_2^T \end{aligned} \quad (5)$$

where $\|W\|_1 = \sum_{ij} |W_{ij}|$ denotes the ℓ_1 -norm of W . It is shown in [18] that the introduction of the sparse prior can improve the accuracy of recovery. Notice that, in the formulation of SLRTI, in order to incorporate the sparse prior, the original matrix is represented in the specific form $X = B_1 W B_2$ using two bases B_1, B_2 . However, this is not a general way to model the sparse property of a matrix, as it needs the transform operator to be separable so that X can be written in the form of matrix multiplication. In addition, SLRTI uses a linearized approximation of the objective function to address the optimization problem, which degrades the accuracy of the solution inevitably.

Motivated by the truncated nuclear norm introduced in the TNNR method [3] and the sparse prior used in SLRTI [18], we propose a novel matrix completion method by simultaneously considering the truncated nuclear norm and a sparse prior that is more general than the one used in SLRTI.

3. Proposed method

In this section, the formulation of the proposed method is presented first, and then the corresponding optimization framework is introduced in detail.

3.1. Problem formulation

In the formulation of our proposed method, the low-rank and sparse priors are both considered to restrict the ill-posed matrix completion problem. Since the truncated nuclear norm is able to provide a better approximation to the low rank function than the nuclear norm [8,3], it is used to model the low-rank prior. For the sparse prior, a more general model, rather than the specific sparse formulation used in SLRTI [18], is employed. In particular, we assume the original matrix X is sparse in a transform domain. Let $\mathcal{T}(\cdot)$ denote the forward transform operator, and the transformed matrix $W = \mathcal{T}(X)$ is assumed to be sparse. This formulation does not require the transform operator to be separable as

in SLRTI, which improves the flexibility of the selection of potential transforms. Specifically, we formulate the matrix completion problem as follows

$$\begin{aligned} \min_X \quad & \|X\|_r + \lambda \|W\|_0 \\ \text{s.t.} \quad & P_\Omega(X) = P_\Omega(M) \\ & W = \mathcal{T}(X), \end{aligned} \quad (6)$$

where $\lambda > 0$, $\|W\|_0$ denotes the ℓ_0 -norm of W and it returns the number of nonzero entries in W . As this formulation uses the Truncated Nuclear Norm and a Sparse Regularizer, the proposed method is named as TNN-SR.

Since the truncated nuclear norm term $\|X\|_r$ and the ℓ_0 -norm term $\|W\|_0$ are not convex, it is challenging to solve (6). To address this issue, the two non-convex terms in the objective function of (6) are converted to other forms. In particular, the ℓ_0 -norm can be relaxed as the ℓ_1 -norm which is convex [21]. The truncated nuclear norm term can be dealt based on the following theorem which has been proven in [8].

Theorem 1. Let $X \in \mathbb{R}^{m \times n}$ be any given matrix and r be any non-negative integer with $r \leq \min(m, n)$. For any matrices $A \in \mathbb{R}^{r \times m}$, $B \in \mathbb{R}^{r \times n}$ satisfying $AA^T = I_{r \times r}$ and $BB^T = I_{r \times r}$, we have

$$\text{Tr}(AXB^T) \leq \sum_{i=1}^r \sigma_i(X),$$

where $I_{r \times r}$ denotes the identity matrix of size $r \times r$.

Suppose the singular value decomposition (SVD) of X is $X = U\Sigma V^T$ with $U = (u_1, u_2, \dots, u_m) \in \mathbb{R}^{m \times m}$, $V = (v_1, v_2, \dots, v_n) \in \mathbb{R}^{n \times n}$, and $\Sigma \in \mathbb{R}^{m \times n}$. Let

$$A = (u_1, u_2, \dots, u_r)^T, \quad B = (v_1, v_2, \dots, v_r)^T, \quad (7)$$

we have

$$\begin{aligned} & \text{Tr}(AXB^T) \\ &= \text{Tr}((u_1, u_2, \dots, u_r)^T U \Sigma V (v_1, v_2, \dots, v_r)) \\ &= \text{Tr}(\text{diag}(\sigma_1(X), \sigma_2(X), \dots, \sigma_r(X))) \\ &= \sum_{i=1}^r \sigma_i(X) \end{aligned} \quad (8)$$

As $AA^T = I_{r \times r}$ and $BB^T = I_{r \times r}$, according to Theorem 1 and Eq. (8), we have

$$\max_{AA^T=I, BB^T=I} \text{Tr}(AXB^T) = \sum_{i=1}^r \sigma_i(X). \quad (9)$$

Therefore, the truncated nuclear norm term $\|X\|_r$ can be rewritten as

$$\begin{aligned} \|X\|_r &= \sum_{i=r+1}^{\min(m,n)} \sigma_i(X) \\ &= \|X\|_* - \max_{AA^T=I, BB^T=I} \text{Tr}(AXB^T) \end{aligned} \quad (10)$$

Using the ℓ_1 -norm as a convex surrogate of the ℓ_0 -norm and Eq. (10), the proposed formulation (6) can be recast as

$$\begin{aligned} \min_X \quad & \|X\|_* - \max_{AA^T=I, BB^T=I} \text{Tr}(AXB^T) + \lambda \|W\|_1 \\ \text{s.t.} \quad & P_\Omega(X) = P_\Omega(M) \\ & W = \mathcal{T}(X). \end{aligned} \quad (11)$$

Even though this converted formulation has avoided the non-convex truncated nuclear norm and ℓ_0 -norm terms, it still cannot be addressed directly by the existing methods.¹

¹ We have implemented the variant where W is recovered directly without X and X is recovered by applying the inverse transform of \mathcal{T} to W . It turns out that the results are very similar to those obtained by the proposed formulation (11). For the reason, this variant and the results are not included in this paper.

3.2. Optimization framework

To solve the proposed optimization problem (11), an iterative approach alternating between two steps is developed. In the first step of the l th iteration, the singular value decomposition is applied to the fixed matrix X_l , i.e., $X_l = U_l \Sigma_l V_l^T$, to calculate A_l and B_l based on U_l , V_l and (7). In the second step, A_l and B_l are fixed, and the matrix X_l is updated by solving the optimization problem as follows

$$\begin{aligned} \min_X \quad & \|X\|_* - \text{Tr}(A_l X B_l^T) + \lambda \|W\|_1 \\ \text{s.t.} \quad & P_\Omega(X) = P_\Omega(M) \\ & W = \mathcal{T}(X). \end{aligned} \quad (12)$$

The optimization framework to address the problem (11) is summarized in Algorithm 1. The algorithm to solve the problem (12), i.e., the details of Step 2 of Algorithm 1, will be introduced in the next subsection.

Algorithm 1 Proposed algorithm: TNN-SR

Input: Observed matrix $M \in \mathbb{R}^{m \times n}$, $\lambda > 0$, $r \leq \min(m, n)$, ϵ

Initialization:

Initialize the iteration counter $l = 1$ and the recovered matrix $X_1 = M$. Perform the following steps iteratively.

Main Steps:

1. Calculate the singular value decomposition of X_l :

$$[U_l, \Sigma_l, V_l] = \text{SVD}(X_l)$$

where $U_l = (u_1, u_2, \dots, u_m) \in \mathbb{R}^{m \times m}$,

$$V_l = (v_1, v_2, \dots, v_n) \in \mathbb{R}^{n \times n}.$$

Compute A_l and B_l :

$$A_l = (u_1, u_2, \dots, u_r)^T,$$

$$B_l = (v_1, v_2, \dots, v_r)^T.$$

2. Solve the following constrained optimization problem:

$$X_{l+1} = \arg \min_X \|X\|_* - \text{Tr}(A_l X B_l^T) + \lambda \|W\|_1$$

$$\text{s.t. } P_\Omega(X) = P_\Omega(M)$$

$$W = \mathcal{T}(X).$$

3. If $\|X_{l+1} - X_l\|_F \leq \epsilon$, let $X^* = X_{l+1}$ and quit the iteration. Otherwise, increase the iteration counter $l = l + 1$ and go back to step 1.

