International Journal of Machine Learning and Cybernetics (2021) 12:1557–1570
https://doi.org/10.1007/s13042-020-01256-7

ORIGINAL ARTICLE

Algorithms of matrix recovery based on truncated Schatten *p*-norm

Chenglin Wen[1] · **Wenchao Qian**[2] · **Qinghua Zhang**[1] · **Feilong Cao**[2]

Received: 22 September 2020 / Accepted: 9 December 2020 / Published online: 3 January 2021

Abstract

In recent years, algorithms to recovery low-rank matrix have become one of the research hotspots, and more corresponding optimization models with nuclear norm have also been proposed. However, nuclear norm is not a good approximation to the rank function. This paper proposes a matrix completion model and a low-rank sparse decomposition model based on truncated Schatten *p*-norm, respectively, which combine Schatten *p*-norm with truncated nuclear norm, so that the models are more flexible. To solve these models, the function expansion method is first used to transform the non-convex optimization models into the convex optimization ones. Then, the two-step iterative algorithm based on alternating direction multiplier method (ADMM) is employed to solve the models. Further, the convergence of the proposed algorithm is proved mathematically. The superiority of the proposed method is further verified by comparing the existing methods in synthetic data and actual images.

Keywords Matrix completion · Matrix decomposition · Sparsity · Schatten *p*-norm · Truncated nuclear norm

1 Introduction

As a special case of matrix recovery, matrix completion addresses to complete the incomplete matrix according to the existing information, which has been widely applied in image restoration [1–4], video denoising [5], and recommendation system [6, 7]. A typical example is Netflix prize problem [8], which can infer the user's favorite degree for other films according to the user's evaluation of some films, and then the movie recommendation is made for other users.

Assuming that $M \in \mathbf{R}^{m\times n}$ is a given low rank or approximate low rank matrix and $X \in \mathbf{R}^{m\times n}$ is a low rank matrix to be recovered, then the completion problem of X can be represented as the following optimization problem:

$$\min_{X} \ \mathrm{rank}(X) \ \ \text{s. t.} \ \ X_{ij} = M_{ij}, (i,j) \in \Omega, \tag{1.1}$$

where $\mathrm{rank}(\cdot)$ denotes the rank of matrix, $\Omega \subset \{1, \dots, m\} \times \{1, \dots, n\}$ is a set of location coordinates for known data. If the sampling operator, $P_\Omega(X)$,

$$[P_\Omega(X)]_{ij} = \begin{cases} X_{ij}, & (i,j) \in \Omega; \\ 0, & \text{otherwise}, \end{cases} \tag{1.2}$$

is introduced, then the problem (1.1) can be denoted as

$$\min_{X} \ \mathrm{rank}(X) \quad \text{s. t.} \ \ P_\Omega(X) = P_\Omega(M). \tag{1.3}$$

Another important problem of matrix recovery is low rank sparse decomposition, also known as robust principal component analysis (RPCA), which has been widely applied in medical image processing [9, 10], video surveillance [11, 12], pattern recognition [13–15], and so on. The main purpose of RPCA is to decompose M into the sum of a low rank matrix X and a sparse matrix E, that is, $M = X + E$, where X and E are unknown matrices. In other words, the optimization problem

$$\min_{X,E} \ \mathrm{rank}(X) + \lambda \|E\|_0 \quad \text{s. t.} \ \ M = X + E \tag{1.4}$$

needs to be solved, where $\|\cdot\|_0$ denotes the ℓ_0 norm of matrix, and λ (> 0) is regularization parameter.

Since the rank function is non-convex and discontinuous, the optimization problems (1.3) and (1.4) are both NP hard problems, which cannot be solved directly by the commonly used optimization algorithms. Fazel [16] proposed a method

✉ Feilong Cao
icteam@163.com

[1] Guangdong Provincial Key Laboratory of Petrochemical Equipment Fault Diagnosis, Guangdong University of Petrochemical Technology, Maoming 525000, Guangdong Province, China

[2] Department of Information and Mathematics Sciences, China Jiliang University, Hangzhou 310018, Zhejiang, People's Republic of China

to replace the rank function in the optimization problems (1.3) and (1.4) by using the nuclear norm (NN) $\|\cdot\|_*$, and the ℓ_1 norm instead of the ℓ_0 norm. Therefore, Eqs. (1.3) and (1.4) are converted to

$$\min_X \ \|X\|_* \quad \text{s. t.} \ P_\Omega(X) = P_\Omega(M), \tag{1.5}$$

and

$$\min_{X,E} \ \|X\|_* + \lambda\|E\|_1 \quad \text{s. t.} \ M = X + E, \tag{1.6}$$

respectively. Here $\|X\|_* = \sum_{i=1}^{\min(m,n)} \sigma_i(X)$ $(\sigma_1(X) \geq \sigma_2(X) \geq \cdots \geq \sigma_{\min(m,n)}(X))$ is NN of matrix X, that is, the sum of all singular values of matrix X, and $\|E\|_1 = \sum_{i=1}^m \sum_{j=1}^n |e_{ij}|$ is the ℓ_1 norm of matrix E, that is, the sum of all the absolute values of elements of matrix E.

Up to now, a series of studies [16–19] have shown that NN can be used as an approximate substitute for rank function. At the same time, some algorithms for solving convex optimization with NN [20–23] have been proposed one after another. This is because in the rank function, all the non-zero singular values have the same effect, and NN adds all the non-zero singular values and minimizes them at the same time, so that the different singular values have different contributions as much as possible. However, NN is not the best approximate substitute for rank function. Although the algorithms based on NN have strong theoretical guarantee, these algorithms can only get the suboptimal solution in practical application.

In 2012, a non-convex minimum optimal substitution for NN is proposed, which is named as Schatten p-norm [6], and defined by $\|X\|_p = \left(\sum_{i=1}^{\min(m,n)} \sigma_i^p(X)\right)^{1/p}$ $(0 < p \leq 1)$. Obviously, when $p = 1$, it is equivalent to the NN. And when p approaches 0, the Schatten p-norm approximates the rank function. In 2013, Hu et al. [1] proposed a new norm called as truncated nuclear norm (TNN): $\|X\|_r = \sum_{i=r+1}^{\min(m,n)} \sigma_i(X)$. Its main idea is to remove the larger singular values of r and add the remaining singular values of $(\min(m,n) - r)$ to reduce the influence of large singular values on low rank. Recently, Gu et al. [24] proposed to replace NN with the weighted nuclear norm (WNN) defined by $\|X\|_{\omega,*} = \sum_{i=1}^{\min(m,n)} \omega_i\sigma_i(X)$ in order to change the influence of singular value on rank function with different weights, and showed that the proposed WNN has a good approximate effect. In [25] and [26], a truncated Schatten p-norm (TSPN) is proposed, which sums only the p power of singular values of $(\min(m,n) - r)$, that is, $\|X\|_r^p = \sum_{i=r+1}^{\min(m,n)} \sigma_i^p(X)$. In [25], a compression sensing model based on truncated Schatten p-norm is proposed, and the alternating direction multiple method (ADMM) was used to solve the model. The proposed method in [25] has a good performance in image restoration. In [26], Chen et al. used truncated Schatten p-norm and proposed an optimization model of human motion recovery, and further showed that the truncated Schatten p-norm has a better effect of approximate rank function than other norms by using a series of experiments.

The motivations of the paper are as follows.

- In the rank function, all the non-zero singular values have the same effect. However, the different singular values should have different contributions as much as possible. So NN is not the best approximate substitute for rank function.
- The main idea of TNN is to remove the larger singular values of r and add the remaining singular values of $(\min(m,n) - r)$ to reduce the influence of large singular values on low rank. It was shown that TNN has better approximate effect than NN.
- When $0 < p < 1$, Schatten p-norm is a non-convex minimum optimal substitution for NN, and when $p = 1$, it is equivalent to the nuclear norm. Specially, when p approximates 0, the Schatten p-norm approximates the rank function.
- Truncated Schatten p-norm sums only the p power of singular values of $(\min(m,n) - r)$, which combines TNN and Schatten p-norm advantages. It has been shown in [25, 26] that Schatten p-norm has a good performance in image restoration and a better effect of approximate rank function than other norms by using a series of experiments.

The main contributions of this paper can be summarized as follows.

- The truncated Schatten p-norm based non-convex regularization models, which are directed against matrix completion and RPCA problems, respectively, are proposed;
- The function expansion method is used to transform the non-convex optimization model into the convex optimization model, and the two-step iterative algorithm of ADMM [27] is utilized to solve the model;
- The convergence of the proposed algorithm is proved mathematically;
- The advantages of the proposed methods are further illustrated by synthetic data, actual image restoration, and background separation experiments.

