Applied Mathematical Modelling 48 (2017) 410–422

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

A non-convex tensor rank approximation for tensor completion

Teng-Yu Ji, Ting-Zhu Huang*, Xi-Le Zhao*, Tian-Hui Ma, Liang-Jian Deng

School of Mathematical Sciences/Resrarch Center for Image and Vision Computing, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, PR China

ARTICLE INFO

Article history:
Received 19 September 2016
Revised 22 March 2017
Accepted 4 April 2017
Available online 8 April 2017

Keywords:
Tensor completion
Low-rank approximation
Non-convex optimization
Alternating direction method of multipliers

ABSTRACT

Low-rankness has been widely exploited for the tensor completion problem. Recent advances have suggested that the tensor nuclear norm often leads to a promising approximation for the tensor rank. It treats the singular values equally to pursue the convexity of the objective function, while the singular values for the practical images have clear physical meanings with different importance and should be treated differently. In this paper, we propose a non-convex logDet function as a smooth approximation for tensor rank instead of the convex tensor nuclear norm and introduce it into the low-rank tensor completion problem. An alternating direction method of multiplier (ADMM)-based method is developed to solve the problem. Experimental results have shown that the proposed method can significantly outperform existing state-of-the-art nuclear norm-based methods for tensor completion.

1. Introduction

In computer vision and image processing, there are many missing data estimation problems, such as image inpainting [1–6], video inpainting [7,8], remote-sensing image reconstruction [9–11], and scan completion [12]. Tensors are high-order generalizations of vectors and matrices [13], thus the above-mentioned problems can be formulated as the tensor completion problem. The missing data can be estimated by local and global approaches [14,15]. Local approaches are used to estimate the missing entries based on the relationship between the adjacent entries, e.g., partial differential equations (PDEs) [1,7] and belief propagation [3]. However, in practice the missing entries depend on the entries located at distinct regions. The global property, also called the non-local self-similarity, has achieved significant success in tensor completion [14–19].

For matrix completion, one popular method is to minimize the matrix rank, which can effectively estimate the missing data by exploiting both the local and global information. However, minimizing the matrix rank is a non-convex and NP-hard problem that may prevent one from obtaining a global minimizer [20]. To tackle this difficulty, the nuclear norm of matrices has been adopted to approximate the rank of matrices [21–25]. Under certain conditions [22–24], the resulting convex optimization problem is equivalent to the rank minimization problem. Beyond the nuclear norm, many other methods, e.g. low-rank matrix factorization [20,26], logDet function [27], weighted nuclear norm minimization (WNNM) [28], and minimax concave penalty (MCP) function [29,30], have been proposed to handle the matrix rank minimization problem.

* Corresponding authors.
E-mail addresses: tengyu_j66@126.com (T.-Y. Ji), tingzhuhuang@126.com, tzhuang@uestc.edu.cn (T.-Z. Huang), xlzhao122003@163.com (X.-L. Zhao), nkmth0307@126.com (T.-H. Ma), liangjian1987112@126.com (L.-J. Deng).

http://dx.doi.org/10.1016/j.apm.2017.04.002
0307-904X/

These methods are all non-convex and work better than those based on the nuclear norm. Despite the theoretical soundness, we conjecture that non-convex approximation for tensor rank could lead to better results, as has been observed in the above-mentioned studies. This is one of the motivation for considering a non-convex function in this paper.

The tensor is the multi-dimensional generalization of the matrix, but is more complex than the matrix. Many definitions of tensors cannot be generalized from the corresponding definitions of matrices, such as the definition of the product between a tensor and another tensor or matrix, the definition of the rank for tensors, and other corresponding operators (e.g., singular value decomposition), etc. To take full advantage of the information in a multi-dimensional tensor, the corresponding definitions for tensors have been defined, e.g., CANDECOMP/PARAFAC (CP) decomposition [31,32], Tucker decomposition [33–35], tensor matrix product, and so on. Based on the CP and Tucker decompositions, two definitions for the tensor rank named CP-rank and n-rank are used for the tensor completion problem. Both decomposition methods require a reliable estimation of the underlying tensor rank. However, it is difficult to estimate the tensor rank accurately.

In addition to CP and Tucker decompositions, there are many other approaches to study the tensor rank [14–18,36–38]. One important technique for tensor completion is to take the unfolding matrices of the tensor into consideration. For example, one can take the ranks of N unfolding matrices for an Nth-order tensor into consideration [14–17,36]. In this case, Liu et al. [14] established a novel definition of the tensor nuclear norm for the tensor n-rank. They proposed the alternating direction method of multipliers (ADMM) to solve the resulting nuclear norm minimization problem. Gandy et al. [15] recently proposed a noise reduction tensor completion model based on the tensor nuclear norm and two algorithms for the problem. One is based on the Douglas–Rachford splitting technique and its dual variant, and the other is based on ADMM.

The matrix nuclear norm-based problem is convex. It treats each singular value equally. However, for the practical image, the singular values have clear physical meanings with different importance and should be treated differently [28]. For instance, the column (or row) vectors of a low-rank matrix often lie in a low-dimensional subspace; the larger singular values are generally associated with the major projection orientations, and thus, they would be better if shrunk less to preserve the major data components. Clearly, the nuclear norm-based methods fail to take advantage of such prior knowledge. The definition of the tensor nuclear norm is the weight sum of the nuclear norm of N unfolding matrices. We can see that the tensor nuclear norm does not consider the prior knowledge. This is the second motivation of this paper.

To overcome the disadvantages of the tensor nuclear norm, the larger singular values should be shrunk less. One way is to assign smaller weights to the larger singular values, as in WNNM [28]. However, this would introduce more parameters, which may lead the method being less robust. In this paper, we use the logarithm of singular values to pursue the importance of the larger singular values. The logarithm operator can make a larger scalar decrease more rapidly than a smaller one; the larger singular value is shrunk less using a smaller weight. In [27], it has been shown that for a symmetric positive semidefinite matrix, the logDet function, which is the sum of the logarithm of singular values, depicts the matrix rank more accurately than the nuclear norm. It works well in many applications, such as matrix completion [39], shadow removal [40], optical flow estimation [41], compressed sensing [42], and image restoration [43,44]. To the best of the authors' knowledge, there is no similar function for tensors. Inspired by these observations, we propose a non-convex surrogate function for tensor rank based on the logDet function instead of using the convex nuclear norm. The non-convex function is defined as the weighted sum of the logDet for each mode unfolding of the tensor. The definition considers the global correlation for each mode, which leads to better performance. Although the resulting problem is non-convex, which means it is difficult to find a global minimizer, we introduce some auxiliary variables and design an ADMM-based method to efficiently solve the proposed model. Extensive experiments are given to demonstrate the effectiveness of the proposed method.

The outline of this paper is as follows. We first review the notations of tensors and related work in Section 2. In Section 3, we describe our motivation details, then we propose the non-convex approximation for tensor rank and establish the tensor completion model based on this definition. In Section 4, an ADMM-based method is proposed. In Section 5, experimental results are reported. Finally, some conclusions are given in Section 6.

2. Notations and related work

2.1. Notations

Following [13], we use non-bold low-case letters for scalars, e.g., x, bold low-case letters for vectors, e.g., $\mathbf{x}$, non-bold upper-case letters for matrices, e.g., X, and bold upper-case letters for tensors, e.g., $\mathbf{X}$. An Nth-order tensor is defined as $\mathbf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, and $x_{i_1,\ldots,i_N}$ is its $(i_1, \ldots, i_N)$th component.

The inner product of two tensors $\mathbf{X}$ and $\mathbf{Y}$ is defined as $\langle \mathbf{X}, \mathbf{Y} \rangle := \sum_{i_1,i_2,\ldots,i_N} x_{i_1,\ldots,i_N} y_{i_1,\ldots,i_N}$. The Frobenius norm is then defined as:

$$\|\mathbf{X}\|_F := \sqrt{\langle \mathbf{X}, \mathbf{X} \rangle}.$$

Fibers are the higher-order analogue of matrix rows and columns. A fiber is defined by fixing every index but one. Fig. 1 shows the mode-n fibers for a third-order tensor. The mode-n unfolding of a tensor $\mathbf{X}$ is denoted as $X_{(n)} = \text{unfold}_n(\mathbf{X}) \in \mathbb{R}^{I_n \times \Pi_{i \neq n} I_i}$, which is a matrix with columns being the mode-n fibers of $\mathbf{X}$ in the lexicographical order. The inverse operator of unfolding is denoted as "fold", i.e., $\mathbf{X} = \text{fold}_n(X_{(n)})$. The n-rank of an Nth-order tensor $\mathbf{X}$, denoted as $\text{rank}_n(\mathbf{X})$, is the rank of $X_{(n)}$, and the rank of $\mathbf{X}$ based on n-rank is defined as an array: $\text{rank}(\mathbf{X}) = (\text{rank}(X_{(1)}), \ldots, \text{rank}(X_{(N)}))$. The tensor $\mathbf{X}$ is

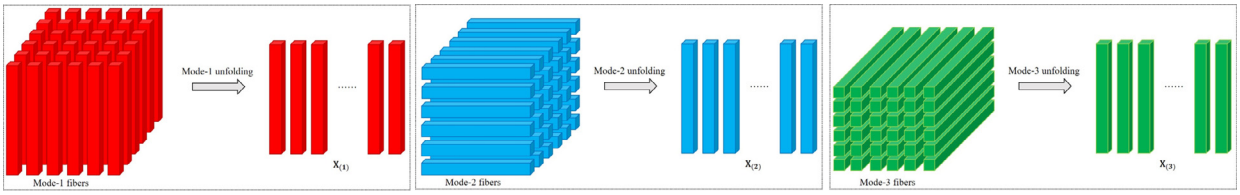

Fig. 1. Mode-n unfoldings of a third-order tensor.

low-rank, if $X_{(n)}$ is low-rank for all n. This definition relates to the Tucker decomposition [33]. Please refer to [13] for its extensive overview.