Output:

The recovered matrix X^* .

3.3. Optimization method to solve the problem (12)

In this section, we propose a new method to solve the optimization problem (12). In the original problem, the variable X is involved in multiple terms of the objective function and both of the two constraints, which makes it difficult to address this problem directly. To make the objective function separable with respect to the variables, a variable splitting technique is used. After that, the ADMM framework [19] is adapted to convert the problem to a sequence of subproblems.

By introducing a variable N , the problem (12) can be converted to the equivalent optimization task as follows

$$\begin{aligned} \min_{X, N, W} \quad & \|X\|_* - \text{Tr}(A_l N B_l^T) + \lambda \|W\|_1 \\ \text{s.t.} \quad & P_\Omega(X) = P_\Omega(M) \\ & N = X \\ & W = \mathcal{T}(X). \end{aligned} \quad (13)$$

Notice that X 's presented in $\text{Tr}(A_l X B_l^T)$ of the original objective function is replaced with N . An additional constraint $N = X$ is also introduced to enforce the equivalence to the original problem (12).

Due to the introduction of the variable N and the corresponding changes in the problem formulation, (13) can be handled with ADMM [19]. Using dual parameters Y and Z , the augmented Lagrangian function of (13) can be written as

$$\begin{aligned} \mathcal{L}(X, N, W, Y, Z) &= \|X\|_* - \text{Tr}(A_l N B_l^T) + \lambda \|W\|_1 \\ &+ \langle Y, N - X \rangle + \frac{\beta}{2} \|N - X\|_F^2 \\ &+ \langle Z, W - \mathcal{T}(X) \rangle + \frac{\beta}{2} \|W - \mathcal{T}(X)\|_F^2, \end{aligned} \quad (14)$$

where $\beta > 0$ is a penalty parameter. Based on the framework of ADMM, the optimization problem (13) can be solved by alternatively updating one variable with the others being fixed. Specifically, in the k th iteration, the variables are updated via the following scheme

$$\begin{cases} X^{k+1} = \arg \min_X \mathcal{L}(X, N^k, W^k, Y^k, Z^k), & \text{(a)} \\ W^{k+1} = \arg \min_W \mathcal{L}(X^{k+1}, N^k, W, Y^k, Z^k), & \text{(b)} \\ N^{k+1} = \arg \min_N \mathcal{L}(X^{k+1}, N, W^{k+1}, Y^k, Z^k), & \text{(c)} \\ Y^{k+1} = Y^k + \beta(N^{k+1} - X^{k+1}), & \text{(d)} \\ Z^{k+1} = Z^k + \beta(W^{k+1} - \mathcal{T}(X^{k+1})). & \text{(e)} \end{cases} \quad (15)$$

In this scheme, the update of X , W , N involves solving subproblems (15)(a)–(15)(c). Fortunately, these subproblems have closed-form solutions as will be detailed as follows.

3.3.1. Update X^{k+1}

In this part, we derive the closed-form solution for calculating X^{k+1} . Specifically, the subproblem as indicated in (15)(a) needs to be addressed, i.e.,

$$\begin{aligned} X^{k+1} &= \arg \min_X \mathcal{L}(X, N^k, W^k, Y^k, Z^k) \\ &= \arg \min_X \|X\|_* + \langle Y, N - X \rangle + \frac{\beta}{2} \|N - X\|_F^2 \\ &\quad + \langle Z, W - \mathcal{T}(X) \rangle + \frac{\beta}{2} \|W - \mathcal{T}(X)\|_F^2 \\ &= \arg \min_X \|X\|_* + \frac{\beta}{2} \left\| N - X + \frac{Y}{\beta} \right\|_F^2 \\ &\quad + \frac{\beta}{2} \left\| W - \mathcal{T}(X) + \frac{Z}{\beta} \right\|_F^2. \end{aligned} \quad (16)$$

Note that X cannot be directly separated from the other variables due to the existence of the transform operator \mathcal{T} in the last term of (16). Therefore, we cannot solve it with the classical singular value shrinkage operator [11] for the nuclear norm minimization problem.

Nevertheless, the Parseval's theorem [22] can be applied here to reformulate the problem, so that the variable X can be isolated from the operator $\mathcal{T}(\cdot)$. In particular, Parseval's theorem indicates that, a unitary transform (e.g., Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Hadamard Transform and Haar-Wavelet Transform) can be energy-conservation, i.e., $\|v\|_F^2 = \|u\|_F^2$, with $v = \mathcal{H}(u)$ and $\mathcal{H}(\cdot)$ being a unitary transform [23]. Based on this theorem, assume $\mathcal{T}(\cdot)$ is a unitary transform and let $\mathcal{G}(\cdot)$ denote its corresponding inverse transform. By applying $\mathcal{G}(\cdot)$ to the last term of (16), we can get

$$\left\| W - \mathcal{T}(X) + \frac{Z}{\beta} \right\|_F^2 = \left\| \mathcal{G}\left(W + \frac{Z}{\beta}\right) - X \right\|_F^2.$$

Hence, we have

$$\begin{aligned} &\frac{\beta}{2} \left\| N - X + \frac{Y}{\beta} \right\|_F^2 + \frac{\beta}{2} \left\| W - \mathcal{T}(X) + \frac{Z}{\beta} \right\|_F^2 \\ &= \frac{\beta}{2} \left\| N - X + \frac{Y}{\beta} \right\|_F^2 + \frac{\beta}{2} \left\| \mathcal{G}\left(W + \frac{Z}{\beta}\right) - X \right\|_F^2 \\ &= \beta \left\| X - \frac{1}{2} \left[N + \frac{Y}{\beta} + \mathcal{G}\left(W + \frac{Z}{\beta}\right) \right] \right\|_F^2 + \xi, \end{aligned}$$

where ξ is a term not related to X .

The update of X^{k+1} in (16) can be further rewritten as

$$\begin{aligned} X^{k+1} &= \arg \min_X \|X\|_* \\ &\quad + \beta \left\| X - \frac{1}{2} \left(N - \frac{Y}{\beta} + \mathcal{G}\left(W + \frac{Z}{\beta}\right) \right) \right\|_F^2. \end{aligned}$$

The above problem has the closed-form solution as follows

$$X^{k+1} = \mathbf{D}_{\frac{1}{2\beta}} \left\{ \frac{1}{2} \left[N + \frac{Y}{\beta} + \mathcal{G}\left(W + \frac{Z}{\beta}\right) \right] \right\}, \quad (17)$$

where $\mathbf{D}_\tau(\cdot)$ is the singular value shrinkage operator [11] defined as

$$\mathbf{D}_\tau(X) = U \mathbf{D}_\tau(\Sigma) V^T \quad (18)$$

with the singular value decomposition of X being $X = U \Sigma V^T$, $\Sigma = \text{diag}(\{\sigma_i\})$ with $1 \leq i \leq \min\{m, n\}$, and $\mathbf{D}_\tau(\Sigma) = \text{diag}(\max\{\sigma_i - \tau, 0\})$.

Considering the constraint $P_\Omega(X) = P_\Omega(M)$ in (13), we fix the observed entries of M and obtain

$$X^{k+1} = P_{\Omega^c}(X^{k+1}) + P_\Omega(M), \quad (19)$$

where Ω^c denotes the indices of the missing entries.