The contents of this paper are arranged as follows. In Sect. 2, we establish regularization models of matrix completion based on truncated Schatten p-norm, corresponding algorithms, and convergence theorem. The corresponding RPCA regularization model and solution method based on truncated Schatten p-norm are described in Sect. 3. Section 4 describes and analyses the experimental results, and through the comparison of experimental results, the advantages of the proposed method are further explained. Section 5 makes a summarization and reduces the conclusion of the whole article. Finally, the details of the proof of the convergence theorem are attached in "Appendix".

2 Regularization of matrix completion based on truncated Schatten *p*-norm (TSPN-MC)

2.1 TSPN-MC model

From [25] and [26], it follows that truncated Schatten p-norm can be written as

$$\begin{aligned}\|X\|_r^p &= \sum_{i=r+1}^{\min(m,n)} (\sigma_i(X))^p \\ &= \min_{A,B} \sum_{i=1}^{\min(m,n)} (1-\sigma_i(B^\top A))(\sigma_i(X))^p, \quad 0 < p \le 1,\end{aligned} \tag{2.1}$$

where $A \in \mathbf{R}^{r\times m}$, and $B \in \mathbf{R}^{r\times n}$. Further, we need the following lemma.

Lemma 2.1 *(see*[25]*) Suppose that the rank of matrix* $X \in \mathbf{R}^{m\times n}$ *is* $r(r \le \min(m,n))$*, and its singular value decomposition is* $X = U\Delta V^\top$*, where* $U = (\mu_1, \dots, \mu_m) \in \mathbf{R}^{m\times m}$ *,* $\Delta \in \mathbf{R}^{m\times n}$*,* $V = (v_1, \dots, v_n) \in \mathbf{R}^{n\times n}$*. Then when* $A = (\mu_1, \dots, \mu_r)^\top \in \mathbf{R}^{r\times m}$*,* $B = (v_1, \dots, v_r)^\top \in \mathbf{R}^{r\times n}$*, and* $0 < p \le 1$*, the optimization problem:*

$$\begin{aligned}\|X\|_r^p = &\min_{A,B} \sum_{i=1}^{\min(m,n)} (1-\sigma_i(B^\top A))(\sigma_i(X))^p \text{ s. t. } AA^\top \\ &= I_{r\times r}, BB^\top = I_{r\times r}\end{aligned} \tag{2.2}$$

has optimal solution.

From Eq. (2.1) and Lemma 2.1, we improve the model of matrix completion (1.3) and obtain the new model based on the truncated Schatten p-norm:

$$\min_X \sum_{i=1}^{\min(m,n)} (1-\sigma_i(B^\top A))(\sigma_i(X))^p \text{ s. t. } P_\Omega(X) = P_\Omega(M), \tag{2.3}$$

where $A \in \mathbf{R}^{r\times m}$, $B \in \mathbf{R}^{r\times n}$, $AA^\top = I_{r\times r}$, $BB^\top = I_{r\times r}$, and $0 < p \le 1$.

2.2 Solving TSPN-MC

Since (2.3) is a non-convex optimization problem, it cannot be solved directly by the usual method. To this end, we first put our hand to transform the model (2.3).

Let $F(\sigma(X)) = \sum_{i=1}^{\min(m,n)}(1-\sigma_i(B^\top A))(\sigma_i(X))^p$, then its derivative with respect to $\sigma(X)$ is

$$\nabla F(\sigma(X)) = \sum_{i=1}^{\min(m,n)} p(1-\sigma_i(B^\top A))(\sigma_i(X))^{p-1}. \tag{2.4}$$

Therefore, we can obtain that the first-order Taylor expansion of $F(\sigma(X))$:

$$\begin{aligned}F(\sigma(X)) &= F(\sigma(X_k)) + \langle \nabla F(\sigma(X_k)), \sigma(X) - \sigma(X_k)\rangle \\ &= \nabla F(\sigma(X_k)) \cdot \sigma(X) \\ &= \sum_{i=1}^{\min(m,n)} p(1-\sigma_i(B^\top A))(\sigma_i(X_k))^{p-1} \cdot \sigma_i(X).\end{aligned} \tag{2.5}$$

Let $\omega_i = p(1-\sigma_i(B^\top A))(\sigma_i(X_k))^{p-1}$, then $F(\sigma(X)) = \sum_{i=1}^{\min(m,n)} \omega_i\sigma_i(X)$, where $W := \{\omega_i\}_{i=1}^{\min(m,n)}$ is a no decreasing weight sequence. As a result, the model (2.3) becomes a model based on weighted nuclear norm:

$$\min_X \sum_{i=1}^{\min(m,n)} \omega_i\sigma_i \text{ s. t. } P_\Omega(X) = P_\Omega(M). \tag{2.6}$$

Consequently, we can use ADMM and divide into two steps to solve the convex optimization model (2.6):

- **Step 1**: Initialize $X_1 = M$, and calculate $X_s = U_s\Delta_s V_s^\top$ in the $(s+1)$-th iteration. Then, A_s and B_s are calculated from U_s and V_s;
- **Step 2**: Fix A_s and B_s, and calculate the weight $W = \{\omega_i\}_{i=1}^r$ of K-th iteration. Then, the ADMM algorithm is used to solve (2.6).

We sort out the specific solution process, and obtain the following Algorithm 1.

Algorithm 1 Two steps iterative algorithm for TSPN-MC

```
Require: Observation matrix $M$, coordinate set $\Omega$, and threshold $\varepsilon_0$.
1: Initialization: $X_1 = M$.
2: In the $(s+1)$-th iteration:
3:     Step 1: Calculate singular value decomposition of $X_s$: $[U_s, \Delta_s, V_s] = SVD(X_s)$,
       where $U_s = (\mu_1, \dots, \mu_m) \in \mathbf{R}^{m \times m}$, $V_s = (\nu_1, \dots, \nu_n) \in \mathbf{R}^{n \times n}$. Compute $A_s$ and $B_s$ as
       $A_s = (\mu_1, \dots, \mu_r)^\top$ and $B_s = (\nu_1, \dots, \nu_r)^\top$.
4:     Step 2: Solve the optimization sub-problem (2.6), and obtain $X_{s+1}$.
5: Until $\|X_{s+1} - X_s\|_F \le \varepsilon_0$.
Ensure: Recovery matrix $X$.
```

The process of using ADMM to solve (2.6) is described in detail below. We first introduce the variable N and rewrite the model (2.6) as

$$\min_X \sum_{i=1}^{\min(m,n)} \omega_i \sigma_i \quad \text{s. t. } X + N = M, P_\Omega(N) = 0. \tag{2.7}$$

Then, we construct augmented Lagrangian function of (2.7) as

$$\begin{aligned}\Gamma(X, N, Y, \mu) = & \sum_{i=1}^{\min(m,n)} \omega_i \sigma_i + \langle Y, M - X - N \rangle \\ & + \frac{\mu}{2} \|M - X - N\|_F^2,\end{aligned} \tag{2.8}$$

where Y is a Lagrangian multiplier, and $\mu > 0$ is a penalty factor.

Fixing X, and updating N follows

$$N_{k+1} = \arg\min_N \frac{\mu_k}{2} \left\| N - \left(M - X_k + \frac{1}{\mu_k} Y_k \right) \right\|_F^2 \quad \text{s. t. } \|P_\Omega(N)\|_F^2 = 0, \tag{2.9}$$

and

$$N_{k+1} = M - X_k + \frac{1}{\mu_k} Y_k. \tag{2.10}$$

Then, fixing N and updating X follows

$$\begin{aligned}X_{k+1} = & \arg\min_X \frac{1}{\mu_k} \sum_{i=1}^{\min(m,n)} \omega_i \sigma_i \\ & + \frac{1}{2} \left\| X - \left(M - N_{k+1} + \frac{1}{\mu_k} Y_k \right) \right\|_F^2.\end{aligned} \tag{2.11}$$

Let $Q_X = M - N_{k+1} + \frac{1}{\mu_k} Y_k$, then (2.11) can be solved by the following Lemma 2.2.

Lemma 2.2 *(see* [24]*) Suppose that the rank of matrix Q is r, and its singular value decomposition is $Q = U \Delta V^\top$, $\Delta = \mathrm{Diag}(\sigma_i(Q))$, $1 \le i \le r$, then for any $\tau > 0$, the optimization problem*

$$X^* = \arg\min_X \tau \sum_{i=1}^{\min(m,n)} \omega_i \sigma_i + \frac{1}{2} \|X - Q\|_F^2, \tag{2.12}$$

has optimum solution:

$$X^* = S_{\omega,\tau}(Q) := U(\Delta - \tau \mathrm{diag}(W))_+ V^\top,$$

where $S_{\omega,\tau}(\cdot)$ is weighted singular value shrink operator, $(\cdot)_+ := \max(\cdot, 0)$, and $W = \{\omega_i\}_{i=1}^r$.