2.2. Related work

In this section, we consider the problem estimating the missing entries using low-rank prior of the underlying tensor. Given an observed tensor $\mathbf{B}$ whose entries in the index set Ω are known but the remaining are missing, find the tensor $\mathbf{X}$ from the following optimization problem:

$$\begin{aligned} \min_{\mathbf{X}} \quad & \text{rank}(\mathbf{X}) \\ \text{s.t.} \quad & \mathbf{X}_\Omega = \mathbf{B}_\Omega. \end{aligned} \tag{1}$$

Recently, Liu et al. [14] developed the definition for the tensor nuclear norm as a surrogate for the tensor n-rank, and they solved the following convex problem as a variant of (1):

$$\begin{aligned} \min_{\mathbf{X}} \quad & \|\mathbf{X}\|_* := \sum_{n=1}^{N} \alpha_n \|X_{(n)}\|_* \\ \text{s.t.} \quad & \mathbf{X}_\Omega = \mathbf{B}_\Omega, \end{aligned} \tag{2}$$

where α_n are constants satisfying $\alpha_n \geq 0$ and $\sum_{n=1}^{N} \alpha_n = 1$. One of the algorithms for solving the problem (2) in [14] is based on ADMM called HaLRTC. Recently, Gandy et al. in [15] studied noisy tensor completion based on nuclear norm and proposed a Douglas-Rachford splitting method.

3. Proposed model

In this section, we explain the motivation and propose the non-convex but smooth surrogate function for the tensor n-rank by using the logDet function rather than the nuclear norm. We also establish the completion model based on this definition.

Our motivation for this model can be divided into three parts. (1) Although the non-convex functions are more difficult, many works have paid more attention to them because of their effectiveness. The non-convex l_p-norm ($0 < p < 1$) [45,46] performs much better than the l_1-norm in approximating the l_0-norm; the low-rank matrix factorization [16,20,26] and MCP function [18,29,30] also perform better than the convex nuclear norm. Therefore, we conjecture that a non-convex approximation for the tensor rank could lead to better results.

(2) The n-rank of a tensor denotes the correlation with respect to the corresponding dimension. A simple description for a third-order tensor unfolding is shown in Fig. 1. We can see that the rank of the mode-n unfolding matrix $X_{(n)}$ denotes the linear correlation between the mode-n fibers. In [14], the rank of $X_{(n)}$ is approximated by the nuclear norm of $X_{(n)}$. Fortunately, the nuclear norm is the tightest convex approximation, and the nuclear norm minimization problem can be easily solved by the singular value thresholding (SVT) [21], which is theoretically sound [47]. The nuclear norm-based methods treat each singular value equally. However, the larger singular values are generally associated with the major information, and hence they should better be shrunk less to preserve the major data information. Clearly, tensor nuclear norm-based completion methods fail to preserve the major data components. (3) To preserve the major data components, the singular values should be treated differently. One way is to use the weighted nuclear norm, but it has to introduce more parameters that may lead the method to be less robust. The logarithm operator can make a larger scalar decrease more rapidly than a smaller one, i.e., the larger singular values are shrunk less using a smaller weight. Given two positive scalars: one is smaller, denoted as s, and the other is larger, denoted as l, $\log(q) = W_q * q$, where $q \in \{s, l\}$. We can see that the weight $|W_l| < |W_s|$, because of $l > s$ when $l, s \notin (1, e)$ where e is the Euler's number (The interval $(1, e)$ is narrower than the range of singular values, thus we can ignore it). Therefore, the larger singular values are shrunk less, and can be treated as more important. In addition, for scalars, the effective of the logDet can be seen in Fig. 2, where we can see that the logDet function can better approximate the rank than the nuclear norm. For vectors, the logDet function leads to the well-known reweighted l_1-norm. In [48] it was shown that the reweighted l_1-norm achieved better performance than the l_1-norm. For matrices, the logDet function also performs better than the nuclear norm and often leads to superior image recovery results [41,42,49].

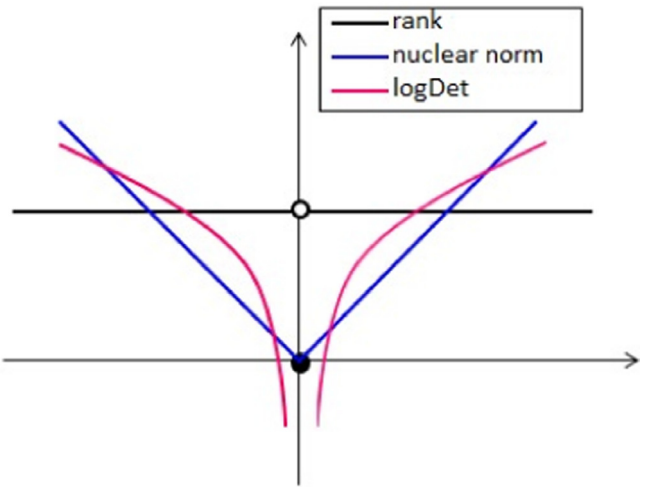

Fig. 2. Comparison of rank, the nuclear norm and the logDet function for scalars.

It has been shown in [27] that for a symmetric positive semidefinite matrix $X \in \mathbb{R}^{n\times n}$, there is a better surrogate, i.e.,

$$E(X,\varepsilon) := \log\ \det(X+\varepsilon I), \tag{3}$$

where $\varepsilon > 0$ is a small constant. Note that the energy function $E(X, \varepsilon)$ is not convex. However, since it is smooth on the positive definite cone, it can be minimized using a local minimization method [27].

In this paper, we propose the smooth surrogate of the tensor n-rank as follows:

$$\text{logDet}(\mathbf{X},\varepsilon) := \sum_{n=1}^{N}\alpha_n \log\ \det((X_{(n)}X_{(n)}^T)^{1/2}+\varepsilon_n I_n), \tag{4}$$

where $\varepsilon = (\varepsilon_1,\ldots,\varepsilon_N)^T$, and α_n are constants satisfying $\alpha_n \geq 0$ and $\sum_{n=1}^{N}\alpha_n = 1$. In essence, the logDet function of a tensor is a linear combination of the logDet function of all matrices unfolded along each mode. Note that when the mode number N is equal to two, the definition of the logDet function (4) is consistent with (3).

Under this definition, we propose the following model for the tensor completion problem:

$$\begin{aligned} \min_{\mathbf{X}}\quad & \sum_{n=1}^{N}\alpha_n L(X_{(n)}) \\ \text{s.t.}\quad & \mathbf{X}_\Omega = \mathbf{B}_\Omega, \end{aligned} \tag{5}$$

where $L(X_{(n)}) = \log\ \det((X_{(n)}X_{(n)}^T)^{1/2}+\varepsilon_n I_n)$ and $\mathbf{B}_\Omega$ denotes the tensor copying the entries from $\mathbf{X}$ in the index set Ω and letting the remaining entries be zeros. For an arbitrarily matrix $X_{(n)} \in \mathbb{R}^{I_n\times\prod_{i\neq n}I_i}$, that is neither square nor positive semidefinite, we consider the $L(X_{(n)})$ function:

$$\begin{aligned} L(X_{(n)}) &= \log\ \det((X_{(n)}X_{(n)}^T)^{1/2}+\varepsilon_n I_n) \\ &= \log\ \det(U_n\Sigma_n^{1/2}U_n^{-1}+\varepsilon_n I_n) \\ &= \log\ \det(\Sigma_n^{1/2}+\varepsilon_n I_n) \\ &= \log\prod_{i=1}^{I_n}(\sigma_i(X_{(n)})+\varepsilon_n) \\ &= \sum_{i=1}^{I_n}\log(\sigma_i(X_{(n)})+\varepsilon_n), \end{aligned} \tag{6}$$

where $X_{(n)}X_{(n)}^T = U_n\Sigma_n U_n^{-1}$, and $\Sigma_n^{1/2} = \text{diag}\{\sigma_1(X_{(n)}),\ldots,\sigma_{I_n}(X_{(n)})\}$ is a diagonal matrix whose diagonal elements are the singular values of $X_{(n)}$. Although $\sum_{i=1}^{I_n}\log(\sigma_i(X_{(n)})+\varepsilon_n)$ is non-convex, we can solve it efficiently using a local minimization method. We can see from (6) that our proposed function shrinks the larger singular values less, so that the important components are preserved.