3.3.2. Update W^{k+1}

To update W^{k+1} , the following subproblem needs to be addressed

$$\begin{aligned} W^{k+1} &= \arg \min_W \mathcal{L}(X^{k+1}, N^k, W, Y^k, Z^k) \\ &= \arg \min_W \lambda \|W\|_1 + \langle Z, W - \mathcal{T}(X) \rangle + \frac{\beta}{2} \|W - \mathcal{T}(X)\|_F^2 \\ &= \arg \min_W \lambda \|W\|_1 + \frac{\beta}{2} \left\| W - \mathcal{T}(X) + \frac{Z}{\beta} \right\|_F^2. \end{aligned}$$

This problem has a closed-form solution given by

$$W^{k+1} = \mathcal{S}_{\frac{\lambda}{\beta}} \left(\mathcal{T}(X) - \frac{Z}{\beta} \right). \quad (20)$$

Here $\mathcal{S}_\tau(\cdot)$ represents the element-wise soft thresholding operator [24] defined by

$$\mathcal{S}_\tau(x) = \text{sgn}(x) \cdot (|x| - \tau), \quad (21)$$

where the function $\text{sgn}(\cdot)$ returns the sign of its operand.

3.3.3. Update N^{k+1}

The update of N^{k+1} involves solving a quadratic minimization problem as follows

$$\begin{aligned} N^{k+1} &= \arg \min_N \mathcal{L}(X^{k+1}, N, W^{k+1}, Y^k, Z^k) \\ &= \arg \min_N -\text{Tr}(A_l N B_l^T) + \langle Y, N - X \rangle + \frac{\beta}{2} \|N - X\|_F^2 \\ &= \arg \min_N -\text{Tr}(A_l N B_l^T) + \frac{\beta}{2} \left\| N - X + \frac{Y}{\beta} \right\|_F^2 \\ &= \arg \min_N \frac{\beta}{2} \left\| N - X + \frac{Y}{\beta} - \frac{A_l^T B_l}{\beta} \right\|_F^2. \end{aligned}$$

Therefore, we have

$$N^{k+1} = X^{k+1} - \frac{Y^k}{\beta} + \frac{A_l^T B_l}{\beta}. \quad (22)$$

3.3.4. Summary of the optimization algorithm

The algorithm to solve the problem (12) is summarized in Algorithm 2.

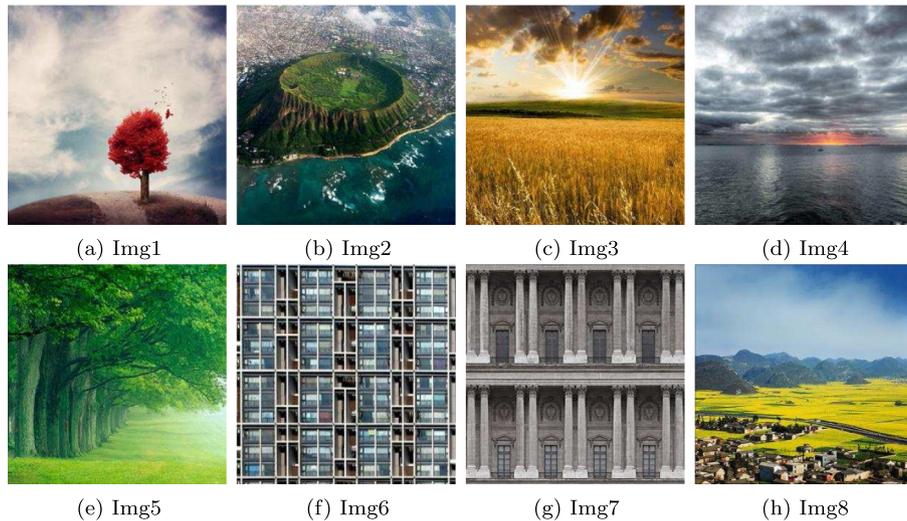


Fig. 1. Test images.

Algorithm 2 Optimization algorithm to solve the problem (12)**Input:** Input matrix M , error support Ω , parameters λ, β , tolerance ϵ .**Initialization:**Initialize the iteration counter $k = 1$, $X_k = N_k = M$, $W = 0$, Y_k as random matrix of size $m \times n$.**Main Steps:**

1. Update X^{k+1} using equations (17), (18) and (19).
2. Update W^{k+1} using equations (20) and (21).
3. Update N^{k+1} using equation (22).
4. $Y^{k+1} = Y^k + \beta (N^{k+1} - X^{k+1})$.
5. $Z^{k+1} = Z^k + \beta (W^{k+1} - \mathcal{T}(X^{k+1}))$.
6. If $\|X^{k+1} - X^k\|_F \leq \epsilon$, let $X = X^{k+1}$ and quit the iteration. Otherwise, increase the iteration counter $k = k + 1$ and go back to step 1.

Output: X .

3.4. Computational complexity

The proposed algorithm involves the SVD of a matrix of size $m \times n$ in the first step, and addresses the constrained optimization problem (12) iteratively in the second step. The computational complexity of the first step is dominated by the SVD operation whose complexity is $\mathcal{O}(\min(mn^2, m^2n))$. In the second step of TNN-SR, the variables are updated alternatively and the time complexity is related to the computational complexity of the transform operator \mathcal{T} . If a transform whose complexity is higher than SVD, e.g. DCT, is used, the update of X and W will be the dominant parts of the second step. In this case, let t denote the number of the iterations of step 2, and the complexity of the second step is $\mathcal{O}(m^2n^2t)$. As a result, the total time complexity of each iteration of the proposed method is dominated by the second step and scales as $\mathcal{O}(m^2n^2t)$.

4. Experimental results

In this section, several experiments are conducted to demonstrate the effectiveness of the proposed TNN-SR algorithm for matrix completion.²

Three state-of-the-art algorithms are used as the baselines: TNNR [8], SLRTI [18], and a recently proposed method named deep matrix factorization (DMF) [25].³ The DMF method formulates the low-rank matrix completion problem as a deep-structure neural network and recovers the matrix by approximating a nonlinear latent variable model. Eight color images of size 300×300 named *Img1*, *Img2*, ..., *Img8*, as shown in Fig. 1, are employed as test images in our experiments. The TNNR, SLRTI, and TNN-SR algorithms are applied to each channel separately and the final results are obtained by combining the results of the three channels. In the DMF algorithm, each image is unfolded to a matrix of 300×900 as in [25].

In the proposed algorithm, unitary transforms that promote the sparsity of the original matrix can be used as the transform operator \mathcal{T} , and DCT is employed as an example in the experiments. It should be noted that although the SLRTI algorithm also uses DCT as the transform to sparsify images, the subproblems of SLRTI is solved by approximating the original objective functions while the subproblems of the proposed TNN-SR algorithm are addressed with closed-form solutions. This difference leads to more effective employment of the transform operator in the proposed algorithm, and more accurate results for matrix completion.

The Peak Signal-to-Noise Ratio (PSNR) is used to evaluate the quality of the recovered images, which is defined by

$$\text{PSNR} = 10 \log_{10} \frac{255^2}{\text{MSE}}, \quad (23)$$

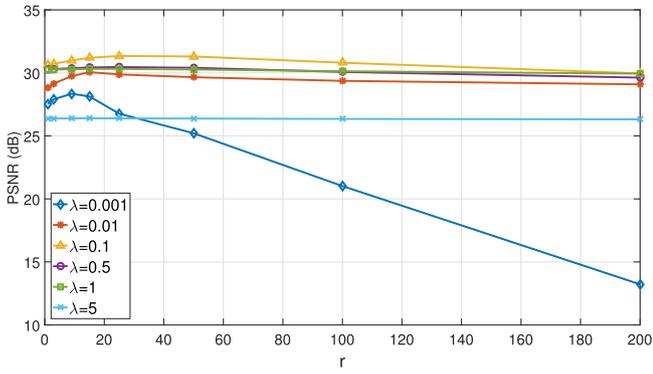
where MSE denotes the mean squared error between the original image and the recovered image. Higher PSNR indicates better recovery performance.