Therefore, from Lemma 2.2 it follows the solution of (2.11):

$$X_{k+1} = S_{\omega, \frac{1}{\mu_k}}(Q_X). \tag{2.13}$$

So updating Y and μ achieves

$$Y_{k+1} = Y_k + \mu_k (M - X_{k+1} - N_{k+1}), \tag{2.14}$$

$$\mu_{k+1} = \min(\rho \mu_k, \mu_{\max}), \tag{2.15}$$

where $\rho (> 1)$ is a constant.

Sorting out the above details obtains the following matrix completion regularization (TSPN-MC) algorithm based on truncated Schatten p-norm 2.

Algorithm 2 ADMM solving TSPN-MC

```
Require: Observation matrix $M$, coordinate set $\Omega$, $A_s$, $B_s$, $p, (0 < p \le 1)$, $\mu_0$, $\mu_{\max}$, $\rho$,
    and $\varepsilon$.
Initialization:   $X_0 = M$, $N_0 = \mathbf{0}$, $Y_0 = \mathbf{0}$, $k = 0$.
In the $(k+1)$-th iteration:
    Step 1:   Calculate weight $W$;
    Step 2:   Fix other variables and compute $N$ from (2.10);
    Step 3:   Fix other variables and compute $X$ from (2.13);
    Step 4:   From (2.14) update Lagrange multiplier $Y$, and from (2.15) update
    penalty factor $\mu$;
Until: $\|X_{k+1} - X_k\|_F \le \varepsilon$.
end while: $X = X_{k+1}$.
Ensure: Recovery matrix $X$.
```

As the end of this section, we give the convergence theorem of Algorithm 2. The details of its proof will be given in "Appendix".

Theorem 2.1 $\{X_k\}$ *obtained by Algorithm 2 is convergent.*

3 Low rank sparse decomposition based on truncated Schatten *p*-norm (TSPN-RPCA)

In the previous section, we introduced the TSPN-MC model and presented solution algorithms. In this section, we build a matrix low-rank sparse decomposition model based on truncated Schatten p-norm and its solution algorithm.

We can rewrite the model (1.4) as

$$\min_{X,E} \sum_{i=1}^{\min(m,n)} (1-\sigma_i(B^\top A))(\sigma_i(X))^p + \lambda\|E\|_1 \quad \text{s. t. } M = X + E. \tag{3.1}$$

Similarly, the TSPN-RPCA model can be solved by two-steps iterative method. Because the first step solution process is the same as the first step solution process of model TSPN-MC, we here only introduces the second step solving process with ADMM. Clearly, the Lagrangian functions of (3.1) can be represented as

$$\Gamma(X,E,Y,\mu) = \sum_{i=1}^{\min(m,n)} (1-\sigma_i(B^\top A))(\sigma_i(X))^p + \lambda\|E\|_1 + \langle Y, M-X-E\rangle + \frac{\mu}{2}\|M-X-E\|_F^2. \tag{3.2}$$

Applying ADMM obtains

$$\begin{cases} E_{k+1} = \arg\min_E \frac{\lambda}{\mu_k}\|E\|_1 + \frac{1}{2}\left\|E - \left(M - X_k + \frac{1}{\mu_k}Y_k\right)\right\|_F^2, \\ X_{k+1} = \arg\min_X \frac{1}{\mu_k}\sum_{i=1}^{\min(m,n)} \omega_i\sigma_i + \frac{1}{2}\left\|X - \left(M - E_{k+1} + \frac{1}{\mu_k}Y_k\right)\right\|_F^2, \\ Y_{k+1} = Y_k + \mu_k(M - X_{k+1} - E_{k+1}), \\ \mu_{k+1} = \min(\rho\mu_k, \mu_{\max}), \rho > 1. \end{cases} \tag{3.3}$$

Since the method of solving X_{k+1}, Y_{k+1}, and μ_{k+1} is the same as that of TSPN-MC model, it is not introduced here. For E_{k1}, it can be solved by the following Lemma 3.1.

Lemma 3.1 *(see [28]) For a given matrix $C \in \mathbf{R}^{m\times n}$, and any $\tau > 0$, the solution of optimization problem*

$$\min_E \tau\|E\|_1 + \frac{1}{2}\|E - C\|_F^2 \tag{3.4}$$

is $E^* = \Theta_\tau[c_{ij}]$, where c_{ij} are the elements of matrixC, $\Theta_\tau[\cdot]$ is shrink operator defined by

$$\Theta_\tau[c_{ij}] := \operatorname{sgn}(c_{ij})\max(|c_{ij}| - \tau, 0). \tag{3.5}$$

Furthermore, the algorithm of TSPN-RPCA solved by ADMM can be summarized in following Algorithm 3.

Table 1 Error results of synthetic data recovery ($ps = 0.1$, $p = 0.4$)

Rank (*pr*/*r*)	0.05/20	0.1/40	0.15/60	0.2/80	0.25/100	0.3/120
TSPN-MC	8.8939×10^{-9}	1.8733×10^{-8}	5.3379×10^{-8}	8.7174×10^{-8}	1.0051×10^{-7}	1.6141×10^{-7}
WNN-MC	6.1500×10^{-8}	8.1654×10^{-8}	1.1556×10^{-7}	1.3454×10^{-7}	1.4448×10^{-7}	1.8280×10^{-7}
TNN-MC	7.3857×10^{-5}	1.0165×10^{-4}	1.4211×10^{-4}	1.4620×10^{-4}	1.6815×10^{-4}	1.9028×10^{-4}
NN-MC	8.7735×10^{-5}	1.3309×10^{-4}	1.5073×10^{-4}	1.6434×10^{-4}	1.9248×10^{-4}	2.1538×10^{-4}

Table 2 Error results of synthetic data recovery ($ps = 0.2$, $p = 0.4$)

Rank (*pr*/*r*)	0.05/20	0.1/40	0.15/60	0.2/80	0.25/100	0.3/120
TSPN-MC	5.4488×10^{-8}	7.5490×10^{-8}	1.0778×10^{-7}	1.4423×10^{-7}	2.0575×10^{-7}	2.3046×10^{-7}
WNN-MC	5.9772×10^{-8}	8.3975×10^{-8}	1.2259×10^{-7}	1.5799×10^{-7}	2.1432×10^{-7}	2.5954×10^{-7}
TNN-MC	1.1857×10^{-4}	1.1754×10^{-4}	1.4968×10^{-4}	1.9382×10^{-4}	2.4690×10^{-4}	3.1079×10^{-4}
NN-MC	1.2036×10^{-4}	1.2072×10^{-4}	1.8260×10^{-4}	1.9133×10^{-4}	2.5129×10^{-4}	3.8911×10^{-4}

Table 3 Error results of synthetic data recovery ($ps = 0.3$, $p = 0.3$)

Rank (*pr*/*r*)	0.05/20	0.1/40	0.15/60	0.2/80	0.25/100	0.3/120
TSPN-MC	7.5754×10^{-8}	1.1156×10^{-7}	1.4874×10^{-7}	2.0458×10^{-7}	2.7272×10^{-7}	3.2947×10^{-7}
WNN-MC	8.5654×10^{-8}	1.2263×10^{-7}	1.6203×10^{-7}	2.2088×10^{-7}	2.7792×10^{-7}	3.9079×10^{-7}
TNN-MC	1.2672×10^{-4}	1.7276×10^{-4}	2.1905×10^{-4}	2.5943×10^{-4}	2.5086×10^{-3}	7.3826×10^{-2}
NN-MC	1.4677×10^{-4}	1.7474×10^{-4}	2.2842×10^{-4}	2.8503×10^{-4}	2.5764×10^{-3}	8.4432×10^{-2}

