Generalized Higher Order Orthogonal Iteration for Tensor Learning and Decomposition

Yuanyuan Liu, Fanhua Shang, *Member, IEEE*, Wei Fan, James Cheng, and Hong Cheng

***Abstract*— Low-rank tensor completion (LRTC) has successfully been applied to a wide range of real-world problems. Despite the broad, successful applications, existing LRTC methods may become very slow or even not applicable for large-scale problems. To address this issue, a novel core tensor trace-norm minimization (CTNM) method is proposed for simultaneous tensor learning and decomposition, and has a much lower computational complexity. In our solution, first, the equivalence relation of trace norm of a low-rank tensor and its core tensor is induced. Second, the trace norm of the core tensor is used to replace that of the whole tensor, which leads to two much smaller scale matrix TNM problems. Finally, an efficient alternating direction augmented Lagrangian method is developed to solve our problems. Our CTNM formulation needs only $O((R^N + NRI)\log(\sqrt{I^N}))$ observations to reliably recover an Nth-order $I \times I \times \cdots \times I$ tensor of n-rank $(r, r, \ldots, r)$, compared with $O(rI^{N-1})$ observations required by those tensor TNM methods ($I \gg R \geq r$). Extensive experimental results show that CTNM is usually more accurate than them, and is orders of magnitude faster.**

***Index Terms*— Alternating direction augmented Lagrangian (ADAL), low-rank tensor, tensor completion, trace-norm minimization (TNM), transductive learning, Tucker decomposition.**

I. Introduction

TENSORS are the higher order generalizations of vectors and matrices. In particular, with the rapid development of modern computing technology in recent years, tensor data are becoming ubiquitous, such as multichannel images and videos, and have become increasingly popular [1], [2]. There are numerous practical applications of tensors in machine learning [3]–[6], signal processing [2], [7]–[9], computer vision [10]–[12], data mining [13]–[16], numerical linear algebra [17], [18], and so on. Due to the problems in the acquisition process, loss of information or costly experiments [19], some entries of observed tensors may be missing. This class of transductive learning problems for low-rank tensor completion (LRTC) has successfully been applied to a wide range of real-world problems, such as multichannel images and videos [10], [12], EEG data [7], retail sales data [20] and hyperspectral data recovery [7], link prediction [4], and multirelational learning [15], [21], [22].

Recently, sparse vector recovery and low-rank matrix completion (LRMC) have intensively been studied [23], [24]. Compared with matrices and vectors, tensors can be used to express more complicated intrinsic structures in higher order data [15], [25]. Liu *et al.* [10] and Goldfarb and Qin [2] indicated that tensor-based methods utilize all information along all dimensions, and are capable of taking full advantage of the high-order structure to provide better understanding and higher precision, while matrix-based methods only consider the constraints along two particular dimensions. Thus, in this paper, we are particularly interested in the simultaneous tensor learning and decomposition (STLD) problem, which is to infer the missing entries in the tensor of the lowest rank just like predicting the missing labels, and simultaneously find its factor components from incomplete observations. To address tensors with missing data, some weighted tensor decomposition (TD) methods, such as the weighted Tucker (WTucker) [26] and weighted CP (WCP) decompositions [19], have successfully been applied to EEG data analysis, nature and hyperspectral images inpainting. However, they are usually sensitive to the given ranks due to their least-squares formulations [6], [12].

Liu *et al.* [10] first extended the trace norm (also called the nuclear norm [27] and Schatten one-norm [28]) regularization for learning of partially observed low-rank tensors. In [12], they presented a more general model, and proposed three efficient algorithms to solve the LRTC problem. In other words, the LRTC problem is converted into a convex combination of the trace-norm minimization (TNM) of all unfoldings along different modes. Some similar algorithms can be found in [7], [9], [29], and [30]. Besides these approaches described above, a number of variations [31] and alternatives [32] have been discussed in the literature. In addition, there are some theoretical developments that guarantee the reconstruction of a low-rank tensor from partial measurements by solving the TNM problem under some reasonable conditions [33]–[35]. Although those TNM methods have successfully been applied in many real-world applications, their algorithms suffer from high computational cost of multiple SVDs using $O(NI^{N+1})$

Manuscript received March 10, 2015; revised July 12, 2015 and October 18, 2015; accepted October 21, 2015. This work was supported in part by the Shun Hing Institute of Advanced Engineering under Grant 8115048, in part by MSRA under Grant 6903555, in part by the General Research Fund under Grant 411211, in part by The Chinese University of Hong Kong under Grant 4055015 and Grant 4055017, in part by the China 973 Fundamental Research and Development Program under Grant 2014CB340304, and in part by the Huawei Project under Grant 7010255. *(Corresponding author: Fanhua Shang.)*

Y. Liu and H. Cheng are with the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong (e-mail: liuyuanyuan0917@hotmail.com; hcheng@se.cuhk.edu.hk).

F. Shang and J. Cheng are with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: shangfanhua@hotmail.com; jcheng@cse.cuhk.edu.hk).

W. Fan is with the Baidu Big Data Research Laboratory, Sunnyvale, CA 94089 USA (e-mail: wei.fan@gmail.com).

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/TNNLS.2015.2496858

time per iteration, where the assumed size of an Nth-order tensor is $I \times I \times \cdots \times I$.

In this paper, we propose a scalable and robust core tensor TNM (CTNM) method for the STLD problems, which has a much lower computational complexity than existing LRTC methods. We first induce the equivalence relation of the trace norm of a low-rank tensor and its core tensor. Then, we formulate two tractable smaller scale matrix TNM models, in which we use the trace norm of the core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ to replace that of the whole tensor of size $I_1 \times I_2 \times \cdots \times I_N$. In other words, our noisy and noiseless CTNM models only involve several much smaller unfoldings of $\mathcal{G}$ ($R_n \ll I_n$, $n = 1, \ldots, N$). Finally, we design an efficient generalized higher order orthogonal iteration (gHOI) algorithm with a convergence guarantee to solve our problems.

Preliminary results have been reported in [6]. Besides providing a more thorough literature review, this paper: 1) extends the partial observation theorem for matrices in [36] and [37] to the higher order cases; 2) further improves the recovery bound in [6], and shows that only $O((R^N + NRI)\log(\sqrt{I^N}))$ observations are sufficient for the noisy CTNM model with $(R, R, \ldots, R)$ to recover an Nth-order $I \times I \times \cdots \times I$ tensor of n-rank $(r, r, \ldots, r)$ with high probability ($I \gg R \geq r$), compared with $O(rI^{N-1})$ observations required by the convex TNM model [31], [34]; 3) presents a graph Laplacian regularized method using auxiliary information induced from the relationships; and 4) provides convincing experimental results to demonstrate the merits of our CTNM method, especially on a large data set (i.e., YouTube).

The rest of this paper is organized as follows. We review some preliminaries in Section II. In Section III, we propose two novel CTNM models for the STLD problems and develop an efficient gHOI algorithm in Section IV. We provide recovery guarantees in Section V. We report empirical results in Section VI. Finally, the conclusion is drawn in Section VII.

II. Notation and Background

As in [1], we denote tensors by calligraphic letters, e.g., $\mathcal{X}$, and matrices by upper case letters, e.g., X. A fiber of $\mathcal{X}$ is a column vector defined by fixing every index of $\mathcal{X}$ but one. The mode-n unfolding (matricization) of an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the matrix, denoted by $\mathcal{X}_{(n)} \in \mathbb{R}^{I_n \times \Pi_{j \neq n} I_j}$, that is obtained by arranging the mode-n fibers to be the columns of $\mathcal{X}_{(n)}$. The inner product of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined as the sum of the product of their entries, i.e., $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1, \ldots, i_N} x_{i_1 \cdots i_N} y_{i_1 \cdots i_N}$, and the Frobenius norm of $\mathcal{X}$ is defined as $\|\mathcal{X}\|_F = (\langle \mathcal{X}, \mathcal{X} \rangle)^{1/2}$.

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: $A \otimes B = [a_{ij} B]_{mp \times nq}$. The mode-$n$ product of a tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with a matrix $C \in \mathbb{R}^{J \times I_n}$, denoted by $\mathcal{A} \times_n C \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$, is defined as

$$(\mathcal{A} \times_n C)_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n = 1}^{I_n} a_{i_1 i_2 \cdots i_N} c_{j i_n}.$$

Table I lists the symbols commonly used in this paper.

TABLE I
Commonly Used Notations

Symbol	Description
$\mathcal{X}$, X, $\mathbf{x}$, x	Tensor, matrix, vector, scalar (respectively)
$\mathcal{X}_{(n)}, \mathcal{M}_{n,(n)}$	Mode-n unfolding of tensors $\mathcal{X}$, $\mathcal{M}_n$
$\mathcal{A} \times_n C$	Mode-n product
$\mathcal{G}, U_n, n = 1, \ldots, N$	Core tensor and factor matrices
$[\![\mathcal{G}; U_1, \ldots, U_N]\!]$	Full multilinear product, $\mathcal{G} \times_1 U_1 \cdots \times_N U_N$
$\otimes$, $\odot$, $\circ$	Kronecker, Hadamard, and outer products
Ω, $\|\Omega\|$	Index set of observed entries, cardinality of Ω
$\mathcal{P}_\Omega$	Projection operator defined behind (2)
$\mathcal{P}_\Omega^\perp$	Complementary operator of $\mathcal{P}_\Omega$
$\langle \mathcal{X}, \mathcal{Y} \rangle$	Inner product of tensors $\mathcal{X}$ and $\mathcal{Y}$
refold$(\cdot)$	Refolding of the matrix into a tensor
$\|\cdot\|_F$, $\|\cdot\|_*$, $\|\cdot\|_2$	Frobenius norm, trace norm, spectral norm

A. Tensor Decompositions and Ranks

The CP decomposition approximates $\mathcal{X}$ by $\sum_{i=1}^R \mathbf{a}_i^1 \circ \mathbf{a}_i^2 \circ \cdots \circ \mathbf{a}_i^N$, where $R > 0$ is a given integer, $\mathbf{a}_i^n \in \mathbb{R}^{I_n}$, and $\circ$ denotes the outer product of vectors. The tensor rank of $\mathcal{X}$, denoted by rank$(\mathcal{X})$, is defined as the smallest value of R such that the approximation holds with equality. Computing the tensor rank of a specific given tensor is NP-hard in general [1], [38], [39]. Fortunately, the multilinear rank (also known as the Tucker rank in [2] and [31]) of $\mathcal{X}$, denoted as n-rank$(\mathcal{X})$, is efficient to compute, and consists of the ranks of all unfoldings as follows.

Definition 1: The n-rank of an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the tuple of the ranks of all unfoldings

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

Given the n-rank$(\mathcal{X})$, the WTucker model proposed in [26] for LRTC problems is formulated as follows:

$$\min_{\{U_n\}, \mathcal{G}} \|\mathcal{W} \odot (\mathcal{T} - [\![\mathcal{G}; U_1, \ldots, U_N]\!])\|_F^2 \tag{1}$$

where $\mathcal{T}$ is a given partially observed Nth-order tensor (that is, the entries of $\mathcal{T}$ in Ω are given, while the remaining entries are missing, where Ω denotes the index set of all observed entries), $[\![\mathcal{G}; U_1, \ldots, U_N]\!] := \mathcal{G} \times_1 U_1 \times_2 \cdots \times_N U_N$, $U_n \in \mathbb{R}^{I_n \times R_n}$ ($n = 1, \ldots, N$), and $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ are called the factor matrices and the core tensor, $\odot$ denotes the Hadamard (elementwise) product, and $\mathcal{W}$ is the indicator tensor: $w_{i_1 i_2 \ldots i_N} = 1$ if $(i_1, i_2, \ldots, i_N) \in \Omega$, and $w_{i_1 i_2 \ldots i_N} = 0$ otherwise. Since R_n is in general much smaller than I_n for $n = 1, \ldots, N$, the storage of the Tucker decomposition form can be significantly smaller than that of the original tensor.

