Trace Norm Regularized CANDECOMP/PARAFAC Decomposition With Missing Data

Yuanyuan Liu, Fanhua Shang, *Member, IEEE*, Licheng Jiao, *Senior Member, IEEE*, James Cheng, and Hong Cheng

***Abstract*—In recent years, low-rank tensor completion (LRTC) problems have received a significant amount of attention in computer vision, data mining, and signal processing. The existing trace norm minimization algorithms for iteratively solving LRTC problems involve multiple singular value decompositions of very large matrices at each iteration. Therefore, they suffer from high computational cost. In this paper, we propose a novel trace norm regularized CANDECOMP/PARAFAC decomposition (TNCP) method for simultaneous tensor decomposition and completion. We first formulate a factor matrix rank minimization model by deducing the relation between the rank of each factor matrix and the mode-*n* rank of a tensor. Then, we introduce a tractable relaxation of our rank function, and then achieve a convex combination problem of much smaller-scale matrix trace norm minimization. Finally, we develop an efficient algorithm based on alternating direction method of multipliers to solve our problem. The promising experimental results on synthetic and real-world data validate the effectiveness of our TNCP method. Moreover, TNCP is significantly faster than the state-of-the-art methods and scales to larger problems.**

***Index Terms*—Alternating direction method of multipliers (ADMM), CANDECOMP/PARAFAC (CP) decomposition, low-rank, tensor completion, trace norm minimization.**

I. Introduction

MULTIWAY data analysis and processing is an important topic in signal processing [1], [2], computer vision [3]–[6], data mining [7], [8], machine learning [9]–[11], numerical linear algebra [12], neuroscience [13], and so on. As the generalization of vectors (i.e., first-order tensors) and matrices (i.e., second-order tensors), higher-order tensors with high dimensionality are becoming increasingly ubiquitous with the rapid development of modern computer technology [14]–[16], such as multichannel images and videos. Compared with vectors and matrices, tensors provide a natural and compact representation for such multiway data, and can be used to express more complicated intrinsic structures in higher-order data [17].

The values of the observed tensors may be missing due to the problems in the acquisition process, loss of information, or costly experiments [18]. In this paper, we are particularly interested in the low-rank tensor completion (LRTC) problem, which is to find a tensor of the (nearly) lowest rank from a subset of the entries of the tensor $f(\mathcal{X}) = b$, where $f(\cdot)$ is the sampling operator, and $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is an Nth-order tensor ($N \geq 3$). The LRTC problem has been successfully applied to a wide range of real-world problems, such as visual data [4], [5], EEG data [13], retail sales data [19], and hyperspectral data analysis [20], [21], social network analysis [8], and link prediction [22]. In the fields of computer vision and image processing, the missing value estimation problem is known as in-painting problems for images or videos [23], [24]. Liu *et al.* [5] indicated that tensor completion utilizes all information along all dimensions, while the matrix completion-based algorithms only consider the constraints along two particular dimensions.

In recent years, sparse vector recovery and low-rank matrix completion problems have been intensively studied [25]–[27]. Though the l_0-norm and the rank minimization have been proven to be strong global constraints and good measures of sparsity [28], the optimization problem involving the l_0-norm or the rank minimization is NP-hard in general due to their discrete nature. The l_1-norm and the trace norm (also known as the nuclear norm) are widely used to approximate the l_0-norm and the rank of a matrix, and the resulting problems are convex optimization problems, respectively. In addition, Candès and Recht [26] and Recht *et al.* [27] have provided theoretical guarantees that the task of the rank minimization problem can be accomplished by solving the trace norm minimization under some reasonable conditions. In fact, the l_1-norm and the trace norm have been shown to be the tightest convex surrogates to the l_0-norm and the rank function, respectively [25], [29].

Manuscript received March 4, 2014; revised August 13, 2014 and November 1, 2014; accepted November 11, 2014. Date of publication December 8, 2014; date of current version October 13, 2015. This work was supported in part by the National Basic Research Program (973 Program) of China, under Grant 2013CB329402, in part by the National Natural Science Foundation of China, under Grant 61072106, Grant 61173090, and Grant 61072108, in part by the National Research Foundation for the Doctoral Program of Higher Education of China, under Grant 20110203110006, in part by the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) under Grant B07048, and in part by the Program for Cheung Kong Scholars and Innovative Research Team in University, under Grant IRT1170. The work of J. Cheng and H. Cheng was supported in part by SHIAE under Grant 8115048, in part by MSRA under Grant 6903555, in part by GRF under 411211, and in part by CUHK Direct under Grant 4055015 and Grant 4055017. This paper was recommended by Associate Editor S. Zafeiriou.

Y. Liu and L. Jiao are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an 710071, China.

F. Shang and J. Cheng are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: fanhuashang@gmail.com).

H. Cheng is with the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong.

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

Digital Object Identifier 10.1109/TCYB.2014.2374695

As the generalization of sparse vector recovery and low-rank matrix completion, the LRTC problem has drawn lots of attention from researchers in the past several years [30]. Tensor decomposition is a type of classical higher-order data analysis methods and gives a concise representation of the underlying structure of the tensor, revealing that the tensor data can be modeled as lying close to a low-dimensional subspace [31]–[33]. Two most popular tensor decomposition methods take the forms: the Tucker decomposition [31] and the CANDECOMP/PARAFAC (CP) decomposition [32]. To address incomplete tensors, their weighted alternating least-squares methods [18], [34] have been successfully applied to EEG data analysis and color image in-painting. However, their performance is usually sensitive to the given ranks due to their least-squares formulations without regularization [35].

Recently, several works [4], [5], [20], [36]–[38] extended the framework of trace norm regularization to the estimation of partially observed low-rank tensor. Liu *et al.* [4] first introduced an extension of trace norm to the LRTC problem. In [5], they defined the trace norm of a tensor as a convex combination of trace norms of its unfolded matrices. In other words, the LRTC problem is converted into a convex combination of each mode unfolded matrix trace norm minimization. However, the tensor trace norm minimization (TTNM) problem has to be solved iteratively and involves multiple singular value decompositions (SVDs) of very large matrices at each iteration. Therefore, those algorithms for TTNM suffer from high computational cost $O(NI^{N+1})$, where the assumed size of an N-th order tensor is $I \times \cdots \times I$.

In this paper, we focus on two issues for the LRTC problem as in [39], i.e., the robustness of the given tensor rank and the computational cost. We propose a novel trace norm regularized CANDECOMP/PARAFAC decomposition (TNCP) method for simultaneous tensor decomposition and completion. We verify, with convincing experimental results on synthetic and real-world data, both the efficiency and effectiveness of our TNCP method. The main contributions of this paper are as follows.

1) We deduce the relation between the rank of each factor matrix and the corresponding unfolding of a tensor, and formulate a factor matrix rank minimization model.
2) We introduce a tractable relaxation of the rank function into our factor matrix rank minimization model, and then obtain a tractable convex combination problem of multiple (much-smaller) factor matrix trace norm minimization.
3) We present an efficient algorithm based on the alternating direction method of multipliers (ADMM) to solve the proposed problem.

The remainder of this paper is organized as follows. We review preliminaries and related work in Sections II and III. In Section IV, we present a novel trace norm regularized CP decomposition model for LRTC, and develop an efficient ADMM algorithm in Section V. Section VI gives the empirical results, and Section VII concludes this paper and points out some potential extensions for future work.

II. Notation

Before reviewing related work, we first introduce basics of tensor notion and terminology. Scalars are denoted by lower-case letters such as i, j, k, and vectors by bold lower-case letters such as a, b, c. Matrices are denoted by upper-case letters, e.g., X, and their entries by lower-case letters, e.g., x_{ij}. Moreover, $\|X\|_*$ denotes the trace norm of the matrix X, i.e., the sum of its singular values. An Nth-order tensor is denoted by a calligraphic letter, e.g., $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, and its entries are denoted by $x_{i_1,\ldots,i_N}$. The order N of a tensor is the number of dimensions, also known as ways or modes. Fibers are the higher-order analog of matrix rows and columns. A fiber is defined by fixing every index but one. The mode-n fibers are all vectors $\mathrm{x}_{i_1,\ldots,i_{n-1},i_{n+1},\ldots,i_N}$ that are obtained by fixing the values of $\{i_1, \ldots, i_N\} \backslash i_n$.

The mode-n unfolding, also known as matricization, of an Nth-order tensor $\mathcal{X}$ is denoted by $X_{(n)} \in \mathbb{R}^{I_n \times \prod_{j \neq n} I_j}$ and a rearrangement of the entries of $\mathcal{X}$ into the matrix $X_{(n)}$ such that the mode-n fiber becomes the row index and all other $(N-1)$ modes become column indices in lexicographical order. The tensor element $(i_1, i_2, \ldots, i_N)$ is mapped to the matrix element (i_n, j), where

$$j = 1 + \sum_{k=1, k \neq n}^{N} (i_k - 1) J_k \text{ with } J_k = \prod_{m=1, m \neq n}^{k-1} I_m.$$

The inner product of two same-sized tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is the sum of the product of their entries

$$< \mathcal{A}, \mathcal{B} > := \sum_{i_1,\ldots,i_N} a_{i_1,\ldots,i_N} b_{i_1,\ldots,i_N}.$$

The Frobenius norm of an Nth-order $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is defined as

$$\|\mathcal{X}\|_F := \sqrt{\sum\nolimits_{i_1=1}^{I_1} \cdots \sum\nolimits_{i_N=1}^{I_N} x_{i_1,\ldots,i_N}^2}.$$

Let A and B be two matrices of size $m \times n$ and $p \times q$, respectively. The Kronecker product of two matrices A and B, denoted by $A \otimes B$, is an $mp \times nq$ matrix given by

$$A \otimes B := \left[a_{ij} B\right]_{mp \times nq}.$$

Let $A = [\mathrm{a}_1\, \mathrm{a}_2\, \cdots\, \mathrm{a}_r]$ and $B = [\mathrm{b}_1\, \mathrm{b}_2\, \cdots\, \mathrm{b}_r]$ be two column matrices of size $m \times r$ and $n \times r$, respectively. Then the Khatri–Rao product of two matrices A and B is defined as the column-wise Kronecker product and represented by $\odot$

$$A \odot B := [\mathrm{a}_1 \otimes \mathrm{b}_1\, \mathrm{a}_2 \otimes \mathrm{b}_2\, \cdots\, \mathrm{a}_r \otimes \mathrm{b}_r].$$

III. Related Work

A. LRTC

For the LRTC problem, Liu *et al.* [4] and Signoretto *et al.* [38] have proposed an extension of low-rank matrix completion concept to tensor data. With an exact analog to the definition of the matrix rank, the rank of a tensor $\mathcal{X}$, denoted as $\mathrm{rank}(\mathcal{X})$, is defined as follows.

