An Iterative Reweighted Method for Tucker Decomposition of Incomplete Tensors

Linxiao Yang, Jun Fang, *Member, IEEE*, Hongbin Li, *Senior Member, IEEE*, and Bing Zeng, *Fellow, IEEE*

***Abstract*—We consider the problem of low-rank decomposition of incomplete tensors. Since many real-world data lie on an intrinsically low-dimensional subspace, tensor low-rank decomposition with missing entries has applications in many data analysis problems such as recommender systems and image inpainting. In this paper, we focus on Tucker decomposition which represents an Nth-order tensor in terms of N factor matrices and a core tensor via multilinear operations. To exploit the underlying multilinear low-rank structure in high-dimensional datasets, we propose a group-based log-sum penalty functional to place structural sparsity over the core tensor, which leads to a compact representation with smallest core tensor. The proposed method is developed by iteratively minimizing a surrogate function that majorizes the original objective function. This iterative optimization leads to an iteratively reweighted least squares algorithm. In addition, to reduce the computational complexity, an over-relaxed monotone fast iterative shrinkage-thresholding technique is adapted and embedded in the iterative reweighted process. The proposed method is able to determine the model complexity (i.e., multilinear rank) in an automatic way. Simulation results show that the proposed algorithm offers competitive performance compared with other existing algorithms.**

***Index Terms*—Tucker decomposition, low rank tensor decomposition, tensor completion, iterative reweighted method.**

I. Introduction

MULTI-DIMENSIONAL data arise in a variety of applications, such as recommender systems [1]–[3], multirelational networks [4], [5], and brain-computer imaging [6], [7]. Tensors (i.e., multiway arrays) provide an effective representation of such data. Tensor decomposition based on low rank approximation is a powerful technique to extract useful information from multiway data as many real-world multiway data are lying on a low dimensional subspace. Compared with matrix factorization, tensor decomposition can capture the intrinsic multi-dimensional structure of the multiway data, which has led to a substantial performance improvement for harmonic retrieval [8], [9], regression/classification [10]–[12], and data completion [3], [13], etc. Tucker decomposition [14] and CANDECOMP/PARAFAC (CP) decomposition [15] are two widely used low-rank tensor decompositions. Specifically, CP decomposes a tensor as a sum of rank-one tensors, whereas Tucker is a more general decomposition which involves multilinear operations between a number of factor matrices and a core tensor. CP decomposition can be viewed as a special case of Tucker decomposition with a super-diagonal core tensor. It is generally believed that Tucker decomposition has a better generalization ability than CP decomposition for different types of data [16]. In many applications, only partial observations of the tensor may be available. It is therefore important to develop efficient tensor decomposition methods for incomplete, sparsely observed data where a significant fraction of entries is missing.

Low-rank decomposition of incomplete tensors has attracted a lot of attention over the past few years and a number of algorithms [17]–[29] were proposed via either optimization techniques or probabilistic model learning. For both CP and Tucker decompositions, the most challenging task is to determine the model complexity (i.e., the rank of the tensor) in the presence of missing entries and noise. It has been shown that determining the CP rank, i.e., the minimum number of rank-one terms in CP decomposition, is an NP-hard problem even for a completely observed tensor [30]. Unfortunately, many existing methods require that the rank of the decomposition is specified *a priori*. To address this issue, a Bayesian method was proposed in [13] for CP decomposition, where a shrinkage prior called as the multiplicative gamma process (MGP) was employed to adaptively learn a concise representation of the tensor. In [26], a sparsity-inducing Gaussian inverse-Gamma prior was placed over multiple latent factors to achieve automatic rank determination. Besides the above Bayesian methods, an optimization-based CP decomposition method was proposed in [18], [19], where the Frobenius-norm of the factor matrices is used as the rank regularization to determine an appropriate number of component tensors.

In addition to the CP rank, another notion of tensor rank is multilinear rank [31], which is defined as the tuple of the ranks of the mode-n unfoldings of the tensor. Multilinear rank is closely related to the Tucker decomposition since the multilinear rank is equivalent to the dimension of the smallest achievable core tensor in Tucker decomposition [32]. To search for a low multilinear rank representation, a tensor nuclear norm, defined as a (weighted) summation of nuclear norms

Manuscript received November 11, 2015; revised March 21, 2016 and May 11, 2016; accepted May 13, 2016. Date of publication May 24, 2016; date of current version July 27, 2016. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Masahiro Yukawa. This work was supported in part by the National Science Foundation of China under Grants 61428103, 61522104, the National Science Foundation under Grant ECCS-1408182, and the Air Force Office of Scientific Research under award number FA9550-16-1-0243.

L. Yang and J. Fang are with the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: yanglinxiao0929@hotmail.com; JunFang@uestc.edu.cn).

H. Li is with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030 USA (e-mail: Hongbin.Li@stevens.edu).

B. Zeng is with the School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China (e-mail: eezeng@uestc.edu.cn).

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/TSP.2016.2572047

of mode-n unfoldings, was introduced to approximate the multilinear rank. Tensor completion and decomposition can then be accomplished by minimizing the tensor nuclear norm. Specifically, an alternating direction method of multipliers (ADMM) was developed in [17], [23] to minimize the tensor nuclear norm with missing data, and encouraging results were reported on visual data. The success of [17] has inspired a number of subsequent works [21], [22], [24], [25], [33], [34] for tensor completion and decomposition based on tensor nuclear norm minimization. Nevertheless, the tensor nuclear norm, albeit effective, is not necessarily the tightest convex envelope of the multilinear rank [21]. Also, the nuclear norm-based methods are sensitive to outliers and the performance is also affected by an increasing number of missing data [35].

In this paper, to automatically achieve a concise Tucker representation, we introduce a notion referred to as the order-$(N-1)$ sub-tensor and propose a group log-sum penalty functional to encourage structural sparsity of the core tensor. Specifically, in the log-sum penalty function, elements in every sub-tensor of the core tensor along each mode are grouped together. Minimizing the group log-sum penalty function thus leads to a structured sparse core tensor with only a few nonzero order-$(N-1)$ sub-tensors along each mode. By removing the zero order-$(N-1)$ sub-tensors, the core tensor shrinks and a compact Tucker decomposition can be obtained. Note that the log-sum function which behaves like the ℓ_0-norm is more sparsity-encouraging than the nuclear norm that is ℓ_1-norm applied to the singular values of a matrix. Thus we expect the group log-sum minimization is more effective than the tensor nuclear norm-minimization in finding a concise representation of the tensor, particularly when a significant number of entries of the tensor are missing. By resorting to a majorization-minimization approach, we develop an iterative reweighted method via iteratively decreasing a surrogate function that majorizes the original log-sum penalty function. The proposed method can determine the model complexity (i.e., multilinear rank) in an automatic way. Also, the over-relaxed monotone fast iterative shrinkage-thresholding technique [36] is adapted and embedded in the iterative reweighted process, which achieves a substantial reduction in computational complexity.

The remainder of this paper is organized as follows. Section II provides notations and basics on tensors. The problem of Tucker decomposition with incomplete entries is formulated as an unconstrained optimization problem in Section III. An iterative reweighted method is developed in Section IV for Tucker decomposition of incomplete tensors. In Section V, the over-relaxed monotone fast iterative shrinkage-thresholding technique is adapted and integrated with the proposed iterative reweighted method, which results in a significant computational complexity reduction. Simulation results are provided in Section VI, followed by concluding remarks in Section VII.