B. LRTC

Recently, Liu *et al.* [12] proposed the following model for LRTC problems:

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

where $\|\mathcal{X}_{(n)}\|_*$ denotes the trace norm of the unfolding $\mathcal{X}_{(n)}$, i.e., the sum of its singular values, α_ns, are prespecified weights, and $\mathcal{P}_\Omega$ is the orthogonal projection operator onto the space spanned by the tensors vanishing outside of Ω so that the $(i_1, i_2, \ldots, i_N)$th entry of $\mathcal{P}_\Omega(\mathcal{X})$ equals to

$x_{i_1 i_2,\ldots,i_N}$ for $(i_1, i_2, \ldots, i_N) \in \Omega$ and zero otherwise. In the presence of noise, we obtain the following formulation:

$$\min_{\mathcal{X}} \sum_{n=1}^{N} \alpha_n \|\mathcal{X}_{(n)}\|_* + \frac{\lambda}{2}\|\mathcal{P}_\Omega(\mathcal{X}) - \mathcal{P}_\Omega(\mathcal{T})\|_F^2 \tag{3}$$

where $\lambda > 0$ is a regularization parameter. Moreover, Tomioka and Suzuki [33] proposed a latent Schatten one-norm model

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

Note that each unfolding $\mathcal{X}_{(n)}$ in (2) and (3) shares the same entries, and accordingly cannot be optimized independently. Therefore, we need to apply variable splitting and introduce an auxiliary variable to each unfolding of $\mathcal{X}$. In other words, many auxiliary variables are introduced to split the interdependent terms such that they can be solved independently. Moreover, all existing TNM algorithms for (2)–(4) involve multiple SVDs of very large unfoldings in each iteration, and thus suffer from high computational cost.

More recently, it has been shown that the sum of trace-norms models mentioned above may substantially be suboptimal [31], [32], especially for higher order tensors. To address this issue, a more square convex model in [31] is given by

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

where $\mathcal{X}_{[j]}$ is defined by $\mathcal{X}_{[j]} := \text{reshape}(\mathcal{X}_{(1)}, \Pi_{n\le j} I_n, \Pi_{n>j} I_n)$, and j is chosen from $\{1, 2, \ldots, N\}$ to make $\prod_{n\le j} I_n$ as close to $\prod_{n>j} I_n$ as possible. If the order of the tensor is more than three, Mu *et al.* [31] showed that model (5) can exactly recover the true low-rank tensor from far fewer observed entries than those required by (2)–(4). However, for the third-order tensors, model (5) is the same as the TNM method for one unfolding, and therefore, its algorithm may not perform as well as those methods for (2)–(4).

III. Core Tensor Trace-Norm Minimization

All existing TNM algorithms for solving (2)–(4) have high computational cost, which limits their applicability to large-scale problems. Moreover, current weighted TD methods require explicit knowledge of the tensor rank or multilinear rank to gain a reliable performance. Motivated by these challenges, we propose two scalable, robust CTNM models, and then achieve two smaller scale matrix TNM problems.

A. CTNM Models

Definition 2: The trace norm of an Nth-order tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ is the average of the trace norms of its all unfoldings along different modes

$$\|\mathcal{X}\|_* = \frac{1}{N}\sum_{n=1}^{N}\|\mathcal{X}_{(n)}\|_*$$

where $\|\mathcal{X}_{(n)}\|_*$ denotes the trace norm of the mode-n unfolding $\mathcal{X}_{(n)}$ for $n = 1, 2, \ldots, N$.

For the imbalance STLD problem (e.g., the multirelational prediction problem in Section VI), some prespecified weights as in [12], $\alpha_n \ge 0$, $n = 1, 2, \ldots, N$ (which satisfy $\sum_n \alpha_n = 1$) can be incorporated into the definition of the tensor trace norm by replacing $(1/N)$. Furthermore, we have Theorem 1 [6].

Theorem 1: Let $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ with n-rank $= (r_1, r_2, \ldots, r_N)$ and $\mathcal{G} \in \mathbb{R}^{R_1\times R_2\times\cdots\times R_N}$ satisfy $\mathcal{X} = [\![\mathcal{G}; U_1, \ldots, U_N]\!]$ and $R_n \ge r_n$, $n = 1, 2, \ldots, N$, then

$$\|\mathcal{X}\|_* = \|\mathcal{G}\|_*$$

where $\|\mathcal{X}\|_*$ and $\|\mathcal{G}\|_*$ are the trace norms of $\mathcal{X}$ and its core tensor $\mathcal{G}$, respectively, and $U_n \in \text{St}(I_n, R_n) := \{U | U^T U = I_{R_n}\}$, which denotes the Stiefel manifold [40], i.e., the set of columnwise orthonormal matrices of size $I_n \times R_n$.

The core tensor $\mathcal{G}$ of size $(R_1, R_2, \ldots, R_N)$ has much smaller size than the whole tensor $\mathcal{X}$ (i.e., $R_n \ll I_n$, $n = 1, 2, \ldots, N$). If the desired low-rank tensor $\mathcal{X}$ of (2) or (3) has the Tucker decomposition form $\mathcal{X} = [\![\mathcal{G}; U_1, \ldots, U_N]\!]$, then according to Theorem 1, the LRTC problems (2) and (3) are reformulated into the following forms:

$$\begin{aligned}&\min_{\mathcal{G},\{U_n\}} \|\mathcal{G}\|_* \\ &\text{s.t., } \mathcal{P}_\Omega([\![\mathcal{G}; U_1, \ldots, U_N]\!]) = \mathcal{P}_\Omega(\mathcal{T}),\ U_n \in \text{St}(I_n, R_n)\end{aligned} \tag{6}$$

$$\begin{aligned}&\min_{\mathcal{G},\{U_n\}} \|\mathcal{G}\|_* + \frac{\lambda}{2}\|\mathcal{P}_\Omega([\![\mathcal{G}; U_1, \ldots, U_N]\!]) - \mathcal{P}_\Omega(\mathcal{T})\|_F^2 \\ &\text{s.t., } U_n \in \text{St}(I_n, R_n).\end{aligned} \tag{7}$$

Our CTNM models (6) and (7) for STLD problems alleviate the SVD computation burden of much larger unfoldings in (2)–(4). In addition, we use the tensor trace-norm regularization terms in (6) and (7) to increase the robustness of the multilinear rank selection, while the WTucker model (1) is usually sensitive to the given ranks $(R_1, R_2, \ldots, R_N)$ [6], [12], [41]. Several matrix rank estimation strategies in [42] and [43] can be used to compute some good values $(r_1, r_2, \ldots, r_N)$ for the multilinear rank of the involved tensor. Therefore, we can set some relatively large integers $(R_1, R_2, \ldots, R_N)$ such that $R_n \ge r_n$, $n = 1, \ldots, N$.

B. Graph Regularization Extension

As our CTNM models are also higher order TD problems, and inspired by the work [44]–[46], we exploit the auxiliary information given as similarity matrices in a regularization model for STLD problems, such as multirelational prediction

$$\begin{aligned}\min_{\mathcal{G},\{U_n\in\text{St}(I_n,R_n)\}} &\|\mathcal{G}\|_* + \eta\sum_{n=1}^{N}\text{Tr}\big(U_n^T L_n U_n\big) \\ &+ \frac{\lambda}{2}\|\mathcal{P}_\Omega([\![\mathcal{G}; U_1, \ldots, U_N]\!]) - \mathcal{P}_\Omega(\mathcal{T})\|_F^2\end{aligned} \tag{8}$$

where $\eta \ge 0$ is a regularization constant, $\text{Tr}(\cdot)$ denotes the matrix trace, L_n is the graph Laplacian matrix, i.e., $L_n = D_n - W_n$, W_n is the weight matrix for the object set, and D_n is the diagonal matrix, whose entries are column sums of W_n, i.e., $(D_n)_{ii} = \sum_j (W_n)_{ij}$.

IV. Optimization Algorithms

In this section, we will propose an alternating direction augmented Lagrangian (ADAL) method[1] to solve the noisy CTNM problem (7), and then extend it to solve the graph regularized CTNM (RCTNM) problem (8). ADAL decomposes a large problem into a series of smaller subproblems, and coordinates the solutions of subproblems to compute the optimal solution. In recent years, it has been shown in [12], [43], [47], and [48] that ADAL is very efficient for some convex or nonconvex optimization problems in various applications.

A. gHOI Algorithm With Rank-Increasing Scheme

Different from existing TNM algorithms in [7], [12], and [33], only several much smaller matrices, $V_n \in \mathbb{R}^{R_n \times \Pi_{j\neq n} R_j}$, are introduced into (7) as the auxiliary variables as well as $\mathcal{Z}$, thus (7) is reformulated into the following equivalent form (see Appendix A for the detailed analysis):

$$\begin{aligned}&\min_{\mathcal{G},\{U_n\},\{V_n\},\mathcal{Z}} \frac{1}{N}\sum_{n=1}^{N}\|V_n\|_* + \frac{\lambda}{2}\|\mathcal{Z}-[\![\mathcal{G};U_1,\ldots,U_N]\!]\|_F^2\\ &\text{s.t., } \mathcal{G}_{(n)}=V_n,\ U_n\in \mathrm{St}(I_n,R_n),\quad \mathcal{P}_\Omega(\mathcal{Z})=\mathcal{P}_\Omega(\mathcal{T}).\end{aligned} \tag{9}$$

The STLD problem (9) can be solved by ADAL, and its partial augmented Lagrangian function is

$$\begin{aligned}&\mathcal{L}_\mu(\mathcal{G},\{U_n\},\{V_n\},\mathcal{Z},\{Y_n\})\\ &\quad=\sum_{n=1}^{N}\left(\frac{\|V_n\|_*}{N}+\langle Y_n,\mathcal{G}_{(n)}-V_n\rangle\right)+\sum_{n=1}^{N}\frac{\mu}{2}\|\mathcal{G}_{(n)}-V_n\|_F^2\\ &\qquad+\frac{\lambda}{2}\|\mathcal{Z}-[\![\mathcal{G};U_1,\ldots,U_N]\!]\|_F^2\end{aligned}$$

where $Y_n, n = 1,\ldots,N$ are the matrices of Lagrange multipliers, and $\mu > 0$ is a penalty parameter. ADAL solves the STLD problem (9) by successively minimizing the Lagrange function $\mathcal{L}_\mu$ over $\{\mathcal{G}, U_1,\ldots,U_N, V_1,\ldots,V_N,\mathcal{Z}\}$, and then updating $\{Y_1,\ldots,Y_N\}$.

Updating $\{\mathcal{G}^{k+1}, U_1^{k+1},\ldots,U_N^{k+1}\}$: The subproblem with respect to $\{U_1,\ldots,U_N\}$ and $\mathcal{G}$ is given by

$$\begin{aligned}\min_{\mathcal{G},\{U_n\in\mathrm{St}(I_n,R_n')\}}&\sum_{n=1}^{N}\frac{\mu^k}{2}\|\mathcal{G}_{(n)}-V_n^k+Y_n^k/\mu^k\|_F^2\\ &+\frac{\lambda}{2}\|\mathcal{Z}^k-[\![\mathcal{G};U_1,\ldots,U_N]\!]\|_F^2\end{aligned} \tag{10}$$

where R_n' is an underestimated rank ($R_n' \le R_n$), and is dynamically adjusted using the rank-increasing scheme in the following. Different from the traditional higher order orthogonal iteration (HOOI) algorithm proposed in [17] for solving the Tucker decomposition problem, we will propose a generalized HOOI (GHOOI) scheme to solve problem (10) in Section IV-B.

[1]This class of algorithms is also known as the alternating direction method of multipliers.