4. Proposed algorithm

In this section, we present the numerical scheme for solving (5). First, use the following first-order Taylor expansion to approximate $L(X_{(n)})$ in (6):

$$\begin{aligned} L_{\omega^k}(X_{(n)}) &\approx \sum_{i=1}^{I_n} \frac{\sigma_i(X_{(n)})}{\sigma_i^k(X_{(n)}) + \varepsilon_n} + \text{constant} \\ &= (\omega^k)^T \sigma + \text{constant}, \end{aligned} \tag{7}$$

where $\sigma_i^k(X_{(n)})$s are the solutions obtained in the kth iteration, $\sigma = (\sigma_1(X_{(n)}), \ldots, \sigma_{I_n}(X_{(n)}))^T$ and $\omega^k = (1/(\sigma_1^k(X_{(n)}) + \varepsilon_n), \ldots, 1/(\sigma_{I_n}^k(X_{(n)}) + \varepsilon_n))^T$. Combining (7) and the following notation

$$\mathcal{X}_{\mathcal{S}}(\mathbf{X}) = \begin{cases} 0, & \text{if } \mathbf{X} \in \mathcal{S}, \\ \infty, & \text{otherwise}, \end{cases}$$

where

$$\mathcal{S} := \{\mathbf{X} \in \mathcal{T}, \quad \mathbf{X}_\Omega = \mathbf{B}_\Omega\},$$

the problem (5) can be rewritten as the following unconstrained problem:

$$\min_{\mathbf{X}} \ \mathcal{X}_{\mathcal{S}}(\mathbf{X}) + \sum_{n=1}^{N} \alpha_n L_{\omega^k}(X_{(n)}). \tag{8}$$

Thus, the problem (8) can be solved efficiently using ADMM [50–56].

Introducing some auxiliary values, the problem (8) can be rewritten as follows:

$$\begin{aligned} \min_{\mathbf{X}, \mathbf{Y}_1, \ldots, \mathbf{Y}_N} \quad & \mathcal{X}_{\mathcal{S}}(\mathbf{X}) + \sum_{n=1}^{N} \alpha_n L_{\omega^k}(Y_{n,(n)}) \\ \text{s.t.} \quad & \mathbf{Y}_1 = \mathbf{X}, \ldots, \mathbf{Y}_N = \mathbf{X}. \end{aligned} \tag{9}$$

For simplicity, we denote the space of the tensors by $\mathcal{T} := \mathbb{R}^{I_1 \times \cdots \times I_N}$, and let $\mathcal{I}_{\mathcal{T}}$ denote the identity operator on $\mathcal{T}$. Define the set $\mathcal{K}$ by the Cartesian product: $\mathcal{K} := \mathcal{T} \times \cdots \times \mathcal{T}(N \text{ terms})$, thus $\mathcal{Y} := (\mathbf{Y}_1, \ldots, \mathbf{Y}_N)^T \in \mathcal{K}$. Thus, the inner product and the corresponding norm on the space $\mathcal{K}$ can be defined as $\langle \mathcal{X}, \mathcal{Y} \rangle := \sum_{i=1}^{N} \langle \mathbf{X}_i, \mathbf{Y}_i \rangle$ and $\|\mathcal{X}\|_{\mathcal{K}} := \sqrt{\sum_{i=1}^{N} \langle \mathbf{X}_i, \mathbf{X}_i \rangle} = \sqrt{\sum_{i=1}^{N} \|\mathbf{X}_i\|_F^2}$. Then, we can define the functions $f : \mathcal{K} \to \mathbb{R}$ and $g : \mathcal{T} \to \mathbb{R}$ as

$$\begin{aligned} f(\mathcal{Y}) &:= \sum_{n=1}^{N} f_n(\mathbf{Y}_n) = \sum_{n=1}^{N} \alpha_n L_{\omega^k}(Y_{n,(n)}) \\ g(\mathbf{X}) &:= \mathcal{X}_{\mathcal{S}}(\mathbf{X}). \end{aligned}$$

The constraints are expressed as follows:

$$\mathcal{Y} = (\mathcal{I}_{\mathcal{T}}, \ldots, \mathcal{I}_{\mathcal{T}})^T \mathbf{X} := G\mathbf{X}.$$

The optimization problem in (9) is well structured since both sets of variables $\mathcal{Y}$ and $\mathbf{X}$ are separated. This allows one to solve $\mathcal{Y}$ and $\mathbf{X}$ in two decoupled subproblems.

By attaching the Lagrangian multiplier $\mathcal{M} = (\mathbf{M}_1, \ldots, \mathbf{M}_N)^T \in \mathcal{K}$ to the linear constraints, the augmented Lagrangian function of (9) is given by

$$\begin{aligned} L(\mathbf{X}, \mathcal{Y}, \mathcal{M}) &= f(\mathcal{Y}) + g(\mathbf{X}) + \langle \mathcal{M}, G\mathbf{X} - \mathcal{Y} \rangle_{\mathcal{K}} + \frac{\beta}{2} \|G\mathbf{X} - \mathcal{Y}\|_{\mathcal{K}}^2 \\ &= \mathcal{X}_{\mathcal{S}}(\mathbf{X}) + \sum_{n=1}^{N} \left(f_n(\mathbf{Y}_n) + \langle \mathbf{M}_n, \mathbf{X} - \mathbf{Y}_n \rangle + \frac{\beta}{2} \|\mathbf{X} - \mathbf{Y}_n\|_F^2 \right), \end{aligned} \tag{10}$$

where β is the penalty parameter for the violation of the linear constraints. More specifically, to approach a solution of (10), the ADMM solves the following subproblems at each iteration:

$$\begin{cases} \text{Step 1:} & \mathbf{X}^{k+1} \in \arg\min L(\mathbf{X}, \mathcal{Y}^k, \mathcal{M}^k) \\ \text{Step 2:} & \mathcal{Y}^{k+1} \in \arg\min L(\mathbf{X}^{k+1}, \mathcal{Y}, \mathcal{M}^k) \\ \text{Step 3:} & \mathcal{M}^{k+1} = \mathcal{M}^k + \beta(G\mathbf{X}^{k+1} - \mathcal{Y}^{k+1}). \end{cases}$$

In Step 1, we solve the following subproblem:

$$\mathbf{X}^{k+1} \in \arg\min_{\mathbf{X}} \left\{ \mathcal{X}_{\mathcal{S}}(\mathbf{X}) + \sum_{n=1}^{N} \langle \mathbf{M}_n^k, \mathbf{X} - \mathbf{Y}_n^k \rangle + \sum_{n=1}^{N} \frac{\beta}{2} \|\mathbf{X} - \mathbf{Y}_n^k\|_F^2 \right\}. \tag{11}$$

It is easy to see that the objective function of (11) is differentiable, thus $\mathbf{X}^{k+1}$ has a closed-form solution:

$$\mathbf{X}^{k+1} = \frac{1}{N\beta}\left(\sum_{n=1}^{N}(\beta\mathbf{Y}_n^k - \mathbf{M}_n^k)\right)_{\Omega^c} + \mathbf{B}, \tag{12}$$

where Ω^c is the complementary set of the index set Ω.

In Step 2, we solve the following subproblem:

$$\mathcal{Y}^{k+1} \in \arg\min_{\mathcal{Y}}\left\{\sum_{n=1}^{N}\left(f_n(\mathbf{Y}_n) + \langle\mathbf{M}_n^k, \mathbf{X}^{k+1} - \mathbf{Y}_n\rangle + \frac{\beta}{2}\|\mathbf{X}^{k+1} - \mathbf{Y}_n\|_F^2\right)\right\}. \tag{13}$$

By noting that the objective function of (13) is a sum of independent non-negative functions, we can solve each $\mathbf{Y}_n^{k+1}$ separately. Thus, let us consider solving the typical variable $\mathbf{Y}_n^{k+1}$:

$$\begin{aligned} Y_{n,(n)}^{k+1} &= \arg\min_{Y_{n,(n)}}\left\{\alpha_n L_{\omega^k}(Y_{n,(n)}) + \langle M_{n,(n)}^k, X_{(n)}^{k+1} - Y_{n,(n)}\rangle + \frac{\beta}{2}\|X_{(n)}^{k+1} - Y_{n,(n)}\|_F^2\right\} \\ &= \arg\min_{Y_{n,(n)}}\left\{\frac{\alpha_n}{\beta}L_{\omega^k}(Y_{n,(n)}) + \frac{1}{2}\|Y_{n,(n)} - X_{(n)}^{k+1} - \frac{1}{\beta}M_{n,(n)}^k\|_F^2\right\}, \end{aligned} \tag{14}$$

where $\omega^k = (1/(\sigma_1(Y_{n,(n)}^k) + \varepsilon_n), \ldots, 1/(\sigma_{I_n}(Y_{n,(n)}^k) + \varepsilon_n))^T$.