Definition 1: The rank of a tensor is the smallest number of rank-one tensors, that generate the tensor as their sum, i.e.,

the smallest R such that

$$\mathcal{X} = \sum_{i=1}^{R} \mathrm{a}_i^1 \circ \mathrm{a}_i^2 \circ \cdots \circ \mathrm{a}_i^N$$

where $\circ$ denotes the outer product of some vectors, that is $(\mathrm{a}_i^1 \circ \mathrm{a}_i^2 \circ \cdots \circ \mathrm{a}_i^N)_{i_1,i_2\ldots,i_N} = [\mathrm{a}_i^1]_{i_1}[\mathrm{a}_i^2]_{i_2} \cdots [\mathrm{a}_i^N]_{i_N}$.

This definition of the rank of a tensor is an extension of the rank of a matrix, but with different properties. One difference is that the rank of a tensor is difficult to handle, as there is no straightforward way to determine the rank of a given tensor. In fact, the problem is NP-hard [14], [40], [41]. Fortunately, the n-rank of a tensor $\mathcal{X}$, denoted as n-rank($\mathcal{X}$), is easy to compute, which consists of the matrix ranks of mode-n unfolding of the tensor.

Definition 2: The n-rank of an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is the tuple of the ranks of the mode-n unfoldings

$$n\text{-rank}(\mathcal{X}) = \big(\text{rank}(X_{(1)}), \text{rank}(X_{(2)}), \ldots, \text{rank}(X_{(N)})\big).$$

According to Definition 2, the LRTC problem with incomplete observations is formulated as a multiobjective problem as in [42]

$$\min_{\mathcal{X}} n\text{-rank}(\mathcal{X}), \quad \text{s.t., } \mathcal{X}_\Omega = \mathcal{T}_\Omega \tag{1}$$

where the entries of $\mathcal{T}$ in the index set Ω are given while the remaining elements are missing. In order to keep things simple, the weighted sum of the ranks of the different unfolded matrices is used to take the place of the n-rank of the involved tensor.

Liu *et al.* [5] and Gandy *et al.* [20] have proposed to minimize the weighted sum of the rank of each unfolding as an objective function

$$\min_{\mathcal{X}} \sum_{n=1}^{N} \alpha_n \text{rank}(X_{(n)}), \quad \text{s.t., } \mathcal{X}_\Omega = \mathcal{T}_\Omega \tag{2}$$

where α_ns are prespecified weights, and $X_{(n)}$ denotes the unfolded matrix along the nth mode. In addition, Gandy *et al.* [20] have presented an unweighted model, i.e., a special case of the model (2), where $\alpha_n = 1$, $n = 1, \ldots, N$. The nonconvex problem (2) can be solved by its convex relaxation replacing the rank of the matrix with the trace norm

$$\min_{\mathcal{X}} \sum_{n=1}^{N} \alpha_n \|X_{(n)}\|_*, \quad \text{s.t., } \mathcal{X}_\Omega = \mathcal{T}_\Omega. \tag{3}$$

In the presence of noise, we obtain a corresponding unconstrained formulation

$$\min_{\mathcal{X}} \sum_{n=1}^{N} \alpha_n \|X_{(n)}\|_* + \frac{\lambda}{2} \|\mathcal{P}_\Omega(\mathcal{X}) - \mathcal{P}_\Omega(\mathcal{T})\|_F^2 \tag{4}$$

where $\mathcal{P}_\Omega$ keeps the entries in Ω and zeros out others, and $\lambda > 0$ is a regularization parameter.

In fact, each mode-n unfolding $X_{(n)}$ shares the same entries and cannot be optimized independently. Note that both discussed models (3) and (4) are difficult to solve due to the interdependent matrix trace norm terms [5]. Hence, we need to perform variable splitting and introduce a separate variable to each unfolding of the tensor $\mathcal{X}$. Recently, Liu *et al.* [5] proposed three efficient algorithms to solve the problem (3). In addition, there are some similar tensor completion methods in [20], [36], and [38]. However, all those algorithms have to be solved iteratively and involve multiple SVDs of very large matrices at each iteration. Besides, many additional variables are introduced to split those interdependent terms in (3) and (4) such that they can be solved independently. Thus, all those algorithms suffer from high computational cost and are even not applicable for large-scale problems.

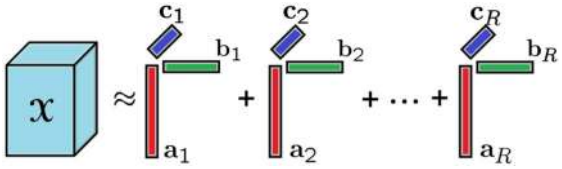


Fig. 1. Illustration of an R-component CP model for a third-order tensor.

More recently, it has been shown that the weighted sum of trace norm model mentioned above can be substantially suboptimal [42], [43]. To address this problem, Mu *et al.* [42] proposed a more "square" convex model for recovering $\mathcal{X}$ as follows:

$$\min_{\mathcal{X}} \|X_{[j]}\|_*, \quad \text{s.t., } \mathcal{X}_\Omega = \mathcal{T}_\Omega \tag{5}$$

where $X_{[j]}$ is defined as

$$X_{[j]} = \text{reshape}\left(X_{(1)}, \prod_{n \le j} I_n, \prod_{n > j} I_n\right)$$

and $j \in \{1, 2, \ldots, N\}$ is chosen to make $\prod_{n \le j} I_n$ as close to $\prod_{n > j} I_n$ as possible. If the order of the involved tensor is no more than three, the model (5) is the same as the trace norm minimization method on the corresponding mode unfolding, hence its algorithm may not perform as well as those algorithms for (3) and (4). However, for a tensor of order higher than three, it has been shown in [42] that the model (5) can exactly recover the tensor from far fewer observed entries than those required by (3) and (4).

B. Tensor Decompositions for Completion

Next, we will introduce two tensor decomposition models for LRTC problems. Acar *et al.* [18] presented a weighted least squares model with missing data

$$\min_{U_n} \|\mathcal{W} * (\mathcal{T} - U_1 \circ U_2 \cdots \circ U_N)\|_F^2 \tag{6}$$

where $*$ denotes the Hadamard (elementwise) product, and $U_n \in \mathbb{R}^{I_n \times R}$ is referred to as the factor matrix which is the combination of the vectors from the rank-one components (e.g., $U_1 = [\mathrm{a}_1, \mathrm{a}_2, \ldots, \mathrm{a}_R]$, as shown in Fig. 1), and R is a positive integer. Moreover, $\mathcal{W}$ is a nonnegative weight tensor with the same size as $\mathcal{T}$

$$w_{i_1,i_2\ldots,i_N} = \begin{cases} 1 & \text{if } t_{i_1,i_2\ldots,i_N} \text{ is known} \\ 0 & \text{otherwise.} \end{cases}$$

In this sense, the approximation generally requires significantly less storage $O(R\Sigma I_n)$ than the original tensor. Hence, we are particularly interested in extending the CP decomposition for LRTC problems.

In [34], the weighted Tucker decomposition model is formulated as follows:

$$\min_{\mathcal{C},U_n} \|\mathcal{W} * (\mathcal{T} - \mathcal{C} \times_1 U_1 \times_2 \cdots \times_N U_N)\|_F^2 \tag{7}$$

where $U_n \in \mathbb{R}^{I_n \times R_n}$, $\mathcal{C} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$ is a core tensor with the given ranks $(R_1, \ldots, R_N)$, and $\times_n$ denotes the n-mode product (please see [14]).

Recently, some extensions of both decomposition methods and corresponding algorithms are developed for tensor estimation problems. An alternating proximal gradient method [44] is proposed for nonnegative tensor factorization and completion. However, for those methods, a suitable rank value needs to be given, and it has been shown that both the Tucker and the CP models are usually sensitive to the given ranks due to their least-squares formulations, and they have poor performance while real-world data have a high rank [5], [35].

IV. Factor Matrix Trace Norm Minimization Model

The major bottleneck of the existing LRTC algorithms for solving the problems (3) and (4) is the high computational cost of multiple SVDs of very large matrices in each iteration. To address this difficulty, we propose a novel model that minimizes the weighted sum of the ranks of factor matrices of the CP decomposition instead of the n-rank of the involved tensor.

A. Rank of Factor Matrices

Let $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ be an Nth-order low-rank tensor with tensor rank r, then the CP form of $\mathcal{X}$ is rewritten as follows:

$$\mathcal{X} = U_1 \circ U_2 \cdots \circ U_N = \sum_{i=1}^{r} \mathrm{u}_1^{(i)} \circ \mathrm{u}_2^{(i)} \cdots \circ \mathrm{u}_N^{(i)} \tag{8}$$

where $U_n = [\mathrm{u}_n^{(1)} \mathrm{u}_n^{(2)} \cdots \mathrm{u}_n^{(r)}] \in \mathbb{R}^{I_n \times r}$ denotes the factor matrix of $\mathcal{X}$ for $n = 1, \ldots, N$.

Theorem 1: Let $X_{(n)}$ be the mode-n unfolding of the tensor $\mathcal{X}$ of rank r, and U_n be the factor matrix for all $n \in \{1, \ldots, N\}$. Then

$$\operatorname{rank}(X_{(n)}) \le \operatorname{rank}(U_n), \quad \forall n = 1, \ldots, N. \tag{9}$$

Proof: Since $\mathcal{X} = U_1 \circ U_2 \cdots \circ U_N$, we have

$$X_{(n)} = U_n (U_N \odot \cdots U_{n+1} \odot U_{n-1} \odot \cdots U_1)^T \quad \forall n = 1, \ldots, N.$$

Thus

$$\begin{aligned} \operatorname{rank}(X_{(n)}) &= \operatorname{rank}\left(U_n (U_N \odot \cdots U_{n+1} \odot U_{n-1} \odot \cdots U_1)^T\right) \\ &\le \operatorname{rank}(U_n), \quad \forall n = 1, \ldots, N. \end{aligned}$$

■

From the above theorem, it is clear that the factor matrices $U_n \in \mathbb{R}^{I_n \times r}$, $n = 1, \ldots, N$, have a much smaller size than the mode-n unfolding $X_{(n)} \in \mathbb{R}^{I_n \times \prod_{j \ne n} I_j}$, while the rank of each factor matrix is an upper bound on the rank of its corresponding unfolding of the tensor. In the following section, we propose a new model that uses the rank of the factor matrices U_n, i.e., $\operatorname{rank}(U_n)$, instead of the mode-n rank of the tensor, i.e., $\operatorname{rank}(X_{(n)})$.