Updating $\{V_1^{k+1},\ldots,V_N^{k+1}\}$: With keeping the other variables fixed, we update V_n^{k+1} by solving the following problem:

$$\min_{V_n}\|V_n\|_*/N+\frac{\mu^k}{2}\|\mathcal{G}_{(n)}^{k+1}-V_n+Y_n^k/\mu^k\|_F^2. \tag{11}$$

Problem (11) is known to have a closed-form solution given by the following so-called matrix shrinkage operator [49]:

$$V_n^{k+1}=\mathrm{prox}_{1/\tau^k}(G_n^k):=U\,\mathrm{diag}\left(\max\left\{\sigma-\frac{1}{\tau^k},0\right\}\right)V^T \tag{12}$$

where $\tau^k = N\mu^k$, $\max\{\cdot,\cdot\}$ should be understood elementwise, and $G_n^k = U\mathrm{diag}(\sigma)V^T$ is the SVD of $G_n^k = \mathcal{G}_{(n)}^{k+1} + Y_n^k/\mu^k$.

Remark 1: Here, only some matrices $G_n^k \in \mathbb{R}^{R_n'\times\Pi_{j\neq n}R_j'}$ of smaller sizes in (12) need to perform SVD. Thus, this step of our gHOI algorithm has a significantly lower computational complexity $O(\sum_n R_n^2\times\Pi_{j\neq n}R_j)$ in the worst case, while the computational complexity of existing convex TNM algorithms for (2)–(4) is $O(\sum_n I_n^2\times\Pi_{j\neq n}I_j)$ at each iteration.

Updating $\mathcal{Z}^{k+1}$: The optimization problem with respect to $\mathcal{Z}$ is formulated as follows:

$$\begin{aligned}&\min_{\mathcal{Z}}\|\mathcal{Z}-[\![\mathcal{G}^{k+1};U_1^{k+1},\ldots,U_N^{k+1}]\!]\|_F^2\\ &\text{s.t., } \mathcal{P}_\Omega(\mathcal{Z})=\mathcal{P}_\Omega(\mathcal{T}).\end{aligned} \tag{13}$$

By introducing a Lagrangian multiplier $\mathcal{Y}$ for $\mathcal{P}_\Omega(\mathcal{Z})=\mathcal{P}_\Omega(\mathcal{T})$, we write the Lagrangian function of (13) as follows:

$$\begin{aligned}\mathcal{F}(\mathcal{Z},\mathcal{Y})=&\|\mathcal{Z}-[\![\mathcal{G}^{k+1};U_1^{k+1},\ldots,U_N^{k+1}]\!]\|_F^2\\ &+\langle\mathcal{Y},\ \mathcal{P}_\Omega(\mathcal{Z})-\mathcal{P}_\Omega(\mathcal{T})\rangle.\end{aligned}$$

Letting $\nabla_{(\mathcal{Z},\mathcal{Y})}\mathcal{F}=0$, the Karush–Kuhn–Tucker (KKT) optimality conditions for (13) are given as follows:

$$\begin{aligned}2\big(\mathcal{Z}-[\![\mathcal{G}^{k+1};U_1^{k+1},\ldots,U_N^{k+1}]\!]\big)+\mathcal{P}_\Omega(\mathcal{Y})=0\\ \mathcal{P}_\Omega(\mathcal{Z})-\mathcal{P}_\Omega(\mathcal{T})=0.\end{aligned} \tag{14}$$

By deriving simply the KKT conditions (14), we have the optimal solution of (13) given by

$$\mathcal{Z}^{k+1}=\mathcal{P}_\Omega(\mathcal{T})+\mathcal{P}_\Omega^\perp\big([\![\mathcal{G}^{k+1};U_1^{k+1},\ldots,U_N^{k+1}]\!]\big) \tag{15}$$

where $\mathcal{P}_\Omega^\perp$ is the complementary operator of $\mathcal{P}_\Omega$.

Rank-Increasing Scheme: The idea of interlacing fixed-rank optimization with adaptive rank-adjusting schemes has appeared recently in the particular context of LRMC [43], [50]. Thus, it is here extended to our algorithm for solving the noisy CTNM problem (7). Considering the fact $\mathcal{L}_{\mu^k}(\mathcal{G}^{k+1},\{U_n^{k+1}\},\{V_n^{k+1}\},\mathcal{Z}^{k+1},\{Y_n^k\}) + \psi^k \le \mathcal{L}_{\mu^k}(\mathcal{G}^k,\{U_n^k\},\{V_n^k\},\mathcal{Z}^k,\{Y_n^k\})+\psi^k$ and $\psi^k = \sum_{n=1}^N\|Y_n^k\|_F^2/(2\mu^k)$, our rank-increasing scheme starts R_n', such that $R_n' \le R_n$. We increase R_n' to $\min(R_n'+\triangle R_n', R_n)$ at iteration $(k+1)$ if

$$\left|1-\frac{\mathcal{L}_{\mu^k}\big(\mathcal{G}^{k+1},\{U_n^{k+1}\},\{V_n^{k+1}\},\mathcal{Z}^{k+1},\{Y_n^k\}\big)+\psi^k}{\mathcal{L}_{\mu^k}\big(\mathcal{G}^k,\{U_n^k\},\{V_n^k\},\mathcal{Z}^k,\{Y_n^k\}\big)+\psi^k}\right|\le\epsilon$$

which $\triangle R_n'$ is a positive integer and ϵ is a small constant. Moreover, we augment $U_n^{k+1} \leftarrow [U_n^k, \widehat{U}_n]$ where $\widehat{H}_n$ has $\triangle R_n'$ randomly generated columns, $\widehat{U}_n = (I - U_n^k(U_n^k)^T)\widehat{H}_n$,

Algorithm 1 Solving CTNM Problem (9) via gHOI

Input: $\mathcal{P}_\Omega(\mathcal{T})$, $(R_1, \ldots, R_N)$, λ and tol.
Initialize: $Y_n^0 = \mathbf{0}, U_n^0 = \text{eye}(I_n, R_n'), V_n^0 = \mathbf{0}$, $n = 1, \ldots, N$, $\mathcal{G}^0 = \mathbf{0}$, $\mu^0 = 10^{-4}$, and $\mu_{\max} = 10^{10}$.
1: **while** not converged **do**
2: Update U_n^{k+1} and $\mathcal{G}^{k+1}$ by (18) and (20).
3: Update V_n^{k+1} and $\mathcal{Z}^{k+1}$ by (12) and (15).
4: Apply the rank-increasing scheme.
5: Update the multiplier Y_n^{k+1} by
$Y_n^{k+1} = Y_n^k + \mu^k(\mathcal{G}_{(n)}^{k+1} - V_n^{k+1})$, $n = 1, \ldots, N$.
6: Update μ^{k+1} by $\mu^{k+1} = \min(\rho\mu^k, \mu_{\max})$.
7: Check the convergence condition,
$\max(\|\mathcal{G}_{(n)}^{k+1} - V_n^{k+1}\|_F^2, n = 1, \ldots, N) < \text{tol}$.
8: **end while**
Output: $[\![\mathcal{G}^{k+1}; U_1^{k+1}, \ldots, U_N^{k+1}]\!]$.

and then orthonormalize $\widehat{U}_n$. Let $\mathcal{V}_n = \text{refold}(V_n^k) \in \mathbb{R}^{R_1' \times \cdots \times R_N'}$, where refold$(\cdot)$ denotes the refolding of the given matrix into a tensor, and $\mathcal{W}_n \in \mathbb{R}^{(R_1'+\triangle R_1') \times \cdots \times (R_N'+\triangle R_N')}$ be augmented as follows: $(\mathcal{W}_n)_{i_1 \cdots i_N} = (\mathcal{V}_n)_{i_1 \cdots i_N}$ for all $i_t \le R_t'$ and $t \in [1, N]$, and $(\mathcal{W}_n)_{i_1 \cdots i_N} = 0$ otherwise. Hence, we set $V_n^k = \mathcal{W}_{n,(n)}$ and augment Y_n^k by the same way. We then update $\mathcal{G}^{k+1}$, V_n^{k+1}, and $\mathcal{Z}^{k+1}$ by (20), (12), and (15), respectively.

Summarizing the analysis above, we develop an efficient ADAL algorithm for solving the STLD problem (9), as outlined in Algorithm 1. Algorithm 1 can be accelerated by adaptively changing μ [51]: 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. The convergence analysis of our algorithm is provided in Section IV-D. In addition, Algorithm 1 can be used to solve the noiseless STLD problem (6).

B. Generalized HOOI

We propose a GHOOI scheme to solve problem (10), while the HOOI problem in [17] can be seen as a special case of (10) when $\mu^k = 0$. Therefore, we extend [17, Th. 4.2] to solve problem (10) as follows.

Theorem 2: Assume a real Nth-order tensor $\mathcal{Z}^k$, then the minimization of (10) is equivalent to the maximization, over the matrices $U_1, U_2, \ldots, U_N$ having orthonormal columns, of the function

$$g(U_1, U_2, \ldots, U_N) = \left\|\lambda\mathcal{M} + \mu^k\mathcal{N}\right\|_F^2 \tag{16}$$

where $\mathcal{N} = \sum_{n=1}^N \text{refold}(V_n^k - Y_n^k/\mu^k)$ and $\mathcal{M} = [\![\mathcal{Z}^k; (U_1)^T, \ldots, (U_N)^T]\!]$.

The detailed proof of the theorem can be found in [6].

Updating $\{U_1^{k+1}, \ldots, U_N^{k+1}\}$: According to Theorem 2, our scheme successively solves U_n $(n = 1, \ldots, N)$ with fixing other variables U_j, $j \ne n$. Imagine that the matrices $U_1, \ldots, U_{n-1}, U_{n+1}, \ldots, U_N$ are fixed and that the optimization problem (16) is thought of as a quadratic expression in the components of the matrix U_n that is being optimized. Considering that the matrix has orthonormal columns, we have

$$\max_{U_n \in \text{St}(I_n, R_n')} \left\|\lambda\mathcal{M}_n \times_n U_n^T + \mu^k\mathcal{N}\right\|_F^2 \tag{17}$$

where $\mathcal{M}_n = \mathcal{Z}^k \times_1 (U_1^{k+1})^T \times_2 \cdots \times_{n-1} (U_{n-1}^{k+1})^T \times_{n+1} (U_{n+1}^k)^T \cdots \times_N (U_N^k)^T$. This is actually the well-known orthogonal procrustes problem [52], whose optimal solution is given by the SVD of $\mathcal{M}_{n,(n)}\mathcal{N}_{(n)}^T$

$$U_n^{k+1} = \text{Oth}\big(\mathcal{M}_{n,(n)}\mathcal{N}_{(n)}^T\big) \tag{18}$$

where $\text{Oth}(Z_n) := U^{(n)}(V^{(n)})^T$, and $U^{(n)}$ and $V^{(n)}$ are obtained by the skinny SVD of Z_n. Repeating the procedure above for different modes leads to an alternating orthogonal procrustes scheme for solving the maximization of problem (17). For any estimate of these factor matrices U_n, $n = 1, \ldots, N$, the optimal solution to problem (10) with respect to $\mathcal{G}$ is updated as follows.