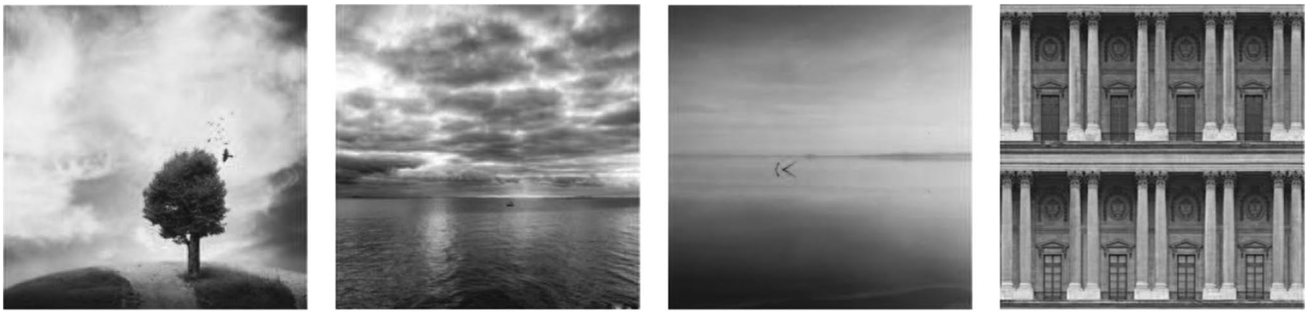
Table 4 Average run time of synthetic data recovery (in seconds/s)

Methods	TSPN-MC	WNN-MC	TNN-MC	NN-MC
Time	66.86	10.43	15.85	3.60

```
Algorithm 3 ADMM solving TSPN-RPCA
Require: $M$, $A_s$, $B_s$, $p, (0 < p \leq 1)$, $\mu_0$, $\mu_{\max}$, $\rho$, $\varepsilon$.
  Initialization:  $X_0 = \mathbf{0}$, $E_0 = \mathbf{0}$, $Y_0 = \mathbf{0}$, $k = 0$.
  In the $(k+1)$-th iteration:
      Step 1: Calculate weight $W$;
      Step 2: Fix other variables and compute $E_{k+1} = \Theta_{\frac{\lambda}{\mu_k}}(M - X_k + \frac{1}{\mu_k}Y_k)$;
      Step 3: Fix other variables and compute $X_{k+1} = S_{\omega, \frac{1}{\mu_k}}(M - E_{k+1} + \frac{1}{\mu_k}Y_k)$;
      Step 4: Update Lagrange multiplier $Y_{k+1} = Y_k + \mu_k(M - X_{k+1} - E_{k+1})$;
      Step 5: Update penalty factor $\mu_{k+1} = \min(\rho\mu_k, \mu_{\max})$;
  Until: $\|X_{k+1} - X_k\|_F \leq \varepsilon$, $\|E_{k+1} - E_k\|_F \leq \varepsilon$.
  and while: $X = X_{k+1}$, $E = E_{k+1}$.
Ensure: Matrix $X$ and $E$.
```

4 Experiments

In this section, the effectiveness of the proposed model and algorithm are verified by a series of experiments on synthetic data and actual images. The simulations of all experiments



(a) Test figure 1 (b) Test figure 2 (c) Test figure 3 (d) Test figure 4

Fig. 1 Experimental images

Table 5 PSNR of test Fig. 1 (dB) ($p = 0.7, r = 8$)

Missing rate	20%	40%	60%	80%	Text masking
TSPN-MC	**40.0583**	**35.2583**	**31.7654**	**27.2461**	**35.3043**
WNN-MC	37.4252	32.9813	29.7691	25.2748	33.1622
TNN-MC	33.3276	31.6651	29.2781	24.9428	31.6663
NN-MC	33.3000	31.5834	28.8288	23.9487	31.2169

Table 6 PSNR of test Fig. 2 (dB) ($p = 0.8, r = 3$)

Missing rate	20%	40%	60%	80%	Text masking
TSPN-MC	**43.7463**	**37.3943**	**32.6607**	**27.2417**	**37.6234**
WNN-MC	43.5335	36.5248	31.8521	27.0981	37.1478
TNN-MC	38.0936	34.9655	31.7271	26.9170	35.0947
NN-MC	38.0575	35.0921	31.4123	25.9986	34.7401

Table 7 PSNR of test Fig. 3 (dB) ($p = 0.9, r = 2$)

Missing rate	20%	40%	60%	80%	Text masking
TSPN-MC	**52.7414**	**48.5528**	**44.0653**	**41.4364**	**50.5905**
WNN-MC	50.2591	45.8884	41.5844	39.0805	48.8553
TNN-MC	42.2772	41.2904	38.9288	35.1127	41.9279
NN-MC	42.2459	40.9211	38.8400	30.7802	41.6521

Table 8 PSNR of test Fig. 4 (dB) ($p = 0.7, r = 4$)

Missing rate	20%	40%	60%	80%	Text masking
TSPN-MC	**37.6958**	**33.1144**	**30.0158**	**26.9119**	**35.3858**
WNN-MC	34.1728	30.0972	27.0882	23.6073	32.7062
TNN-MC	25.3034	24.1951	23.0382	20.0014	25.2862
NN-MC	25.4173	24.0664	22.9235	19.9547	25.2516

are carried out in a MATLAB R2014a environment running on an Intel(R), Core(TM) i3-4150, CPU @ 3.5GHz with 4 GB main memory.

4.1 Application of TSPN-MC in image restoration

Firstly, the experiments results of the methods NN-MC [17], TNN-MC [1], WNN-MC [29], and TSPN-MC on synthetic data are compared, and the feasibility of the proposed algorithm is illustrated. Then, the effectiveness of the four methods is compared by the experimental results on actual image restoration.

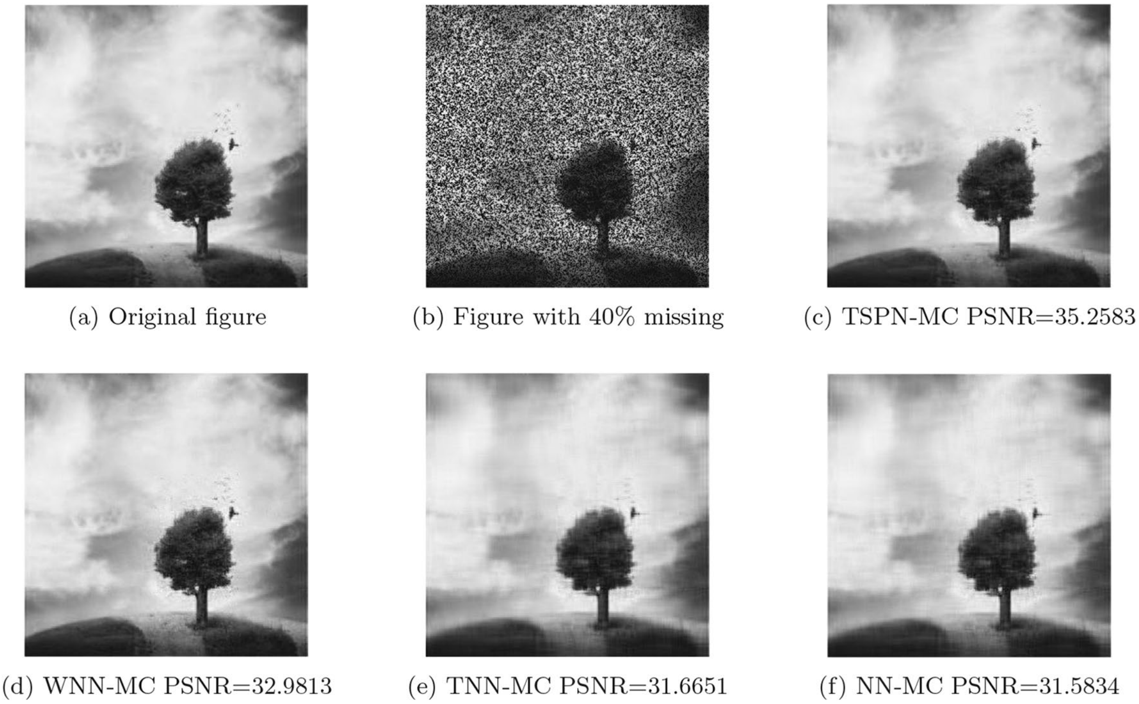
4.1.1 Experimental results on synthetic data

Following the generated method of synthetic data in [29], the generated data obeys the Gaussian distribution $\mathcal{N}(0, 1)$. We set the matrix size as $m = n = 400$, and take $ps = 0.1$, 0.2, and 0.3, respectively. So the number of missing data in the matrix is $(m \times n \times ps)$. Then, we select the value of $pr \in [0.05, 0.3]$ for each ps at an interval of 0.05, and calculate the rank $r = pr \times m$ of the data matrix. In order to illustrate the performance of the model, the matrix recovery error is used as an evaluation index, which is defined by

$$\text{Error} = \frac{\|X - M\|_F}{\|M\|_F}. \tag{4.1}$$

For the data generated by each group of (ps, pr) values, we conduct 10 tests, and then take the average as the final result for that group. For the parameter p in TSPN-MC, depending on the different synthetic data, its specific value will be indicated in the results of the experiment.