B. Our Model

Suppose the unknown tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is low rank, our rank minimization model based on the CP decomposition for LRTC can be expressed as follows:

$$\begin{aligned} &\min_{\mathcal{X},\, U_n} \sum_{n=1}^{N} \alpha_n \operatorname{rank}(U_n) \\ &\text{s.t., } \mathcal{X}_\Omega = \mathcal{T}_\Omega, \quad \mathcal{X} = U_1 \circ U_2 \cdots \circ U_N \end{aligned} \tag{10}$$

where $U_n \in \mathbb{R}^{I_n \times R}$ for $n = 1, \ldots, N$, and R denotes a upper bound of the tensor rank and is a positive integer. Moreover, the factor matrix rank minimization is a relaxation form of the mode-n rank minimization of the involved tensor. In addition, Bro and Kiers [45] have provided some tensor rank estimation strategies to compute a good value r for the rank of the tensor. Thus, we only set a relatively large integer R such that $R \ge r$.

Due to the discrete nature of the rank, the model (10) can be relaxed by replacing the rank function with the trace norm as follows:

$$\begin{aligned} &\min_{\mathcal{X},\, U_n} \sum_{n=1}^{N} \alpha_n \|U_n\|_* \\ &\text{s.t., } \mathcal{X}_\Omega = \mathcal{T}_\Omega, \quad \mathcal{X} = U_1 \circ U_2 \cdots \circ U_N. \end{aligned} \tag{11}$$

Furthermore, the relaxation version of (11) is formulated as follows:

$$\begin{aligned} &\min_{\mathcal{X},\, U_n} \sum_{n=1}^{N} \alpha_n \|U_n\|_* + \frac{\lambda}{2} \|\mathcal{X} - U_1 \circ U_2 \cdots \circ U_N\|_F^2 \\ &\text{s.t.,} \quad \mathcal{X}_\Omega = \mathcal{T}_\Omega. \end{aligned} \tag{12}$$

The model (12) is called the TNCP method for simultaneous tensor decomposition and completion. We only need to perform SVDs on some much smaller scale factor matrices, and thus our TNCP method is very efficient. Meanwhile, TNCP is much more robust to the given tensor rank R, which will be confirmed by our experimental results in Section VI.

V. Optimization Algorithm

Recently, it has been shown in [28], [46], and [47] that the ADMM is very efficient for some convex or nonconvex programming problems for various applications. We also refer to [5], [20], [48], [49], and [50] for some recently exploited applications of ADMM. In this paper, we also propose an ADMM algorithm for solving the proposed model (12).

Following [5], some auxiliary variables with much smaller sizes, $M_n \in \mathbb{R}^{I_n \times R}$, $n = 1, \ldots, N$, are introduced into our model (12), and then the problem (12) is reformulated into the following equivalent form:

$$\begin{aligned} &\min_{\mathcal{X},\, U_n, M_n} \sum_{n=1}^{N} \alpha_n \|M_n\|_* + \frac{\lambda}{2} \|\mathcal{X} - U_1 \circ \cdots \circ U_N\|_F^2 \\ &\text{s.t., } \mathcal{X}_\Omega = \mathcal{T}_\Omega,\ M_n = U_n,\ n = 1, \ldots, N. \end{aligned} \tag{13}$$

A. Solving Scheme

The partial augmented Lagrangian function of (13) is

$$\begin{aligned}\mathcal{L}_\mu &(U_1, \ldots, U_N, M_1, \ldots, M_N, \mathcal{X}, Y_1, \ldots, Y_N) \\ &= \sum_{n=1}^{N} \alpha_n \|M_n\|_* + \frac{\lambda}{2} \|\mathcal{X} - U_1 \circ U_2 \cdots \circ U_N\|_F^2 \\ &\quad + \sum_{n=1}^{N} \left(\langle Y_n,\, M_n - U_n \rangle + \frac{\mu}{2} \|M_n - U_n\|_F^2 \right) \end{aligned} \tag{14}$$

where $Y_n \in \mathbb{R}^{I_n \times R}$ is the matrix of Lagrange multipliers for $n = 1, \ldots, N$, and $\mu > 0$ is a penalty parameter. We present an ADMM iterative scheme to successively minimize $\mathcal{L}_\mu$ over $(\{U_1, \ldots, U_N\}, \{M_1, \ldots, M_N\}, \mathcal{X})$ and then update $\{Y_1, \ldots, Y_N\}$ as follows:

$$\min_{\{U_1, \ldots, U_N\}} \mathcal{L}_{\mu^k} \left(U_1 \ldots U_N, M_1^k \ldots M_N^k, \mathcal{X}^k, Y_1^k \ldots Y_N^k \right) \tag{15}$$

$$\min_{\{M_1, \ldots, M_N\}} \mathcal{L}_{\mu^k} \left(U_1^{k+1} \ldots U_N^{k+1}, M_1 \ldots M_N, \mathcal{X}^k, Y_1^k \ldots Y_N^k \right) \tag{16}$$

$$\begin{aligned} \min_{\mathcal{X}} \mathcal{L}_{\mu^k} &\left(U_1^{k+1} \ldots U_N^{k+1}, M_1^{k+1} \ldots M_N^{k+1}, \mathcal{X}, Y_1^k \ldots Y_N^k \right) \\ \text{s.t.,} \quad & \mathcal{X}_\Omega = \mathcal{T}_\Omega \end{aligned} \tag{17}$$

$$Y_n^{k+1} = Y_n^k + \mu^k \left(M_n^{k+1} - U_n^{k+1} \right), \quad n = 1, \ldots, N. \tag{18}$$

Updating $\{U_1^{k+1}, \ldots, U_N^{k+1}\}$: To update the variables $(U_1, \ldots, U_N)$, the optimization problem (15) is rewritten as follows:

$$\begin{aligned} \min_{U_n} \frac{\lambda}{2} &\left\| \mathcal{X}^k - U_1 \circ \cdots \circ U_N \right\|_F^2 \\ &+ \sum_{n=1}^{N} \frac{\mu^k}{2} \left\| U_n - M_n^k - Y_n^k / \mu^k \right\|_F^2. \end{aligned} \tag{19}$$

To update U_n, $n = 1, \ldots, N$, with fixing the other variables, then the problem (19) becomes a smooth optimization problem. Let $B_n = (U_N^k \odot \cdots U_{n+1}^k \odot U_{n-1}^{k+1} \odot \cdots U_1^{k+1})^T$, then the resulting subproblem with respect to U_n is formulated as follows:

$$\min_{U_n} \frac{\lambda}{2} \left\| U_n B_n - X_{(n)}^k \right\|_F^2 + \frac{\mu^k}{2} \left\| U_n - M_n^k - Y_n^k / \mu^k \right\|_F^2. \tag{20}$$

Thus, U_n is efficiently updated by solving the optimization problem (20)

$$U_n^{k+1} = \left(\lambda X_{(n)}^k B_n^T + \mu^k M_n^k + Y_n^k \right) \left(\lambda B_n B_n^T + \mu^k I \right)^{-1}. \tag{21}$$

Updating $\{M_1^{k+1}, \ldots, M_N^{k+1}\}$: To update the variables $M_n\,(n = 1, \ldots, N)$ with fixing other variables, the optimization problem (16) is reformulated concretely as follows:

$$\min_{M_n} \alpha_n \|M_n\|_* + \frac{\mu^k}{2} \left\| M_n - U_n^{k+1} + Y_n^k / \mu^k \right\|_F^2. \tag{22}$$

Following [51], a closed-form solution to the problem (22) can be obtained easily as follows:

$$M_n^{k+1} = \mathrm{SVT}_{\alpha_n / \mu^k} \left(U_n^{k+1} - Y_n^k / \mu^k \right) \tag{23}$$

where $\mathrm{SVT}_\delta(A) = U \mathrm{diag}(\{(\sigma - \delta)_+\}) V^T$ is a singular value thresholding (SVT) operator, the SVD of A is given by $A = U \mathrm{diag}(\{\sigma_i\}_{1 \le i \le r}) V^T$, $t_+ = \max(0, t)$, and $\max(\cdot, \cdot)$ should be understood element-wise. The computational complexity of the SVT operator on $U_n^{k+1} - Y_n^k / \mu^k$ is $O(I_n R^2)$. Thus, our TNCP method has a significantly lower complexity $O(\sum_n I_n R^2)$ for the soft-thresholding operation than the other TTNM algorithms, which require to perform SVDs on the much larger unfolding $X_{(n)}$ with size of $I_n \times \prod_{j \ne n} I_j$ $(n = 1, \ldots, N)$ in each iteration, and then have a much higher computational complexity of $O(\sum_n I_n^2 \times \prod_{j \ne n} I_j)$.

Updating $\mathcal{X}^{k+1}$: To update the variable $\mathcal{X}$, we have the following subproblem:

$$\min_{\mathcal{X}} \left\| \mathcal{X} - U_1^{k+1} \circ \cdots \circ U_N^{k+1} \right\|_F^2, \quad \text{s.t., } \mathcal{X}_\Omega = \mathcal{T}_\Omega. \tag{24}$$

By introducing Lagrangian multiplier $\mathcal{Q} \in \mathbb{R}^{I_1 \times I_2 \ldots \times I_N}$ for the constraint $\mathcal{X}_\Omega = \mathcal{T}_\Omega$, we write the Lagrangian function of (24) as follows:

$$F(\mathcal{X}, \mathcal{Q}) = \left\| \mathcal{X} - U_1^{k+1} \circ \cdots \circ U_N^{k+1} \right\|_F^2 + \langle \mathcal{Q},\ \mathcal{X}_\Omega - \mathcal{T}_\Omega \rangle.$$

Letting $\nabla_{(\mathcal{X}, \mathcal{Q})} F = 0$, we have the Karush–Kuhn–Tucker (KKT) conditions

$$\begin{aligned} 2(\mathcal{X} - U_1^{k+1} \circ \cdots \circ U_N^{k+1}) + \mathcal{Q}_\Omega &= 0 \\ \mathcal{X}_\Omega - \mathcal{T}_\Omega &= 0. \end{aligned}$$

By deriving simply the KKT conditions, we have

$$\mathcal{X}^{k+1} = \mathcal{P}_\Omega(\mathcal{T}) + \mathcal{P}_{\Omega^C} \left(U_1^{k+1} \circ \cdots \circ U_N^{k+1} \right) \tag{25}$$

where Ω^C is the complement of Ω, i.e., the set of indexes of the unobserved entries.