Updating $\mathcal{G}^{k+1}$: The optimization problem (10) with respect to $\mathcal{G}$ can be rewritten as

$$\min_{\mathcal{G}} \sum_{n=1}^N \frac{\mu^k}{2}\left\|\mathcal{G}_{(n)} - V_n^k + Y_n^k/\mu^k\right\|_F^2 + \frac{\lambda}{2}\left\|\mathcal{Z}^k - [\![\mathcal{G}; U_1^{k+1}, \ldots, U_N^{k+1}]\!]\right\|_F^2. \tag{19}$$

Problem (19) is a smooth convex optimization problem, thus we can obtain a closed-form solution given by

$$\mathcal{G}^{k+1} = \frac{\lambda[\![\mathcal{Z}^k; \big(U_1^{k+1}\big)^T, \ldots, \big(U_N^{k+1}\big)^T]\!] + \mu^k\mathcal{N}}{\lambda + N\mu^k}. \tag{20}$$

C. Extension to RCTNM

Algorithm 1 can be extended to solve our RCTNM problem (8), where the main difference is that the subproblem with respect to $\{U_1, \ldots, U_N\}$ is formulated as follows:

$$\min_{\mathcal{G}, \{U_n \in \text{St}(I_n, R_n')\}} \sum_{n=1}^N \frac{\mu^k}{2}\left\|\mathcal{G}_{(n)} - V_n^k + Y_n^k/\mu^k\right\|_F^2 + \frac{\lambda}{2}\|\mathcal{Z}^k - [\![\mathcal{G}; U_1, \ldots, U_N]\!]\|_F^2 + \eta\sum_{n=1}^N \text{Tr}\big(U_n^T L_n U_n\big). \tag{21}$$

Similar to the derivation of [6, Th. 2], U_n can be solved by minimizing the following cost function:

$$F(U_1, \ldots, U_N) = -\frac{g(U_1, \ldots, U_N)}{\lambda + N\mu^k} + \eta\sum_{n=1}^N \text{Tr}\big(U_n^T L_n U_n\big).$$

To update U_n^{k+1}, the approximate procedure is given by

$$\begin{aligned} &-\frac{\left\|\lambda\mathcal{M}_n \times_n U_n^T + \mu^k\mathcal{N}\right\|_F^2}{\lambda + N\mu^k} + \eta\text{Tr}\big(U_n^T L_n U_n\big) \\ &\quad = -\frac{1}{\lambda + N\mu^k}(\langle U_n,\ \mathcal{P}(U_n)\rangle + 2\langle U_n,\ Q_n\rangle) + c \\ &\quad \approx -\frac{1}{\lambda + N\mu^k}\langle U_n,\ \mathcal{P}\big(U_n^k\big) + 2Q_n\rangle + c \end{aligned}$$

TABLE II
COMPLEXITIES PER ITERATION OF MAJOR COMPUTATIONS IN LRTC ALGORITHMS

Algorithms	Complexity
WTucker [26]	$O(2(N+1)RI^N)$
WCP [19]	$O(2(N+1)RI^N)$
TNM algorithms [12], [30], [33]	$O(NI^{N+1})$
CTNM	$O((N+1)RI^N)$

where c is a constant, $Q_n = \lambda\mu^k\mathcal{M}_{n,(n)}\mathcal{N}^T_{(n)}$, and $\mathcal{P}(U_n) = [\lambda^2\mathcal{M}_{n,(n)}\mathcal{M}^T_{n,(n)} - (\lambda + N\mu^k)\eta L_n]U_n$. Therefore, we consider the following maximization problem to update U_n^{k+1}:

$$\max_{U_n \in \mathrm{St}(I_n, R'_n)} \langle U_n,\ \mathcal{P}(U_n^k) + 2Q_n\rangle.$$

Following [52], we have:

$$U_n^{k+1} = \mathrm{Oth}\big(\mathcal{P}\big(U_n^k\big) + 2Q_n\big). \tag{22}$$

Furthermore, by repeating the procedure above for different modes, we can update the other factor matrices.

D. Convergence and Complexity Analysis

In this part, we first provide the convergence analysis of Algorithm 1 for solving the STLD problem (7).

Theorem 3: Let $\{(\mathcal{G}^k, U_1^k, \ldots, U_N^k, V_1^k, \ldots, V_N^k, \mathcal{Z}^k)\}$ be a sequence generated by Algorithm 1, then we have the following conclusions.

1) $\{\mathcal{G}^k\}$, $\{(U_1^k, \ldots, U_N^k)\}$, $\{(V_1^k, \ldots, V_N^k)\}$, $\{\mathcal{Z}^k\}$ are all Cauchy sequences.
2) If $\lim_{k\to\infty} \mu^k(V_n^{k+1} - V_n^k) = 0$, $n = 1, \ldots, N$, then the accumulation point of $\{(\mathcal{G}^k, U_1^k, \ldots, U_N^k)\}$ satisfies the KKT conditions for problem (7).

The proof of Theorem 3 may refer to [6]. Theorem 3 ensures that the feasibility of each solution produced by Algorithm 1 has been assessed.

Moreover, we compare the computational complexity of our CTNM method to some related methods. In this comparison, we assume that the input tensor is of size $I \times I \times \cdots \times I$, and the given ranks are $R_1 = \cdots = R_N = R$. The time complexity of performing operator (12) is $O(NR^{N+1})$. The time complexities of some multiplication operators in (15) and (18) are $O(RI^N)$ and $O(NRI^N)$, respectively. Hence, the total time complexity of CTNM is $O((N+1)RI^N)$. Table II summarizes complexities of the two weighted TD algorithms [19], [26] and the three convex tensor TNM algorithms [12], [30], [33].

V. RECOVERY GUARANTEES

In this section, we extend the partial observation theorem for matrices in [36] and [37] to higher order tensors, which involves a tensor covering number argument and the Hoeffding inequality for sampling without replacement [53]. Moreover, we also analyze the statistical performance of our model (7). For simplicity of discussion, we assume that the true Nth-order tensor $\mathcal{D}$ is of size $I \times I \times \cdots \times I$, and has multilinear rank $(r, r, \ldots, r)$ throughout this section, even though our analysis can easily be extended to the more general case, i.e., $\mathcal{D} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ with n-rank $(r_1, r_2, \ldots, r_N)$.

A. Covering Number of Low-Rank Tensors

To provide the recovery guarantee for higher order tensors, we need to extend the covering number argument for low-rank matrices in [36] and [37] to higher order tensors, as stated in Theorem 4 and Lemma 1.

Theorem 4 (Covering Number for Low-Rank Tensors of Bounded Size): Let $\mathcal{S}_R = \{\mathcal{X} \in \mathbb{R}^{I\times I\times\cdots\times I} \,|\, n\text{-rank}(\mathcal{X}) \preceq (R, R, \ldots, R),\ \|\mathcal{X}\|_F \le \gamma\}$. Then, there exists an ϵ-net $\overline{\mathcal{S}}_R$ with the covering number $|\overline{\mathcal{S}}_R|$ for the Frobenius norm obeying

$$|\overline{\mathcal{S}}_R| \le (3\gamma\,(N+1)/\epsilon)^{R^N + NRI}. \tag{23}$$

To prove Theorem 4, we first give Lemma 1, which uses the triangle inequality to characterize the combined effects of perturbations in the factors of the generalized Tucker decomposition in (7).

Lemma 1: Let $\mathcal{G}, \overline{\mathcal{G}} \in \mathbb{R}^{R_1\times R_2\times\cdots\times R_N}$ and $U_n, \overline{U}_n \in \mathbb{R}^{I_n\times R_n}$ with $U_n^T U_n = I_{R_n}$, $\overline{U}_n^T \overline{U}_n = I_{R_n}$, $\|\mathcal{G}\|_F \le \gamma$ and $\|\overline{\mathcal{G}}\|_F \le \gamma$. Then

$$\begin{aligned}\|[\![\mathcal{G}; U_1, \ldots, U_N]\!] - [\![\overline{\mathcal{G}}; \overline{U}_1, \ldots, \overline{U}_N]\!]\|_F \\ \le \|\mathcal{G} - \overline{\mathcal{G}}\|_F + \gamma \sum_n \|U_n - \overline{U}_n\|_2.\end{aligned} \tag{24}$$

The lemma is in essence the same as [31, Lemma 2], and has the only difference of the ranges of $\|\mathcal{G}\|_F$ and $\|\overline{\mathcal{G}}\|_F$: instead of having $\|\mathcal{G}\|_F = 1$ and $\|\overline{\mathcal{G}}\|_F = 1$, we have $\|\mathcal{G}\|_F \le \gamma$ and $\|\overline{\mathcal{G}}\|_F \le \gamma$. Using Lemma 1, we construct an ε-net for $\mathcal{S}_R$ by building $\epsilon/(N+1)$-nets for each of the $(N+1)$ factors $\{U_n\}$ and $\mathcal{G}$, and give the proof of Theorem 4 in the following.

Proof: Following [36, Lemma 3.1] and [31, Lemma 3], we construct an ϵ-net $\overline{\mathcal{S}}_R$ by building $\epsilon/(N+1)$-nets for each of the $(N+1)$ factors to approximate $\mathcal{S}_R$ to within distance ϵ with respect to the Frobenius norm.

Let $\mathcal{W} = \{\mathcal{G} \in \mathbb{R}^{R\times R\times\cdots\times R} \,|\, \|\mathcal{G}\|_F \le \gamma\}$ and $\mathcal{O} = \{U \in \mathbb{R}^{I\times R} \,|\, U^T U = I_R\}$. Clearly, for any $\mathcal{G} \in \mathcal{W}$, $\|\mathcal{G}\|_F \le \gamma$, as shown in [36, Sec. III], there exists an $\epsilon/(N+1)$-net $\overline{\mathcal{W}}$ covering $\mathcal{W}$ with respect to the Frobenius norm, such that

$$|\overline{\mathcal{W}}| \le \frac{\mathrm{Vol}\big(\mathcal{W} + \frac{\mathcal{D}}{2}\big)}{\mathrm{Vol}\big(\frac{\mathcal{D}}{2}\big)}$$

where $(\mathcal{D}/2)$ is an $\epsilon/2(N+1)$ ball (with the Frobenius norm) and $\mathcal{W} + (\mathcal{D}/2) = \{\mathcal{G} + \mathcal{B} | \mathcal{G} \in \mathcal{W}, \mathcal{B} \in \mathcal{D}\}$. By $\|\mathcal{G}\|_F \le \gamma$, then $\mathcal{W} + (\mathcal{D}/2)$ is contained in the $\gamma + (\epsilon/2(N+1))$ ball, and thus

$$|\overline{\mathcal{W}}| \le \left(\frac{\gamma + \frac{\epsilon}{2(N+1)}}{\frac{\epsilon}{2(N+1)}}\right)^{R^N} \le \left(\frac{\gamma + \frac{\gamma}{2}}{\frac{\epsilon}{2(N+1)}}\right)^{R^N} = \left(\frac{3\gamma\,(N+1)}{\epsilon}\right)^{R^N}.$$

The second inequality is due to the fact that $(\epsilon/N+1) < \gamma$ [29]. Similarly, for any $U_n \in \mathcal{O}$, $\|U_n\|_2 = 1$, there exists an $\epsilon/(\gamma\,(N+1))$-net $\overline{\mathcal{O}}$ covering $\mathcal{O}$ with respect to the spectral norm, such that $|\overline{\mathcal{O}}| \le (3\gamma\,(N+1)/\epsilon)^{RI}$.

Let $\overline{\mathcal{S}}_R = \{[\![\overline{\mathcal{G}}; \overline{U}_1, \ldots, \overline{U}_N]\!] \,|\, \overline{\mathcal{G}} \in \overline{\mathcal{W}}, \overline{U}_n \in \overline{\mathcal{O}}\}$. Clearly, $|\overline{\mathcal{S}}_R| \le (3\gamma\,(N+1)/\epsilon)^{R^N + NRI}$. It remains to show that $\overline{\mathcal{S}}_R$ is indeed an ϵ-net covering $\mathcal{S}_R$ with respect to the Frobenius norm. For any given $\mathcal{X} = [\![\mathcal{G}; U_1, \ldots, U_N]\!] \in \mathcal{S}_R$, where $\mathcal{G} \in \mathcal{W}$ and $U_n \in \mathcal{O}$, there exist $\overline{\mathcal{G}} \in \overline{\mathcal{W}}$ and $\overline{U}_n \in \overline{\mathcal{O}}$,

such that $\|\mathcal{G}-\overline{\mathcal{G}}\|_F \leq \epsilon/(N+1)$ and $\|U_n-\overline{U}_n\|_2 \leq \epsilon/(\gamma(N+1))$. By Lemma 1, we have

$$\begin{aligned}\|\mathcal{X}-\overline{\mathcal{X}}\|_F &= \|[\![\mathcal{G};U_1,\ldots,U_N]\!]-[\![\overline{\mathcal{G}};\overline{U}_1,\ldots,\overline{U}_N]\!]\|_F \\ &\leq \|\mathcal{G}-\overline{\mathcal{G}}\|_F+\sum_n \gamma\|U_n-\overline{U}_n\|_2 \leq \epsilon.\end{aligned}$$

In other words, $\overline{\mathcal{X}}=[\![\overline{\mathcal{G}};\overline{U}_1,\ldots,\overline{U}_N]\!] \in \overline{\mathcal{S}}_R$ is within ϵ-distance from $\mathcal{X}$. ∎

B. Partial Observation Theorem

We give the partial observation theorem for higher order tensor recovery, which involves the covering number argument in Theorem 4 and the Hoeffding inequality for sampling without replacement [53], as stated in Theorem 4.