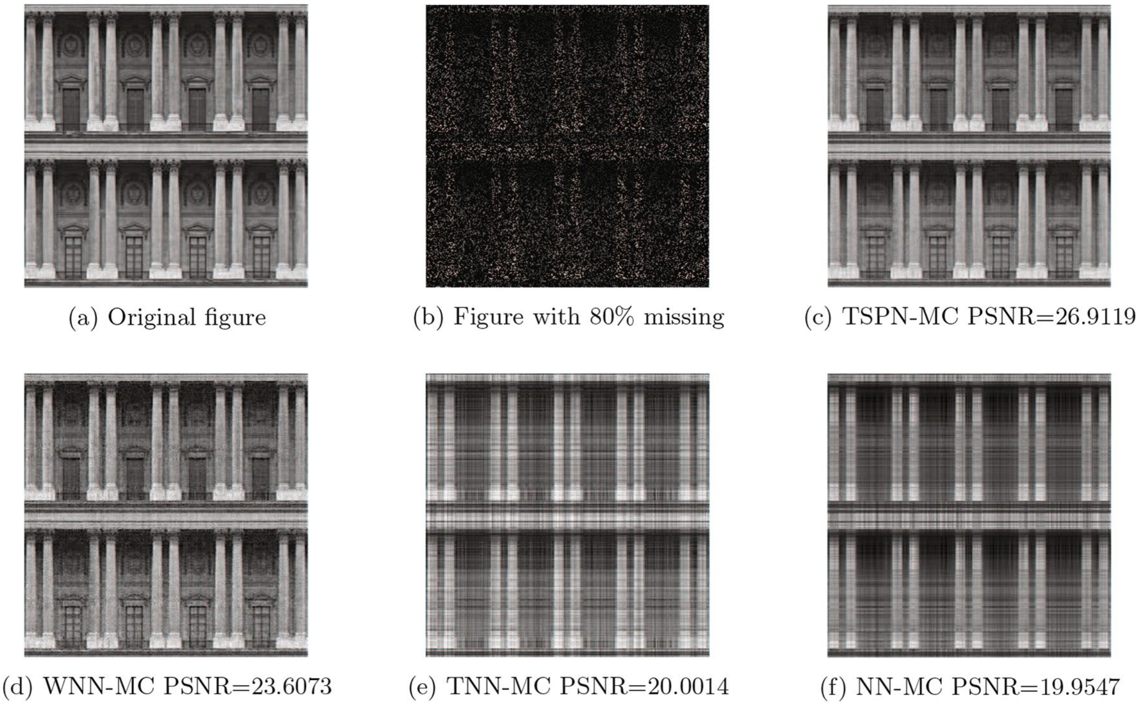
From the error results shown in Tables 1, 2, and 3, we can see that TSPN-MC and WNN-MC are better than the other two methods in recovery. However, the recovery error of



(a) Original figure (b) Figure with 40% missing (c) TSPN-MC PSNR=35.2583

(d) WNN-MC PSNR=32.9813 (e) TNN-MC PSNR=31.6651 (f) NN-MC PSNR=31.5834

Fig. 2 Restoration results of image with 40% missing

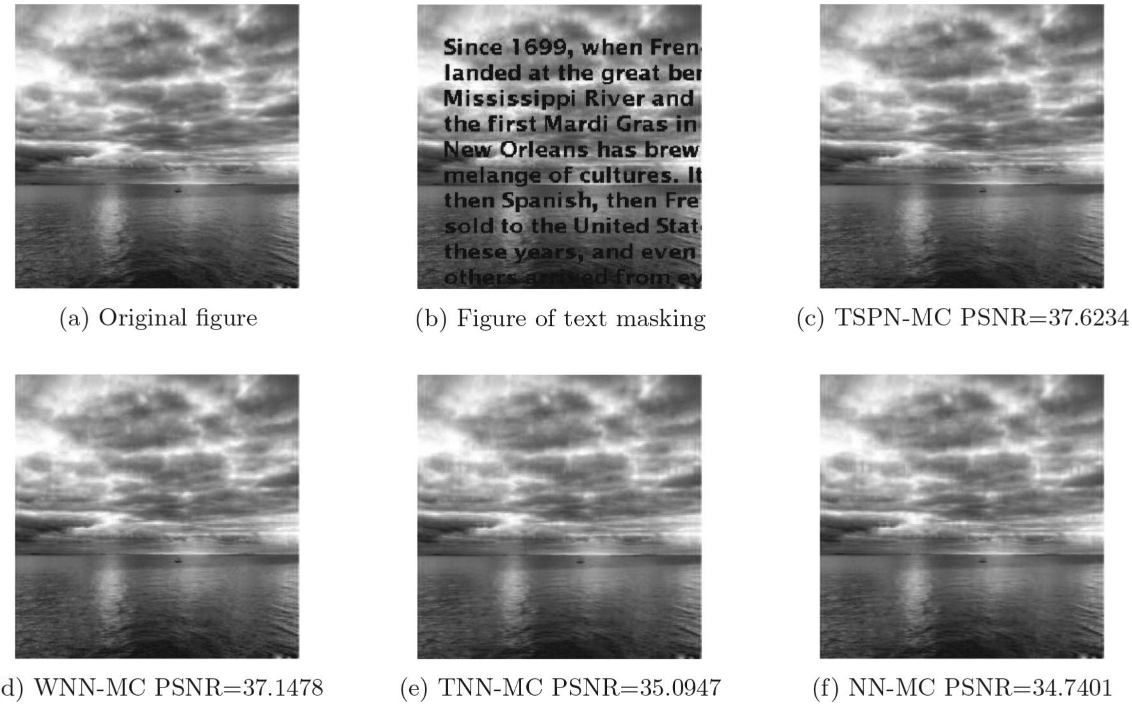


(a) Original figure (b) Figure with 80% missing (c) TSPN-MC PSNR=26.9119

(d) WNN-MC PSNR=23.6073 (e) TNN-MC PSNR=20.0014 (f) NN-MC PSNR=19.9547

Fig. 3 Restoration results of image with 80% missing

TSPN-MC is the smallest, and it has better performance than the other three methods. On the other hand, with the increase of missing data and the rank of synthetic data matrix, the error fluctuation of TSPN-MC is small and has certain stability. Table 4 shows the average time of four methods to run on synthetic data. Because the singular value decomposition



(a) Original figure

(b) Figure of text masking

(c) TSPN-MC PSNR=37.6234

(d) WNN-MC PSNR=37.1478

(e) TNN-MC PSNR=35.0947

(f) NN-MC PSNR=34.7401

Fig. 4 Restoration results of text masking

Table 9 Error result of synthetic data recovery ($m = n = 100$, $r = 0.1n$, $spr = 0.05$)

Algorithm	Rank	Sparsity	Totalerr	LRerr	SPerr	Iterations	Time
TSPN-RPCA	10	500	$\mathbf{2.0644 \times 10^{-12}}$	$\mathbf{2.4056 \times 10^{-12}}$	$\mathbf{1.6351 \times 10^{-12}}$	66	0.7712
LRSD-TNN	10	500	7.1300×10^{-12}	8.2567×10^{-12}	5.7247×10^{-12}	52	0.3626
IALM	10	500	1.4074×10^{-5}	1.1020×10^{-5}	1.6618×10^{-5}	16	0.1006
ALM	10	569	2.2319×10^{-6}	2.0108×10^{-6}	2.4350×10^{-6}	4	0.2740
APG	10	610	1.7455×10^{-3}	1.9036×10^{-3}	1.5622×10^{-3}	65	0.2448

Table 10 Error result of synthetic data recovery ($m = n = 500$, $r = 0.1n$, $spr = 0.05$)

Algorithm	Rank	Sparsity	Totalerr	LRerr	SPerr	Iterations	Time
TSPN-RPCA	50	12500	$\mathbf{4.0898 \times 10^{-12}}$	$\mathbf{5.4730 \times 10^{-}12}$	$\mathbf{2.8245 \times 10^{-}12}$	256	105.2832
LRSD-TNN	50	12500	1.2572×10^{-11}	1.6712×10^{-11}	8.8245×10^{-12}	54	16.7335
IALM	50	12500	1.0831×10^{-5}	9.0085×10^{-6}	1.1883×10^{-5}	15	1.4774
ALM	50	12831	1.9819×10^{-6}	2.6286×10^{-6}	1.3985×10^{-6}	3	6.0768
APG	50	13791	1.7202×10^{-3}	2.1184×10^{-3}	1.3957×10^{-3}	65	6.7533

Table 11 Error result of synthetic data recovery ($m = n = 900$, $r = 0.1n$, $spr = 0.05$)

Algorithm	Rank	Sparsity	Totalerr	LRerr	SPerr	Iterations	Time
TSPN-RPCA	90	40500	$\mathbf{5.3732 \times 10^{-12}}$	$\mathbf{7.2534 \times 10^{-12}}$	$\mathbf{3.6905 \times 10^{-12}}$	184	433.0768
LRSD-TNN	90	40500	1.2608×10^{-11}	1.7063×10^{-11}	8.6050×10^{-12}	57	86.4996
IALM	90	40500	1.0334×10^{-5}	8.5755×10^{-6}	1.1320×10^{-5}	15	7.6584
ALM	90	41113	1.1651×10^{-6}	1.3543×10^{-6}	1.0254×10^{-6}	3	30.8631
APG	90	44267	1.7130×10^{-3}	2.1293×10^{-3}	1.3809×10^{-3}	65	34.0963

was used three times in the computation weight W, so the running time of TSPN-MC is longer than the other three methods.