Based on the above analysis, we develop an ADMM algorithm for the tensor decomposition and completion problem (12), as outlined in Algorithm 1.[1] This algorithm can also be accelerated by adaptively changing μ. An efficient strategy [28], [52] is to let $\mu = \mu^0$ (the initialization in Algorithm 1) and increase μ^k iteratively by $\mu^{k+1} = \rho \mu^k$, where $\rho \in (1.0, 1.1]$ in general and μ^0 is a small constant. Moreover, the stability and efficiency of our TNCP algorithm are verified by experiments in Section VI.

B. Convergence Analysis

Nonetheless, the proposed model (12) is nonconvex. In the following, we will present the convergence analysis of Algorithm 1 for solving the problem (12).

Theorem 2: Let $(\{M_1^k, \ldots, M_N^k\}, \{U_1^k, \ldots, U_N^k\}, \mathcal{X}^k)$ be a sequence generated by Algorithm 1, then we have the following conclusions.

1) $\{M_1^k, \ldots, M_N^k\}$, $\{U_1^k, \ldots, U_N^k\}$ and $\mathcal{X}^k$ are Cauchy sequences.
2) If $\lim_{k \to \infty} \mu^k (M_n^k - M_n^{k-1}) = 0$, $n = 1, \ldots, N$, $(\{U_1^k, \ldots, U_N^k\}, \mathcal{X}^k)$ converges to a KKT point of the problem (12).

[1] rand denotes a MATLAB function that generates a matrix of uniformly distributed random numbers between 0 and 1.

Algorithm 1 Solving the TNCP Model (12) Via ADMM

Input: $\mathcal{P}_\Omega(\mathcal{T})$, R and λ.

Output: $Y_n^0 = M_n^0 = 0$, $U_n^0 = \text{rand}(I_n, R)$, $n = 1, \ldots, N$, $\mu^0 = 10^{-6}$, $\mu_{\max} = 10^{10}$, $\rho = 1.10$, and $\text{tol} = 10^{-5}$.

while not converged **do**
  for $n = 1 : N$ **do**
    Update U_n^{k+1} by (21);
    Update M_n^{k+1} by (23);
  end for
  Update $\mathcal{X}^{k+1}$ by (25);
  for $n = 1 : N$ **do**
    Update Y_n^{k+1} by $Y_n^{k+1} = Y_n^k + \mu^k(M_n^{k+1} - U_n^{k+1})$;
  end for
  Update μ^{k+1} by $\mu^{k+1} = \min(\rho\mu^k, \mu_{\max})$;
  Check the convergence condition,
    $\max\{\|M_n^{k+1} - U_n^{k+1}\|_F, n = 1, \ldots, N\} < \text{tol}$;
end while

Output: $\mathcal{X}$ and U_n, $n = 1, \ldots, N$.

The proof sketch of Theorem 2 is similar to that in [28]. We first prove the boundedness of multipliers and some variables of Algorithm 1, and then we analyze the convergence of Algorithm 1. To prove the boundedness, we first give the following lemmas.

Lemma 1 [28], [29]: Let $\mathcal{X}$ be a real Hilbert space endowed with an inner product $\langle\cdot\rangle$ and a corresponding norm $\|\cdot\|$ such as the trace norm, and $y \in \partial\|x\|$, where $\partial\|\cdot\|$ denotes the subgradient. Then $\|y\|^* = 1$ if $x \neq 0$, and $\|y\|^* \leq 1$ if $x = 0$, where $\|\cdot\|^*$ is the dual norm of $\|\cdot\|$. For example, the dual norm of the trace norm is the spectral norm, $\|\cdot\|_2$, i.e., the largest singular value.

Lemma 2: Let $Y_n^{k+1} = Y_n^k + \mu^k(M_n^{k+1} - U_n^{k+1})$, then the sequences $\{M_n^k\}$, $\{Y_n^k\}$ and $\{U_n^k\}$, $n = 1, \ldots, N$, produced by Algorithm 1 are bounded.

Proof: Let $\mathcal{M}^k = (M_1^k, \ldots, M_N^k)$, $\mathcal{U}^k = (U_1^k, \ldots, U_N^k)$ and $\mathcal{Y}^k = (Y_1^k, \ldots, Y_N^k)$. By the optimality condition of (19) for any $n \in \{1, \ldots, N\}$, we have

$$0 \in \partial_{M_n}\mathcal{L}_{\mu^k}\left(\mathcal{U}^{k+1}, \mathcal{M}^{k+1}, \mathcal{X}^k, \mathcal{Y}^k\right)$$

i.e., $Y_n^k + \mu^k(M_n^{k+1} - U_n^{k+1}) \in \alpha_n\partial\|M_n^{k+1}\|_*$, and $Y_n^{k+1} \in \alpha_n\partial\|M_n^{k+1}\|_*$. By Lemma 1, we have $\|Y_n^{k+1}\|_2 \leq \alpha_n$. Hence, the sequence $\{Y_n^k\}$ is bounded for all $n \in \{1, \ldots, N\}$.

By the iteration procedure, we have

$$\begin{aligned}
&\mathcal{L}_{\mu^k}\left(\mathcal{M}^{k+1}, \mathcal{X}^{k+1}, \mathcal{U}^{k+1}, \mathcal{Y}^k\right)\\
&\quad\leq \mathcal{L}_{\mu^k}\left(\mathcal{M}^k, \mathcal{X}^k, \mathcal{U}^k, \mathcal{Y}^k\right)\\
&\quad= \mathcal{L}_{\mu^{k-1}}\left(\mathcal{M}^k, \mathcal{X}^k, \mathcal{U}^k, \mathcal{Y}^{k-1}\right) + \beta^k\sum_{n=1}^N\left\|Y_n^k - Y_n^{k-1}\right\|_F^2
\end{aligned}$$

where $\beta^k = (\mu^{k-1} + \mu^k)/2(\mu^{k-1})^2$ and $\mu^k = \rho\mu^{k-1}$. Furthermore, we have

$$\sum_{k=1}^\infty \frac{\mu^{k-1} + \mu^k}{2\left(\mu^{k-1}\right)^2} = \frac{\rho(\rho+1)}{2\mu^0(\rho-1)} < \infty.$$

Hence, $\{\mathcal{L}_{\mu^{k-1}}(\mathcal{M}^k, \mathcal{X}^k, \mathcal{U}^k, \mathcal{Y}^{k-1})\}$ is upper-bounded due to the boundedness of $\{Y_n^k\}$ for $n = 1, \ldots, N$.

$$\begin{aligned}
&\sum_{n=1}^N \alpha_n\left\|M_n^k\right\|_* + \frac{\lambda}{2}\left\|\mathcal{X}^k - U_1^k \circ U_2^k \cdots \circ U_N^k\right\|_F^2\\
&= \mathcal{L}_{\mu^{k-1}}\left(\mathcal{M}^k, \mathcal{X}^k, \mathcal{U}^k, \mathcal{Y}^{k-1}\right) - \frac{1}{2}\sum_{n=1}^N \frac{\left\|Y_n^k\right\|_F^2 - \left\|Y_n^{k-1}\right\|_F^2}{\mu^{k-1}}
\end{aligned}$$

is upper-bounded. Thus, $\{M_n^k\}$, $n = 1, \ldots, N$, are bounded.

By $U_n^k = M_n^k - (Y_n^k - Y_n^{k-1})/\mu^{k-1}$, and $\{M_n^k\}$, $\{Y_n^k\}$, $n = 1, \ldots, N$, are bounded, then $\{U_n^k\}$, $n = 1, \ldots, N$, are also bounded. ∎

We prove Theorem 2 as follows.

Proof: 1) By $(M_n^{k+1} - U_n^{k+1}) = (\mu^k)^{-1}(Y_n^{k+1} - Y_n^k)$, the boundedness of $\{Y_n^k\}$ and $\lim_{k\to\infty}\mu^k = \infty$, we have

$$\lim_{k\to\infty}\left\|M_n^{k+1} - U_n^{k+1}\right\|_F = 0, \text{ for } n = 1, \ldots, N. \tag{26}$$

Thus, $(\{M_1^k, \ldots, M_N^k\}, \{U_1^k, \ldots, U_N^k\})$ approaches to a feasible solution.

Furthermore, we prove that the sequences $\{U_n^k\}$, $n = 1, \ldots, N$, are all Cauchy sequences.

By $Y_n^k = Y_n^{k-1} + \mu^{k-1}(M_n^k - U_n^k)$ and $\mu^k = \rho\mu^{k-1}$, then the optimality condition of (20) with respect to U_n^{k+1} is rewritten as follows:

$$\begin{aligned}
&\lambda\left(U_n^{k+1}B_n - X_{(n)}^k\right)B_n^T + \mu^k\left(U_n^{k+1} - M_n^k - \frac{Y_n^k}{\mu^k}\right)\\
&\quad= \lambda\left(U_n^{k+1}B_n - X_{(n)}^k\right)B_n^T + \mu^k\left(U_n^{k+1} - U_n^k\right)\\
&\qquad+ \mu^k\left(U_n^k - M_n^k - \frac{Y_n^{k-1}}{\mu^{k-1}} + \frac{Y_n^{k-1}}{\mu^{k-1}} - \frac{Y_n^k}{\mu^k}\right)\\
&\quad= \mu^k\left(U_n^{k+1} - U_n^k\right) + \lambda\left(U_n^{k+1}B_n - X_{(n)}^k\right)B_n^T\\
&\qquad+ \rho Y_n^{k-1} - (\rho+1)Y_n^k = 0.
\end{aligned} \tag{27}$$

By (27), we have

$$\begin{aligned}
&U_n^{k+1} - U_n^k\\
&\quad= \frac{(\rho+1)Y_n^k - \rho Y_n^{k-1} - \lambda\left(U_n^{k+1}B_n - X_{(n)}^k\right)B_n^T}{\mu^k}.
\end{aligned} \tag{28}$$

By the boundedness of $\{U_n^k\}$ in Lemma 2, thus B_n is bounded. Furthermore, we have $\|U_n^{k+1} - U_n^k\|_F = O((\mu^k)^{-1})$ and $\sum_{k=1}^\infty(\mu^{k-1})^{-1} = \rho/(\mu^0(\rho-1)) < \infty$. Hence, $\{U_n^k\}$ is a Cauchy sequence, and it has a limit point, U_n^∞, for all $n \in \{1, \ldots, N\}$.