For solving (14), a lemma [42] has to be introduced.

Lemma 1. *Given* $X_{(n)} \in \mathbb{R}^{I_n \times \prod_{i\neq n} I_i}$ *and* $0 \le \omega_1^k \le \cdots \le \omega_{I_n}^k$, *a minimizer to*

$$\min_{X_{(n)}} \frac{1}{2}\|X_{(n)} - M\|_F^2 + \tau L_{\omega^k}(X_{(n)}), \tag{15}$$

is given by the weighted SVT*operator* $T_{\omega^k,\tau}(M)$*:*

$$T_{\omega^k,\tau}(M) = U(\Sigma - \tau\, diag(\omega^k))_+ V^T, \tag{16}$$

where $U\Sigma V^T$ *is the* SVD*of* M.

Let $\tau = \alpha_n/\beta$, then we can obtain $Y_{n,(n)}^{k+1}$ by applying Lemma 1, namely,

$$\begin{aligned} Y_{n,(n)}^{k+1} &= T_{\omega^k,\tau}(X_{(n)}^{k+1} + \frac{1}{\beta}M_{n,(n)}^k) \\ &= U^k(\Sigma^k - \tau\,\mathrm{diag}(\omega^k))_+(V^k)^T, \end{aligned} \tag{17}$$

where $U^k\Sigma^k(V^k)^T$ is the SVD of $X_{(n)}^{k+1} + \frac{1}{\beta}M_{n,(n)}^k$.

As the above derivation is available for any $n \in \{1, 2, \ldots, N\}$, $\mathcal{Y}^{k+1}$ is therefore

$$\begin{aligned} \mathcal{Y}^{k+1} &= (\mathbf{Y}_1^{k+1}, \ldots, \mathbf{Y}_N^{k+1})^T \\ &= (\mathrm{fold}_1(Y_{1,(1)}^{k+1}), \ldots, \mathrm{fold}_N(Y_{N,(N)}^{k+1}))^T. \end{aligned} \tag{18}$$

Based on the previous derivation, we develop the following ADMM iterative scheme for the model (5), as outlined in Algorithm 1.

Algorithm 1 Solving the model (5) via ADMM.

Input: Nth-order tensor $\mathbf{B}$, index set Ω, parameters β and ε.
Initialize: $\mathcal{Y} = 0$, $\mathcal{M} = 0$, tol $= 10^{-5}$, and $K = 500$.

```
for k = 1 to K do
  Update X^{k+1} by (12)
    for n = 1 to N do
      Update Y_n^{k+1} by (17)
    end for
  Update the multiplier M^{k+1} by
    M^{k+1} = M^k + β(GX^{k+1} − Y^{k+1})
  Check the convergence condition
    ‖X^{k+1} − X^k‖_F / ‖X^k‖_F < tol
end for
```

Output: $\mathbf{X}$.

Algorithm 1 can be accelerated by adaptively changing β. One of the efficient strategies [51] is to increase β iteratively by a multiple t, i.e., $\beta^{k+1} = t\beta^k$, where β^0 is the input of β in Algorithm 1, and $t \in [1.1, 1.2]$. However, the proposed model (5) is non-convex, and the proof of the convergence properties of the ADMM in theory is still an open issue [52].

Table 1
RSE ($\times 10^{-6}$) comparison of the results for different dimensions with different sampling rates.

$\mathbf{T} = \mathbb{R}^{20\times20\times20},\ r = 2$					$\mathbf{T} = \mathbb{R}^{20\times20\times20\times20},\ r = 2$				
Algorithm	SR	RSE	Iter	Time (s)	Algorithm	SR	RSE	Iter	Time (s)
HaLRTC	10%	–	–	–	HaLRTC	10%	–	–	–
	30%	33.1	143	0.199		30%	5.85	78	2.099
	50%	7.02	75	0.131		50%	**3.84**	33	0.884
ADMM-TR(E)	10%	–	–	–	ADMM-TR(E)	10%	–	–	–
	30%	73.9	132	0.175		30%	11.7	76	2.313
	50%	9.70	74	0.101		50%	6.40	33	0.982
Our method	10%	**252**	209	0.540	Our method	10%	**13.3**	102	5.459
	30%	**6.32**	166	0.432		30%	**4.95**	50	2.659
	50%	**3.72**	57	0.144		50%	4.49	46	2.481
$\mathbf{T} = \mathbb{R}^{20\times20\times20\times20\times20},\ r = 2$					$\mathbf{T} = \mathbb{R}^{20\times30\times40\times50},\ r = 2$				
Algorithm	SR	RSE	Iter	Time (s)	Algorithm	SR	RSE	Iter	Time (s)
HaLRTC	10%	–	–	–	HaLRTC	10%	784	220	45.663
	30%	7.88	66	37.798		30%	7.12	59	12.722
	50%	8.89	36	22.936		50%	5.19	35	7.621
ADMM-TR(E)	10%	–	–	–	ADMM-TR(E)	10%	927	195	47.880
	30%	29.0	61	35.101		30%	12.8	60	14.855
	50%	11.3	35	21.358		50%	7.16	36	9.018
Our method	10%	**9.00**	79	126.808	Our method	10%	**7.55**	147	97.605
	30%	**5.51**	62	110.767		30%	**5.22**	69	46.453
	50%	**4.47**	38	67.547		50%	**5.11**	48	31.920

Table 2
RSE ($\times 10^{-6}$) comparison of the results for different n-rank.

n-rank	HaLRTC	ADMM-TR(E)	Our method
(2, 2, 2)	18.8	7.08	**3.54**
(4, 4, 4)	12.2	7.08	**4.92**
(6, 6, 6)	11.8	13.9	**4.84**
(2, 4, 6)	12.1	22.1	**5.60**

5. Numerical experiments

In this section, we compare our method with HaLRTC [14] and ADMM-TR(E) [15]. Both belong to the nuclear norm method. All the tests are performed under Windows 7 and Matlab Version 8.2.0.701 (R2013b) running on a desktop PC with an Inter Core i3-4160 CPU at 3.60 GHz and 4 GB of memory.

The masked data $\mathbf{B}$ are obtained by random sampling. The weights α_n $(n = 1, \ldots, N)$ describe the importance of rank$(X_{(n)})$, i.e., the α_n will be set larger if the mode-n fibers are more correlative. In this paper, we mainly study the effects of the logDet function and nuclear norm, so we set the common variables the same, namely, the constants α_n are set to be $1/N$. For simplicity, we set the parameters $\varepsilon_1 = \cdots = \varepsilon_N = \varepsilon$. Their values are related to the pixel values of $\mathbf{X}$: (1) the magnitude of ε is 10^{-5}, 10^{-6}, or 10^{-7} according to the different ranges of pixel values of $\mathbf{X}$; (2) if the range of pixel values of $\mathbf{X}$ is in [0, 1], then $\varepsilon = m \times 10^{-5}$, where $m = 1, 2, \ldots, 9$ ($m = 1$ in our experiments).

5.1. Synthetic data

In this section, we test the proposed model on synthetic data. The test synthetic data are generated by Tucker decomposition:

$$\mathbf{X} = \mathbf{C} \times_1 A_1 \times_2 \cdots \times_N A_N,$$

where $\mathbf{C} \in \mathbb{R}^{r_1 \times r_2 \times \cdots \times r_N}$ is the core tensor and $A_n \in \mathbb{R}^{I_n \times r_n}$ for $n = 1, 2, \ldots, N$ are the factor matrices. The core tensor and the factor matrices are all standard independent and identically distributed (i.i.d) Gaussian entries. In our experiments, the core tensor $\mathbf{C}$ and the factor matrices $A_n \in \mathbb{R}^{I_n \times r_n}$ are generated by MATLAB commands `randn`$(r_1, \ldots, r_N)$ and `randn`(I_n, r_n), respectively. Thus, the n-rank for the generated tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is $(r_1, \ldots, r_N)$. Furthermore, we set $r_1 = r_2 = \cdots = r_N = r$ for simplicity, and the order of the tensor varies from three to five. The index set Ω is generated uniformly at random, and we sample a few entries from $\mathbf{X}$ at the positions in Ω.