II. Notations and Basics on Tensors

We first provide a brief review on tensor decompositions. A tensor is the generalization of a matrix to higher dimensions, also known as ways or modes. Vectors and matrices can be viewed as special cases of tensors with one and two modes, respectively. Throughout this paper, we use symbols $\otimes$, $\circ$ and $*$ to denote the Kronecker, outer and Hadamard product, respectively.

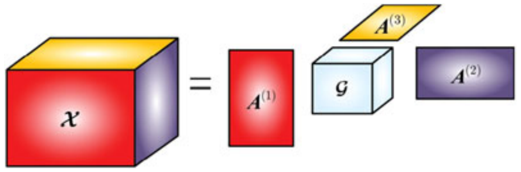


Fig. 1. The Tucker decomposition of a three-order tensor.

Let $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ denote an Nth order tensor with its $(i_1, \ldots, i_N)$th entry denoted by $\mathcal{X}_{i_1 \cdots i_N}$. Here the order N of a tensor is the number of dimensions. Fibers are the higher-order analogue of matrix rows and columns. The mode-n fibers of $\boldsymbol{\mathcal{X}}$ are I_n-dimensional vectors obtained by fixing every index but i_n (see references e.g. [16], [32], [37]). Slices are two-dimensional sections of a tensor, defined by fixing all but two indices. Unfolding or matricization is an operation that turns a tensor into a matrix. Specifically, the mode-n unfolding of a tensor $\boldsymbol{\mathcal{X}}$, denoted as $\boldsymbol{X}_{(n)}$, arranges the mode-n fibers to be the columns of the resulting matrix [32]. For notational convenience, we also use the notation $\text{unfold}_n(\boldsymbol{\mathcal{X}})$ to denote the unfolding operation along the n-th mode. The n-mode product of $\boldsymbol{\mathcal{X}}$ with a matrix $\boldsymbol{A} \in \mathbb{R}^{J \times I_n}$ is denoted by $\boldsymbol{\mathcal{X}} \times_n \boldsymbol{A}$ and is of size $I_1 \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N$, with each mode-$n$ fiber multiplied by the matrix $\boldsymbol{A}$ [32], i.e.,

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_n \boldsymbol{A} \Leftrightarrow \boldsymbol{Y}_{(n)} = \boldsymbol{A}\boldsymbol{X}_{(n)}. \tag{1}$$

The CP decomposition decomposes a tensor into a sum of rank-one component tensors [32], i.e.,

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \boldsymbol{a}_r^{(1)} \circ \boldsymbol{a}_r^{(2)} \circ \cdots \circ \boldsymbol{a}_r^{(N)}, \tag{2}$$

where $\boldsymbol{a}_r^{(n)} \in \mathbb{R}^{I_n}$ and R is referred to as the rank of the tensor. Elementwise, we have

$$\mathcal{X}_{i_1 i_2 \cdots i_N} = \sum_{r=1}^{R} \lambda_r a_{i_1 r}^{(1)} a_{i_2 r}^{(2)} \cdots a_{i_N r}^{(N)}. \tag{3}$$

The Tucker decomposition can be considered as a high order principle component analysis. It decomposes a tensor into a core tensor multiplied by a factor matrix along each mode. The Tucker decomposition of an N-th order tensor $\boldsymbol{\mathcal{X}}$ can be written as [32]

$$\begin{aligned}\boldsymbol{\mathcal{X}} &= \boldsymbol{\mathcal{G}} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \cdots \times_N \boldsymbol{A}^{(N)} \\ &= \sum_{r_1=1}^{R_1} \sum_{r_2=2}^{R_2} \cdots \sum_{r_N=1}^{R_N} \mathcal{G}_{r_1 r_2 \cdots r_N} \boldsymbol{a}_{r_1}^{(1)} \circ \boldsymbol{a}_{r_2}^{(2)} \circ \cdots \circ \boldsymbol{a}_{r_N}^{(N)},\end{aligned} \tag{4}$$

where $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ is the core tensor, and $\boldsymbol{A}^{(n)} \triangleq [\boldsymbol{a}_1^{(n)} \ldots \boldsymbol{a}_{R_n}^{(n)}] \in \mathbb{R}^{I_n \times R_n}$ denotes the factor matrix along the n-th mode (see Fig. 1).

The inner product of two tensors with the same size is defined as [32]

$$\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle = \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_N=1}^{I_N} x_{i_1 i_2 \ldots i_N} y_{i_1 i_2 \ldots i_N}. \tag{5}$$

The Frobenius norm of a tensor $\boldsymbol{\mathcal{X}}$ is square root of the inner product with itself, i.e.,

$$\| \boldsymbol{\mathcal{X}} \|_F = \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}} \rangle^{\frac{1}{2}}. \tag{6}$$

Also, for notational convenience, the sequential Kronecker product of a set of matrices in a reversed order is defined and denoted by

$$\bigotimes_n \boldsymbol{A}^{(n)} = \boldsymbol{A}^{(N)} \otimes \boldsymbol{A}^{(N-1)} \otimes \cdots \otimes \boldsymbol{A}^{(1)},$$

$$\bigotimes_{n \neq k} \boldsymbol{A}^{(n)} = \boldsymbol{A}^{(N)} \otimes \cdots \otimes \boldsymbol{A}^{(k+1)} \otimes \boldsymbol{A}^{(k-1)} \otimes \cdots \otimes \boldsymbol{A}^{(1)}.$$

An Nth order tensor $\boldsymbol{\mathcal{X}}$ multiplied by factor matrices $\{\boldsymbol{A}^{(k)}\}_{k=1}^{N}$ along each mode is denoted by

$$\boldsymbol{\mathcal{X}} \prod_{n=1}^{N} \times_n \boldsymbol{A}^{(n)} = \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \cdots \times_N \boldsymbol{A}^{(N)},$$

while the tensor $\boldsymbol{\mathcal{X}}$ multiplied by the factor matrices along every mode except the k-th mode is denoted as

$$\boldsymbol{\mathcal{X}} \prod_{n \neq k} \times_n \boldsymbol{A}^{(n)} = \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{A}^{(1)} \cdots \times_{k-1} \boldsymbol{A}^{(k-1)} \times_{k+1} \boldsymbol{A}^{(k+1)} \cdots \times_N \boldsymbol{A}^{(N)}.$$

With these notations, vectorization and unfolding of a tensor which admits a Tucker decomposition (4) can be expressed as [16]

$$\mathrm{vec}(\boldsymbol{\mathcal{X}}) = \Big(\bigotimes_n \boldsymbol{A}^{(n)} \Big) \mathrm{vec}(\boldsymbol{\mathcal{G}}), \tag{7}$$

$$\boldsymbol{X}_{(n)} = \boldsymbol{A}^{(n)} \boldsymbol{G}_{(n)} \Big(\bigotimes_{k \neq n} \boldsymbol{A}^{(k)} \Big)^T. \tag{8}$$

III. Problem Formulation

Let $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be an incomplete Nth order tensor, with its entry $\mathcal{Y}_{i_1 i_2 \ldots i_N}$ observed if $\mathcal{O}_{i_1 i_2 \ldots i_N} = 1$, where $\boldsymbol{\mathcal{O}} \in \{0, 1\}^{I_1 \times I_2 \times \cdots \times I_N}$ is a binary tensor of the same size as $\boldsymbol{\mathcal{Y}}$ and indicates which entries of $\boldsymbol{\mathcal{Y}}$ are missing or observed. Given the observed data, our objective is to find a Tucker decomposition which has a minimum model complexity and meanwhile fits the observed data, or to be more precise, seek a Tucker representation such that the data can be represented by a smallest core tensor. Since the dimension of the smallest achievable core tensor is unknown *a priori*, we need to develop a method that can achieve automatic model determination. To this objective, we first introduce a new notion called as order-$(N-1)$ sub-tensor.