4.1. Random mask

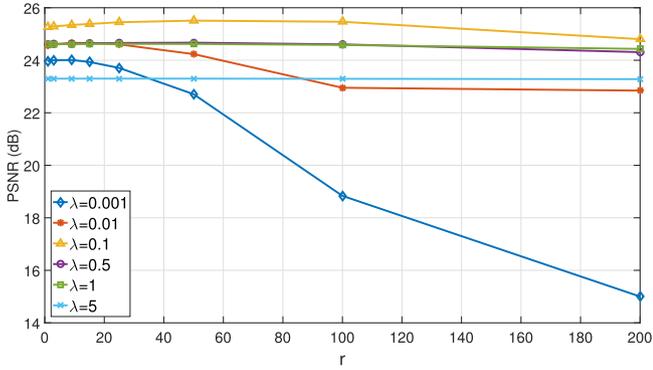
First, experiments for completing matrices with randomly missing entries are performed. In particular, some pixels of the test images are covered randomly and the algorithms are employed to recover the original images. The ratios of the missing entries are varied from 10% to 90%.

² The code of the proposed method has been uploaded to <https://github.com/jd0710/TNN-SR>.

³ The code of TNNR is downloaded from <https://sites.google.com/site/zjuyaohu/>. The code of SLRTI is implemented by ourselves, as it is not available online. We thank the authors of [25] for sharing the code of DMF via email.



(a) Img1, Missing Ratio = 70%



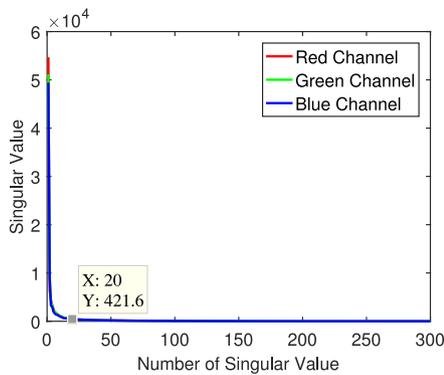
(b) Img2, Missing Ratio = 20%

Fig. 2. The PSNR values (in dB) obtained by the proposed TNN-SR algorithm using different λ 's and r 's with β being fixed. Specifically, $\beta = 10^{-3}$, $\lambda = \{0.001, 0.01, 0.1, 0.5, 1, 5\}$, and $r = \{1, 3, 9, 15, 25, 50, 100, 200\}$. (a) Img1, Missing Ratio = 70%; (b) Img2, Missing Ratio = 20%.

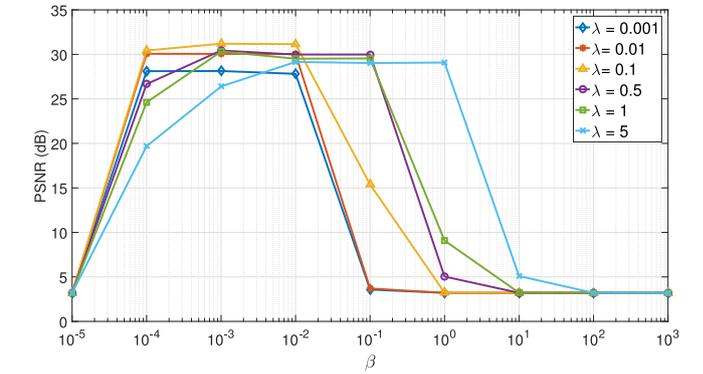
4.1.1. Simulations with different parameters of TNN-SR

As the settings of the parameters λ , β and r are important to the performance of the proposed TNN-SR algorithm, simulations with different parameters are performed to investigate the influence of the settings of these parameters.

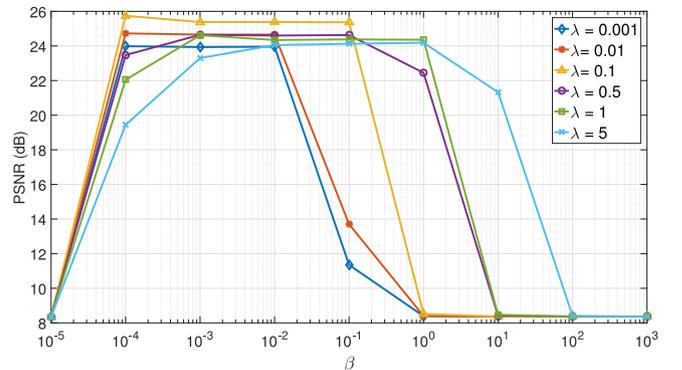
Firstly, the settings of $\lambda = \{0.001, 0.01, 0.1, 0.5, 1, 5\}$ and $r = \{1, 3, 9, 15, 25, 50, 100, 200\}$ are tested with β fixed as 10^{-3} . The PSNR results for recovering Img1 and Img2 with random masks are plotted in Fig. 2. It can be found that proper settings of λ are critical to obtain good results, and 0.1 is a favorable choice in most cases. Proper settings of r are integers between [1, 20]. The singular values of the three channels of Img1 and Img2 are shown in Fig. 3. As can be seen, the top 20 singular



(a) Img1



(a) Img1, Missing Ratio = 70%



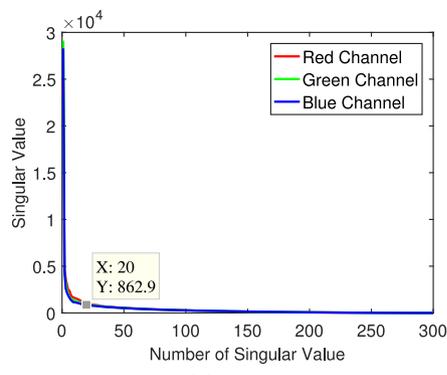
(b) Img2, Missing Ratio = 20%

Fig. 4. The PSNR values (in dB) obtained by the proposed TNN-SR algorithm using different λ 's and β 's with r being fixed. Specifically, $r = 15$, $\lambda = \{0.001, 0.01, 0.1, 0.5, 1, 5\}$, and $\beta = \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10, 100, 1000\}$. (a) Img1, Missing Ratio = 70%; (b) Img2, Missing Ratio = 20%.

values dominate the information, which is consistent with the proper settings of r in the proposed algorithm.

The value of r is then fixed as 15, and different values of λ and β are tested to demonstrate the influence of these two parameters. Specifically, $\lambda = \{0.001, 0.01, 0.1, 0.5, 1, 5\}$ and $\beta = \{10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 0.1, 1, 10, 100, 1000\}$ are tested. The results for Img1 and Img2 are shown in Fig. 4. It can be observed that the trends of the results with varying β 's are consistent for different λ 's, and proper settings of β are between $[10^{-3}, 10^{-2}]$.

The parameters β and r are fixed as $\beta = 10^{-3}$ and $r = 15$ respectively, the results of the test images with different $\lambda = \{0.001, 0.01, 0.1, 0.5, 1, 5\}$ are shown in Fig. 5, where “MR” in the legends is short for “Missing



(b) Img2

Fig. 3. The singular values of the red channel, the blue channel and the green channel.