4.1.2 Experimental results on image restoration

We select four images with the size of 300×300. By comparing the peak signal-to-noise ratio (PSNR), we verify the image restoration effect of TSPN-MC under the condition of missing rate of $20\%, 40\%, 60\%, 80\%$, and text masking, respectively, to illustrate the performance advantages of TSPN-MC. As usual, the PSNR is defined by

$$\mathrm{PSNR} = 20 \log \frac{255}{\sqrt{\frac{1}{mn}\|X - M\|_F^2}}, \tag{4.2}$$

where $X \in \mathbf{R}^{m\times n}, M \in \mathbf{R}^{m\times n}$ represents the original image and the restored image, respectively. Since the parameter settings of different images are different, we will mark the parameter p and truncated rank r of each image in the experimental results. The test images used in this article are shown in Fig. 1.

From PSNR shown in Tables 5, 6, 7 and 8, it can be concluded that the restoration effect of TSPN-MC is better than that of the other three methods, and even with the increase of the missing rate, the PSNR value is still decreasing. On the whole, TSPN-MC's PSNR is higher than other methods, and the recovery effect is better. In order to further illustrate the advantages of the proposed method, Figs. 2, 3, and 4 show the restoration visual effects under different missing or masking conditions, respectively. From these Figures, we can clearly see that the image restored by TSPN-MC is closer to the original image and clearer than that of other methods. To sum up, TSPN-MC model has better recovery performance.

4.2 Application of TSPN-RPCA in background subtraction

4.2.1 Experimental results on synthetic data

In this section, the recovery effects of TSPN-RPCA and LRSD-TNN [30], IALM [21], ALM [21] and APG [31] are compared by generating synthetic data, and the effectiveness of the proposed algorithm is illustrated. First of all, we are generating a synthetic matrix with rank r and Sparsity degree spr: $M_0 = X_0 + E_0$. Here the low rank matrix can be written as $X_0 = HG^\top$, where $H \in \mathbf{R}^{m\times r}$, and $G \in \mathbf{R}^{n\times r}$ satisfies Gaussian distribution $\mathcal{N}(0, 1)$. And the sparse matrix E_0 is a pulse sparse matrix with uniformly distribution on $[-t, t]$, where $t = \max(|X_0|)$, and the number of elements in the matrix is $(spr \times m \times n)$. In the experiment, p is set as 0.2.

In order to illustrate the accuracy of synthetic data recovery, we compare the errors of different methods, including recovery total error Totalerr, low rank matrix error LRerr, and sparse matrix error SPerr. These errors are defined by

$$\mathrm{Totalerr} = \frac{\|M - M_0\|_F}{\|M_0\|_F}, \tag{4.3}$$

$$\mathrm{LRerr} = \frac{\|X - X_0\|_F}{\|X_0\|_F}, \tag{4.4}$$

$$\mathrm{SPerr} = \frac{\|E - E_0\|_F}{\|E_0\|_F}, \tag{4.5}$$

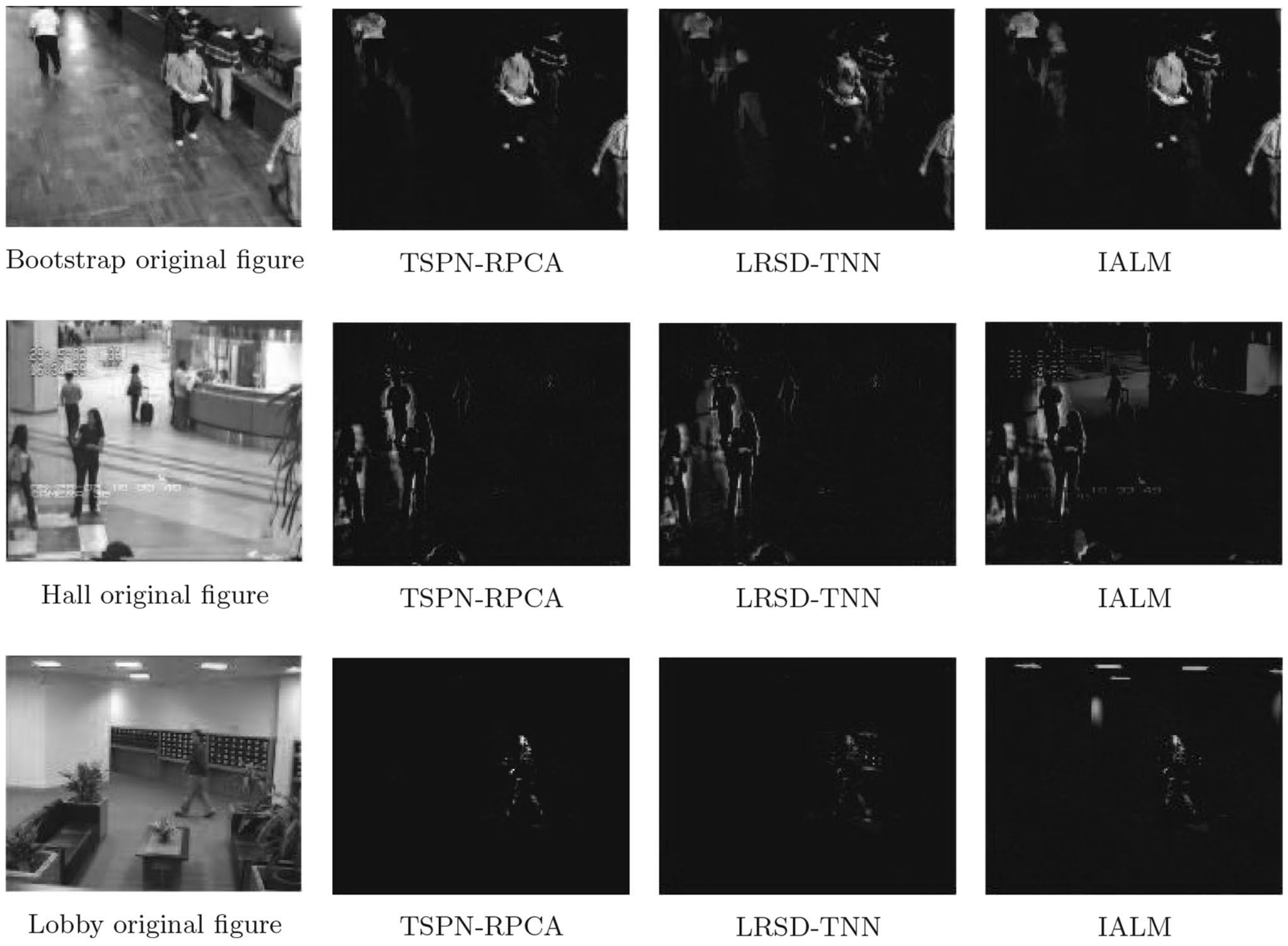
respectively, where M, X, and E are restored matrices.