Definition: Order-$(N-1)$ sub-tensor is defined as a new tensor obtained by fixing only one index of the original tensor. Let $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ be an Nth order tensor. The ith $(1 \leq i \leq I_n)$ sub-tensor along the nth mode of $\boldsymbol{\mathcal{Z}}$, denoted as $\boldsymbol{\mathcal{Z}}_{(n,i)}$, is an $(N-1)$th order tensor of size $I_1 \times I_2 \times \cdots I_{n-1} \times I_{n+1} \cdots \times I_N$, and its $(j_1, \ldots, j_{n-1}, j_{n+1}, \ldots, j_N)$th entry is given by $\mathcal{Z}_{j_1, \ldots, j_{n-1}, i, j_{n+1}, \ldots, j_N}$. For tensors with three modes, i.e., $N = 3$, order-$(N-1)$ sub-tensors reduce to slices, although order-$(N-1)$ sub-tensors are generally different from slices.

Clearly, $\boldsymbol{\mathcal{Z}}$ consists of I_n order-$(N-1)$ sub-tensors along its nth mode. If some order-$(N-1)$ sub-tensors along the nth mode become zero, then the dimension of $\boldsymbol{\mathcal{Z}}$ along the nth mode is reduced accordingly. Suppose the data tensor $\boldsymbol{\mathcal{Y}}$ has a Tucker decomposition

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \prod_{n=1}^{N} \times_n \boldsymbol{A}^{(n)}. \tag{9}$$

Unfolding $\boldsymbol{\mathcal{Y}}$ along the nth mode, we have

$$\begin{aligned} \boldsymbol{Y}_{(n)} &= \boldsymbol{A}^{(n)} \boldsymbol{X}_{(n)} \Big(\bigotimes_{k \neq n} \boldsymbol{A}^{(k)} \Big)^T \\ &= \sum_{i=1}^{I_n} \boldsymbol{a}_i^{(n)} \boldsymbol{x}_{i,\cdot}^{(n)} \Big(\bigotimes_{k \neq n} \boldsymbol{A}^{(k)} \Big)^T, \end{aligned} \tag{10}$$

where $\boldsymbol{a}_i^{(n)}$ is the ith column of $\boldsymbol{A}^{(n)}$ and $\boldsymbol{x}_{i,\cdot}^{(n)}$ denotes the ith row of $\boldsymbol{X}_{(n)}$. Clearly, $\boldsymbol{x}_{i,\cdot}^{(n)}$ is the vectorization of the ith sub-tensor along the nth mode of $\boldsymbol{\mathcal{X}}$. If $\boldsymbol{x}_{i,\cdot}^{(n)}$ is a zero vector, i.e., the corresponding order-$(N-1)$ sub-tensor is a zero tensor, both $\boldsymbol{a}_i^{(n)}$ and $\boldsymbol{x}_{i,\cdot}^{(n)}$ have no contribution to $\boldsymbol{\mathcal{Y}}$ and can thus be removed. Inspired by this insight, sparsity can be enforced over each sub-tensor (along each mode) of the core tensor such that the observed data can be represented by a structural sparsest core tensor with only a few nonzero sub-tensors over all modes. By removing those zero sub-tensors along each mode (the associated columns of the factor matrices are disabled and can be removed as well), the core tensor shrinks to a smaller one and a compact Tucker decomposition can be obtained. The problem can be formulated as

$$\begin{aligned} \min_{\boldsymbol{\mathcal{X}}, \{\boldsymbol{A}^{(n)}\}} \quad & \sum_{n=1}^{N} \| \boldsymbol{z}_n \|_0 \\ \text{s.t.} \quad & \left\| \boldsymbol{\mathcal{O}} * \left(\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}} \prod_{n=1}^{N} \times_n \boldsymbol{A}^{(n)} \right) \right\|_F^2 \leq \varepsilon, \end{aligned} \tag{11}$$

where ε is an error tolerance parameter related to noise statistics, and $\boldsymbol{z}_n$ is an I_n-dimensional vector with its ith entry given by

$$z_{n,i} \triangleq \| \boldsymbol{\mathcal{X}}_{(n,i)} \|_F. \tag{12}$$

It should be noted that since there is usually no knowledge about the size of smallest core tensor, the dimensions of the core tensor are predefined to be the same as the original tensor, i.e., $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$. The term $\| \boldsymbol{z}_n \|_0$ specifies the number of nonzero sub-tensors along the nth mode of tensor $\boldsymbol{\mathcal{X}}$. Since the number of nonzero sub-tensors along the nth mode is equivalent to the dimension of mode-n of the core tensor, the above optimization yields a smallest (in terms of sum of dimensions of all modes) core tensor. The optimization (11), however, is an NP-hard problem. Thus, alternative sparsity-promoting functions which are more computationally efficient in finding the structural sparse core tensor are desirable. In this paper, we consider the use of the log-sum sparsity-encouraging functional. Log-sum penalty function has been extensively used for sparse signal recovery and was shown to be more sparsity-encouraging than the ℓ_1-norm [38]–[41]. Replacing the ℓ_0-norm in (11) with the log-sum functional leads to

$$\min_{\boldsymbol{\mathcal{X}},\{\boldsymbol{A}^{(n)}\}} \quad \sum_{n=1}^{N}\sum_{i=1}^{I_n}\log\left(\left\|\boldsymbol{\mathcal{X}}_{(n,i)}\right\|_F^2+\epsilon\right)$$
$$\text{s.t.} \quad \left\|\boldsymbol{\mathcal{O}}*\left(\boldsymbol{\mathcal{Y}}-\boldsymbol{\mathcal{X}}\prod_{n=1}^{N}\times_n\boldsymbol{A}^{(n)}\right)\right\|_F^2\leq\varepsilon, \tag{13}$$

where ϵ is a small positive parameter to ensure the logarithmic function is well-defined. Note that in our formulation, coefficients are grouped according to sub-tensors and different sub-tensors may have overlapping entries. This is similar to the overlapping group LASSO method [42] with entries shared among multiple groups, while different from the conventional group-LASSO method [43] in which entries are grouped into a number of non-overlapping subsets. To make the problem more tractable, we convert the constrained optimization (13) into the following unconstrained optimization problem by resorting to the Tikhonov regularization method