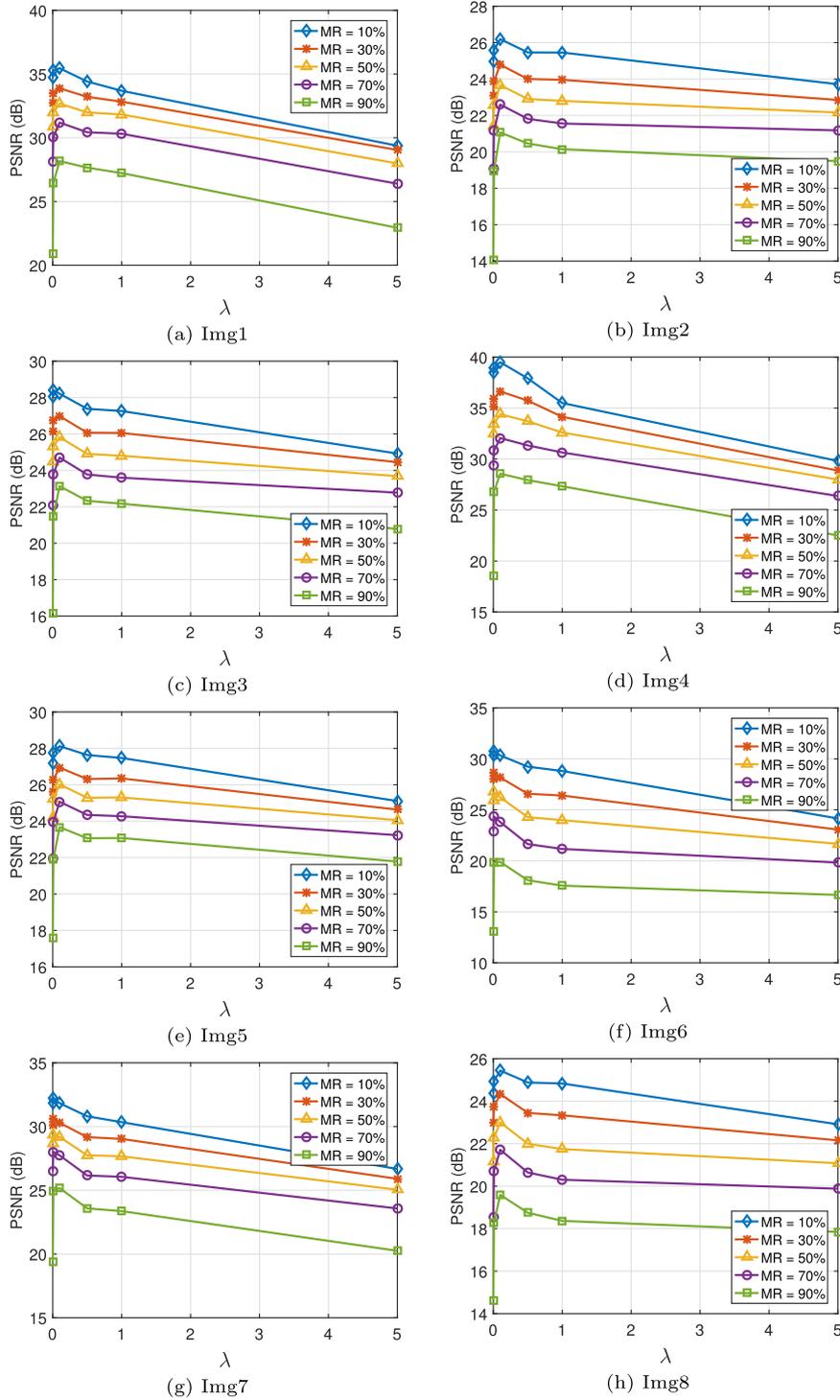


Fig. 5. The PSNR values (in dB) obtained by the proposed TNN-SR algorithm using different λ 's with $\beta = 10^{-3}$ and $r = 15$. $\lambda = \{0.001, 0.01, 0.1, 0.5, 1, 5\}$.

Ratio". We can see that for most images $\lambda = 0.1$ can achieve the best performance, and for Img6 and Img7 $\lambda = 0.01$ obtains the best results.

4.1.2. Simulations with adaptive penalty parameter

In the proposed TNN-SR algorithm, the penalty parameter β in the Lagrangian function (14) is fixed as in the traditional ADMM approach [19]. However, in real applications, a dynamical β is usually preferred to speed up the convergence of the algorithm [8,26]. Based on this, an adaptive penalty (AP) is used to replace the fixed β in the TNN-SR algorithm, which results in a modified version referred to as TNN-SR-AP. In particular, the following adaptive update step [8] for the

penalty parameter β is added after the update of all variables, i.e., step 5 of Algorithm 2

$$\beta^{k+1} = \min(\beta_{max}, \gamma \beta^k) \tag{24}$$

where β_{max} is an upper bound of β^k and $\gamma \geq 1$ is the coefficient to increase β . The value of γ is given as

$$\gamma = \begin{cases} \gamma_0, & \text{if } \frac{\beta^k \max\{\|X^{k+1} - X^k\|_F, \|N^{k+1} - N^k\|_F\}}{\|M\|_F} < \kappa \\ 1, & \text{otherwise} \end{cases} \tag{25}$$

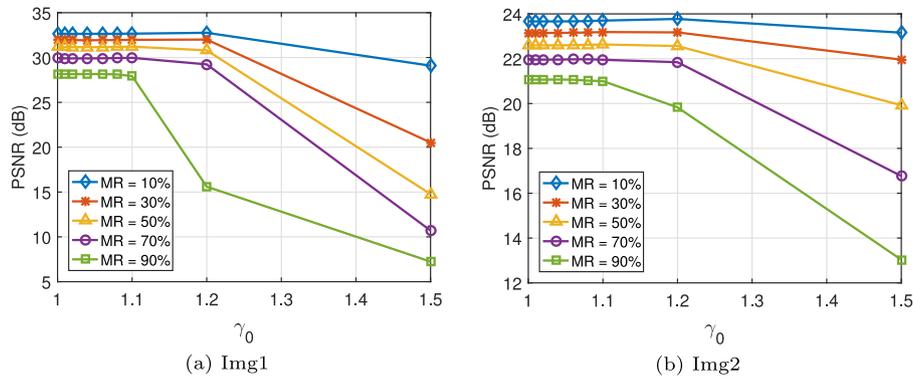


Fig. 6. The PSNR values (in dB) obtained by TNN-SR-AP where an adaptive penalty parameter is adapted based on Eqs. (24) and (25). The parameters are set as: $r = 15$, $\lambda = 0.1$, $\beta^1 = 10^{-3}$, $\beta_{max} = 10^{10}$, $\kappa = 10^{-3}$, and $\gamma_0 = \{1, 1.01, 1.02, 1.04, 1.06, 1.08, 1.1, 1.2, 1.5\}$.