The quality of the estimated tensor is measured by the relative squared error (RSE), which is defined by

$$\text{RSE} = \frac{\|\mathbf{X} - \tilde{\mathbf{X}}\|_F}{\|\tilde{\mathbf{X}}\|_F},$$

Fig. 3. The recovered results for video "suzie". From left to right: the original frame, the masked frame, and the estimated results by HaLRTC, ADMM-TR(E), and our method. From top to bottom: the sampling rates are 10%, 30%, and 50%, respectively.

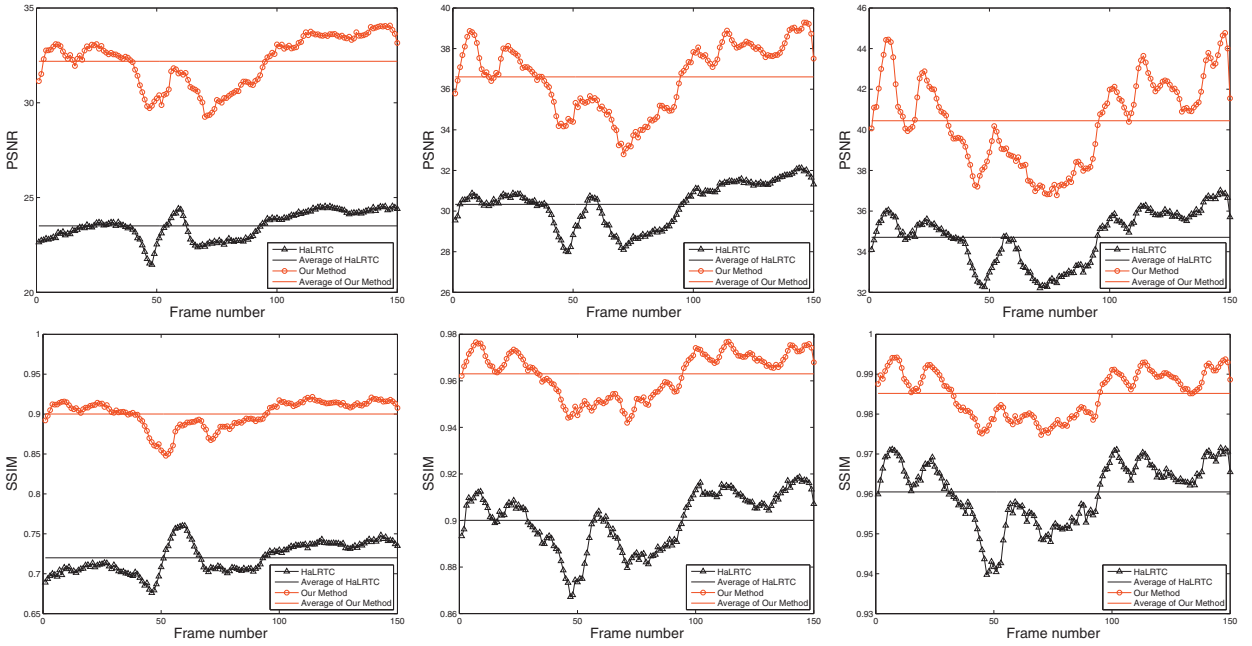

Fig. 4. The PSNR values and the SSIE values recovered by HaLRTC and our method for every frame. From left to right: the sampling rates are 10%, 30%, and 50%, respectively.

where $\tilde{\mathbf{X}}$ and $\mathbf{X}$ are the original tensor and the estimated tensor, respectively.

We test four synthetic data of size $20\times20\times20$, $20\times20\times20\times20$, $20\times20\times20\times20\times20$, and $20\times30\times40\times50$ with different sampling rates (SR), i.e., SR = 10%, SR = 30%, and SR = 50%. The results are summarized in Table 1. In this table, "Iter" and "Time (s)" denote the number of the iterative step and the MATLAB CPU time, respectively, and "—" denotes that the RSE value is greater than 5×10^{-1}, which means the corresponding method fails. From this table, one can make the following observations: (1) the larger the sampling rate, the better the results recovered by all methods; (2) for different dimensional tensors, our results are the best; (3) our method can recover a better result for both low sampling rates and high sampling rates, but the HaLRTC and the ADMM-TR(E) methods fail for the 10% sampling rate; (4) our method costs more cpu time than the HaLRTC and the ADMM-TR(E) methods, because our method is non-convex and uses the singular value decomposition twice per iteration.

Table 3
PSNR and SSIE comparison of the results.

Video	SR (%)	HaLRTC		ADMM-TR(E)		Our method	
		PSNR	SSIE	PSNR	SSIE	PSNR	SSIE
akiyo	10	25.26	0.7976	25.25	0.7975	**34.52**	**0.9454**
	30	32.32	0.9454	32.32	0.9454	**42.09**	**0.9881**
	50	37.92	0.9833	37.92	0.9833	**47.10**	**0.9660**
bridge	10	31.06	0.8450	31.06	0.8450	**37.69**	**0.9288**
	30	36.61	0.9355	36.61	0.9355	**39.97**	**0.9537**
	50	39.84	0.9635	39.84	0.9635	**42.13**	**0.9708**
carphone	10	22.37	0.7019	22.36	0.7017	**30.75**	**0.8816**
	30	29.17	0.8962	29.17	0.8962	**35.46**	**0.9540**
	50	33.74	0.9573	33.74	0.9573	**38.86**	**0.9781**
claire	10	26.81	0.8760	26.81	0.8760	**36.03**	**0.9562**
	30	34.21	0.9642	34.21	0.9642	**42.29**	**0.9856**
	50	39.43	0.9866	39.43	0.9866	**46.69**	**0.9936**
coastguard	10	20.50	0.5207	20.50	0.5206	**25.77**	**0.7471**
	30	24.54	0.7605	24.54	0.7605	**30.15**	**0.9010**
	50	28.19	0.8879	28.19	0.8879	**33.97**	**0.9588**
container	10	22.63	0.7742	22.62	0.7741	**31.40**	**0.9517**
	30	28.89	0.9222	28.89	0.9222	**39.98**	**0.9800**
	50	34.49	0.9719	34.49	0.9720	**46.88**	**0.9941**
foreman	10	19.66	0.5165	19.66	0.5166	**28.50**	**0.8320**
	30	25.94	0.8039	25.94	0.8039	**34.14**	**0.9426**
	50	30.79	0.9226	30.79	0.9226	**38.47**	**0.9776**
hall	10	22.52	0.7527	22.52	0.7526	**30.56**	**0.8930**
	30	29.05	0.9184	29.05	0.9184	**35.62**	**0.9568**
	50	33.68	0.9654	33.68	0.9654	**39.07**	**0.9792**
highway	10	26.45	0.7355	26.45	0.7355	**31.59**	**0.8737**
	30	30.85	0.8853	30.85	0.8853	**34.71**	**0.9295**
	50	34.29	0.9443	34.29	0.9443	**37.08**	**0.9587**
news	10	20.99	0.6977	20.99	0.6977	**29.07**	**0.8610**
	30	27.17	0.8902	27.17	0.8902	**34.26**	**0.9514**
	50	32.03	0.9585	32.03	0.9585	**38.31**	**0.9805**
salesman	10	23.68	0.6511	23.68	0.6510	**31.78**	**0.8948**
	30	30.10	0.8888	30.10	0.8888	**36.60**	**0.9614**
	50	34.81	0.9589	34.81	0.9589	**40.60**	**0.9840**
escalator	10	17.87	0.5537	17.88	0.5538	**23.43**	**0.8012**
	30	23.18	0.8411	23.18	0.8411	**27.44**	**0.9128**
	50	27.15	0.9339	27.15	0.9339	**30.52**	**0.9594**

Next, we study the influence of ranks. In this example, we test the data of size $50\times50\times50$, and the n-rank of the tensor is set to be (2, 2, 2), (6, 6, 6), (10, 10, 10), and (2, 4, 6). The sampling rate is set to be 30% in this test. We display the results in Table 2. From this table, we can see that the results recovered by our method are better than those from the other methods.