$$\min_{\boldsymbol{\mathcal{X}},\{\boldsymbol{A}^{(n)}\}} L\left(\boldsymbol{\mathcal{X}},\left\{\boldsymbol{A}^{(n)}\right\}\right)=\sum_{n=1}^{N}\sum_{i=1}^{I_n}\log\left(\left\|\boldsymbol{\mathcal{X}}_{(n,i)}\right\|_F^2+\epsilon\right)$$
$$+\lambda_1\left\|\boldsymbol{\mathcal{O}}*\left(\boldsymbol{\mathcal{Y}}-\boldsymbol{\mathcal{X}}\prod_{n=1}^{N}\times_n\boldsymbol{A}^{(n)}\right)\right\|_F^2+\lambda_2\sum_{n=1}^{N}\left\|\boldsymbol{A}^{(n)}\right\|_F^2, \tag{14}$$

where λ_1 is a parameter controlling the tradeoff between the sparsity of the core tensor and the fitting error, the last term is a Frobenius norm imposed on the factor matrices $\{\boldsymbol{A}^{(n)}\}$ in order to avoid a trivial solution $\{\boldsymbol{\mathcal{X}}\to 0,\{\boldsymbol{A}^{(n)}\}\to\infty\}$, and λ_2 is the associated regularization parameter. The choices of λ_1 and λ_2 will be discussed later in our paper. Note that, due to the nonconvexity of the objective function, the optimization problems (13) and (14) may not be strictly equivalent even if the last term is excluded.

The above optimization (14) can be viewed as searching for a low multilinear rank representation of the observed data. Multilinear rank, also referred to as n-rank, of an N-order tensor $\boldsymbol{\mathcal{X}}$ is defined as the tuple of the ranks of the mode-n unfoldings, i.e.,

$$n\text{-rank}\triangleq\left\{\text{rank}(\boldsymbol{X}_{(1)}),\text{rank}\left(\boldsymbol{X}_{(2)}\right),\ldots,\text{rank}\left(\boldsymbol{X}_{(N)}\right)\right\}. \tag{15}$$

It can be shown that n-rank is equivalent to the dimensions of the smallest achievable core tensor in Tucker decomposition [32]. Therefore the optimization (14) can also be used for recovery of incomplete low n-rank tensors. Existing methods (e.g., [23], [33]) for low n-rank completion employ a tensor nuclear-norm, defined as a (weighted) summation of nuclear-norms of mode-n unfoldings, to approximate the n-rank and achieve a low n-rank representation. Our formulation, instead, uses the logarithm of Frobenius-norms of order-$(N-1)$ sub-tensors to promote a low n-rank representation.

IV. Proposed Iterative Reweighted Method

We resort to a bounded optimization approach, also known as the majorization-minimization (MM) approach [44], to solve the optimization (14). The idea of the MM approach is to iteratively minimize a simple surrogate function majorizing the given objective function. It can be shown that through iteratively minimizing the surrogate function, the iterative process yields a non-increasing objective function value and eventually converges to a stationary point of the original objective function.

To obtain an appropriate surrogate function for (14), we first find a suitable surrogate function for the log-sum function. The following inequality is commonly used to derive a surrogate function for the log-sum term [38], [40]

$$\log\left(\left\|\boldsymbol{\mathcal{X}}_{(n,i)}\right\|_F^2+\epsilon\right)\leq\frac{\left\|\boldsymbol{\mathcal{X}}_{(n,i)}\right\|_F^2+\epsilon}{\left\|\boldsymbol{\mathcal{X}}_{(n,i)}^{[t]}\right\|_F^2+\epsilon}$$
$$+\log\left(\left\|\boldsymbol{\mathcal{X}}_{(n,i)}^{[t]}\right\|_F^2+\epsilon\right)-1. \tag{16}$$

The above inequality holds valid for any $\boldsymbol{\mathcal{X}}^{[t]}$ and the equality is attained when $\boldsymbol{\mathcal{X}}=\boldsymbol{\mathcal{X}}^{[t]}$. To facilitate our subsequent presentation, we use the supscript $(\,\cdot\,)^{[t]}$ to denote an estimate of $(\,\cdot\,)$ obtained from the tth iteration of the iterative process. From (16), we know that $f(\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{X}}^{[t]})$ defined in (17) is a surrogate function majorizing the log-sum functional, i.e.,

$$\sum_{n=1}^{N}\sum_{i=1}^{I_n}\log\left(\left\|\boldsymbol{\mathcal{X}}_{(n,i)}\right\|_F^2+\epsilon\right)\leq f(\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{X}}^{[t]}),$$

in which

$$f(\boldsymbol{\mathcal{X}}|\boldsymbol{\mathcal{X}}^{[t]})\triangleq\langle\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{D}}^{[t]}*\boldsymbol{\mathcal{X}}\rangle$$
$$+\sum_{n=1}^{N}\sum_{i=1}^{I_n}\log\left(\left\|\boldsymbol{\mathcal{X}}_{(n,i)}^{[t]}\right\|_F^2+\epsilon\right)-\sum_{n=1}^{N}I_n, \tag{17}$$

where $\boldsymbol{\mathcal{D}}^{[t]}$ is a tensor of the same size of $\boldsymbol{\mathcal{X}}$, with its $(i_1,i_2,\ldots,i_N)$th element given by

$$\mathcal{D}_{i_1i_2\ldots i_N}^{[t]}=\sum_{n=1}^{N}\left(\left\|\boldsymbol{\mathcal{X}}_{(n,i_n)}^{[t]}\right\|_F^2+\epsilon\right)^{-1}. \tag{18}$$

Thus we can readily verify that an appropriate surrogate function majorizing the objective function (14) is given as

$$Q(\boldsymbol{\mathcal{X}},\{\boldsymbol{A}^{(n)}\}|\boldsymbol{\mathcal{X}}^{[t]})=\lambda_1\left\|\boldsymbol{\mathcal{O}}*\left(\boldsymbol{\mathcal{Y}}-\boldsymbol{\mathcal{X}}\prod_{n=1}^{N}\times_n\boldsymbol{A}^{(n)}\right)\right\|_F^2$$
$$+\langle\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{D}}^{[t]}*\boldsymbol{\mathcal{X}}\rangle+\lambda_2\sum_{n=1}^{N}\left\|\boldsymbol{A}^{(n)}\right\|_F^2+c, \tag{19}$$

where c is a constant. That is,

$$L\left(\boldsymbol{\mathcal{X}},\{\boldsymbol{A}^{(n)}\}\right)\leq Q(\boldsymbol{\mathcal{X}},\{\boldsymbol{A}^{(n)}\}|\boldsymbol{\mathcal{X}}^{[t]}) \tag{20}$$

with the equality attained when $\boldsymbol{\mathcal{X}}=\boldsymbol{\mathcal{X}}^{[t]}$.

Solving (14) now reduces to minimizing the surrogate function (19) iteratively. Minimization of the surrogate function, however, is still difficult since it involves a joint search over the core tensor $\boldsymbol{\mathcal{X}}$ and the associated factor matrices $\{\boldsymbol{A}^{(n)}\}_{n=1}^{N}$.

Nevertheless, we will show that through iteratively decreasing (not necessarily minimizing) the surrogate function, the iterative process also results in a non-increasing objective function value and eventually converges to a stationary point of $L(\boldsymbol{\mathcal{X}}, \{\boldsymbol{A}^{(n)}\})$. Decreasing the surrogate function is much easier since we only need to alternatively minimize the surrogate function (19) with respect to each variable while keeping other variables fixed. Such an alternating minimization strategy was also used in other tensor decomposition works (e.g. [7], [45], [46]). Details of this alternating procedure are provided below.