Theorem 5: Let $\mathcal{L}(\mathcal{X}) = (1/\sqrt{I^N})\|\mathcal{X}-\widehat{\mathcal{X}}\|_F$ and $\widehat{\mathcal{L}}(\mathcal{X})=(1/\sqrt{|\Omega|})\|\mathcal{P}_\Omega(\mathcal{X}-\widehat{\mathcal{X}})\|_F$ be the actual and empirical loss function, respectively, where $\mathcal{X}, \widehat{\mathcal{X}} \in \mathbb{R}^{I\times I\times\cdots\times I}$. Furthermore, assume entrywise constraint $\max_{n_1,n_2,\ldots,n_N}|\mathcal{X}_{n_1n_2\cdots n_N}| \leq \beta$. Then, for all tensors $\mathcal{X}$ with n-rank$(\mathcal{X}) \preceq (R,R,\ldots,R)$, with probability greater than $1-2\exp(-I)$, there exists a fixed constant C, such that

$$\sup_{\mathcal{X}\in\mathcal{S}_R}|\widehat{\mathcal{L}}(\mathcal{X})-\mathcal{L}(\mathcal{X})| \leq C\beta\left(\frac{(R^N+NRI)\log(\sqrt{I^N})}{|\Omega|}\right)^{1/4}.$$

Indeed, the proof sketch for Theorem 5 follows that of [37, Th. 2], and their main difference is that the covering number argument in Theorem 4 for higher order tensors is used to replace that of [37, Lemma A2] for low-rank matrices.

Proof: Following the proof procedure of [37, Th. 2] and using the Hoeffding inequality theorem and Theorem 4, we have:

$$\begin{aligned}&\sup_{\mathcal{X}\in\mathcal{S}_R}|\widehat{\mathcal{L}}(\mathcal{X})-\mathcal{L}(\mathcal{X})| \\ &\leq \frac{2\epsilon}{\sqrt{|\Omega|}}+\left(\frac{M^2}{2}\frac{(2R^N+2NRI)\log(3\beta\sqrt{I^N}(N+1)/\epsilon)}{|\Omega|}\right)^{1/4} \\ &\leq \frac{6\beta(N+1)}{\sqrt{|\Omega|}}+2\beta\left(\frac{(R^N+NRI)\log(\sqrt{I^N})}{|\Omega|}\right)^{1/4} \\ &= \left(2+\frac{6(N+1)}{[|\Omega|(R^N+NRI)\log(\sqrt{I^N})]^{1/4}}\right)\beta \\ &\quad\times\left(\frac{(R^N+NRI)\log(\sqrt{I^N})}{|\Omega|}\right)^{1/4}\end{aligned}$$

where $\gamma=\sqrt{I^N}\beta \geq \|\mathcal{X}\|_F$, $M:=\max_{n_1,\ldots,n_N}(\mathcal{X}_{n_1\ldots n_N}-\widehat{\mathcal{X}}_{n_1\ldots n_N})^2 \leq (2\beta)^2$, and $\epsilon=3\beta(N+1)$. Hence, C can be set to $2+6(N+1)/[|\Omega|(R^N+NRI)\log(\sqrt{I^N})]^{1/4}$. ∎

C. Recovery Bound

In this part, we will show that when sufficiently many entries are sampled, the KKT point of Algorithm 1 is stable, i.e., it recovers a tensor close to the ground-truth one. We assume that the observed tensor $\mathcal{T}\in\mathbb{R}^{I\times I\times\cdots\times I}$ can be decomposed as a true tensor $\mathcal{D}$ with n-rank $(r,\ldots,r)$ and a random gaussian noise $\mathcal{E}$, whose entries are independently drawn from $\mathcal{N}(0,\sigma^2)$, i.e., $\mathcal{T}=\mathcal{D}+\mathcal{E}$. The root mean square error (RMSE) is a frequently used measure of the difference between the tensor $\mathcal{X}=[\![\mathcal{G};U_1,\ldots,U_N]\!]$ recovered by gHOI and the true one

$$\text{RMSE} := \frac{1}{\sqrt{I^N}}\|\mathcal{D}-[\![\mathcal{G};U_1,\ldots,U_N]\!]\|_F. \tag{25}$$

Definition 3: The operator $\mathcal{P}_S$ is defined as follows: $\mathcal{P}_S(\mathcal{X})=P_{U_N}\ldots P_{U_1}(\mathcal{X})$, where $P_{U_n}(\mathcal{X})=\mathcal{X}\times_n(U_nU_n^T)$.

Using the partial observation theorem for higher order tensors, we further improve [6, Th. 4] as follows.

Theorem 6: Let $(\mathcal{G},U_1,U_2,\ldots,U_N)$ satisfy the KKT conditions for problem (7) with regularization constant $\lambda=\sqrt{R}/\|\mathcal{P}_\Omega(\mathcal{E})\|_F$ and given multilinear rank $R_1=\cdots=R_N=R$. Then, there exists an absolute constant C, such that with probability at least $1-2\exp(-I)$

$$\begin{aligned}\text{RMSE} \leq \frac{\|\mathcal{E}\|_F}{\sqrt{I^N}}+C\beta\left(\frac{(R^N+NRI)\log(\sqrt{I^N})}{|\Omega|}\right)^{\frac{1}{4}} \\ +\frac{\|\mathcal{P}_\Omega(\mathcal{E})\|_F}{C_1\sqrt{|\Omega|}}\end{aligned} \tag{26}$$

where $\beta=\max_{i_1,\ldots,i_N}|\mathcal{T}_{i_1\ldots i_N}|$, and $C_1=(\|\mathcal{P}_S\mathcal{P}_\Omega(\mathcal{T}-\mathcal{X})\|_F/\|\mathcal{P}_\Omega(\mathcal{T}-\mathcal{X})\|_F)$.

Remark 2: The detailed proof of the theorem can be found in Appendix B. Moreover, the analysis of lower boundedness of C_1 can be found in [6]. When the samples size $|\Omega| \gg (R^N+NRI)\log(\sqrt{I^N})$, the second term diminishes, and the RMSE is essentially bounded by the average magnitude of entries of the noise tensor $\mathcal{E}$, that is, our CTNM method is stable. In addition, when $\mathcal{E}=0$, $\mathcal{X}$ can be obtained using the noiseless model (6). Our formulation (6) needs only $O(R^N+NRI)$ observations to exactly recover all $\mathcal{D}\in\mathcal{S}_r$ ($r\leq R$. The conclusion is a routine extension of [31, Th. 1]), while $O(rI^{N-1})$ observations are required for recovering the true tensor by those convex TNM methods [7], [12], [29], [30], [34] (as stated in [31, Th. 3]), which will be confirmed by the experimental results in Section VI.

VI. Experiments

In this section, we evaluate the effectiveness and efficiency of our CTNM method on synthetic and real-world data for face reconstruction, medical image inpainting, and multirelational prediction, where our algorithm was implemented in MATLAB 7.11, using the Tensor Toolbox version 2.6 [54] for handling some of the tensor operations. Except for multirelational prediction, all the other experiments were performed on an Intel(R) Core (TM) i5-4570 (3.20 GHz) PC running Windows 7 with 8-GB main memory.

A. Synthetic Tensor Completion

Following [12], we generated some low-rank tensors $\mathcal{D}\in\mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$, which we used as the ground-truth data. The synthetic tensors follow the Tucker model, i.e., $\mathcal{D}=\mathcal{G}\times_1 U_1\times_2\cdots\times_N U_N$, where the order of the tensors varies from three to five, and r is usually set to 10. The subset of observed entries of the input tensors should be uniformly

TABLE III

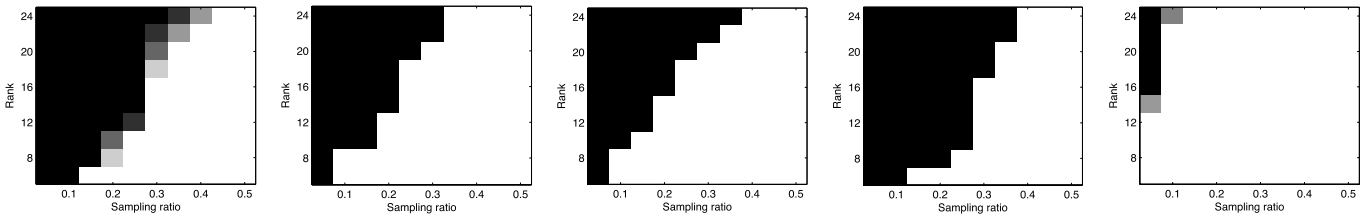
RSE AND RUNNING TIME (SECONDS) COMPARISON ON SYNTHETIC TENSORS. (a) TENSOR SIZE: $30 \times 30 \times 30 \times 30 \times 30$. (b) TENSOR SIZE: $60 \times 60 \times 60 \times 60$

(a)

	WTucker		WCP		FaLRTC		Latent		CTNM	
SR	RSE±std.	Time	RSE±std.	Time	RSE±std.	Time	RSE±std.	Time	RSE±std.	Time
10%	0.4982±2.3e-2	2163.05	0.5003±3.6e-2	4359.23	0.6744±2.7e-2	1575.78	0.6268±5.0e-2	8324.17	**0.1031±1.2e-2**	**132.95**
30%	0.1562±1.7e-2	2226.67	0.3364±2.3e-2	3949.57	0.3153±1.4e-2	1779.59	0.2443±1.2e-2	8043.83	**0.0269±6.0e-3**	**112.67**
50%	0.0490±9.3e-3	2652.90	0.0769±5.0e-3	3260.86	0.0365±6.2e-4	2024.52	0.0559±7.7e-3	8263.24	**0.0055±1.3e-3**	**126.50**

(b)

	WTucker		WCP		FaLRTC		Latent		CTNM	
SR	RSE±std.	Time	RSE±std.	Time	RSE±std.	Time	RSE±std.	Time	RSE±std.	Time
10%	0.2319±3.6e-2	1437.61	0.4766±9.4e-2	1586.92	0.4927±1.6e-2	562.15	0.5061±4.4e-2	5075.82	**0.0381±4.2e-4**	**36.64**
30%	0.0143±2.8e-3	1756.95	0.1994±6.0e-3	1696.27	0.1694±2.5e-3	603.49	0.1872±7.5e-3	5559.17	**0.0026±3.1e-4**	**31.37**
50%	0.0079±6.2e-4	2534.59	0.1335±4.9e-3	1871.38	0.0602±5.8e-4	655.69	0.0583±9.7e-4	6086.63	**0.0006±0.1e-4**	**32.29**



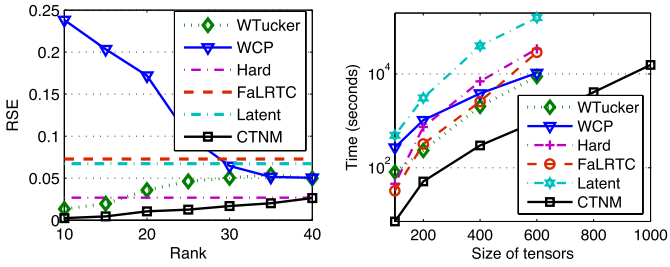
(a) (b) (c) (d) (e)

Fig. 1. Phase transition plots on the third-order tensors. White region: 100% success in all experiments. Black region: 0% success. Gray region: some intermediate value. (a) WTucker. (b) Hard. (c) FaLRTC. (d) Latent. (e) CTNM.

and randomly chosen, and they are recovered with various sampling ratios (SRs) by our noiseless CTNM model (6), two weighted TD algorithms: WTucker[2] [26] and WCP[3] [19], and three convex LRTC algorithms: FaLRTC[4] [12], Latent[5] [33], and Hard[6] [30]. We set the multilinear rank $R_1 = \cdots = R_N = \lfloor 1.2r \rfloor$ for WTucker and CTNM, and the tensor rank $R = 40$ for WCP. For FaLRTC, the weights α_n are set to be $\alpha_n = 1/N$, and the smoothing parameters are set to be $5\alpha_n/I_n$, $n = 1, \ldots, N$. For Hard, we let $\tau = 10^2$ and $\lambda_1 = \lambda_2 = \lambda_3 = 1$. We set the tolerance value to tol $= 10^{-4}$ for all these algorithms, and the maximal number of iterations to maxiter $= 50$ for WTucker and WCP, and maxiter $= 500$ for the other methods. The relative square error (RSE) of the recovered tensor $\mathcal{X}$ is defined by RSE $:= \|\mathcal{X} - \mathcal{D}\|_F / \|\mathcal{D}\|_F$.