5.2. Real data

In this section, we test the proposed method on video, hyperspectral images, and MRI. We quantitatively evaluate the quality of the estimated tensor via peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) index [57]. Given a ground-truth tensor $\tilde{\mathbf{X}}$, the PSNR of an estimated tensor $\mathbf{X}$ is computed by the standard formula

$$\mathrm{PSNR}(\mathbf{X},\tilde{\mathbf{X}}) = 10\log_{10}\frac{n\tilde{\mathbf{X}}_{\max}^2}{\|\tilde{\mathbf{X}}-\mathbf{X}\|_F^2},$$

where n denotes the number of the pixels in the tensor, and $\tilde{\mathbf{X}}_{\max}$ is the maximum pixel value of the original tensor. Given an original matrix $\tilde{X}$, the SSIM of the estimated matrix X is defined by Eq. (13) of [57]

$$\mathrm{SSIM}(X,\tilde{X}) = \frac{(2\mathrm{mean}(X)\mathrm{mean}(\tilde{X})+c_1)(2\mathrm{cov}(X,\tilde{X})+c_2)}{(\mathrm{mean}(X)^2+\mathrm{mean}(\tilde{X})^2+c_1)(\mathrm{std}(X)^2+\mathrm{std}(\tilde{X})^2+c_2)},$$

where mean(A), std(A), and cov(A, B) denote the mean value of A, the standard deviation of A, and the covariance of A and B, respectively, and c_1 and c_2 are constants. For the real tensor data, we calculate the SSIM of its mode-1 unfolding as the SSIM value of the tensor.

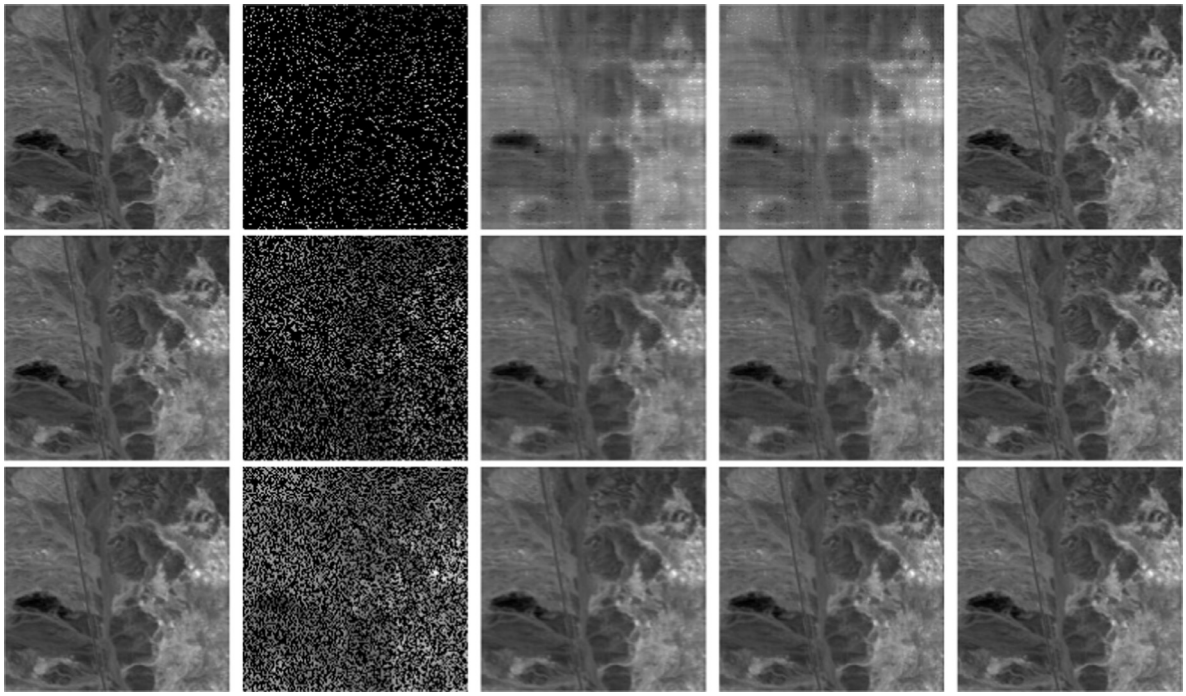

Fig. 5. The recovered results for hyperspectral images. From left to right: the original frame, the masked frame, and the estimated results by HaLRTC, ADMM-TR(E), and our method. From top to bottom: the sampling rates are 10%, 30%, and 50%, respectively.

5.2.1. Video

In this section, we study the performance of HaLRTC, ADMM-TR(E), and the proposed method. The test videos are downloaded from http://trace.eas.asu.edu/yuv/, which are in color in the YUV format. In the experiments, the videos are converted into RGB. The color video is a fourth-order tensor[1]. In Fig. 3, we illustrate the recovered results of one frame of "suzie" of size $144\times176\times3\times150$ by HaLRTC, ADMM-TR(E), and the proposed method with different sampling rates. From the figure, we see that the proposed method obtains higher-quality results for all sampling rates. In particular, for the 10% sampling rate, the HaLRTC and ADMM-TR(E) methods cannot recover clear details, whereas our results are satisfactory. The PSNR and SSIE[2] values against the frame number are plotted in Fig. 4. It can be seen that our method performs better in terms of PSNR and SSIE values for every frame and all sampling rates. Moreover, more videos are also tested, and the results are summarized in Table 3. From this table, it can be seen that the results recovered by the proposed method are always better than those of other methods.

5.2.2. Hyperspectral image

In this section, the proposed method is applied to the hyperspectral image recovery. The hyperspectral data used in the experiments are a 150×150 pixels subset of Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Cuprite data[3] with 188 spectral bands, see [58–60] for more details. We test three different sampling rates: 10%, 30%, and 50%. The results recovered by HaLRTC, ADMM-TR(E), and the proposed method are shown in Fig. 5. It is clear that our results are visually better than those from HaLRTC and ADMM-TR(E): for the 10% sampling rate case, our method can recover satisfactory results, but the results of HaLRTC and ADMM-TR(E) are blurry; for the 30% sampling rate case, our results contain more details. Fig. 6 shows the PSNR and SSIE values with respect to the frame number. We can see that our method obtains better PSNR and SSIE values for every frame as compared with HaLRTC.

5.2.3. MRI

In this section, we test the performance of HaLRTC, ADMM-TR(E), and the proposed method on MRI data. The test MRI data[4] are a third-order tensor of size $181\times217\times181$. In this study, 70% of the data elements are removed at random. The results observed from different directions are shown in Fig. 7. Clearly, our results are sharper and visually better than those of HaLRTC and ADMM-TR(E). To be specific, our method can reconstruct more details and sharper edges and obtain clearer

[1] In our test, if the number of the video frames is larger than 150, we use the first 150 frames.
[2] Each frame of this video is in color, we calculate the SSIM of color images after converting it to grayscale.
[3] Available from http://aviris.jpl.nasa.gov/html/aviris.freedata.html
[4] Available from http://brainweb.bic.mni.mcgill.ca/brainweb/selection_normal.html

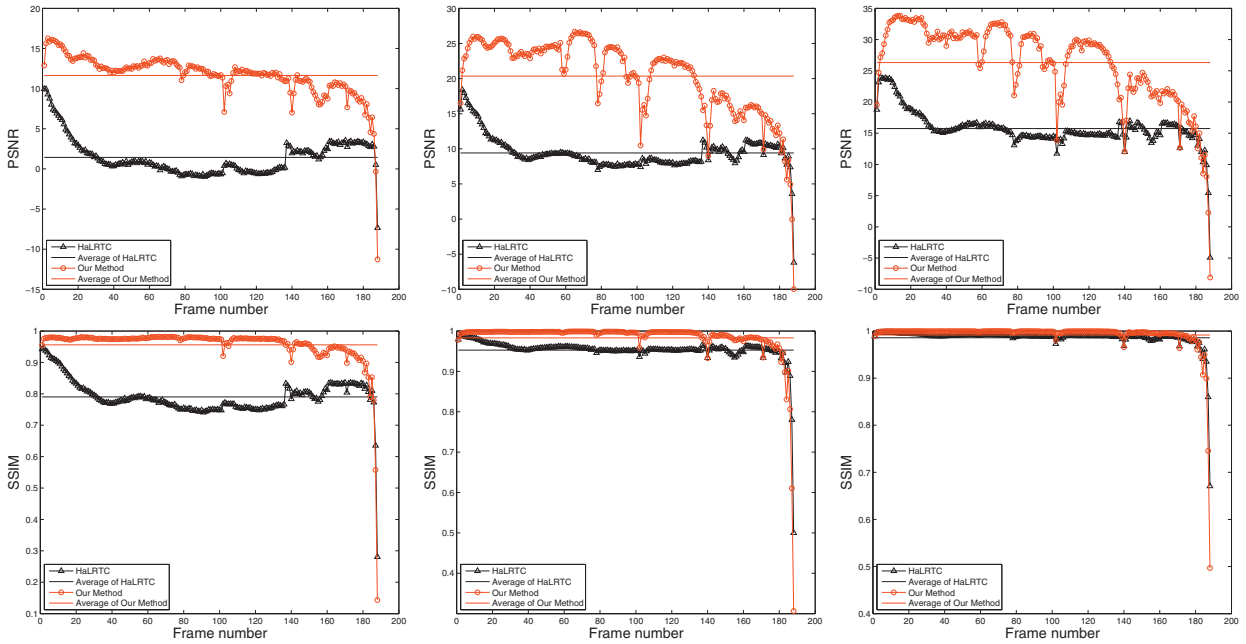

Fig. 6. The PSNR values and the SSIE values recovered by HaLRTC and our method for every frame. From left to right: the sampling rates are 10%, 30%, and 50%, respectively.