First, we minimize the surrogate function (19) with respect to the core tensor $\boldsymbol{\mathcal{X}}$, given the factor matrices $\{\boldsymbol{A}^{(n)}\}$ fixed. The problem reduces to

$$\min_{\boldsymbol{\mathcal{X}}} \lambda_1 \left\| \boldsymbol{\mathcal{O}} * \left(\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}} \prod_{n=1}^{N} \times_n \boldsymbol{A}^{(n)} \right) \right\|_F^2 + \langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{D}}^{[t]} * \boldsymbol{\mathcal{X}} \rangle. \tag{21}$$

Let $\boldsymbol{x} \triangleq \text{vec}(\boldsymbol{\mathcal{X}})$. The above optimization can be expressed as

$$\min_{\boldsymbol{x}} \lambda_1 \left\| \boldsymbol{\Sigma} \Big(\boldsymbol{y} - \Big(\bigotimes_n \boldsymbol{A}^{(n)} \Big) \boldsymbol{x} \Big) \right\|_2^2 + \boldsymbol{x}^T \boldsymbol{D}^{[t]} \boldsymbol{x}, \tag{22}$$

where $\boldsymbol{\Sigma} \triangleq \text{diag}(\text{vec}(\boldsymbol{\mathcal{O}}))$ and $\boldsymbol{D}^{[t]} \triangleq \text{diag}(\text{vec}(\boldsymbol{\mathcal{D}}^{[t]}))$. For notational simplicity, define

$$\boldsymbol{H} \triangleq \Big(\bigotimes_n \boldsymbol{A}^{(n)} \Big). \tag{23}$$

The optimal solution to (22) can be easily obtained as

$$\boldsymbol{x} = \Big(\boldsymbol{H}^T \boldsymbol{\Sigma} \boldsymbol{H} + \lambda_1^{-1} \boldsymbol{D}^{[t]} \Big)^{-1} \boldsymbol{H}^T \boldsymbol{\Sigma} \boldsymbol{y}. \tag{24}$$

Next, we discuss minimizing the surrogate function (19) with respect to the factor matrix $\boldsymbol{A}^{(n)}$, given that the core tensor $\boldsymbol{\mathcal{X}}$ and the rest of factor matrices $\{\boldsymbol{A}^{(k)}\}_{k \neq n}$ fixed. Ignoring terms independent of $\boldsymbol{A}^{(n)}$ and unfolding the tensor along the nth mode, we arrive at

$$\begin{aligned} \min_{\boldsymbol{A}^{(n)}} \quad & \lambda_1 \left\| \boldsymbol{O}_{(n)} * \left(\boldsymbol{Y}_{(n)} - \boldsymbol{A}^{(n)} \boldsymbol{X}_{(n)} \Big(\bigotimes_{k \neq n} \boldsymbol{A}^{(k)} \Big)^T \right) \right\|_F^2 \\ & + \lambda_2 \left\| \boldsymbol{A}^{(n)} \right\|_F^2 . \end{aligned} \tag{25}$$

Clearly, the optimization can be decomposed into a set of independent tasks, with each task optimizing each row of $\boldsymbol{A}^{(n)}$. Specifically, let $\boldsymbol{y}_{i,\cdot}^{(n)}$ denote the ith row of $\boldsymbol{Y}_{(n)}$, $\boldsymbol{a}_{i,\cdot}^{(n)}$ denote the ith row of $\boldsymbol{A}^{(n)}$, and $\boldsymbol{\Sigma}_i^n \triangleq \text{diag}(\boldsymbol{o}_{i,\cdot}^{(n)})$, with $\boldsymbol{o}_{i,\cdot}^{(n)}$ being the ith row of $\boldsymbol{O}_{(n)}$. The optimization of each row of $\boldsymbol{A}^{(n)}$ can be written as

$$\begin{aligned} \min_{\boldsymbol{a}_{i,\cdot}^{(n)}} \quad & \lambda_1 \left\| \boldsymbol{\Sigma}_i^n \left(\boldsymbol{y}_{i,\cdot}^{(n)} - \boldsymbol{a}_{i,\cdot}^{(n)} \boldsymbol{X}_{(n)} \Big(\bigotimes_{k \neq n} \boldsymbol{A}^{(k)} \Big)^T \right) \right\|_2^2 \\ & + \lambda_2 \left\| \boldsymbol{a}_{i,\cdot}^{(n)} \right\|_2^2 , \end{aligned} \tag{26}$$

whose optimal solution can be readily given as

$$\boldsymbol{a}_{i,\cdot}^{(n)} = \lambda_1 \boldsymbol{y}_{i,\cdot}^{(n)} \boldsymbol{\Sigma}_i^n \boldsymbol{\Phi} \big(\lambda_1 \boldsymbol{\Phi}^T \boldsymbol{\Sigma}_i^n \boldsymbol{\Phi} + \lambda_2 \boldsymbol{I} \big)^{-1}, \tag{27}$$

in which

$$\boldsymbol{\Phi} \triangleq \Big(\bigotimes_{k \neq n} \boldsymbol{A}^{(k)} \Big) \boldsymbol{X}_{(n)}^T .$$

Note that $\boldsymbol{\Phi}$ can be more efficiently calculated from $\text{unfold}_n(\boldsymbol{\mathcal{X}} \prod_{k \neq n} \boldsymbol{A}^{(k)})$.

Thus far we have shown how to minimize the surrogate function (19) with respect to each variable while keeping other variables fixed. Given a current estimate of the core tensor and the associated factor matrices $\{\boldsymbol{\mathcal{X}}^{[t]}, \{(\boldsymbol{A}^{(n)})^{[t]}\}_{n=1}^{N}\}$, this alternating procedure is guaranteed to find a new estimate $\{\boldsymbol{\mathcal{X}}^{[t+1]}, \{(\boldsymbol{A}^{(n)})^{[t+1]}\}_{n=1}^{N}\}$ such that

$$Q(\boldsymbol{\mathcal{X}}^{[t+1]}, \{(\boldsymbol{A}^{(n)})^{[t+1]}\} | \boldsymbol{\mathcal{X}}^{[t]}) \leq Q(\boldsymbol{\mathcal{X}}^{[t]}, \{(\boldsymbol{A}^{(n)})^{[t]}\} | \boldsymbol{\mathcal{X}}^{[t]}). \tag{28}$$