Table III shows the average recovery accuracy (RSE) and running time (seconds) of ten independent runs, where the order of tensors varies from four to five. In the table, CTNM can consistently yield much more accurate solutions, and outperform the other algorithms in terms of both RSE and efficiency. The empirical performance of all these LRTC methods can be charted using phase transition plots, which use gray-scale colors to depict how likely a certain kind of low-rank tensors can be recovered by those algorithms for a range of different ranks and SRs. If the relative error RSE $\leq 10^{-2}$, we declare the incomplete tensor $\mathcal{T}$ to be recovered. Fig. 1 shows the phase transition plots of all those algorithms, except WCP, on the third-order tensors

[2]http://www.lair.irb.hr/ikopriva/marko-filipovi.html
[3]http://www.sandia.gov/~tgkolda/TensorToolbox/index-2.6.html
[4]http://www.cs.rochester.edu/u/jliu/publications.html
[5]http://ttic.uchicago.edu/~ryotat/softwares/tensor/
[6]https://sites.google.com/site/marcosignoretto/codes



(a) (b)

Fig. 2. Comparison of all these methods in terms of RSE and computational time (in seconds and in logarithmic scale) on the third-order tensors by varying given (a) ranks or (b) tensor sizes.

of size $100 \times 100 \times 100$, where the x-axis denotes the SR varying from 5% to 50% with increment 5%, and the y-axis corresponds to the multilinear rank r_n, $n = 1, 2, 3$ varying from 6 to 24 with increment 2. For each setting, ten independent trials were run. In the figure, we can observe that CTNM performs significantly better than the other methods.

To further evaluate the robustness of our CTNM method with respect to the given tensor rank changes, we conduct some experiments on the synthetic data of size $100 \times 100 \times 100$, and illustrate the recovery results of all these methods with 20% SR, where the rank parameter of CTNM, WTucker, and WCP is chosen from $\{10, 15, \ldots, 40\}$. The average RSE results of ten independent runs are shown in Fig. 2(a), from which we can see that CTNM performs much more robust with respect to multilinear ranks than both WTucker and WCP. This confirms that our CTNM model with trace-norm regularization can provide a good low-rank

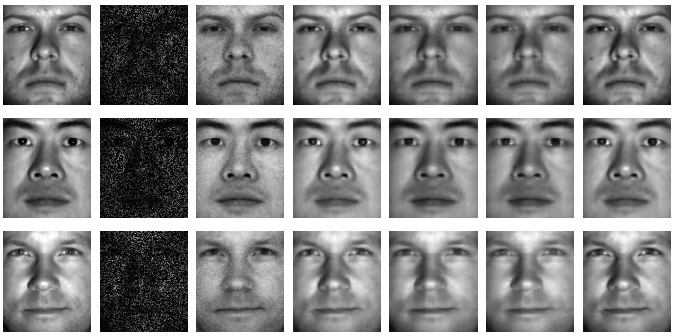


Fig. 3. Face reconstruction results with 20% SR. From left column to right column: original images, input images (black pixels denote missing entries), and reconstruction results by ALM, WTucker, Hard, FaLRTC, and CTNM.

estimation of the observed tensor in the presence of missing data.

Moreover, we report the running time of our CTNM method and the other methods on the third-order tensors with varying sizes, as shown in Fig. 2(b). We can see that the running time of WTcuker, WCP, Hard, Latent, and FaLRTC dramatically grows with the increase of tensor sizes, whereas the running time of CTNM only increases slightly. Furthermore, WTcuker, WCP, Hard, Latent, and FaLRTC could not yield experimental results on the two largest problems with the sizes of 800 and 1000, because they ran out of memory. This shows that CTNM has a very good scalability and can address large-scale problems. Notice that because Latent has similar recovery accuracy to FaLRTC and Hard, and converges too slowly, we do not consider it in the following.

B. Face Reconstruction

In this part, we apply various LRTC methods to face reconstruction problems and compare their performance. In the experiments, we use a part of extended Yale face database B [55], which consists of 320 frontal face images of the first five classes, and each subset contains 64 images with varying illumination conditions and heavily shadows. The resolution of all images is 192×168, and the intensity of each pixel was normalized to [0, 1], then the pixel values were used to form a third-order tensor of size $192 \times 168 \times 320$. The input tensors are generated by setting 90% or 80% of randomly selected pixels of each image as missing entries. The tolerance value of these methods is set to tol $= 10^{-4}$. For WTucker and CTNM, we set the multilinear rank $R_1 = R_2 = R_3 = 30$. We could not apply WCP and Latent to the problem, because they have long computational time (more than an hour).

Fig. 3 shows some examples of face images, input images with missing entries and their reconstruction faces by the augmented Lagrange multiplier (ALM) method [51], WTucker, Hard, FaLRTC, and CTNM, respectively. Note that ALM is a well-known LRMC method for solving the problem, where the pixel values of each image are converted to a vector of dimension 32 256. Moreover, we report the average reconstruction results, including the recovery accuracy (RSE) and running time (seconds), with 10% or 20% SR in Table IV. From the results, we can observe that all those LRTC methods

TABLE IV
RSE AND TIME COST (SECONDS) COMPARISON FOR FACE RECONSTRUCTION

	10%		20%	
Methods	RSE	Time	RSE	Time
ALM	0.3818	318.89	0.3472	407.71
WTucker	0.1521	1816.44	0.0942	1927.23
Hard	0.2098	949.18	0.1105	614.30
FaLRTC	0.1807	586.83	0.1007	465.74
CTNM	**0.1029**	**66.14**	**0.0763**	**57.32**

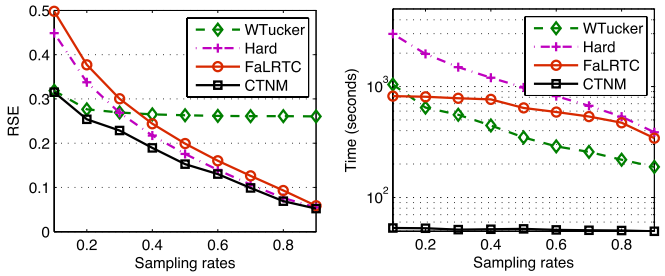


Fig. 4. Comparison of WTucker, Hard, FaLRTC, and CTNM in terms of RSE (left) and computational time (right) in seconds and in logarithmic scale.

perform significantly better than ALM in terms of RSE. This further confirms that those LRTC methods can utilize much more information contained in higher order tensors than the LRMC method, as stated in [12]. In addition, CTNM outperforms the other methods in terms of both RSE and efficiency. In particular, the lower SR, the more obvious the improvement is. Except ALM, which gives the poorest recovery accuracy, all the compared methods are ∼8–30 times slower than CTNM.

C. Medical Images Inpainting

In this part, we apply our CTNM method for the medical image inpainting and decomposition problem, and compare CTNM against WTucker, FaLRTC, and Hard on the BRAINIX date set. This data set is from the OsirX repository,[7] and consists of 100 images of size 256×256. Thus, it is represented as a third-order tensor. The tolerance value of these methods is fixed at tol $= 10^{-4}$. For WTucker and CTNM, we set the multilinear rank $R_1 = R_2 = R_3 = 40$.

We report the recovery accuracy (RSE) and running time (seconds) on the BRAINIX data set with various SRs, as shown in Fig. 4. It is clear that CTNM consistently performs better than the other methods in terms of both RSE and efficiency. In particular, when the SR is low (e.g., 10%), CTNM reaches significantly smaller RSE results than FaLRTC and Hard. Moreover, it is ∼20 times faster than WTucker and FaLRTC, and >50 times faster than Hard. By increasing the SR, the RSE results of three trace-norm regularized methods: 1) FaLRTC; 2) Hard; and 3) CTNM, dramatically reduce, whereas that of WTucker decreases slightly. This can be explained as follows: the former three methods can be viewed as the trace-norm regularized method, in which the trace-norm term can effectively avoid over-fitting

[7] http://www.osirix-viewer.com/datasets/

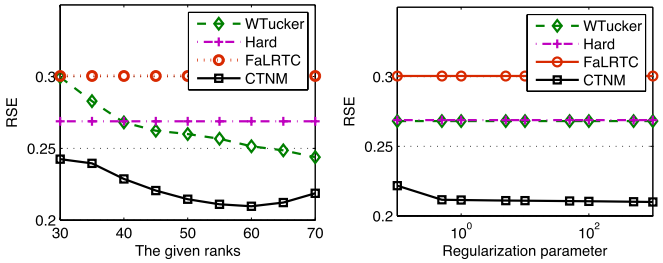
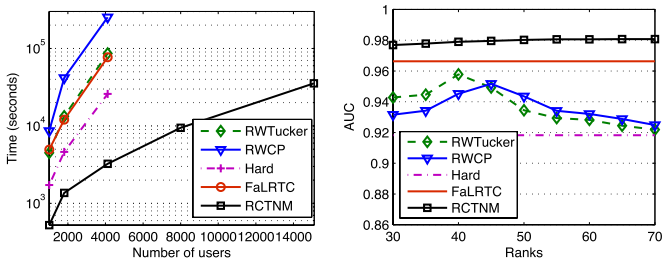


Fig. 5. Recovery results of CTNM against its parameters on the BRAINIX data set with 30% SR. Left: RSE versus the given ranks. Right: RSE versus the regularization parameter λ.



(a) (b)

Fig. 6. (a) Time cost and (b) prediction accuracy comparison on the YouTube data set. For each data set, we use 20% for training. Note that RWTucker, RWCP, FaLRTC, and Hard could not run for sizes $\{8,000, 15,088\}$ due to run-time exceptions.

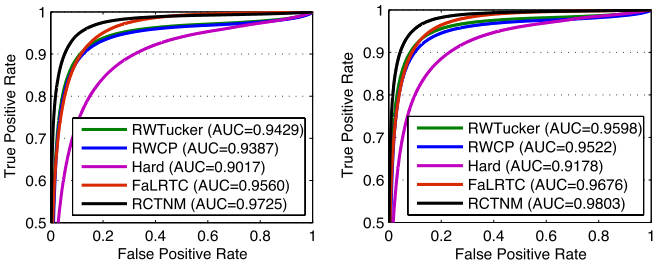


Fig. 7. Average ROC curves showing the performance of multirelational prediction methods with 10% (left) and 20% (right) training data, respectively.

phenomenon, while WTucker is a least-square method, which inevitably suffers from overfitting.

Moreover, we also evaluate the robustness of our CTNM method with respect to its parameters: the given multilinear ranks and the regularization parameter λ on the BRAINIX data set, as shown in Fig. 5, from which we can see that CTNM is very robust against its parameter variations, and consistently outperforms the other method in terms of RSE in all these parameter settings. Note that the regularization parameter λ is chosen from $\{0.1, 0.5, 1, \ldots, 10^3\}$. These results further confirm that our TD model with trace-norm regularization is much more robust with respect to the given multilinear rank than WTucker.