Similarly, $\{M_n^k\}$, $n = 1, \ldots, N$, and $\{\mathcal{X}^k\}$ are also Cauchy sequences.

2) The KKT conditions of (12) are

$$\begin{aligned}
&0 \in \alpha_n\partial\left\|U_n^*\right\|_* + \lambda\left(U_n^*B_n^* - X_{(n)}^*\right)\left(B_n^*\right)^T,\ \forall n \in \{1, \ldots, N\}\\
&\mathcal{X}_\Omega^* = \mathcal{T}_\Omega, \quad \mathcal{X}_{\Omega^C}^* = \left(U_1^* \circ \ldots \circ U_N^*\right)_{\Omega^C}
\end{aligned}$$

where $B_n^* = (U_N^* \odot \ldots U_{n+1}^* \odot U_{n-1}^* \ldots \odot U_1^*)^T$.

TABLE I
COMPLEXITIES PER ITERATION OF MAJOR COMPUTATIONS IN TENSOR COMPLETION ALGORITHMS

Algorithms	Complexity
WCP [18]	$O(2(N+1)R\Pi_{j=1}^{N} I_j)$
WTucker [34]	$O(2(N+1)R\Pi_{j=1}^{N} I_j)$
TTNM algorithms [5], [11], [33]	$O(\sum_{n=1}^{N} I_n \Pi_{j=1}^{N} I_j)$
TNCP	$O((N+1)R\Pi_{j=1}^{N} I_j)$

According to Algorithm 1, the first-order optimal condition of (20) at the $(k+1)$th iteration is

$$0 = \lambda \left(U_n^{k+1} B_n - X_{(n)}^k\right) B_n^T + \mu^k \left(U_n^{k+1} - M_n^k - Y_n^k/\mu^k\right). \tag{29}$$

The first-order optimal condition of the problem (22) is

$$0 \in \alpha_n \partial \left\|M_n^{k+1}\right\|_* + \mu^k \left(U_n^{k+1} - M_n^{k+1} - Y_n^k/\mu^k\right). \tag{30}$$

Since $\{M_n^k\}$, $\{U_n^k\}$, $n = 1, \ldots, N$, and $\{\mathcal{X}^k\}$ are all Cauchy sequences, M_n^∞, U_n^∞, $n = 1, \ldots, N$, and $\mathcal{X}^\infty$ are their limit points, respectively. By the result 1), we have $M_n^\infty = U_n^\infty$ for $n = 1, \ldots, N$. By (29) and (30), we have

$$0 \in \alpha_n \partial \left\|U_n^\infty\right\|_* + \lambda \left(U_n^\infty B_n^\infty - \mathcal{X}_{(n)}^\infty\right),\ n = 1, \ldots, N. \tag{31}$$

By (25), we have $\mathcal{X}_\Omega^\infty = \mathcal{T}_\Omega$ and $\mathcal{X}_{\Omega^C}^\infty = (U_1^\infty \circ \ldots \circ U_N^\infty)_{\Omega^C}$. Hence, the sequence $(\{U_1^k, \ldots, U_N^k\}, \mathcal{X}^k)$ generated by Algorithm 1 converges the KKT point of (12). ■

C. Complexity Analysis

We analyze the time complexity of our TNCP method as follows. For the LRTC problem (12), the main running time of Algorithm 1 is consumed by performing SVD for the SVT operator and some multiplications. The time complexity of performing SVD is $O(R^2 \sum_n I_n)$. Moreover, the time complexities of computing B_n and U_n in (21), and $\mathcal{X}$ in (25) are $O(R \sum_{i=1}^{N-1} \Pi_{j=N-i(j\neq n)}^{N} I_j + R\Pi_{j=1}^{N} I_j)$ and $O(R\Pi_{j=1}^{N} I_j)$. Thus, the total time complexity of Algorithm 1 is $O(T(N+1)R\Pi_{j=1}^{N} I_j)$, where T is the number of iterations. Moreover, Table I summarizes complexities of major computations in the two related tensor decomposition algorithms and the three TTNM algorithms. From Table I, we can see that although WCP [18] and WTucker [34] have time complexity similar to our TNCP method, they are much slower in practice than our TNCP method due to their Polak–Ribiere nonlinear conjugate gradient algorithm with a time-consuming line search [53]. From the space complexity view, since the decomposition rank R is in general smaller than I_n, $n = 1, \ldots, N$, the storage $O(R\Sigma_n I_n)$ of our TNCP decomposition form can be significantly smaller than that of the original tensor.

D. Connections to Existing Approaches

Our TNCP model (12) can be reformulated as follows:

$$\min_{U_n} \frac{1}{\lambda} \sum_{n=1}^{N} \alpha_n \|U_n\|_* + \frac{1}{2} \|\mathcal{W} * (\mathcal{T} - U_1 \circ U_2 \cdots \circ U_N)\|_F^2. \tag{32}$$

Thus, our model (12) is also a trace norm regularized least squares problem. When letting $\lambda \to \infty$, the model (32) degenerates to the weighted CP (WCP) model (6) in [18]. In other words, the weighted least squares model (6) is a special case of our TNCP method. Moreover, Allen *et al.* [54] proposed a sparse CP least squares method with l_1-norm penalties on each of factor matrix for structured tensor data. In this sense, our TNCP method is also a fast higher-order tensor decomposition method in the presence of missing data and gives a concise representation of the latent structure of incomplete tensors. When letting $\mathcal{W} = 1$, the model (32) degenerates to a trace norm regularized CP decomposition problem. Hence, the traditional CP decomposition method can be seen as a special case of TNCP by setting $\lambda \to \infty$ and $\mathcal{W} = 1$.

VI. EXPERIMENTS

In this section, we evaluate the effectiveness and efficiency of our TNCP method for LRTC problems on both synthetic and real-world tensor data, including link prediction, natural images, brain MRI images, and hyperspectral images in-painting. Except for large-scale link prediction, all the other experiments were performed with MATLAB 7.11 on an Intel Core 2 Duo (3.0 GHz) PC running Windows 7 with 2-GB main memory.

A. Synthetic Data

In this part, we generated a low-n-rank tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, which we used as ground truth data. The tensor data follows the Tucker model, i.e., $\mathcal{T} = \mathcal{C} \times_1 U_1 \times_2 \cdots \times_N U_N$, where the entries of the core tensor $\mathcal{C} \in \mathbb{R}^{r \times r \cdots \times r}$ are from a uniform distribution in the range [0, 1], and the entries of $U_n \in \mathbb{R}^{I_n \times r}$ are random samples drawn from a uniform distribution in the range $[-0.5, 0.5]$. With this construction, the n-rank of $\mathcal{T}$ equals $(r, r, \ldots, r)$ almost surely. The order of the tensors varies from three to five, and r is set to 5. Furthermore, we randomly sample a few entries from $\mathcal{T}$ and recover the whole tensor with various sampling rates (SRs) by our TNCP method and four state-of-the-art algorithms including weighted Tucker (WTucker)[2] [34], WCP [18], Latent[3] [33], and FaLRTC[4] [5].

We set tol $= 10^{-5}$ and maxiter $= 1000$ for all these algorithms. In the implementation of our TNCP method, we set the regularization parameter $\lambda = 10$. For TNCP and FaLRTC, the weight α_n is set to $1/N$ for all $n \in \{1, \ldots, N\}$, and the smoothing parameters of FaLRTC are set to $\mu_n = 5\alpha_n/I_n$. The other parameters of FaLRTC are set to their default values. For TNCP, WTucker, and WCP, the tensor rank is set to $R = 10$. The relative square error (RSE) of the recovered tensor $\mathcal{X}$ is given by RSE:$= \|\mathcal{X} - \mathcal{T}\|_F / \|\mathcal{T}\|_F$.

The average results (RSE and time cost) of ten independent runs are shown in Table II, where the order of tensor data varies from four to five, and SR is set to 20%, 50%, or

[2]http://www.lair.irb.hr/ikopriva/marko-filipovi.html
[3]http://ttic.uchicago.edu/~ryotat/softwares/tensor/
[4]http://pages.cs.wisc.edu/~ji-liu/

TABLE II
RSE AND TIME COST (SECONDS) COMPARISON ON SYNTHETIC DATA

(a) Tensor size: $20 \times 20 \times 20 \times 20 \times 20$

	WTucker		WCP		FaLRTC		Latent		TNCP	
SR	RSE	Time	RSE	Time	RSE	Time	RSE	Time	RSE	Time
20%	2.26e-01	4071.76	4.48e-01	2634.85	4.65e-01	1198.77	4.83e-01	8691.94	**1.58e-01**	**209.39**
50%	6.34e-02	2451.53	1.25e-01	2463.57	1.14e-01	821.46	1.67e-01	9634.13	**5.79e-02**	**232.65**
80%	3.17e-02	3034.17	7.26e-02	2396.24	5.77e-02	852.21	2.98e-02	12410.22	**2.73e-02**	**260.25**

(b) Tensor size: $30 \times 30 \times 30 \times 30$

	WTucker		WCP		FaLRTC		Latent		TNCP	
SR	RSE	Time	RSE	Time	RSE	Time	RSE	Time	RSE	Time
20%	1.85e-01	502.93	9.72e-02	708.04	4.48e-01	359.43	3.50e-01	2380.75	**9.48e-02**	**34.59**
50%	6.59e-02	439.52	7.13e-02	645.63	1.16e-01	512.85	6.37e-02	2582.56	**5.63e-02**	**36.35**
80%	4.84e-02	490.86	5.40e-02	669.63	6.48e-02	473.20	2.93e-02	2694.43	**1.85e-02**	**39.81**

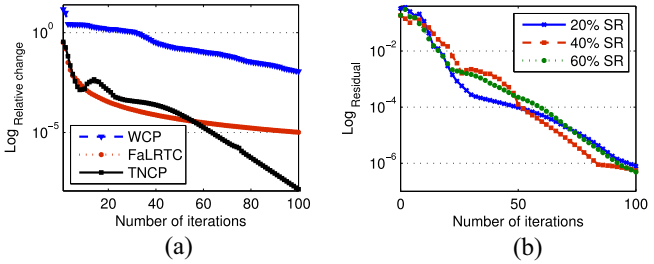


Fig. 2. Convergence behaviors of WCP, FaLRTC, and TNCP on the synthetic tensor data of size $80 \times 80 \times 80$. (a) Log-value of the relative change with 20% SR. (b) Log-value of the residual.