In the following, we further show that the new estimate $\{\boldsymbol{\mathcal{X}}^{[t+1]}, \{(\boldsymbol{A}^{(n)})^{[t+1]}\}_{n=1}^{N}\}$ results in a non-increasing objective function value

$$\begin{aligned} & L\left(\boldsymbol{\mathcal{X}}^{[t+1]}, \left\{(\boldsymbol{A}^{(n)})^{[t+1]}\right\}\right) \\ = \; & L\left(\boldsymbol{\mathcal{X}}^{[t+1]}, \left\{(\boldsymbol{A}^{(n)})^{[t+1]}\right\}\right) \\ & - Q\left(\boldsymbol{\mathcal{X}}^{[t+1]}, \left\{(\boldsymbol{A}^{(n)})^{[t+1]}\right\} | \boldsymbol{\mathcal{X}}^{[t]}\right) \\ & + Q\left(\boldsymbol{\mathcal{X}}^{[t+1]}, \left\{(\boldsymbol{A}^{(n)})^{[t+1]}\right\} | \boldsymbol{\mathcal{X}}^{[t]}\right) \\ \leq \; & L\left(\boldsymbol{\mathcal{X}}^{[t]}, \left\{(\boldsymbol{A}^{(n)})^{[t]}\right\}\right) - Q\left(\boldsymbol{\mathcal{X}}^{[t]}, \left\{(\boldsymbol{A}^{(n)})^{[t]}\right\} | \boldsymbol{\mathcal{X}}^{[t]}\right) \\ & + Q\left(\boldsymbol{\mathcal{X}}^{[t+1]}, \left\{(\boldsymbol{A}^{(n)})^{[t+1]}\right\} | \boldsymbol{\mathcal{X}}^{[t]}\right) \\ \leq \; & L\left(\boldsymbol{\mathcal{X}}^{[t]}, \left\{(\boldsymbol{A}^{(n)})^{[t]}\right\}\right) - Q\left(\boldsymbol{\mathcal{X}}^{[t]}, \left\{(\boldsymbol{A}^{(n)})^{[t]}\right\} | \boldsymbol{\mathcal{X}}^{[t]}\right) \\ & + Q\left(\boldsymbol{\mathcal{X}}^{[t]}, \left\{(\boldsymbol{A}^{(n)})^{[t]}\right\} | \boldsymbol{\mathcal{X}}^{[t]}\right) \\ = \; & L\left(\boldsymbol{\mathcal{X}}^{[t]}, \left\{(\boldsymbol{A}^{(n)})^{[t]}\right\}\right), \end{aligned} \tag{29}$$

where the first inequality follows from the fact that $Q(\boldsymbol{\mathcal{X}}, \{\boldsymbol{A}^{(n)}\} | \boldsymbol{\mathcal{X}}^{[t]}) - L(\boldsymbol{\mathcal{X}}, \{\boldsymbol{A}^{(n)}\})$ attains its minimum when $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{X}}^{[t]}$, and the second inequality comes from (28). We see that through iteratively decreasing (not necessarily minimizing) the surrogate function, the objective function $L(\boldsymbol{\mathcal{X}}, \{\boldsymbol{A}^{(n}\})$ is guaranteed to be non-increasing at each iteration.

For clarity, we summarize our algorithm as follows.

Remark: We discuss the computational complexity of the proposed method. The main computational task of our proposed algorithm at each iteration involves calculating a new estimate of $\boldsymbol{\mathcal{X}}^{[t]}$ and $\{(\boldsymbol{A}^{(n)})^{[t]}\}$. Specifically, the update of the core tensor $\boldsymbol{\mathcal{X}}$ involves computing an inverse of a $(\prod_n I_n) \times (\prod_n I_n)$ matrix (cf. (24)), which has a computational complexity of order $O(\prod_n I_n^3)$ and scaling polynomially with the data size. The computational complexity associated with the update of the ith row of $\boldsymbol{A}^{(n)}$ is of order $O(I_n^3 + (\sum_{k \neq n} I_k) \prod_k I_k)$ (cf. (27)), where the term $O((\sum_{k \neq n} I_k) \prod_k I_k)$ comes from the

```
Iterative Reweighted Algorithm for Tucker Decomposition
1. Given initial estimates $\{(\boldsymbol{A}^{(n)})^{[0]}\}$, $\boldsymbol{\mathcal{X}}^{[0]}$ and a
   pre-selected regularization parameter $\lambda_1$ and $\lambda_2$.
2. At iteration $t = 0, 1, \ldots$: Based on the estimate $\boldsymbol{\mathcal{X}}^{[t]}$,
   construct the surrogate function as depicted in (19).
   Search for a new estimate of $\{(\boldsymbol{A}^{(n)})^{[t+1]}\}$
   and $\boldsymbol{\mathcal{X}}^{[t+1]}$ via (24) and (27), respectively.
3. Go to Step 2 if $\| \boldsymbol{\mathcal{X}}^{[t+1]} - \boldsymbol{\mathcal{X}}^{[t]}\|_F > \eta$, where $\eta$ is a
   prescribed tolerance value; otherwise stop.
```

computation of $\boldsymbol{\Phi}$ and scales linearly with the data size, and the term $O(I_n^3)$ is related to the inverse of an $I_n \times I_n$ matrix. Since all rows of $\boldsymbol{A}^{(n)}$ share a same $\boldsymbol{\Phi}$, the computational complexity of updating $\boldsymbol{A}^{(n)}$ is of order $O(I_n^4 + (\sum_{k \neq n} I_k) \prod_k I_k)$. We see that the overall computational complexity at each iteration is dominated by $O(\prod_n I_n^3)$, which scales polynomially with the data size of the tensor $\boldsymbol{\mathcal{Y}}$, and makes the algorithm unsuitable for many real-world applications involving large dimensions. To address this issue, we resort to a computationally efficient algorithm, namely, an over-relaxed monotone fast iterative shrinkage-thresholding algorithm (MFISTA) [36], to solve the optimization (22). A directly calculation of (24) is no longer needed and a significant reduction in computational complexity can be achieved.

V. A Computationally Efficient Iterative Reweighted Algorithm

It is well known that first order methods [36], [47] based on function values and gradient evaluations are often practically most feasible options to solve many large-scale optimization problems. One famous first order method is the fast iterative shrinkage-thresholding algorithm (FISTA) [47]. It has a convergence rate of $O(1/k^2)$ for the minimization of the sum of a smooth and a possibly nonsmooth convex function, where k denotes the iteration counter. Later on in [36], an over-relaxed monotone FISTA (MFISTA) was proposed to overcome some limitations inherent in the FISTA. Specifically, the over-relaxed MFISTA guarantees the monotone decreasing in the function values, which has been shown to be helpful in many practical applications. In addition, the over-relaxed MFISTA admits a variable stepsize in a broader range than FISTA while keeping the same convergence rate. In the following, we first provide a brief review of the over-relaxed MFISTA, and then discuss how to extend the technique to solve our problem.

A. Review of Over-Relaxed MFISTA

Consider the general convex optimization problem [36]:

$$\min_{\boldsymbol{x}} F(\boldsymbol{x}) = f(\boldsymbol{x}) + g(\boldsymbol{x}),$$

where f is a smooth convex function with the Lipschitz continuous gradient $L(f)$, and g is a convex but possibly nonsmooth function. The over-relaxed MFISTA scheme is summarized as follows (details can be found in [36]). Given $\boldsymbol{x}^{[0]} = \boldsymbol{w}^{[1]}, \eta^{[1]} = 1, \delta \in (0, 2)$ and $\beta \in (0, (2-\delta)/L(f)]$, the sequence $\{\boldsymbol{x}^{[t]}\}$ is given by

$$\boldsymbol{z}^{[t]} = \operatorname{prox}_{\beta g}\left(\boldsymbol{w}^{[t]} - \beta \nabla f\left(\boldsymbol{w}^{[t]}\right)\right), \tag{30}$$

$$\boldsymbol{x}^{[t]} = \arg\min\left\{F(\boldsymbol{z}) | \boldsymbol{z} \in \left\{\boldsymbol{z}^{[t]}, \boldsymbol{x}^{[t-1]}\right\}\right\}, \tag{31}$$