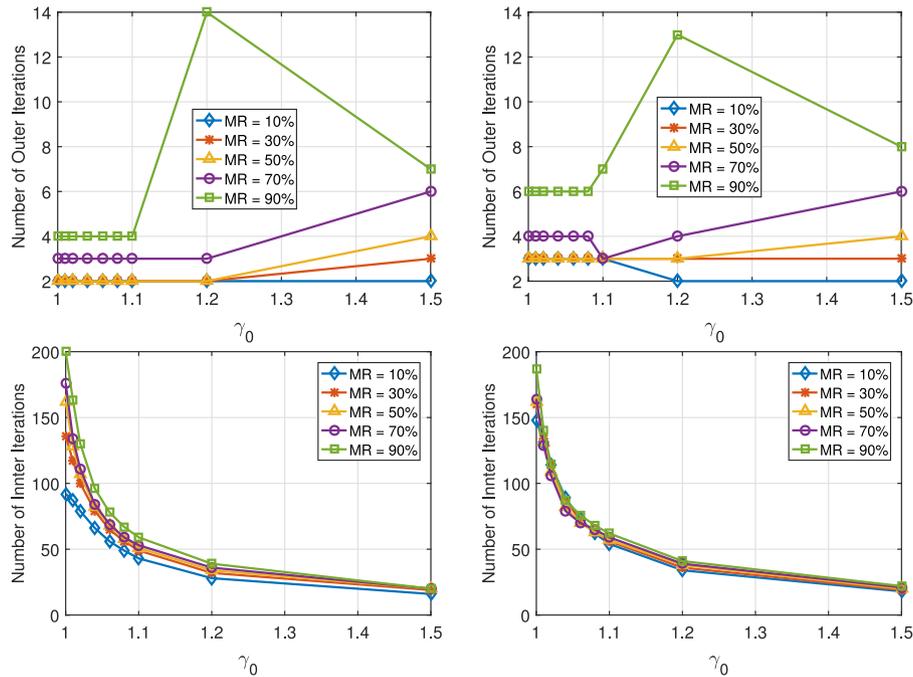


Fig. 7. The number of outer iterations (top) and the number of inner iterations (bottom) used by the TNN-SR-AP algorithm with respect to the choice of γ_0 . Left column: Img1. Right column: Img2.

where $\gamma_0 > 1$ and $\kappa > 0$ are constants. Using this adaptive update strategy, if the difference between (X^{k+1}, N^{k+1}) and (X^k, N^k) is small enough, β is updated as $\gamma_0 \beta^k$ in the $(k + 1)$ -th iteration for accelerating the speed of convergence. Note that the TNN-SR algorithm can be seen as a special case of TNN-SR-AP with $\gamma_0 = 1$.

In this adaptive update approach, the selection of γ_0 is critical. We set $r = 15$, $\lambda = 0.1$, $\beta^1 = 10^{-3}$, $\beta_{max} = 10^{10}$ and $\kappa = 10^{-3}$, and test different values of $\gamma_0 = \{1, 1.01, 1.02, 1.04, 1.06, 1.08, 1.1, 1.2, 1.5\}$. The results of TNN-SR-AP are shown in Fig. 6. It can be seen that proper settings of γ_0 lead to similar results of TNN-SR, but too large values of γ_0 may degrade the performance. Recommended range of γ_0 is $[1, 1.1]$.

To investigate the improvement of the convergence speed resulting from the adaptive update strategy in TNN-SR-AP, the numbers of the outer iterations and the inner iterations of TNN-SR-AP for recovering Img1 and Img2 are shown in Fig. 7. Note that the outer iteration is in the main framework to address the main problem (11) and the inner iteration refers to the process to address the subproblem (12) involved in the second step of the main process. From Fig. 7, it can be seen that with the increase of γ_0 the numbers of the inner iterations required decrease while the numbers of the outer iterations remain similar. This

demonstrates the speedup of the convergence due to the employment of the adaptive update strategy.

4.1.3. Results as compared with baseline algorithms

The parameters of TNN-SR are set as $\lambda = 0.1$, $\beta = 10^{-3}$, $r = 15$, and $\epsilon = 10^{-3}$. The parameters of TNN-SR-AP are set as $\gamma = 1.1$, $\beta^1 = 10^{-3}$, $\beta_{max} = 10^{10}$ and $\kappa = 10^{-3}$ with the common parameters remaining the same as in TNN-SR. The parameters of the baseline algorithms are tuned empirically to achieve the best performance.

The PSNR values of the images recovered by different algorithms are shown in Fig. 8. Some recovery examples with 90% and 70% missing ratios are shown in Fig. 9 and Fig. 10, respectively.⁴ From Fig. 8, it can be found that the results of TNN-SR and TNN-SR-AP are very similar in different cases, and the images recovered by the proposed algorithms have much higher PSNR values, as compared with the results of the baseline algorithms, especially for images with large pixel missing ratios. For example, with missing ratio 90%, both TNN-SR and SLRTI

⁴ Note that as the recovery images of TNN-SR-AP are very similar to the results of TNN-SR, the recovery samples of TNN-SR-AP are not presented.

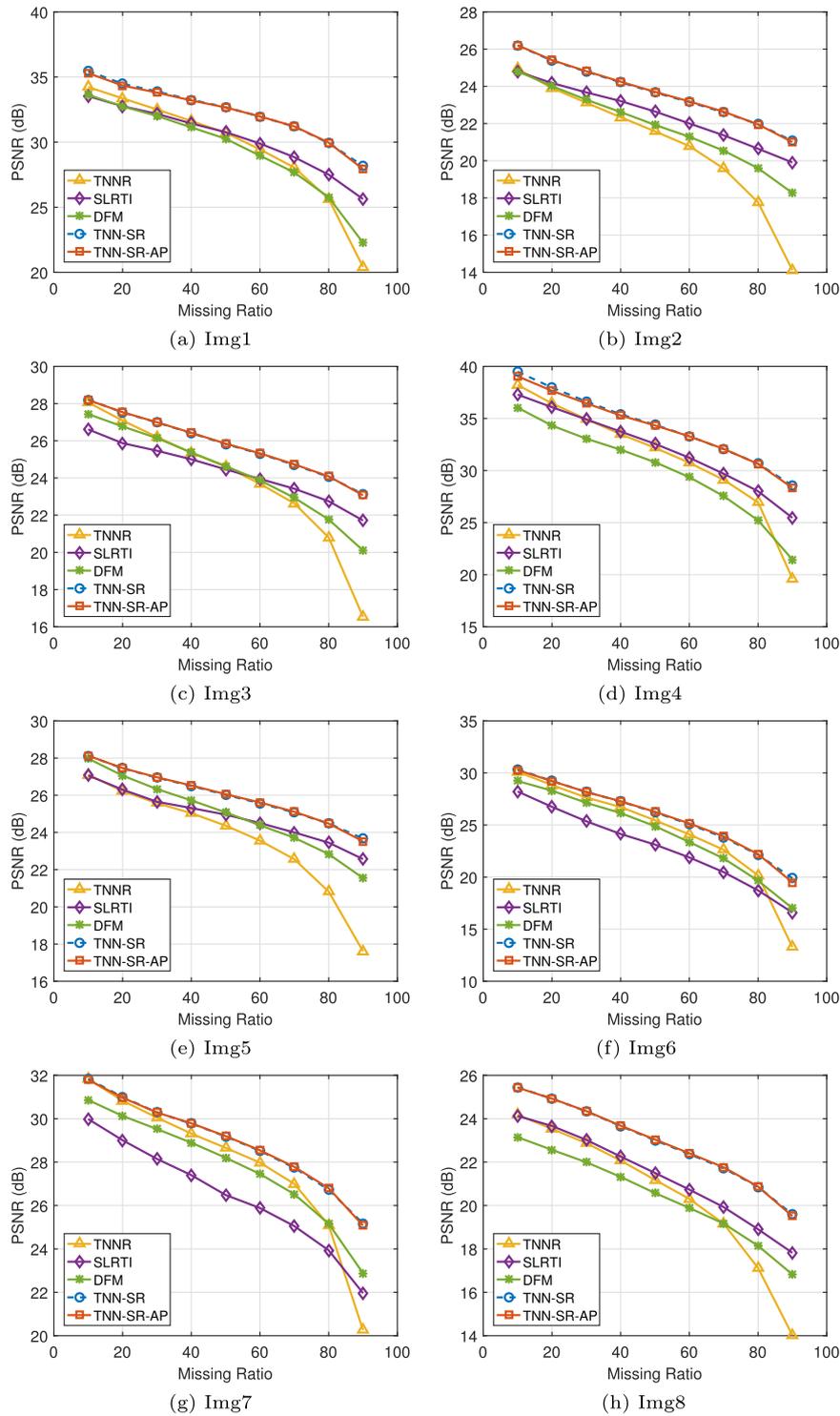


Fig. 8. The PSNR values of the recovered images under different missing pixel ratios (random mask) using different algorithms.

algorithms obtain higher PSNR values than the TNNR algorithm, which indicates that the sparse regularizer plays a great role for repairing the missing entries. The DMF algorithm outperforms TNNR and SLRTI in some cases, but its performance is not as good as the proposed algorithm. The recovery results shown in Figs. 9 and 10 also demonstrate the superiority of the proposed algorithm in terms of visual effect.