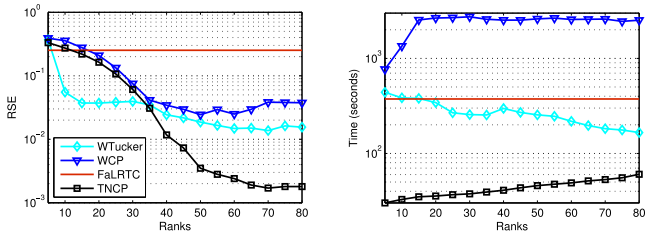


Fig. 3. RSE and time cost (seconds) of WTucker, WCP, FaLRTC, and our TNCP method versus the given tensor ranks.

80%. From the results, we can see that our TNCP method usually yields much more accurate solutions using less time, and often outperforms the other algorithms in terms of RSE and efficiency. Notice that because latent converges too slowly, we do not consider it in the following experiments.

We also study the convergence behaviors of WCP, FaLRTC, and our TNCP method on the synthetic data of size $80 \times 80 \times 80$ with the given tensor rank $R = 10$, as shown in Fig. 2, where the ordinate is the log-value of the relative change of $\mathcal{X}^k$ generated by WCP, FaLRTC, and TNCP, or the log-value of the residual of $\max\{\|M_n^{k+1} - U_n^{k+1}\|_F,\ n = 1, \ldots, N\}$ generated by TNCP with different SRs: 20%, 40%, or 60%, and the abscissa denotes the number of iteration. We can observe that the relative change of our TNCP method drops much more quickly, and converges much faster than WCP and FaLRTC.

To further evaluate the robustness of our TNCP method with respect to the given tensor rank R, we conduct some experiments on the rank-$(10, 10, 10)$ synthetic data of size $100 \times 100 \times 100$, and illustrate the recovery results of FaLRTC, WTucker, WCP, and TNCP with 30% SR, where the rank parameter R for the latter three is chosen from $\{5, 10, \ldots, 80\}$. The average RSE results and time cost of ten independent runs are shown in Fig. 3, from which we can see that as the number of the given tensor rank increases, our TNCP method performs much better than WTucker, WCP, and FaLRTC in terms of RSE. This also confirms that our model with trace norm regularization can provide a better estimation of a low-rank tensor. Moreover, our TNCP method is much faster than the other methods.

B. Large-Scale Network Data

In this part, we examine our TNCP method on a real-world network data set, the YouTube data set[5] [55]. YouTube is currently the most popular video sharing web site, which allows users to interact with each other in various forms such as contacts, subscriptions, sharing favorite videos, etc. In total, this data set contains 848 003 users, with 15 088 users sharing all of the information types, and includes five-dimensions of interactions: contact network, co-contact network, co-subscription network, co-subscribed network, and favorite network. Additional information about the data can be found in [55]. We run these experiments on a machine with 6-core Intel Xeon 2.4 GHz CPU and 64 GB memory.

We address the link prediction problem as the LRTC problem. For our TNCP method, we set the regularization parameter $\lambda = 10$. The tolerance value of TNCP, WTucker, WCP, Hard[6] [11] and FaLRTC is fixed at tol $= 10^{-5}$. For TNCP and FaLRTC, α_n are set to $[1, 1, 10^{-3}]$, and the smoothing parameters of the latter are set to $\mu_n = 5\alpha_n/I_n$, $n = 1, 2, 3$. For Hard, we let $\tau = 10^4$ and $\lambda_1 = \lambda_2 = \lambda_3 = 1$. For TNCP and WCP, the tensor rank is set to $R = 40$, and $R_1 = R_2 = 40$ and $R_3 = 5$ for WTucker.

We use the 15 088 users who share all of the information types and have five-dimensions of interactions in our experiments. So the data size is $15\,088 \times 15\,088 \times 5$. We first report the average running time (seconds) over ten independent runs in Fig. 4, when we vary the number of users. Our TNCP algorithm runs significantly faster than FaLRTC, WTucker, WCP,

[5]http://leitang.net/heterogeneous_network.html
[6]https://sites.google.com/site/marcosignoretto/codes

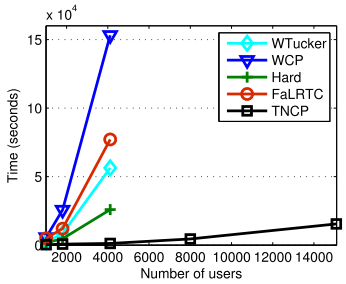


Fig. 4. Comparison of computational time (seconds) on the YouTube data set. For each dataset, we use 20% for training. Note that the other four methods including WTucker, WCP, Hard, and FaLRTC could not run sizes {8000, 15 088} due to runtime exceptions.

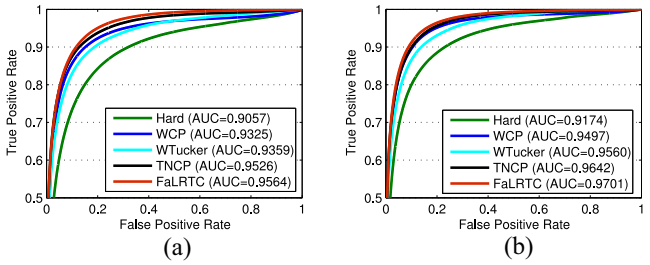


Fig. 5. Average ROC curves showing the performance of link prediction methods with 10% and 20% training data, respectively. (a) 10% train. (b) 20% train.

and Hard. The running time of TNCP increases only slightly when the number of users increases. This shows that our TNCP method has very good scalability and can address large-scale problems. In contrast, the running time of WTucker, WCP, FaLRTC, and Hard increases dramatically. They could not yield experimental results within 48 h when the number of users is 8000 or 15 088.

As the other methods cannot finish running when the problem size is large, we choose 4117 users who have more than ten interactions to form a data set of size $4117 \times 4117 \times 5$. We randomly select 10% or 20% entries as the training set, and the remainder as the testing data. We report the average prediction accuracy [the score area under the receiver operating characteristic curve (AUC)] over ten independent runs in Fig. 5. We can see that two trace norm regularized tensor completion algorithms, TNCP and FaLRTC, outperform WTucker, and WCP in terms of the prediction accuracy. Our TNCP method can achieve very similar performance compared with FaLRTC in terms of AUC.

C. Natural Images

In this section, we apply our TNCP method to in-painting of RGB colored images, each of which is represented as a third-order tensor. Our TNCP method is used to estimate missing data in natural color images used in [5], whose size is $493 \times 517 \times 3$. The tolerance value of TNCP, WTucker, WCP, Hard and FaLRTC is fixed at $\text{tol} = 10^{-5}$, while the parameter of FPCA[7] [56] (one state-of-the-art low-rank matrix completion algorithm) is 10^{-4}. For TNCP and FaLRTC, the weights are set to $\alpha_1 = \alpha_2 = 1$ and $\alpha_3 = 10^{-3}$. Besides, for Hard we let $\tau = 10^4$ and $\lambda_1 = \lambda_2 = \lambda_3 = 1$.

[7]http://www1.se.cuhk.edu.hk/~sqma/FPCA.html

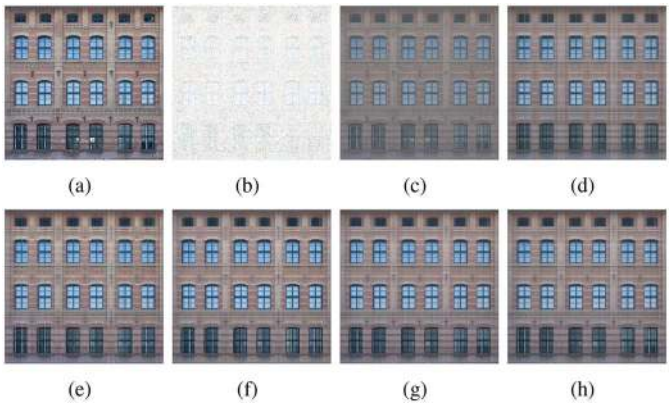


Fig. 6. Comparison of the recovered results of WTucker, WCP, FPCA, Hard, FaLRTC, and TNCP. (a) Original image. (b) 10% randomly sampled image. (c) WTucker, RSE: 0.1872, Time: 830.41 s. (d) WCP, RSE: 0.1546, Time: 1536.19 s. (e) FPCA, RSE: 0.1526, Time: 649.31 s. (f) Hard, RSE: 0.1286, Time: 2362.37 s. (g) FaLRTC with $\mu = 5$, RSE: 0.1223, Time: 2079.58 s. (h) TNCP with $\lambda = 10$ and $R = 40$, RSE: 0.1278, Time: 210.91 s.

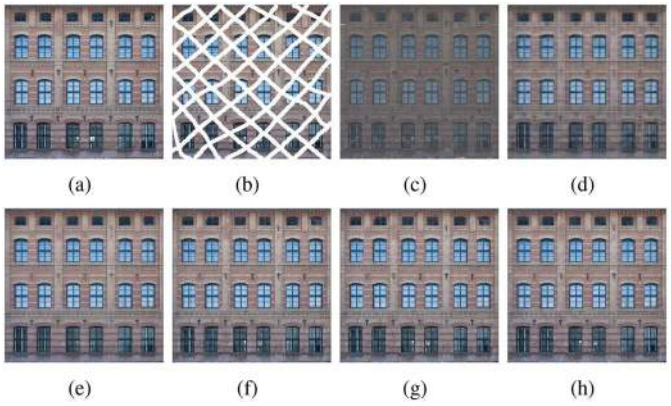


Fig. 7. Comparison of the recovered results of WTucker, WCP, FPCA, Hard, FaLRTC, and TNCP. (a) Original image. (b) Deterministically masked image. (c) WTucker, RSE: 0.1314, Time: 763.56 s. (d) WCP, RSE: 0.1452, Time: 1644.87 s. (e) FPCA, RSE: 0.1102, Time: 988.05 s. (f) Hard, RSE: 0.0774, Time: 1005.40 s. (g) FaLRTC with $\mu = 5$, RSE: 0.0750, Time: 1875.23 s. (h) TNCP with $\lambda = 10$ and $R = 40$, RSE: 0.0789, Time: 194.67 s.