$$\eta^{[t+1]} = \frac{1 + \sqrt{1 + 4\left(\eta^{[t]}\right)^2}}{2}, \tag{32}$$

$$\begin{aligned}\boldsymbol{w}^{[t+1]} = {} & \boldsymbol{x}^{[t]} + \frac{\eta^{[t]}}{\eta^{[t+1]}}\left(\boldsymbol{z}^{[t]} - \boldsymbol{x}^{[t]}\right) \\ & + \frac{\eta^{[t]} - 1}{\eta^{[t+1]}}\left(\boldsymbol{x}^{[t]} - \boldsymbol{x}^{[t-1]}\right) \\ & + \frac{\eta^{[t]}}{\eta^{[t+1]}}(1 - \delta)\left(\boldsymbol{w}^{[t]} - \boldsymbol{z}^{[t]}\right),\end{aligned} \tag{33}$$

where $\nabla f(\boldsymbol{x})$ denotes the gradient of $f(\boldsymbol{x})$, and the proximal operator is defined as [36]

$$\operatorname{prox}_{\beta g}(\boldsymbol{x}) : \boldsymbol{z} = \arg\min_{\boldsymbol{z}} \left\{ g(\boldsymbol{z}) + \frac{1}{2\beta} \| \boldsymbol{z} - \boldsymbol{x} \|_2^2 \right\}. \tag{34}$$

It was proved in [36] that the sequence $\{\boldsymbol{x}^{[t]}\}$ is guaranteed to monotonically decrease the objective function $F(\boldsymbol{x})$ and the convergence rate is $O(1/k^2)$. Since (22) is convex, the over-relaxed MFISTA can be employed to efficiently solve (22).

B. Solving (22) via the Over-Relaxed MFISTA

Consider the optimization (22). Let $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ respectively represent the data fitting and regularization terms, i.e.,

$$\begin{aligned} f(\boldsymbol{x}) &= \lambda_1 \|\boldsymbol{\Sigma}(\boldsymbol{y} - \boldsymbol{H}\boldsymbol{x})\|_F^2, \\ g(\boldsymbol{x}) &= \boldsymbol{x}^T \boldsymbol{D} \boldsymbol{x}. \end{aligned}$$

Recalling that $\boldsymbol{H}$ is defined in (23). To apply the over-relaxed MFISTA, we need to compute $\nabla f(\boldsymbol{x})$, $\operatorname{prox}_{\beta g}(\boldsymbol{x})$, and determine the value of β. The gradient of $f(\boldsymbol{x})$ can be easily computed as

$$\nabla f(\boldsymbol{x}) = 2\lambda_1 \boldsymbol{H}^T \boldsymbol{\Sigma} \boldsymbol{H} \boldsymbol{x} - 2\lambda_1 \boldsymbol{H}^T \boldsymbol{\Sigma} \boldsymbol{y}, \tag{35}$$

which can also be expressed in a tensor form as

$$\begin{aligned}\nabla f(\boldsymbol{\mathcal{X}}) = 2\lambda_1 & \left(\boldsymbol{\mathcal{O}} * \left(\boldsymbol{\mathcal{X}} \prod_{n=1}^{N} \times_n \boldsymbol{A}^{(n)} - \boldsymbol{\mathcal{Y}} \right) \right) \\ & \times \prod_{n=1}^{N} \times_n \left(\boldsymbol{A}^{(n)}\right)^T.\end{aligned} \tag{36}$$

Such a tensor representation enables a more efficient computation of $\nabla f(\boldsymbol{x})$. The proximal operation $\operatorname{prox}_{\beta g}(\boldsymbol{x})$ defined in (34) can be readily obtained as

$$\begin{aligned}\boldsymbol{z} &= \arg\min_{\boldsymbol{z}} \left\{ g(\boldsymbol{z}) + \frac{1}{2\beta} \|\boldsymbol{z} - \boldsymbol{x}\|_2^2 \right\} \\ &= (2\beta \boldsymbol{D} + \boldsymbol{I})^{-1} \boldsymbol{x}.\end{aligned} \tag{37}$$

Note that since $\boldsymbol{D}$ is a diagonal matrix, the inverse of $2\beta\boldsymbol{D} + \boldsymbol{I}$ is easy to calculate. We now discuss the choice of β in the MFISTA. As mentioned earlier, β has to be smaller than $(2 - \delta)/L(f)$;

otherwise convergence of the scheme cannot be guaranteed. Recalling that the Lipschitz continuous gradient $L(f)$ is defined as any constant which satisfies the following inequality [48]

$$\parallel \nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y}) \parallel \leq L(f) \parallel \boldsymbol{x} - \boldsymbol{y} \parallel, \quad \text{for every } \boldsymbol{x}, \boldsymbol{y},$$

where $\parallel \cdot \parallel$ denotes the standard Euclidean norm. Hence it is easy to verify that

$$L(f) = 2\lambda_1 \lambda_{\max}\left(\boldsymbol{H}^T \boldsymbol{\Sigma} \boldsymbol{H}\right) \tag{38}$$

is a Lipschitz constant of $\nabla f(\boldsymbol{x})$, where $\lambda_{\max}(\boldsymbol{X})$ denotes the largest eigenvalue of the matrix $\boldsymbol{X}$. Note that $\boldsymbol{H}^T \boldsymbol{\Sigma} \boldsymbol{H}$ is of dimension $(\prod_n I_n) \times (\prod_n I_n)$. Calculation of $L(f)$, therefore, requires tremendous computational efforts. To circumvent this issue, we seek an upper bound of $L(f)$ that is easier to compute. Such an upper bound can be obtained by noticing that $\boldsymbol{\Sigma}$ is a diagonal matrix with its diagonal element equal to zero or one

$$\begin{aligned} L(f) &= 2\lambda_1 \lambda_{\max}\left(\boldsymbol{H}^T \boldsymbol{\Sigma} \boldsymbol{H}\right) \\ &\overset{(a)}{\leq} 2\lambda_1 \lambda_{\max}\left(\boldsymbol{H}^T \boldsymbol{H}\right) \\ &\overset{(b)}{=} 2\lambda_1 \prod_{n=1}^{N} \lambda_{\max}\left(\boldsymbol{A}^{(n)T} \boldsymbol{A}^{(n)}\right) \triangleq \tilde{L}, \end{aligned} \tag{39}$$

where (a) follows from the fact that $\boldsymbol{H}^T\boldsymbol{H} - \boldsymbol{H}^T \boldsymbol{\Sigma} \boldsymbol{H}$ is positive semi-definite, and (b) comes from the Kronecker product's properties

$$\boldsymbol{H}^T\boldsymbol{H} = \left(\bigotimes_n \boldsymbol{A}^{(n)}\right)^T \left(\bigotimes_n \boldsymbol{A}^{(n)}\right) = \bigotimes_n \left(\boldsymbol{A}^{(n)T}\boldsymbol{A}^{(n)}\right)$$

and

$$\operatorname{eig}\left(\bigotimes_n \left(\boldsymbol{A}^{(n)T}\boldsymbol{A}^{(n)}\right)\right) = \bigotimes_n \operatorname{eig}\left(\boldsymbol{A}^{(n)T}\boldsymbol{A}^{(n)}\right),$$

in which $\operatorname{eig}(\boldsymbol{X})$ is a vector consisting of the eigenvalues of matrix $\boldsymbol{X}$. Since $(2-\delta)/\tilde{L} \leq (2-\delta)/L(f)$, β can be chosen from $(0, (2-\delta)/\tilde{L}]$, without affecting the convergence rate of the over-relaxed MFISTA. The calculation of $\tilde{L}$ is much easier than $L(f)$ as the dimension of the matrix involved in the eigenvalue decomposition has been significantly reduced.