The running time of the compared algorithms for recovering Img1 and Img2 with random masks at different missing ratios is shown in Fig. 11. It can be seen that the running time of the proposed TNN-SR

and TNN-SR-AP algorithms is comparable to the baselines, and TNN-SR-AP is faster than TNN-SR as fewer iterations are required. The DMF algorithm requires more time than other algorithms.

4.2. Text mask

In this part, we would like to test the performance of the algorithms for removing text mask. It is hard to deal with text mask, as the corruptions of the pixels are not randomly distributed and some important

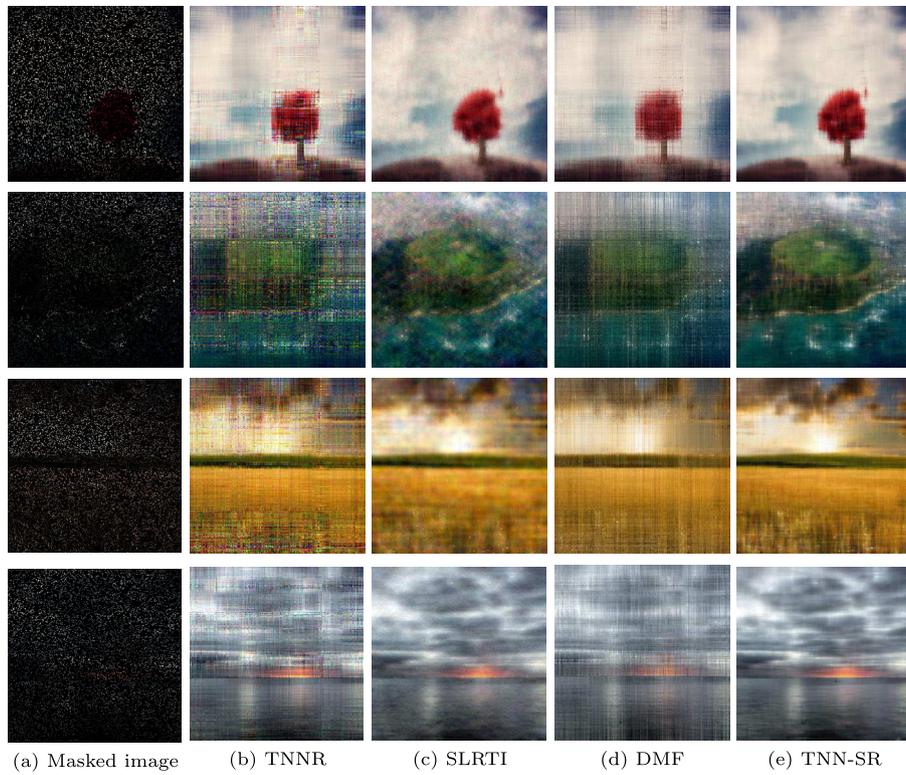


Fig. 9. The completion results with 90% pixels missing. (a) Masked image. (b)–(e) Recovered images by TNNR, SLRTI, DMF, and the proposed TNN-SR algorithm, respectively.

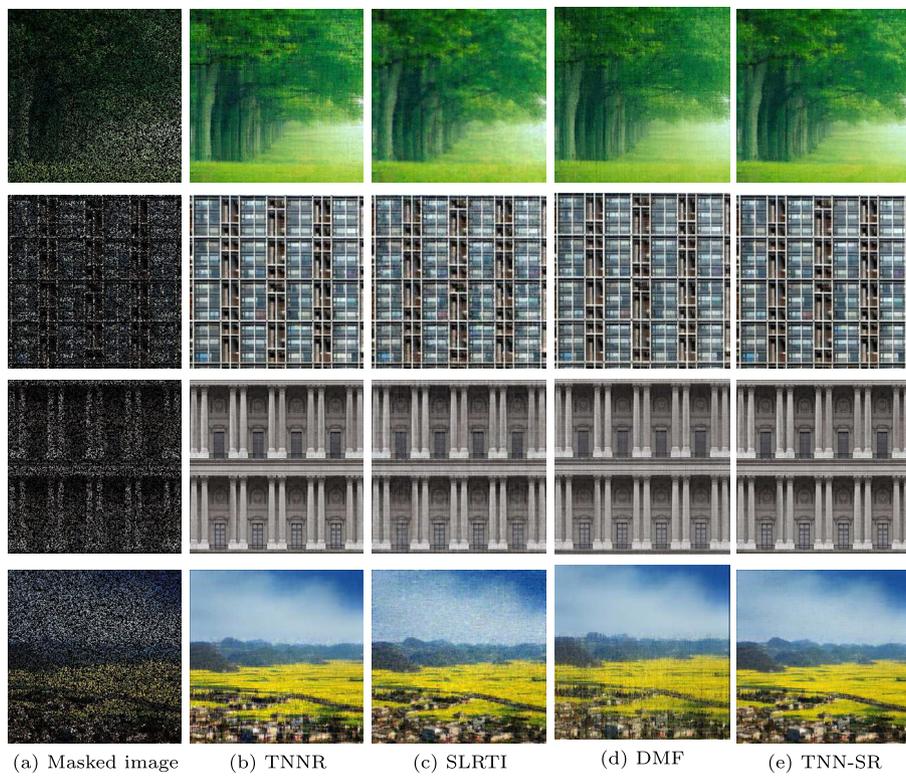


Fig. 10. The completion results with 70% pixels missing. (a) Masked image. (b)–(e) Recovered images by TNNR, SLRTI, DMF, and the proposed TNN-SR algorithm, respectively.

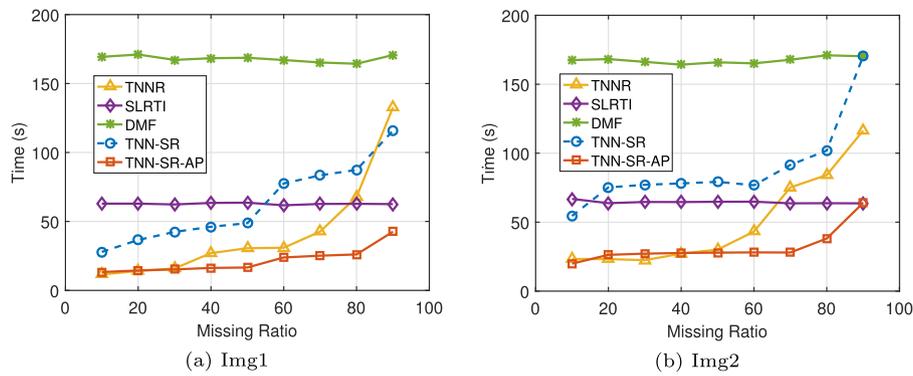


Fig. 11. Running time of the compared algorithms for recovering the images with random masks at different missing ratios. (a) *Img1*. (b) *Img2*.