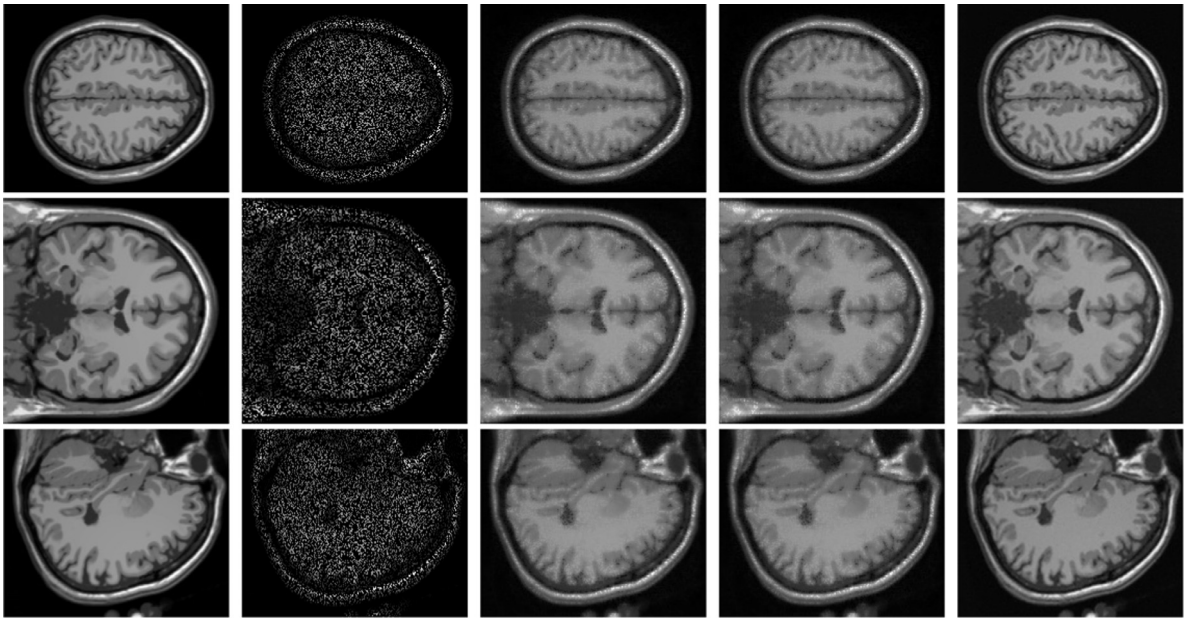

Fig. 7. The recovered results for three directions. From left to right: the original data, the masked data, and the estimated results by HaLRTC, ADMM-TR(E), and our method. From top to bottom: the results for front direction, the side direction, and the top direction, respectively. The sampling rate is 30%.

results in the homogenous regions. Fig. 8 shows the PSNR and SSIM values of every frame for three directions. We note that our method obtains better results for every frame than those recovered by HaLRTC.

6. Conclusions

In this paper, we have proposed a non-convex definition for the tensor rank. Compared with the convex tensor nuclear norm proposed by Liu et al. [14], our definition was better at approximating the tensor rank. We introduced it into low-rank tensor completion and developed an ADMM-based algorithm. Finally, we compared our method with HaLRTC

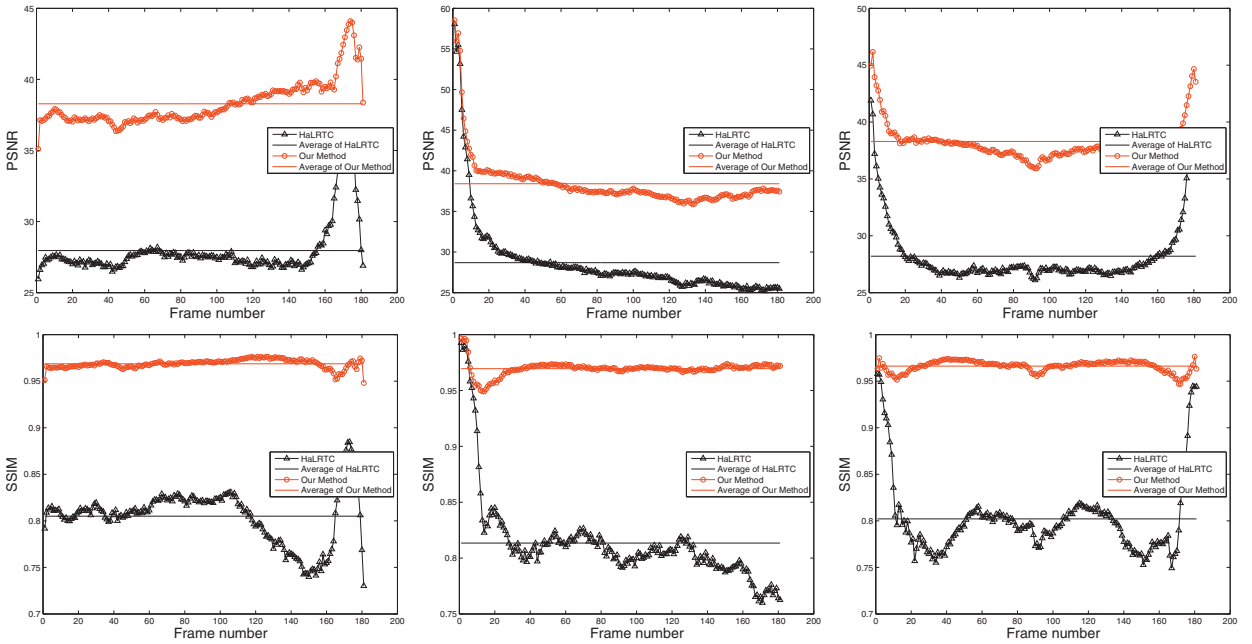

Fig. 8. The PSNR values and the SSIE values recovered by HaLRTC and our method for every frame. From left to right: the numerical values for front direction, the side direction, and the top direction, respectively. The sampling rate is 30%.

and ADMM-TR(E) on synthetic and real data with different sampling rates. The experimental results have shown that the proposed method was more effective, especially for the 10% sampling rate.

Acknowledgments

The authors would like to express their great thankfulness to the reviewers for their much helpful suggestions for revising this paper. The authors would like to express their great thankfulness to Dr. J. Liu for sharing the codes of the HaLRTC algorithm. This research is supported by 973 Program (2013CB329404), NSFC (61370147, 61402082).

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '00, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 2000, pp. 417–424.
[2] A. Criminisi, P. Perez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, IEEE Trans. Image Process. 13 (9) (2004) 1200–1212.
[3] N. Komodakis, Image completion using global optimization, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1, 2006, pp. 442–452.
[4] Q. Cheng, H. Shen, L. Zhang, P. Li, Inpainting for remotely sensed images with a multichannel nonlocal total variation model, IEEE Trans. Geosci. Remote Sens. 52 (1) (2014) 175–187.
[5] H. Shen, X. Li, L. Zhang, D. Tao, C. Zeng, Compressed sensing-based inpainting of aqua moderate resolution imaging spectroradiometer band 6 using adaptive spectrum-weighted sparse bayesian dictionary learning, IEEE Trans. Geosci. Remote Sens. 52 (2) (2014) 894–906.
[6] L.-J. Deng, T.-Z. Huang, X.-L. Zhao, Exemplar-based image inpainting using a modified priority definition, PLoS ONE 10 (10) (2015) 1–18.
[7] M. Bertalmio, A.L. Bertozzi, G. Sapiro, Navier-stokes, fluid dynamics, and image and video inpainting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, 2001. I–355–362
[8] T. Korah, C. Rasmussen, Spatiotemporal inpainting for recovering texture maps of occluded building facades, IEEE Trans. Image Process. 16 (9) (2007) 2262–2271.
[9] X. Li, H. Shen, L. Zhang, H. Li, Sparse-based reconstruction of missing information in remote sensing images from spectral/temporal complementary information, ISPRS J. Photogramm. Remote Sens. 106 (2015) 1–15.
[10] X. Li, H. Shen, H. Li, L. Zhang, Patch matching-based multitemporal group sparse representation for the missing information reconstruction of remote-sensing images, IEEE J. Select. Top. Appl. Earth Obs. Remote Sens. 9 (8) (2016) 3629–3641.
[11] H. Shen, X. Li, Q. Cheng, C. Zeng, G. Yang, H. Li, L. Zhang, Missing information reconstruction of remote sensing data: a technical review, IEEE Geosci. Remote Sens. Mag. 3 (3) (2015) 61–85.
[12] M. Pauly, N.J. Mitra, J. Giesen, M.H. Gross, L.J. Guibas, Example-based 3D scan completion, in: Symposium on Geometry Processing, 2005, pp. 23–32.
[13] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (3) (2009) 455–500.
[14] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 208–220.
[15] S. Gandy, B. Recht, I. Yamada, Tensor completion and low-n-rank tensor recovery via convex optimization, Inverse Probl. 27 (2) (2011) 025010.
[16] Y. Xu, R. Hao, W. Yin, Z. Su, Parallel matrix factorization for low-rank tensor completion, Inverse Probl. Imaging 9 (2) (2015) 601–624.
[17] T.-Y. Ji, T.-Z. Huang, X.-L. Zhao, T.-H. Ma, G. Liu, Tensor completion using total variation and low-rank matrix factorization, Inf. Sci. 326 (2016) 243–257.
[18] W. Cao, Y. Wang, C. Yang, X. Chang, Z. Han, Z. Xu, Folded-concave penalization approaches to tensor completion, Neurocomputing 152 (2015) 261–273.