We present the recovery results of our TNCP method, WTucker, WCP, Hard, FaLRTC, and FPCA in two cases of 10% randomly sampled images and deterministically masked images, as shown in Figs. 6 and 7, respectively, from which we can see that the three LRTC approaches including TNCP, Hard, and FaLRTC are significantly better than FPCA in terms of RSE, where FPCA performs three matrix completion tasks on three channels: red, green, and blue, respectively. This further confirms that those LRTC methods can utilize much more information contained in higher-order data than matrix completion methods can, as stated in [5]. Moreover, the four trace norm regularized methods including FPCA, Hard, FaLRTC and TNCP consistently outperform the two weighted tensor decomposition methods including WTucker and WCP. In addition, our TNCP method can achieve almost similar performance with Hard and FaLRTC both in visually quality and in terms of RSE. More importantly, our TNCP method is much faster than the other methods, and is nearly ten times faster

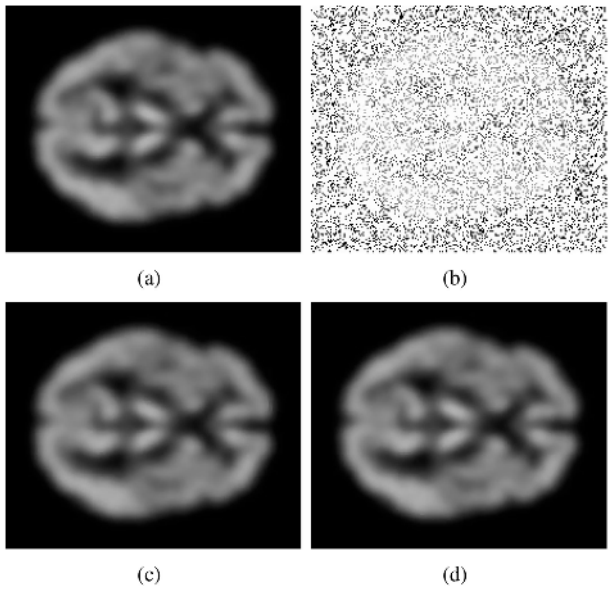


Fig. 8. Comparison of the recovery results of FaLRTC and TNCP on one slice of the brain MRI data. (a) Original image. (b) 20% randomly sampled image. (c) FaLRTC. (d) TNCP.

TABLE III
RSE AND TIME COST (SECONDS) COMPARISON ON THE BRAIN MRI DATA

	NCP		FaLRTC		TNCP	
SR	RSE	Time	RSE	Time	RSE	Time
20%	7.12e-2	1001.26	1.78e-2	4032.35	2.16e-2	389.48
50%	6.92e-2	931.79	3.53e-3	3525.79	3.24e-3	367.35
80%	6.58e-2	847.71	2.09e-4	3275.42	3.15e-4	358.38

than FaLRTC, and at least five and three times faster than Hard and FPCA, respectively.

D. MRI Images

In this part, we compare our TNCP method with FaLRTC and a nonnegative CP completion (NCP) method [44] on the brain MRI image data used in [5], whose size is $181 \times 217 \times 181$. Since its ranks unfolded along three modes are 164, 198, and 165, respectively, then they can decrease to 35, 42, and 36 if removing small singular values less than 1 percent of its Frobenius norm. Thus, the generated data is approximately a low-rank tensor. The tolerance value of TNCP, NCP, and FaLRTC is fixed at tol $= 10^{-5}$. For TNCP and FaLRTC, α_n are set to $\alpha_n = 1/3$, $n = 1, 2, 3$. The regularized parameter of TNCP is set to 10. In addition, we set the tensor rank $R = 50$ for TNCP and NCP.

Fig. 8 shows the recovery results on the brain MRI data set with 20% SR, and we only show one of the slices exemplarily. In addition, Table III shows the recovery accuracy (RSE) and running time (seconds) of NCP, FaLRTC, and TNCP with different SRs: 20%, 50%, and 80%, respectively. From these results, we can see that our TNCP method is much faster than FaLRTC with the almost similar recovery accuracy and visually quality. Moreover, TNCP significantly outperforms NCP in terms of both recovery accuracy and efficiency.

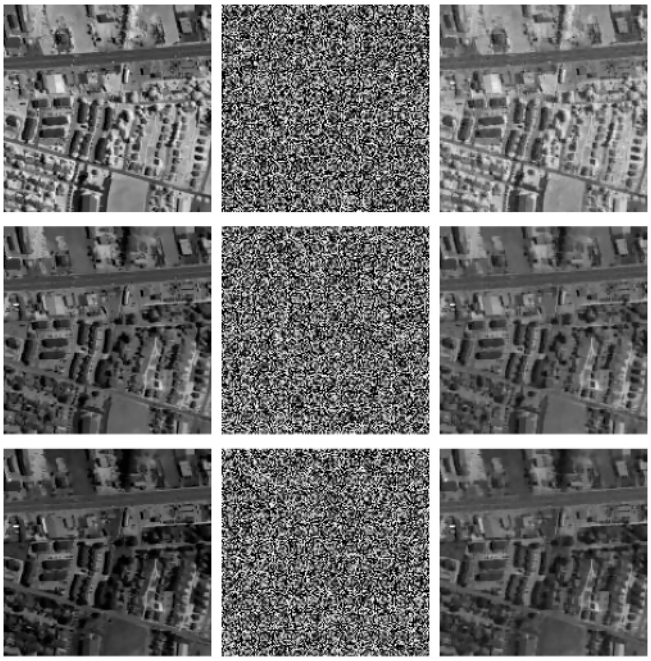


Fig. 9. Hyperspectral data recovery results of our TNCP method with 30% SR: only three selected slices are shown. Left: original images. Middle: 30% sampling images. Right: recovered results of TNCP.

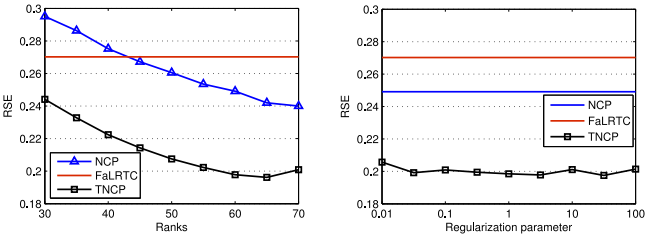


Fig. 10. Recovery results of TNCP against its parameters on the urban hyperspectral data. Left: RSE versus the given ranks. Right: RSE versus the regularization parameter λ.

E. Hyperspectral Images

Finally, we apply our TNCP method on the urban hyperspectral image data, which is a $150 \times 150 \times 210$ hyperspectral cube from the Army Geospatial Center of the U.S. Army Corps of Engineers.[8] The tolerance value of TNCP, NCP, and FaLRTC is fixed at tol $= 10^{-5}$. For TNCP and FaLRTC, α_n is set to $1/3$ for all $n \in \{1, 2, 3\}$. The regularized parameter of TNCP is set to 10. In addition, we set the tensor rank $R = 60$ for TNCP and NCP. The recovery results for three of the bands are shown in Fig. 9. Moreover, we report the recovery accuracy (RSE) and running time (seconds) of these three methods with different SRs: 20%, 40%, and 60%, as listed in Table IV, from which we can see that our TNCP method consistently outperforms NCP in terms of both RSE and efficiency. Furthermore, our TNCP method significantly performs better than FaLRTC in terms of RSE, and is nearly five times faster than FaLRTC. We also evaluate the robustness of our TNCP method with respect to its parameters: the given tensor ranks and the regularization parameter λ, as shown in Fig. 10,

[8] http://www.agc.army.mil/hypercube/

TABLE IV
RSE AND TIME COST (SECONDS) COMPARISON ON THE URBAN HYPERSPECTRAL DATA

	NCP		FaLRTC		TNCP	
SR	RSE	Time	RSE	Time	RSE	Time
20%	0.2516	672.74	0.5931	2837.22	0.2011	467.09
40%	0.2475	628.95	0.4748	2548.02	0.1678	443.61
60%	0.2437	595.14	0.4025	2125.16	0.1357	468.07

from which we can see that our TNCP method is very robust against its parameter variations.

VII. CONCLUSION

In this paper, we proposed a trace norm regularized CP decomposition method for simultaneous tensor completion and decomposition. We first used a factor matrix rank minimization to replace the rank minimization of each unfolding of involved tensors. Then, we relaxed the weighted sum of each factor matrix rank into a tractable convex surrogate, and then obtained a much smaller-scale factor matrix trace norm optimization problem. Finally, we developed an efficient ADMM algorithm to solve the proposed problem. Our convincing experimental results on synthetic data and real-world data verified both the efficiency and effectiveness of our TNCP method.

Our TNCP method can address large-scale tensor completion and decomposition problems, and is much robust to the given tensor rank. In the future, our TNCP method can also be extended to the nonnegative CP tensor decomposition problem [57] as follows:

$$\min_{U_n} \sum_{n=1}^{N} \alpha_n \|U_n\|_* + \frac{\lambda}{2}\|\mathcal{T} - U_1 \circ U_2 \cdots \circ U_N\|_F^2 \quad \text{s.t.,} \quad U_n \geq 0,\ n = 1, \ldots, N. \tag{33}$$

Moreover, we are interested in exploring ways to regularize our model with auxiliary information as in [58] and [59], such as graph Laplacian of relationships among data and position information contained in images. We will also apply our model to address a variety of robust tensor recovery problems, i.e., higher-order robust PCA [30] and robust tensor completion.

ACKNOWLEDGMENT

The authors would like to thank the Editor-in-Chief, the handling Associate Editor, and the anonymous reviewers for their support and constructive comments on this paper.

REFERENCES

[1] L. De Lathauwer and J. Vandewalle, "Dimensionality reduction in higher-order signal processing and rank-(R1, R2, ..., RN) reduction in multilinear algebra," *Linear Algebra Appl.*, vol. 391, pp. 31–55, Jan. 2004.

[2] A. Cichocki *et al.*, "Tensor decompositions for signal processing applications," *IEEE Signal Process. Mag.*, to be published.

[3] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and Gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.

[4] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Kyoto, Japan, 2009, pp. 2114–2121.

[5] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.

[6] A. Barmpoutis, "Tensor body: Real-time reconstruction of the human body and avatar synthesis from RGB-D," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1347–1356, Oct. 2013.

[7] M. Mørup, "Applications of tensors (multiway array) factorizations and decompositions in data mining," *WIREs: Data Min. Knowl. Disc.*, vol. 1, no. 1, pp. 24–40, 2011.