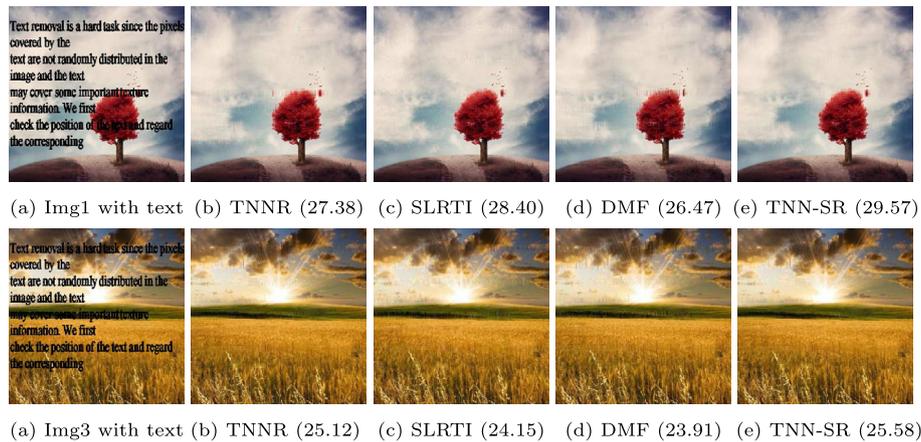


Fig. 12. The completion results with text mask. (a) Noisy image. (b)–(e) Recovered images by TNNR, SLRTI, DMF, and the proposed TNN-SR algorithm, respectively. The PSNR values (in dB) are in the parentheses.

texture information may be covered by the text. The experiments for text removal are performed on *Img1* and *Img3*.

The recovered images are shown in Fig. 12. It can be seen that the proposed algorithm can recover the missing pixels caused by text mask noise very well, and it achieves better visual effect than the baseline algorithms. Furthermore, we can also observe that higher PSNR values are obtained by the proposed TNN-SR method, as compared to the baseline algorithms. Specifically, for *Img1*, the PSNR values obtained by TNNR, SLRTI, DMF, and TNN-SR are 27.38 dB, 28.40 dB, 26.47 dB, and 29.57 dB, respectively. For *Img3*, the PSNR values obtained by these algorithms are 25.12 dB, 24.15 dB, 23.91 dB, and 25.58 dB, respectively. From both the visual effect and the numerical PSNR values, it can be demonstrated that the performance of the proposed algorithm is better than the baseline algorithms.

5. Conclusion

In this paper, we have proposed a novel matrix completion algorithm based on low-rank and sparse priors. Specifically, the truncated nuclear norm is employed to approximate the rank of the matrix, rather than the nuclear norm used in most existing approaches, to obtain a more accurate approximation. The sparse prior is exploited by an ℓ_1 -norm regularizer based on a transform operator, which is a general form to model the sparse property of the underlying matrix. We have also proposed an optimization method consisting of two steps, and the ADMM framework is adapted to solve the subproblem in the second step. Experimental results for recovering images corrupted by random mask and text mask have shown the superiority of the proposed algorithm in comparison with two baseline algorithms.

Acknowledgments

This work was supported by the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province of China (17KJB510025). The authors thank the Associate Editor and the anonymous reviewers for their contributions to improving the quality of the paper.

References

- [1] N. Komodakis, Image completion using global optimization, in: 2006 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2006, pp. 442–452.
- [2] W. Li, L. Zhao, D. Xu, D. Lu, Efficient image completion method based on alternating direction theory, in: 2013 IEEE International Conference on Image Processing, ICIP, IEEE, 2013, pp. 700–703.
- [3] H. Ji, C. Liu, Z. Shen, Y. Xu, Robust video denoising using low rank matrix completion, in: 2010 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2010, pp. 1791–1798.
- [4] R.S. Cabral, F. Torre, J.P. Costeira, A. Bernardino, Matrix completion for multi-label image classification, in: Advances in Neural Information Processing Systems, 2011, pp. 190–198.
- [5] R. Cabral, F.D.I. Torre, J.P. Costeira, A. Bernardino, Matrix completion for weakly-supervised multi-label image classification, IEEE Trans. Pattern Anal. Mach. Intell. 37 (1) (2015) 121–135.
- [6] E.J. Candès, B. Recht, Exact matrix completion via convex optimization, Found. Comput. Math. 9 (6) (2009) 717–772.
- [7] H. Steck, Training and testing of recommender systems on data missing not at random, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2010, pp. 713–722.
- [8] Y. Hu, D. Zhang, J. Ye, X. Li, X. He, Fast and accurate matrix completion via truncated nuclear norm regularization, IEEE Trans. Pattern Anal. Mach. Intell. 35 (9) (2013) 2117–2130.
- [9] P. Jain, P. Netrapalli, S. Sanghavi, Low-rank matrix completion using alternating minimization, in: Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, ACM, 2013, pp. 665–674.

- [10] B. Vandereycken, Low-rank matrix completion by Riemannian optimization, *SIAM J. Optim.* 23 (2) (2013) 1214–1236.
- [11] J.-F. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, *SIAM J. Optim.* 20 (4) (2010) 1956–1982.
- [12] E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis?, *J. ACM* 58 (3) (2011) 11–49.
- [13] J. Wright, A. Ganesh, S. Rao, Y. Peng, Y. Ma, Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization, in: Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, A. Culotta (Eds.), *Advances in Neural Information Processing Systems*, vol. 22, Curran Associates, Inc., 2009, pp. 2080–2088.
- [14] K.-C. Toh, S. Yun, An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems, *Pac. J. Optim.* 6 (3) (2009) 615–640.
- [15] D. Zhang, Y. Hu, J. Ye, X. Li, X. He, Matrix completion by truncated nuclear norm regularization, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, 2012, pp. 2192–2199.
- [16] J. Yang, J. Wright, T.S. Huang, Y. Ma, Image super-resolution via sparse representation, *IEEE Trans. Image Process.* 19 (11) (2010) 2861–2873.
- [17] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T.S. Huang, S. Yan, Sparse representation for computer vision and pattern recognition, *Proc. IEEE* 98 (6) (2010) 1031–1044.
- [18] X. Liang, X. Ren, Z. Zhang, Y. Ma, Repairing sparse low-rank texture, in: *Proceedings of the 12th European Conference on Computer Vision - Volume Part V*, Springer-Verlag, 2012, pp. 482–495.
- [19] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends® Mach. Learn.* 3 (1) (2011) 1–122.
- [20] M. Fazel, *Matrix Rank Minimization with Applications*, (Ph.D. thesis), Stanford University, 2002.
- [21] S.S. Chen, D.L. Donoho, M.A. Saunders, Atomic decomposition by basis pursuit, *SIAM Rev.* 43 (1) (2001) 129–159.
- [22] N. Merhav, R. Kresch, Approximate convolution using DCT coefficient multipliers, *IEEE Trans. Circuits Syst. Video Technol.* 8 (4) (1998) 378–385.
- [23] P. Porwik, A. Lisowska, The Haar-wavelet transform in digital image processing: Its status and achievements, *Mach. Graph. Vis.* 13 (1–2) (2004) 79–98.
- [24] D.L. Donoho, I.M. Johnstone, Adapting to unknown smoothness via wavelet shrinkage, *J. Amer. Statist. Assoc.* (1995) 1200–1224.
- [25] J. Fan, J. Cheng, Matrix completion by deep matrix factorization, *Neural Netw.* 98 (2017) 34–41.
- [26] Z. Lin, R. Liu, Z. Su, Linearized alternating direction method with adaptive penalty for low-rank representation, *Adv. Neural Inf. Process. Syst.* (2011) 612–620.