D. Multirelational Prediction

Finally, we apply our CTNM method to incorporate auxiliary information induced from the relationships, and examine our graph RCTNM method for multirelational prediction problems on a real-world network data set, YouTube[8] [56]. 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, and so on. 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, cocontact network, cosubscription network, cosubscribed network, and favorite network. Additional information about the data can be found in [56]. We ran these experiments on a machine with 6-core Intel Xeon 2.4 GHz CPU and 64 GB memory. For graph regularized WTucker (RWTucker) and regularized WCP (RWCP) [44], and our RCTNM method, we set the multilinear rank $R_1 = R_2 = 40$ and $R_3 = 5$. For FaLRTC and RCTNM, the weights α_n are set to be $\{0.4998, 0.4998, 0.0004\}$. The tolerance value of all these methods is fixed at tol $= 10^{-4}$.

We use the 15 088 users who share all of the information types and have five-dimensions of interactions in our experiments, thus the data size is $15\,088 \times 15\,088 \times 5$. We first report the average running time (seconds) of all these algorithms in Fig. 6(a), where the number of users is gradually increased. RCTNM is much faster than the other methods, and its running time increases only slightly when the number of users increases. On the contrary, the running time of RWTucker, RWCP, FaLRTC, and Hard increases dramatically. They could not yield experimental results within 48 h when the number of users is 8000 or 15 088. This shows that RCTNM has a very good scalability and can address large-scale problems. Moreover, we illustrates the prediction accuracy (the score area under the receiver operating characteristic curve, AUC) of RWTucker, RWCP, and RCTNM against their rank parameter in Fig. 6(b), from which we can observe that RCTNM consistently outperforms than the other methods in all those settings.

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 subset of size $4117 \times 4117 \times 5$. We randomly select 10% and 20% of the entries, respectively, as the training set, and the remainder as the testing data. We report the average prediction accuracy (AUC) over ten independent runs in Fig. 7 for both of the SRs, 10% and 20% of the entries, respectively. The results show that RCTNM performs significantly better than the other methods in terms of prediction accuracy. Moreover, RCTNM based on 20% of the entries further improves the prediction accuracy compared with RCTNM based on 10% of the entries.

VII. Conclusion and Future Work

In this paper, we proposed an efficient and scalable CTNM method for STLD problems. First, we induced the equivalence relation of the trace norm of a low-rank tensor and its core tensor. Then, we used the trace norm of the involved core tensor to replace that of the whole tensor, and then attained two much smaller scale matrix TNM problems. Therefore, our

[8]http://leitang.net/

CTNM method has a much lower computational complexity. Finally, we developed an efficient ADAL algorithm with a convergence guarantee. Moreover, we also established the theoretical recovery bound for our noisy CTNM model using random Gaussian measurements. Compared with $O(rI^{N-1})$ observations required by existing convex tensor TNM methods, our significantly improved sampling complexity, $O((R^N + NRI)\log(\sqrt{I^N}))$, is sufficient to recover a low-rank tensor with high probability, which has been confirmed by the convincing experimental results of our CTNM method.

Our CTNM method is significantly faster than the state-of-the-art LRTC methods. In the future, we will explore discriminant information as in [57]–[59] or incorporate active learning techniques [60] to further improve classification accuracy for pattern recognition problems. In addition, we will apply our CTNM models to address a variety of robust TD and representation learning problems, e.g., higher order robust PCA [2] and K-SVD [61].

APPENDIX A

Assume that $(\mathcal{G}^*, \{U_n^*\}, \{V_n^*\}, \mathcal{Z}^*)$ is a critical point of (9). Since $\mathcal{Z}^*$ satisfies the KKT condition of (9), and according to the similar derivation for (14) and (15), the following holds:

$$\mathcal{Z}^* = \mathcal{P}_\Omega(\mathcal{T}) + \mathcal{P}_\Omega^\perp([\![\mathcal{G}^*; U_1^*, \ldots, U_N^*]\!]).$$

Thus, we have

$$\begin{aligned}
&[\![\mathcal{G}^*; U_1^*, \ldots, U_N^*]\!] - \mathcal{Z}^* \\
&\quad = \mathcal{P}_\Omega([\![\mathcal{G}^*; U_1^*, \ldots, U_N^*]\!]) - \mathcal{P}_\Omega(\mathcal{Z}^*) \\
&\qquad + \mathcal{P}_\Omega^\perp([\![\mathcal{G}^*; U_1^*, \ldots, U_N^*]\!]) - \mathcal{P}_\Omega^\perp(\mathcal{Z}^*) \\
&\quad = \mathcal{P}_\Omega([\![\mathcal{G}^*; U_1^*, \ldots, U_N^*]\!]) - \mathcal{P}_\Omega(\mathcal{T}). \qquad (27)
\end{aligned}$$

Let $\mathcal{Q}^* = [\![\mathcal{G}^*; U_1^*, \ldots, U_N^*]\!] - \mathcal{Z}^*$ and $B_n^* = \mathcal{G}_{(n)}^*(U_N^* \otimes \cdots \otimes U_{n+1}^* \otimes U_{n-1}^* \otimes \cdots \otimes U_1^*)^T$ $(n = 1, 2, \ldots, N)$, then the following holds:

$$\|\mathcal{Q}^*\|_F^2 = \|U_n^* B_n^* - \mathcal{Z}_{(n)}^*\|_F^2. \qquad (28)$$

Thus, the critical point $(\mathcal{G}^*, \{U_n^*\}, \{V_n^*\}, \mathcal{Z}^*)$ of problem (9) satisfies the following KKT conditions:

$$\begin{aligned}
&0 \in \frac{1}{N}\partial\|V_n^*\|_* - \Lambda_n, \quad n = 1, 2, \ldots, N \\
&\lambda[\![\mathcal{Q}^*; (U_1^*)^T, \ldots, (U_N^*)^T]\!] + \sum_{n=1}^N \text{refold}(\Lambda_n) = 0 \\
&\mathcal{Q}_{(n)}^*(B_n^*)^T + U_n^*\Gamma_n = 0, \quad n = 1, 2, \ldots, N \\
&\mathcal{Z}^* = \mathcal{P}_\Omega(\mathcal{T}) + \mathcal{P}_\Omega^\perp([\![\mathcal{G}^*; U_1^*, \ldots, U_N^*]\!]) \qquad (29)
\end{aligned}$$

where $U_n^* \in \text{St}(I_n, R_n)$, $\Lambda_n \in \mathbb{R}^{R_n \times \Pi_{j\neq n} R_j}$ and $\Gamma_n \in \mathbb{R}^{R_n \times R_n}$ denote the Lagrange multipliers for all $n \in \{1, 2, \ldots, N\}$, and the derivation of (3) in (29) is similar to [62]. By (27) and $V_n^* = \mathcal{G}_{(n)}^*$, (29) is reformulated as follows:

$$\begin{aligned}
0 \in \frac{\sum_{n=1}^N \text{refold}(\partial\|\mathcal{G}_{(n)}^*\|_*)}{N} + \lambda[\![\mathcal{R}^*; (U_1^*)^T, \ldots, (U_N^*)^T]\!] \\
\mathcal{R}_{(n)}^*(B_n^*)^T + U_n^*\Gamma_n = 0, \quad n = 1, 2, \ldots, N \qquad (30)
\end{aligned}$$

where $\mathcal{R}^* = \mathcal{P}_\Omega([\![\mathcal{G}^*; U_1^*, \ldots, U_N^*]\!]) - \mathcal{P}_\Omega(\mathcal{T}) = \mathcal{Q}^*$. It is easy to see that (30) is the KKT conditions for problem (7), that is, $(\mathcal{G}^*, \{U_n^*\})$ is also the critical point of (7).

On the other hand, if $(\mathcal{G}^*, \{U_n^*\})$ is a critical point of (7), and let $\mathcal{Z}^* = \mathcal{P}_\Omega(\mathcal{T}) + \mathcal{P}_\Omega^\perp([\![\mathcal{G}^*; U_1^*, \ldots, U_N^*]\!])$ and $V_n^* = \mathcal{G}_{(n)}^*$, then we can know that $(\mathcal{G}^*, \{U_n^*\}, \{V_n^*\}, \mathcal{Z}^*)$ is also the critical point of problem (9). Hence, problem (7) is equivalent to problem (9).

APPENDIX B
PROOF OF THEOREM 6

To prove Theorem 6, we first give Lemma 2 [51].

Lemma 2: Let $\mathcal{H}$ be a real Hilbert space endowed with an inner product $\langle\cdot,\cdot\rangle$ and a corresponding norm $\|\cdot\|$ (e.g., the trace norm), and $y \in \partial\|x\|$, where $\partial\|\cdot\|$ denotes the subgradient of the norm. Then, $\|y\|^* = 1$ if $x \neq 0$, and $\|y\|^* \leq 1$ if $x = 0$, where $\|\cdot\|^*$ is the dual norm of $\|\cdot\|$.

Proof of Theorem 6:

Proof: Let $\mathcal{X} = [\![\mathcal{G}; U_1, \ldots, U_N]\!]$, we first need to bound $\|\mathcal{T} - \mathcal{X}\|_F$. By $C_1 = (\|\mathcal{P}_S\mathcal{P}_\Omega(\mathcal{T} - \mathcal{X})\|_F / \|\mathcal{P}_\Omega(\mathcal{T} - \mathcal{X})\|_F)$, then we have

$$\begin{aligned}
&\frac{\|\mathcal{T} - \mathcal{X}\|_F}{\sqrt{I^N}} \\
&\leq \left|\frac{\|\mathcal{T}-\mathcal{X}\|_F}{\sqrt{I^N}} - \frac{\|\mathcal{P}_S\mathcal{P}_\Omega(\mathcal{T} - \mathcal{X})\|_F}{C_1\sqrt{|\Omega|}}\right| + \frac{\|\mathcal{P}_S\mathcal{P}_\Omega(\mathcal{T} - \mathcal{X})\|_F}{C_1\sqrt{|\Omega|}} \\
&= \left|\frac{\|\mathcal{T} - \mathcal{X}\|_F}{\sqrt{I^N}} - \frac{\|\mathcal{P}_\Omega(\mathcal{T} - \mathcal{X})\|_F}{\sqrt{|\Omega|}}\right| + \frac{\|\mathcal{P}_S\mathcal{P}_\Omega(\mathcal{T} - \mathcal{X})\|_F}{C_1\sqrt{|\Omega|}}.
\end{aligned}$$

Let $\varphi(\Omega) = |(1/\sqrt{I^N})\|\mathcal{T} - \mathcal{X}\|_F - (1/\sqrt{|\Omega|})\|\mathcal{P}_\Omega(\mathcal{T} - \mathcal{X})\|_F|$, then we need to bound $\varphi(\Omega)$. Since $n\text{-rank}(\mathcal{X}) \preceq (R, R, \ldots, R)$, $\mathcal{X} \in \mathcal{S}_R$, and by Theorem 5, then with probability greater than $1 - 2\exp(-I)$, there exists a fixed constant C, such that

$$\begin{aligned}
\sup_{\mathcal{X}\in\mathcal{S}_R} \varphi(\Omega) &= \left|\frac{\|\mathcal{X} - \mathcal{T}\|_F}{\sqrt{I^N}} - \frac{\|\mathcal{P}_\Omega(\mathcal{X}) - \mathcal{P}_\Omega(\mathcal{T})\|_F}{\sqrt{|\Omega|}}\right| \\
&\leq C\beta\left(\frac{(R^N + NRI)\log(\sqrt{I^N})}{|\Omega|}\right)^{\frac{1}{4}}. \qquad (31)
\end{aligned}$$

To bound the gap $\|\mathcal{T} - \mathcal{X}\|_F$, we next need to bound $\|\mathcal{P}_S\mathcal{P}_\Omega(\mathcal{T} - \mathcal{X})\|_F$. Since $(\mathcal{G}, U_1, \ldots, U_N)$ is a KKT point of problem (7), the first-order optimal condition for problem (7) with respect to $\mathcal{G}$ is written as follows:

$$\lambda[\![\mathcal{P}_\Omega(\mathcal{T} - \mathcal{X});\ U_1^T, \ldots, U_N^T]\!] \in \sum_{n=1}^N \text{refold}(\partial\|\mathcal{G}_{(n)}\|_*)/N$$

where $[\![\mathcal{P}_\Omega(\mathcal{T} - \mathcal{X});\ U_1^T, \ldots, U_N^T]\!] := \mathcal{P}_\Omega(\mathcal{T} - \mathcal{X}) \times_1 U_1^T \times_2 \cdots \times_N U_N^T$.