[8] J. Sun *et al.*, "MultiVis: Content-based social network exploration through multi-way visual analysis," in *Proc. SIAM Int. Conf. Data Min. (SDM)*, Denver, CO, USA, 2009, pp. 1063–1074.

[9] X. Li, S. Lin, S. Yan, and D. Xu, "Discriminant locally linear embedding with high-order tensor data," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 2, pp. 342–352, Apr. 2008.

[10] X. Geng, K. Smith-Miles, Z. Zhou, and L. Wang, "Face image modeling by multilinear subspace analysis with missing values," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 3, pp. 881–892, Jun. 2011.

[11] M. Signoretto, Q. T. Dinh, L. D. Lathauwer, and J. Suykens, "Learning with tensors: A framework based on convex optimization and spectral regularization," *Mach. Learn.*, vol. 94, no. 3, pp. 303–351, 2014.

[12] L. Lathauwer, B. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.

[13] M. Mørup, L. K. Hansen, C. S. Herrmann, J. Parnas, and S. M. Arnfred, "Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG," *Neuroimage*, vol. 29, no. 3, pp. 938–947, 2006.

[14] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[15] E. Acar and B. Yener, "Unsupervised multiway data analysis: A literature survey," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 1, pp. 6–20, Jan. 2009.

[16] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "A survey of multilinear subspace learning for tensor data," *Pattern Recognit.*, vol. 44, no. 7, pp. 1504–1551, 2011.

[17] J. Liu, J. Liu, P. Wonka, and J. Ye, "Sparse non-negative tensor factorization using columnwise coordinate descent," *Pattern Recognit.*, vol. 45, no. 1, pp. 649–656, 2012.

[18] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations with missing data," in *Proc. SIAM Int. Conf. Data Min. (SDM)*, Singapore, 2010, pp. 701–711.

[19] H. Shan, A. Banerjee, and R. Natarajan, "Probabilistic tensor factorization for tensor completion," Univ. Minnesota, Minneapolis, MN, USA, Tech. Rep. TR11-026, Oct. 2011.

[20] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Probl.*, vol. 27, no. 2, 2011, Art. ID 025010.

[21] Y. Liu and F. Shang, "An efficient matrix factorization method for tensor completion," *IEEE Signal Process. Lett.*, vol. 20, no. 4, pp. 307–310, Apr. 2013.

[22] Y. K. Yilmaz, A. T. Cemgil, and U. Simsekli, "Generalized coupled tensor factorization," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Granada, Spain, 2011, pp. 2151–2159.

[23] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. 27th Annu. Conf. Comput. Graph. Interact. Technol.*, 2000, pp. 417–424.

[24] T. Korah and C. Rasmussen, "Spatiotemporal inpainting for recovering texture maps of occluded building facades," *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2262–2271, Sep. 2007.

[25] E. J. Candès, and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[26] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, pp. 717–772, 2009.

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

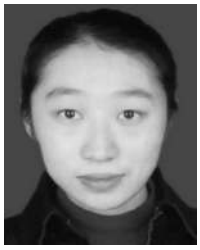
[28] Z. Lin, M. Chen, and L. Wu, "The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices," Univ. Illinois, Urbana-Champaign, Champaign, IL, USA, Tech. Rep. UILU-ENG-09-2215, Oct. 2009.

[29] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, Stanford University, Stanford, CA, USA, 2002.

[30] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 1, pp. 225–253, 2014.

[31] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, pp. 279–311, Sep. 1966.

[32] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an 'explanatory' multi-modal factor analysis," *UCLA Work. Pap. Phonetics*, vol. 16, no. 1, pp. 1–84, 1970.
[33] R. Tomioka and T. Suzuki, "Convex tensor decomposition via structured Schatten norm regularization," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Stateline, NV, USA, 2013, pp. 1331–1339.
[34] M. Filipovic and A. Jukic, "Tucker factorization with missing data with application to low-n-rank tensor completion," *Multidim. Syst. Sign. Process.*, to be published.
[35] Y. Liu, F. Shang, W. Fan, J. Cheng, and H. Cheng, "Generalized higher-order orthogonal iteration for tensor decomposition and completion," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Montreal, QC, Canada, 2014.
[36] L. Yang, Z. Huang, and X. Shi, "A fixed point iterative method for low-n-rank tensor pursuit," *IEEE Trans. Signal Process.*, vol. 61, no. 11, pp. 2952–2962, Jun. 2013.
[37] Z. Shi, J. Han, T. Zheng, and J. Li, "Guarantees of augmented trace norm models in tensor recovery," in *Proc. 23rd Int. Joint Conf. Artif. Intell. (IJCAI)*, Beijing, China, 2013, pp. 1670–1676.
[38] M. Signoretto, L. De Lathauwer, and J. A. Suykens, "Nuclear norms for tensors and their use for convex multilinear estimation," ESAT-SISTA, K. U. Leuven, Leuven, Belgium, Tech. Rep. 10-186, 2010.
[39] Y. Liu, F. Shang, H. Cheng, J. Cheng, and H. Tong, "Factor matrix trace norm minimization for low-rank tensor completion," in *Proc. SIAM Int. Conf. Data Min. (SDM)*, Philadelphia, PA, USA, 2014, pp. 866–874.
[40] J. Hastad, "Tensor rank is NP-complete," *J. Algorithms*, vol. 11, no. 4, pp. 644–654, 1990.
[41] C. J. Hillar and L. H. Lim, "Most tensor problems are NP hard," *J. ACM*, vol. 45, no. 6, Nov. 2013, Art. ID 45.
[42] C. Mu, B. Huang, J. Wright, and D. Goldfarb, "Square deal: Lower bounds and improved relaxations for tensor recovery," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, Beijing, China, 2014, pp. 73–81.
[43] B. Romera-Paredes and M. Pontil, "A new convex relaxation for tensor completion," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Stateline, NV, USA, 2013, pp. 2967–2975.
[44] Y. Xu and W. Yin, "A block coordinate descent method for multi-convex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imag. Sci.*, vol. 6, no. 3, pp. 1758–1789, 2013.
[45] R. Bro and H. A. L. Kiers, "A new efficient method for determining the number of components in PARAFAC models," *J. Chemometrics*, vol. 17, no. 5, pp. 274–286, 2003.
[46] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
[47] F. Shang, Y. Liu, J. Cheng, and H. Cheng, "Robust principal component analysis with missing data," in *Proc. 23rd Int. Conf. Inf. Knowl. Manage. (CIKM)*, Shanghai, China, 2014, pp. 1149–1158.
[48] C. Chen, B. He, and X. Yuan, "Matrix completion via an alternating direction method," *IMA J. Numer. Anal.*, vol. 32, no. 1, pp. 227–245, 2012.
[49] Y. Liu, L. C. Jiao, and F. Shang, "A fast tri-factorization method for low-rank matrix recovery and completion," *Pattern Recognit.*, vol. 46, no. 1, pp. 163–173, 2013.
[50] Y. Liu, L. C. Jiao, and F. Shang, "An efficient matrix factorization based low-rank representation for subspace clustering," *Pattern Recognit.*, vol. 46, no. 1, pp. 284–292, 2013.
[51] J. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, 2010.
[52] F. Shang, Y. Liu, and J. Cheng, "Generalized higher-order tensor decomposition via parallel ADMM," in *Proc. 28th AAAI Conf. Artif. Intell. (AAAI)*, Quebec City, QC, Canada, 2014, pp. 1279–1285.
[53] M. F. Møller, "A scaled conjugate gradient algorithm for fast supervised learning," *Neural Netw.*, vol. 6, no. 4, pp. 525–533, 1993.
[54] G. Allen, "Sparse higher-order principal components analysis," in *Proc. 15th Int. Conf. Artif. Intell. Statist. (AISTATS)*, La Palma, Spain, 2012, pp. 27–36.
[55] L. Tang, X. Wang, and H. Liu, "Uncovering groups via heterogeneous interaction analysis," in *Proc. 9th IEEE Int. Conf. Data Min. (ICDM)*, Miami, FL, USA, 2009, pp. 503–512.
[56] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and Bregman iterative methods for matrix rank minimization," *Math. Program.*, vol. 128, no. 1, pp. 321–353, 2011.
[57] J. Kim, Y. He, and H. Park, "Algorithms for nonnegative matrix and tensor factorizations: A unified view based on block coordinate descent framework," *J. Glob. Optim.*, vol. 58, no. 2, pp. 285–319, 2014.
[58] A. Narita, K. Hayashi, R. Tomioka, and H. Kashima, "Tensor factorization using auxiliary information," *Data Min. Knowl. Disc.*, vol. 25. no. 2, pp. 298–324, 2012.
[59] Y. Chen, C. Hsu, and H. Liao, "Simultaneous tensor decomposition and completion using factor priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 577–591, Mar. 2014.



Yuanyuan Liu received the Ph.D. degree in pattern recognition and intelligent system from Xidian University, Xi'an, China, in 2013.

She is currently a Post-Doctoral Research Fellow with the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong. Her current research interests include image processing, pattern recognition, and machine learning.

Fanhua Shang (M'14) received the Ph.D. degree in circuits and systems from Xidian University, Xi'an, China, in 2012.

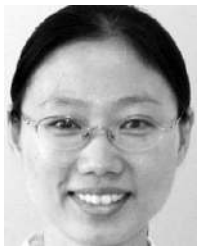
He is currently a Post-Doctoral Research Fellow with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. He was a Post-Doctoral Research Associate with the Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA. His current research interests include machine learning, data mining, pattern recognition, and computer vision.

Licheng Jiao (SM'89) received the B.S. degree from Shanghai Jiaotong University, Shanghai, China, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 1982, 1984, and 1990, respectively.

He is currently a Distinguished Professor with the School of Electronic Engineering, Xidian University, Xi'an. His current research interests include signal and image processing, natural computation, machine learning, and intelligent information processing. He has led approximately 40 important scientific research projects and has published over ten monographs and 150 papers in international journals and conferences.

Prof. Jiao was the recipient of the second prize of the National Natural Science Award in 2013. He is the Chairman of the Awards and Recognition Committee and an Executive Committee Member of the Chinese Association of Artificial Intelligence.

James Cheng is an Assistant Professor with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. His current research interests include distributed computing systems, large-scale network analysis, temporal networks, and big data.



Hong Cheng is an Associate Professor with the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong. Her current research interests include data mining, machine learning, and database systems.