Table 12 Error result of synthetic data recovery ($m = n = 500, r = 0.1n, spr = 0.1$)

Algorithm	Rank	Sparsity	Totalerr	LRerr	SPerr	Iterations	Time
TSPN-RPCA	50	25000	$\mathbf{8.0848 \times 10^{-12}}$	$\mathbf{1.2357 \times 10^{-11}}$	$\mathbf{5.5165 \times 10^{-12}}$	228	95.3320
LRSD-TNN	50	25000	1.5299×10^{-11}	2.3283×10^{-11}	1.0527×10^{-11}	67	19.1362
IALM	50	25000	1.4438×10^{-5}	1.5334×10^{-5}	1.4062×10^{-5}	17	1.6279
ALM	50	26399	5.5170×10^{-6}	8.4826×10^{-6}	3.7189×10^{-6}	3	8.3070
APG	50	27714	1.5834×10^{-3}	2.0695×10^{-3}	1.3404×10^{-3}	67	6.6725

Table 13 Error result of synthetic data recovery ($m = n = 500, r = 0.05n, spr = 0.05$)

Algorithm	Rank	Sparsity	Totalerr	LRerr	SPerr	Iterations	Time
TSPN-RPCA	25	12500	$\mathbf{4.2025 \times 10^{-12}}$	$\mathbf{5.6113 \times 10^{-12}}$	$\mathbf{2.6757 \times 10^{-12}}$	70	29.9138
LRSD-TNN	25	12500	8.0121×10^{-12}	1.0706×10^{-11}	5.0889×10^{-12}	43	14.0390
IALM	25	12500	4.9002×10^{-6}	3.2649×10^{-6}	5.8429×10^{-6}	12	0.8867
ALM	25	12500	4.4272×10^{-7}	5.5029×10^{-7}	3.3888×10^{-7}	3	3.8708
APG	25	12538	1.8928×10^{-3}	2.0150×10^{-3}	1.7942×10^{-3}	64	6.5124


Bootstrap original figure TSPN-RPCA LRSD-TNN IALM

Hall original figure TSPN-RPCA LRSD-TNN IALM

Lobby original figure TSPN-RPCA LRSD-TNN IALM

Fig. 5 Different video front and rear scene separation effect diagram (from top to bottom is Bootstrap, Hall, Lobby)

We first fix the rank of matrix as $r = 0.1n$, sparsity degree as $spr = 0.05$, and three sets of data of $m = n = 100, 500, 900$ are generated respectively. The matrix recovery experiments are then carried out in five methods, and the results are shown in Tables 9, 10, and 11. By analyzing the results in the Tables, we know that the error of TSPN-RPCA is the smallest, the accuracy is higher than the other four methods, and the original matrix can be recovered more accurately, and the rank and sparsity of the restored matrix are the same as those of the initial matrix. Even with the increase of matrix dimension, the recovery error of TSPN-RPCA can be maintained at a stable level. Because TSPN-RPCA involves matrix singular value decomposition many times, the time consumption is larger than other methods, especially when the matrix dimension is larger.

On the other hand, we fix matrix size as $m = n = 500$. When setting $r = 0.1n, spr = 0.1$ and $r = 0.05n, spr = 0.05$, we respectively illustrate the advantages of the proposed method. The results are shown in Tables 12 and 13. By comparing the results of Tables 10 and 12, we find that the recovery accuracy of TSPN-RPCA decreases when the sparsity of matrix is reduced, but it is still superior to other methods. In addition, when the rank of matrix decreases, comparison Tables 10 and 13 finds that the recovery accuracy of TSPN-RPCA does not fluctuate much, and time and the number of iterations are obviously reduced. In a word, TSPN-RPCA has better recovery effect and better performance than other methods.

4.2.2 Experimental results on background subtraction

In this subsection, the background subtraction experiment is carried out through the actual video, and the superior performance of the proposed method is further verified by the comparative experiment of LRSD-TNN, IALM, and TSPN-RPCA.

For a given video, each frame can be regarded as a column of elements of the matrix. We chose three different videos [30]: Bootstrap, Hall, and Lobby, respectively. We select 16 frames of each video and convert it into a matrix with $(mn \times 16)$ dimension, where m, n is the dimension size of each video. Due to the limited space of the article, this paper only shows the effect diagram of the separation prospect. It can be seen from the visual diagram Fig. 5 that the moving target in the video can be clearly separated by the proposed method. The separation effect of the other two methods is relatively poor, and the light in the background and the "ghost shadow" formed by the moving target appear. In this experiment, the parameter value of TSPN-RPCA is set as $p = 0.2$, and the truncation rank is taken as $r = 3$.

5 Conclusion

This paper studied matrix completion (TSPN-MC) and matrix low rank sparse decomposition (TSPNR-PCA) based on truncated Schatten p-norm. Combined with the characteristics and advantages of truncated norm and Schatten p-norm, the flexibility of the model and its effectiveness in practical problems are enhanced by adjusting the value of $p(0 < p \leq 1)$. We first used Taylor expansion to transform the proposed TSPN-MC and TSPN-RPCA nonconvex models into convex optimization models. Then, the popular alternating direction multiplier method (ADMM) was used to solve the models. In particular, the convergence of the proposed algorithm is proved. A series of comparative experiments further verified the effectiveness and superiority of the proposed method.

It should be pointed out that in the solution of the models, the singular value of the matrix needs to be decomposed and calculated, which is bound to increase the time cost of the experiment. Therefore, in the future research, we should pay attention to how to find new methods to reduce the time consumption caused by singular value decomposition (SVD). In addition, for the selection of truncated rank, we will consider using adaptive method to determine its value in order to reduce the repetition times of the experiment.

Acknowledgements This work was supported by the National Natural Science Foundation of China under Grant 61933013.

Appendix

The proof of Theorem 2.1 Let $Q_X = M - N_{k+1} + \frac{1}{\mu_k} Y_k$, then in $(k+1)$-th iteration, its singular value decomposition is $[U_k, \Delta_k, V_k] = SVD(Q_X)$. From Lemma 2.2, it follows that $X_{k+1} = U_k \Lambda_k V_k^{\top}$, where $\Lambda_k = \{(\Delta_k - \frac{1}{\mu_k} W)_+\}$. So from (2.14) it follows that

$$\begin{aligned}\|Y_{k+1}\|_F &= \left\|Y_k + \mu_k(M - X_{k+1} - N_{k+1})\right\|_F \\ &= \mu_k \left\| \frac{1}{\mu_k} Y_k + M - X_{k+1} - N_{k+1} \right\|_F \\ &= \mu_k \left\| U_k \Delta_k V_k^{\top} - U_k \Lambda_k V_k^{\top} \right\|_F \\ &= \mu_k \|\Delta_k - \Lambda_k\|_F \\ &\leq \mu_k \left\| \frac{W}{\mu_k} \right\|_F = \|W\|_F,\end{aligned} \tag{5.1}$$

which shows that $\{Y_k\}$ is bounded. Recalling the definition of augmented Lagrangian function (2.8), for the solution of $(k+1)$-th iteration $\{X_{k+1}, N_{k+1}\}$, there is

$$\Gamma(X_{k+1}, N_{k+1}, Y_k, \mu_k) \leq \Gamma(X_k, N_k, Y_k, \mu_k).$$

Noticing $Y_{k+1} = Y_k + \mu_k(M - X_{k+1} - N_{k+1})$, we have

$$\begin{aligned}&\Gamma(X_k, N_k, Y_k, \mu_k) \\ &= \Gamma(X_k, N_k, Y_{k-1}, \mu_{k-1}) + \frac{\mu_k - \mu_{k-1}}{2} \|M - X_k - N_k\|_F^2 \\ &\quad + \langle Y_k - Y_{k-1}, M - X_k - N_k \rangle \\ &= \Gamma(X_k, N_k, Y_{k-1}, \mu_{k-1}) + \frac{\mu_k - \mu_{k-1}}{2} \left\| \frac{1}{\mu_{k-1}} (Y_k - Y_{k-1}) \right\|_F^2 \\ &\quad + \langle Y_k - Y_{k-1}, \frac{1}{\mu_{k-1}} (Y_k - Y_{k-1}) \rangle \\ &= \Gamma(X_k, N_k, Y_{k-1}, \mu_{k-1}) + \frac{\mu_k + \mu_{k-1}}{2\mu_{k-1}^2} \|(Y_k - Y_{k-1})\|_F^2.\end{aligned} \tag{5.2}$$

Therefore,

$$\begin{aligned}&\Gamma(X_{k+1}, N_{k+1}, Y_k, \mu_k) \\ &\leq \Gamma(X_1, N_1, Y_0, \mu_0) + \frac{\mu_k + \mu_{k-1}}{2\mu_{k-1}^2} \|(Y_k - Y_{k-1})\|_F^2.\end{aligned} \tag{5.3}$$

Since $\{Y_k\}$ is bound, and there is $\mu_{k+1} = \min(\rho\mu_k, \mu_{\max})$, $\Gamma(X_{k+1}, N_{k+1}, Y_k, \mu_k)$ is bounded.