[19] Q. Zhao, D. Meng, X. Kong, Q. Xie, W. Cao, Y. Wang, Z. Xu, A novel sparsity measure for tensor recovery, in: IEEE International Conference on Computer Vision (ICCV), 2015, pp. 271–279.
[20] Z. Wen, W. Yin, Y. Zhang, Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm, Math. Program. Comput. 4 (4) (2012) 333–361.
[21] J.-F. Cai, E.J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Opt. 20 (4) (2010) 1956–1982.
[22] E.J. Candès, B. Recht, Exact matrix completion via convex optimization, Found. Comput. Math. 9 (6) (2009).
[23] E.J. Candès, T. Tao, The power of convex relaxation: near-optimal matrix completion, IEEE Trans. Inf. Theory 56 (5) (2010) 2053–2080.
[24] B. Recht, M. Fazel, P.A. Parrilo, Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization, SIAM Rev. 52 (3) (2010) 471–501.
[25] S. Ma, D. Goldfarb, L. Chen, Fixed point and Bregman iterative methods for matrix rank minimization, Math. Program. 128 (1) (2011).
[26] W. He, H. Zhang, L. Zhang, H. Shen, Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration, IEEE Trans. Geosci. Remote Sens. 54 (1) (2016) 178–188.
[27] M. Fazel, H. Hindi, S.P. Boyd, Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices, in: Proceedings of American Control Conference, volume 3, 2003, pp. 2156–2162.
[28] S. Gu, L. Zhang, W. Zuo, X. Feng, Weighted nuclear norm minimization with application to image denoising, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[29] C.-H. Zhang, Nearly unbiased variable selection under minimax concave penalty, Ann. Stat. 38 (2) (2010) 894–942.
[30] T.T. Cai, A. Zhang, Sharp RIP bound for sparse signal and low-rank matrix recovery, Appl. Comput. Harmonic Anal. 35 (1) (2013) 74–93.
[31] R.A. Harshman, Foundations of the parafac procedure: models and conditions for an “explanatory” multi-modal factor analysis, in: UCLA Working Papers in Phonetics, 1970.
[32] E. Acar, D.M. Dunlavy, T.G. Kolda, M. Mørup, Scalable tensor factorizations with missing data, in: SIAM International Conference on Data Mining (SDM), SIAM, 2010, pp. 701–712.
[33] L.R. Tucker, Some mathematical notes on three-mode factor analysis, Psychometrika 31(3) 279–311.
[34] M. Filipović, A. Jukić, Tucker factorization with missing data with application to low-n-rank tensor completion, Multidimens. Syst. Signal Process. 26 (3) (2015) 677–692.
[35] W. Cao, Y. Wang, J. Sun, D. Meng, C. Yang, A. Cichocki, Z. Xu, Total Variation Regularized Tensor RPCA for Background Subtraction From Compressive Measurements, IEEE Trans. Image Process. 25 (9) (2016) 4075–4090.
[36] T.-H. Ma, T.-Z. Huang, X.-L. Zhao, Group-based image decomposition using 3-d cartoon and texture priors, Inf. Sci. 328 (2016) 510–527.
[37] Y. Peng, D. Meng, Z. Xu, C. Gao, Y. Yang, B. Zhang, Decomposable nonlocal tensor dictionary learning for multispectral image denoising, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.
[38] D. Meng, B. Zhang, Z. Xu, L. Zhang, C. Gao, Robust low-rank tensor factorization by cyclic weighted median, Sci. China Inf. Sci. 58 (5) (2015) 1–11.
[39] K. Mohan, M. Fazel, Iterative reweighted algorithms for matrix rank minimization, J. Mach. Learn. Res. 13 (1) (2012) 3441–3473.
[40] Y. Deng, Q. Dai, R. Liu, Z. Zhang, S. Hu, Low-rank structure learning via nonconvex heuristic recovery, IEEE Trans. Neural Netw. Learn. Syst. 24 (3) (2013) 383–396.
[41] W. Dong, G. Shi, X. Hu, Y. Ma, Nonlocal sparse and low-rank regularization for optical flow estimation, IEEE Trans. Image Process. 23 (10) (2014) 4527–4538.
[42] W. Dong, G. Shi, X. Li, Y. Ma, F. Huang, Compressive sensing via nonlocal low-rank regularization, IEEE Trans. Image Process. 23 (8) (2014) 3618–3632.
[43] Y. Peng, J. Suo, Q. Dai, W. Xu, Reweighted low-rank matrix recovery and its application in image restoration, IEEE Trans. Cybern. 44 (12) (2014) 2418–2430.
[44] M. Malek-Mohammadi, M. Babaie-Zadeh, M. Skoglund, Iterative concave rank approximation for recovering low-rank matrices, IEEE Trans. Signal Process. 62 (20) (2014) 5213–5226.
[45] O. Taheri, S.A. Vorobyov, Sparse channel estimation with L_p-norm and reweighted L_1-norm penalized least mean squares, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011, pp. 2864–2867.
[46] Z. Xu, H. Zhang, Y. Wang, X. Chang, Y. Liang, $L_{1/2}$ regularization, Sci. China Inf. Sci. 53 (6) (2010) 1159–1169.
[47] E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis? J. ACM 58 (3) (2011) 11:1–11:37.
[48] E.J. Candès, M.B. Wakin, S.P. Boyd, Enhancing sparsity by reweighted l1 minimization, J. Fourier Anal. Appl. 14 (5) (2008) 877–905.
[49] M. Dorpinghaus, N. Gaffke, L.A. Imhof, R. Mathar, A log-det inequality for random matrices, SIAM J. Matrix Anal. Appl. 36 (3) (2015) 1164–1179.
[50] 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.
[51] Z. Lin, M. Chen, Y. Ma, The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices, 2010. ArXiv preprint arXiv:1009.5055.
[52] B. He, M. Tao, X. Yuan, Alternating direction method with Gaussian back substitution for separable convex programming, SIAM J. Opt. 22 (2) (2012) 313–340.
[53] X.-L. Zhao, F. Wang, M.K. Ng, A new convex optimization model for multiplicative noise and blur removal, SIAM J. Imaging Sci. 7 (1) (2014) 456–475.
[54] X.-L. Zhao, W. Wang, T.-Y. Zeng, T.-Z. Huang, M.K. Ng, Total variation structured total least squares method for image restoration, SIAM J. Sci. Comput. 35 (6) (2013) B1304–B1320.
[55] L.-J. Deng, W. Guo, T.-Z. Huang, Single image super-resolution by approximated heaviside functions, Inf. Sci. 348 (2016) 107–123.
[56] C. Chen, M.K. Ng, X.-L. Zhao, Alternating direction method of multipliers for nonlinear image restoration problems, IEEE Trans. Image Process. 24 (1) (2015) 33–43.
[57] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.
[58] X.-L. Zhao, F. Wang, T.-Z. Huang, M.K. Ng, R.J. Plemmons, Deblurring and sparse unmixing for hyperspectral images, IEEE Trans. Geosci. Remote Sens. 51 (7) (2013) 4045–4058.
[59] M.-D. Iordache, J.M. Bioucas-Dias, A. Plaza, Sparse unmixing of hyperspectral data, IEEE Trans. Geosci. Remote Sens. 49 (6) (2011) 2014–2039.
[60] M.-D. Iordache, J.M. Bioucas-Dias, A. Plaza, Total variation spatial regularization for sparse hyperspectral unmixing, IEEE Trans. Geosci. Remote Sens. 50 (11) (2012) 4484–4502.