In other words, there exist $\{P_n \in \mathbb{R}^{R \times R^{N-1}},\ n = 1, \ldots, N\}$, such that

$$\lambda P_n \in \partial\|\mathcal{G}_{(n)}\|_*/N, \quad n = 1, 2, \ldots, N \qquad (32a)$$

$$[\![\mathcal{P}_\Omega(\mathcal{T} - \mathcal{X});\ U_1^T, \ldots, U_N^T]\!] = \sum_{n=1}^N \text{refold}(P_n). \qquad (32b)$$

Using Lemma 2 and (32a), we obtain

$$\lambda \|P_n\|_2 \leq 1/N$$

where $\|\cdot\|_2$ is the spectral norm and the dual norm of the trace norm. According to $\text{rank}(P_n) \leq R$, we have

$$\|P_n\|_F \leq \sqrt{R}\|P_n\|_2 \leq \frac{\sqrt{R}}{N\lambda}. \tag{33}$$

By (32b) and (33), we obtain

$$\begin{aligned}\|\mathcal{P}_S\mathcal{P}_\Omega(\mathcal{T}-\mathcal{X})\|_F &= \left\|[\![\mathcal{P}_\Omega(\mathcal{T}-\mathcal{X});\, U_1^T,\ldots,U_N^T]\!]\right\|_F \\ &\leq \sum_{n=1}^{N}\|\text{refold}(P_n)\|_F = \sum_{n=1}^{N}\|P_n\|_F \leq \frac{\sqrt{R}}{\lambda}. \end{aligned}\tag{34}$$

By (31) and (34), we have

$$\begin{aligned}\text{RMSE} &= \frac{\|\mathcal{D}-\mathcal{X}\|_F}{\sqrt{I^N}} \leq \frac{\|\mathcal{E}\|_F}{\sqrt{I^N}} + \frac{\|\mathcal{T}-\mathcal{X}\|_F}{\sqrt{I^N}} \\ &\leq \frac{\|\mathcal{E}\|_F}{\sqrt{I^N}} + \varphi(\Omega) + \frac{\|\mathcal{P}_S\mathcal{P}_\Omega(\mathcal{T}-\mathcal{X})\|_F}{C_1\sqrt{|\Omega|}} \\ &\leq \frac{\|\mathcal{E}\|_F}{\sqrt{I^N}} + C\beta\left(\frac{(R^N+NRI)\log(\sqrt{I^N})}{|\Omega|}\right)^{\frac{1}{4}} \\ &\quad + \frac{\|\mathcal{P}_\Omega(\mathcal{E})\|_F}{C_1\sqrt{|\Omega|}}.\end{aligned}$$

This completes the proof. ∎

REFERENCES

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

[2] 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.

[3] D. Tao, X. Li, X. Wu, and S. 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] Y. K. Yılmaz, A. T. Cemgil, and U. Şimşekli, "Generalised coupled tensor factorisation," in *Proc. Adv. Neural Inf. Process. Syst.*, Granada, Spain, Dec. 2011, pp. 2151–2159.

[5] Z. Lai, W. K. Wong, Y. Xu, C. Zhao, and M. Sun, "Sparse alignment for robust tensor learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 10, pp. 1779–1792, Oct. 2014.

[6] 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.*, Montreal, QC, Canada, Dec. 2014, pp. 1763–1771.

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

[8] 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.

[9] L. Yang, Z.-H. 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.

[10] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep./Nov. 2009, pp. 2114–2121.

[11] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imag. Sci.*, vol. 6, no. 3, pp. 1758–1789, 2013.

[12] 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.

[13] J. Sun, S. Papadimitriou, C.-Y. Lin, N. Cao, S. Liu, and W. Qian, "MultiVis: Content-based social network exploration through multi-way visual analysis," in *Proc. SIAM Int. Conf. Data Mining*, Denver, CO, USA, Apr. 2009, pp. 1063–1074.

[14] M. Mørup, "Applications of tensor (multiway array) factorizations and decompositions in data mining," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 1, no. 1, pp. 24–40, 2011.

[15] 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 Mining*, Philadelphia, PA, USA, Apr. 2014, pp. 866–874.

[16] J. Zhong, J. Lu, Y. Liu, and J. Cao, "Synchronization in an array of output-coupled Boolean networks with time delay," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2288–2294, Dec. 2014.

[17] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1324–1342, 2000.

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

[19] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations with missing data," in *Proc. SIAM Int. Conf. Data Mining*, Columbus, OH, USA, Apr. 2010, pp. 701–711.

[20] H. Shan, A. Banerjee, and R. Natarajan, "Probabilistic tensor factorization for tensor completion," Dept. Comput. Sci., Univ. Minnesota, Minneapolis, MN, USA, Tech. Rep. TR 11-026, Oct. 2011.

[21] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Proc. 28th Int. Conf. Mach. Learn.*, Bellevue, WA, USA, Jun. 2011, pp. 809–816.

[22] Y. Liu, F. Shang, L. Jiao, J. Cheng, and H. Cheng, "Trace norm regularized CANDECOMP/PARAFAC decomposition with missing data," *IEEE Trans. Cybern.*, vol. 45, no. 11, pp. 2437–2448, Nov. 2015.

[23] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

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

[25] 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.

[26] M. Filipović and A. Jukić, "Tucker factorization with missing data with application to low-n-rank tensor completion," *Multidimensional Syst. Signal Process.*, vol. 26, no. 3, pp. 677–692, 2015.

[27] M. Fazel, "Matrix rank minimization with applications," Ph.D. dissertation, Dept. Elect. Eng., Stanford Univ., Stanford, CA, USA, Mar. 2002.

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

[29] M. Signoretto, L. De Lathauwer, and J. A. K. Suykens, "Nuclear norms for tensors and their use for convex multilinear estimation," ESAT-SISTA, Katholieke Univ. Leuven, Leuven, Belgium, Tech. Rep. 10-186, 2010.

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

[31] 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.*, Beijing, China, Jun. 2014, pp. 73–81.

[32] B. Romera-Paredes and M. Pontil, "A new convex relaxation for tensor completion," in *Proc. Adv. Neural Inf. Process. Syst.*, Stateline, NV, USA, Dec. 2013, pp. 2967–2975.

[33] R. Tomioka and T. Suzuki, "Convex tensor decomposition via structured Schatten norm regularization," in *Proc. Adv. Neural Inf. Process. Syst.*, Stateline, NV, USA, Dec. 2013, pp. 1331–1339.

[34] R. Tomioka, T. Suzuki, K. Hayashi, and H. Kashima, "Statistical performance of convex tensor decomposition," in *Proc. Adv. Neural Inf. Process. Syst.*, Granada, Spain, Dec. 2011, pp. 972–980.

[35] 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.*, Beijing, China, Aug. 2013, pp. 1670–1676.

[36] E. J. Candès and Y. Plan, "Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements," *IEEE Trans. Inf. Theory*, vol. 57, no. 4, pp. 2342–2359, Apr. 2011.

[37] Y.-X. Wang and H. Xu, "Stability of matrix factorization for collaborative filtering," in *Proc. 29th Int. Conf. Mach. Learn.*, Edinburgh, U.K., Jun. 2012, pp. 417–424.

[38] J. Håstad, "Tensor rank is NP-complete," *J. Algorithms*, vol. 11, no. 4, pp. 644–654, 1990.

[39] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-hard," *J. ACM*, vol. 60, no. 6, 2013, Art. ID 45.

[40] A. Edelman, T. A. Arias, and S. T. Smith, "The geometry of algorithms with orthogonality constraints," *SIAM J. Matrix Anal. Appl.*, vol. 20, no. 2, pp. 303–353, 1998.

[41] F. Shang, Y. Liu, and J. Cheng, "Generalized higher-order tensor decomposition via parallel ADMM," in *Proc. 28th AAAI Conf. Artif. Intell.*, Québec, QC, Canada, Jul. 2014, pp. 1279–1285.

[42] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from a few entries," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2980–2998, Jun. 2010.

[43] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Math. Program. Comput.*, vol. 4, no. 4, pp. 333–361, 2012.

[44] A. Narita, K. Hayashi, R. Tomioka, and H. Kashima, "Tensor factorization using auxiliary information," *Data Mining Knowl. Discovery*, vol. 25, no. 2, pp. 298–324, 2012.

[45] F. Shang, L. C. Jiao, and F. Wang, "Graph dual regularization non-negative matrix factorization for co-clustering," *Pattern Recognit.*, vol. 45, no. 6, pp. 2237–2250, 2012.

[46] Y.-L. Chen, C.-T. Hsu, and H.-Y. M. Liao, "Simultaneous tensor decomposition and completion using factor priors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 577–591, Mar. 2014.

[47] 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.

[48] 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.

[49] J.-F. 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.

[50] Y. Xu, R. Hao, W. Yin, and Z. Su, "Parallel matrix factorization for low-rank tensor completion," *Inverse Problems Imag.*, vol. 9, no. 2, pp. 601–624, 2015.

[51] Z. Lin, M. Chen, L. Wu, and Y. Ma, "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, Nov. 2009.

[52] H. Nick, *Matrix Procrustes Problems*. Manchester, U.K.: Univ. Manchester, 1995.

[53] R. J. Serfling, "Probability inequalities for the sum in sampling without replacement," *Ann. Statist.*, vol. 2, no. 1, pp. 39–48, 1974.

[54] B. W. Bader and T. G. Kolda. (2012). *MATLAB Tensor Toolbox Version 2.5*. [Online]. Available: http://www.sandia.gov/~tgkolda/TensorToolbox/

[55] K.-C. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.

[56] L. Tang, X. Wang, and H. Liu, "Uncovering groups via heterogeneous interaction analysis," in *Proc. 9th IEEE Int. Conf. Data Mining*, Miami, FL, USA, Dec. 2009, pp. 503–512.

[57] 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.

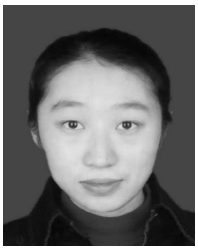
[58] S.-J. Wang, J. Yang, M.-F. Sun, X.-J. Peng, M.-M. Sun, and C.-G. Zhou, "Sparse tensor discriminant color space for face verification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 6, pp. 876–888, Jun. 2012.

[59] J. Xu, G. Yang, Y. Yin, H. Man, and H. He, "Sparse-representation-based classification with structure-preserving dimension reduction," *Cognit. Comput.*, vol. 6, no. 3, pp. 608–621, 2014.

[60] A. Krishnamurthy and A. Singh, "Low-rank matrix and tensor completion via adaptive sampling," in *Proc. Adv. Neural Inf. Process. Syst.*, Stateline, NV, USA, Dec. 2013, pp. 836–844.

[61] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[62] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix tri-factorizations for clustering," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Philadelphia, PA, USA, Aug. 2006, pp. 126–135.



Yuanyuan Liu received the Ph.D. degree in pattern recognition and intelligent systems 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.

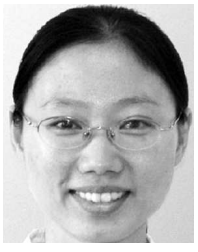
Fanhua Shang (M'14) received the Ph.D. degree in circuits and systems from Xidian University, Xi'an, China, in 2012.

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

Wei Fan received the Ph.D. degree in computer science from Columbia University, New York, NY, USA, in 2001.

He was with the IBM T. J. Watson Research Center, Hawthorne, NY, USA, from 2000 to 2012. He was with the Noah's Ark Laboratory, Huawei Technologies, Hong Kong, from 2013 to 2014. He is currently the Director and Deputy Head of the Baidu Big Data Research Laboratory, Sunnyvale, CA, USA. His current research interests include machine learning, data mining, database, and security.

James Cheng is currently 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 currently 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.