And because

$$\begin{aligned}\sum_{i=1}^{\min(m,n)} \omega_i\sigma_i(X_k) = \|X_k\|_{W,*} &= \Gamma(X_k, N_k, Y_{k-1}, \mu_{k-1}) \\ &+ \frac{\mu_{k-1}}{2}\left(\frac{1}{\mu_{k-1}^2}\|Y_{k-1}\|_F^2 - \|M - X_k - N_k + \frac{1}{\mu_{k-1}}\|Y_{k-1}\|_F^2\right) \\ &= \Gamma(X_k, N_k, Y_{k-1}, \mu_{k-1}) - \frac{1}{2\mu_{k-1}}(\|Y_k\|_F^2 - \|Y_{k-1}\|_F^2),\end{aligned} \tag{5.4}$$

$\{X_k\}$ is bounded. From $N_{k+1} = M - X_k + \frac{1}{\mu_k}Y_k$, it can be deduced that $\{N_k\}$ is also bounded. So $\{X_k, N_k, Y_k\}$ has at least one point of accumulation, and

$$\lim_{k\to+\infty} \|M - X_{k+1} - N_{k+1}\|_F = \lim_{k\to+\infty} \frac{1}{\mu_k}\|Y_{k+1} - Y_k\|_F = 0.$$

We prove the convergence of X_k below. Because

$$\begin{cases} X_k = U_{k-1}\Lambda_{k-1}V_{k-1}^{\top}, \\ X_{k+1} = M - N_{k+1} - \frac{1}{\mu_k}(Y_{k+1} - Y_k), \\ N_k = M - X_k - \frac{1}{\mu_{k-1}}(Y_k - Y_{k-1}), \\ N_{k+1} = M - X_k + \frac{1}{\mu_k}Y_k. \end{cases} \tag{5.5}$$

we have

$$\begin{aligned}&\lim_{k\to+\infty} \|X_{k+1} - X_k\|_F \\ &= \lim_{k\to+\infty} \left\|M - N_{k+1} - \frac{1}{\mu_k}(Y_{k+1} - Y_k) - X_k\right\|_F \\ &= \lim_{k\to+\infty} \left\|M - N_{k+1} - \frac{1}{\mu_k}(Y_{k+1} - Y_k) - X_k + \left(N_k + \frac{1}{\mu_{k-1}}Y_{k-1}\right) - \left(N_k + \frac{1}{\mu_{k-1}}Y_{k-1}\right)\right\|_F \\ &\le \lim_{k\to+\infty} \left\|M + \frac{1}{\mu_{k-1}}Y_{k-1} - N_k - X_k\right\|_F \\ &\quad + \left\|N_k - N_{k+1} + \frac{1}{\mu_k}(Y_k - Y_{k+1}) - \frac{1}{\mu_{k-1}}Y_{k-1})\right\|_F \\ &\le \lim_{k\to+\infty} \|\Delta_{k-1} - \Lambda_{k-1}\|_F + \|N_k - N_{k+1}\|_F \\ &\quad + \left\|\frac{1}{\mu_k}(Y_k - Y_{k+1}) - \frac{1}{\mu_{k-1}}Y_{k-1})\right\|_F \\ &= \lim_{k\to+\infty} \|\Delta_{k-1} - \Lambda_{k-1}\|_F + \left\|\frac{1}{\mu_{k-1}}(Y_{k-1} - Y_k) - \frac{1}{\mu_k}Y_k)\right\|_F + \left\|\frac{1}{\mu_k}(Y_k - Y_{k+1}) - \frac{1}{\mu_{k-1}}Y_{k-1})\right\|_F \\ &= 0.\end{aligned} \tag{5.6}$$

This completes the proof of Theorem 2.1. □

References

1. Hu Y, Zhang D, Ye J, Li X, He X (2013) Fast and accurate matrix completion via truncated nuclear norm regularization. IEEE Trans Pattern Anal Mach Intell 35(9):2117–2130
2. Fan J, Chow TW (2017) Matrix completion by least-square, low-rank, and sparse self-representations. Pattern Recognit 71:290–305
3. Yu Y, Peng J, Yue S (2018) A new nonconvex approach to low-rank matrix completion with application to image inpainting. Multidimens Syst Signal Process 1:1–30
4. Qian W, Cao F (2019) Adaptive algorithms for low-rank and sparse matrix recovery with truncated nuclear norm. Int J Mach Learn Cybern 10(6):1341–1355
5. Ji H, Liu C, Shen Z, Xu Y (2010) Robust video denoising using low rank matrix completion. In: Proc IEEE Conf Comp Vis Pattern Recognit (CVPR), pp 1791–1798
6. Nie F, Wang H, Cai X, Huang H, Ding C (2012). Robust matrix completion via joint Schatten *p*-norm and l_p-norm minimization. In: Proc Int Comp Data Mining (ICDM), pp 566–574
7. Lu J, Liang G, Sun J, Bi J (2016) A sparse interactive model for matrix completion with side information. In: Proc Adv Neural Inf Proc Syst (NIPS), pp 4071–4079
8. Bennett J, Lanning S (2007) The netflix prize. Proc KDD Cup Workshop 2007:35–38
9. Otazo R, Candès E, Sodickson DK (2015) Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components. Magn Res Med 73(3):1125–1136
10. Ravishankar S, Moore BE, Nadakuditi RR, Fessler JA (2017) Low-rank and adaptive sparse signal (LASSI) models for highly accelerated dynamic imaging. IEEE Trans Med Imaging 36(5):1116–1128
11. Javed S, Bouwmans T, Shah M, Jung SK (2017) Moving object detection on RGB-D videos using graph regularized spatiotemporal RPCA. In: Proc Int Conf Image Anal Proc (ICIAP), pp 230–241
12. Dang C, Radha H (2015) RPCA-KFE: key frame extraction for video using robust principal component analysis. IEEE Trans Image Process 24(11):3742–3753
13. Jin M, Li R, Jiang J, Qin B (2017) Extracting contrast-filled vessels in X-ray angiography by graduated RPCA with motion coherency constraint. Pattern Recognit 63:653–666
14. Werner R, Wilmsy M, Cheng B, Forkert ND (2016) Beyond cost function masking: RPCA-based non-linear registration in the context of VLSM. In: Proc Int Workshop Pattern Recognit Neuroimaging (PRNI), pp 1–4
15. Song Z, Cui K, Cheng G (2019) Image set face recognition based on extended low rank recovery and collaborative representation. Int J Mach Learn Cybern. https://doi.org/10.1007/s13042-019-00941-6
16. Fazel M (2002) Matrix rank minimization with applications. Doctoral dissertation, PhD thesis, Stanford University
17. Candès EJ, Recht B (2009) Exact matrix completion via convex optimization. Found Comput Math 9(6):717–772
18. Oymak S, Mohan K, Fazel M, Hassibi B (2011) A simplified approach to recovery conditions for low rank matrices. In: Proc Int Symp Inf Theory (ISIT), pp 2318–2322
19. Li G, Yan Z (2019) Reconstruction of sparse signals via neurodynamic optimization. Int J Mach Learn Cybern 10(1):15–26
20. Lin Z, Ganesh A, Wright J, Wu L, Chen M, Ma Y (2009) Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix. Comput Adv Multisens Adapt Process 61(6):1–18
21. Lin Z, Chen M, Ma Y (2010) The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. arXiv preprint arXiv: 1009.5055
22. Toh KC, Yun S (2010) An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. Pac J Optim 6(3):615–640
23. Yang AY, Sastry SS, Ganesh A, Ma Y (2010) Fast ℓ_1-minimization algorithms and an application in robust face recognition: a review. In: Proc 17th Int Conf Image Process, pp 1849–1852
24. Gu S, Zhang L, Zuo W, Feng X (2014) Weighted nuclear norm minimization with application to image denoising. In: Proc Conf Comp Vis Pattern Recognit, pp 2862–2869
25. Feng L, Sun H, Sun Q, Xia G (2016) Image compressive sensing via Truncated Schatten-*p* Norm regularization. Signal Proc Image Commun 47:28–41
26. Chen B, Sun H, Xia G, Feng L, Li B (2018) Human motion recovery utilizing truncated Schatten *p*-norm and kinematic constraints. Inf Sci 450:89–108
27. Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. Found Trends Mach Learn 3(1):1–122
28. Candès EJ, Li X, Ma Y, Wright J (2011) Robust principal component analysis? J ACM 58(3):1–37
29. Gu S, Xie Q, Meng D, Zuo W, Feng X, Zhang L (2017) Weighted nuclear norm minimization and its applications to low level vision. Int J Comput Vis 121(2):183–208
30. Cao F, Chen J, Ye H, Zhao J, Zhou Z (2017) Recovering low-rank and sparse matrix based on the truncated nuclear norm. Neural Netw 85:10–20
31. Beck A, Teboulle M (2009) A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J Imaging Sci 2(1):183–202

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.