Algorithm 1: Iterative Re-weighted Algorithm For Incomplete Tensor Decomposition.

```
Input: $\mathcal{Y}$, $\mathcal{O}$, $\delta$, $\lambda_1$ and $\lambda_2$.
Output: $\mathcal{X}$, $\{A^{(n)}\}_{n=1}^{N}$, $\mathcal{Y}$ and multilinear rank.
Initialize $\mathcal{X}$, $\{A^{(n)}\}$, $\mathcal{D}$;
while not converge do
  Calculate $\tilde{L}$ using (39) and select $\beta$
  from $(0, (2-\delta)/\tilde{L}]$;
  Set $x^{[0]} = \mathrm{vec}(\mathcal{X})$;
  for $t = 1$ to $t_{\max}$ do
    Calculate the gradient of $f(\mathrm{tensor}(x^{[t]}))$ using
    (36) and the proximal operation $\mathrm{prox}_{\beta g}(x)$ using
    (37);
    Update $x^{[t]}$ using (30), (31), (32), (33);
  end for
  Set $\mathcal{X} = \mathrm{tensor}(x^{[t_{\max}]})$;
  for $n = 1$ to $N$ do
    for $i = 1$ to $I_n$ do
      Update $a_{i,\cdot}^{(n)}$ using (27);
    end for
  end for
  Remove the zero order-$(N-1)$ sub-tensors of $\mathcal{X}$
  and corresponding columns of $\{A^{(n)}\}$ (an
  optional procedure);
end while
Reconstruct $\mathcal{Y}$ using estimated $\mathcal{X}$ and $\{A^{(n)}\}$;
Estimate multilinear rank by count the nonzero order-
  $(N-1)$ sub-tensors of estimated $\mathcal{X}$ along each mode.
```

Remarks: We see that the dominant operation in solving (22) via the over-relaxed MFISTA is the evaluation of gradient (36), which has a computational complexity of order $O((\sum_n I_n)\prod_n I_n)$ that scales linearly with the data size of the observed tensor. Thus a significant reduction in computational complexity is achieved as compared with a direct calculation of (24). In addition, our proposed iterative reweighted method only needs to decrease (not necessarily minimize) the surrogate function at each iteration. Therefore when applying the over-relaxed MFISTA to solve (22), there is no need to wait until convergence is achieved. Only a few iterations are enough since the over-relaxed MFISTA guarantees a monotome decreasing in the function values. This further reduces the computational complexity of the proposed algorithm.

For clarity, we now summarize the proposed computationally efficient iterative reweighted algorithm as follows.

VI. Simulation Results

In this section, we conduct experiments to illustrate the performance of our proposed iterative reweighted Tucker decomposition method (referred to as IRTD). In our simulations, we set $\delta = 0.1$, $\beta = (2-\delta)/\tilde{L}$ and $\lambda_2 = 1$. In fact, our proposed algorithm is insensitive to the choices of these parameters. The choice of λ_1 is more critical than the others, and a suitable choice of λ_1 depends on the noise level and the data missing ratio. Empirical results suggest that stable recovery performance can be achieved when λ_1 is set in the range $[0.1, 2]$. The factor matrices and the core tensor are initialized by decomposing the observed tensor (the missing elements are set to zero) with high order singular value decomposition [49]. The over-relaxed MFISTA performs one hundred iterations to update the core tensor, i.e., $t_{\max} = 100$. In the iterative process, some sub-tensors of the core tensor keep decreasing to a small value but will not become exact zero, which is particularly the case for the noisy case. Hence a threshold is needed to prune these negligible sub-tensors. In our algorithm, this pruning operation is conducted at each iteration, and a sub-tensor is removed if its Frobenius norm is smaller than a scaled maximum Frobenius norm of all sub-tensors along the same mode, i.e.,

$$\left\|\mathcal{X}_{(n,i)}\right\|_F \leq \gamma \max_j \left\{\left\|\mathcal{X}_{(n,j)}\right\|_F\right\}, \tag{40}$$

where γ is set to 0.05 for the noisy case and 10^{-5} for the noiseless case. For our proposed algorithm, we continue the iterative process until the difference between the reconstructed tensors of successive iterations is negligible, i.e.,

$$\| \mathcal{Y}^{[t+1]} - \mathcal{Y}^{[t]} \|_F < 10^{-5}. \tag{41}$$

We compare our method with several existing state-of-the-art tensor decomposition/completion methods, namely, a CP decomposition-based tensor completion method (also referred to as the low rank tensor imputation (LRTI)) which uses the Frobenius-norm of the factor matrices as the rank regularization [18], a tensor nuclear-norm based tensor completion method [23] which is also referred to as the high accuracy low rank tensor completion (HaLRTC) method, and a Tucker factorization method based on pre-specified multilinear rank [22] which is referred to as the WTucker method. Since all competing algorithms exploit the intrinsic low-rank structure of the tensor, all of them are able to address both data completion and denoising problems being considered in our experiments. It should be noted that the LRTI requires to set a regularization parameter λ to control the tradeoff between the rank and the data fitting error, the HaLRTC method is unable to provide an explicit multilinear rank estimate, and the WTucker method requires an over-estimated multilinear rank. All the parameters used for competing algorithms are tuned carefully to ensure the best performance is achieved. Explicit values of the parameters of respective algorithms are also provided in our paper. Codes of our proposed algorithm along with other competing algorithms are available at `http://www.junfang-uestc.net/codes/TD.rar`.

A. Synthetic and Chemometrics Data

In this subsection, we carry out experiments on synthetic and chemometrics data. Two sets of synthetic data are generated and both of them are third-order tensors of size $32 \times 32 \times 32$. The first tensor is generated according to the CP model which is a summation of six equally-weighted rank-one tensors, with all of the factor matrices drawn from a normal distribution. Thus the true rank is 6 or (6, 6, 6) in a multilinear rank form. The other tensor is generated based on the Tucker decomposition model, with a random core tensor of size (3, 4, 5) multiplied by random factor matrices along each mode. Clearly, the groundtruth for the multilinear rank is (3, 4, 5). Two chemometrics data sets are also considered in our simulations. One is the Amino Acid data set [50] of size $5 \times 201 \times 61$ and the other is the flow injection data set [15] of size $12 \times 100 \times 89$.

For each data set, we consider two cases with 50% or 80% entries missing in our simulations, where the missing entries are randomly selected. The observed entries are corrupted by zero mean Gaussian noise and the signal-to-noise ratio (SNR) is set to different levels. The tensor reconstruction performance is evaluated by two metrics. The first is the total normalized mean squared error (NMSE(T)) which is defined as $\| \mathcal{Y} - \widehat{\mathcal{Y}} \|_F / \| \mathcal{Y} \|_F$, where $\mathcal{Y}$ and $\widehat{\mathcal{Y}}$ denote the true tensor and the estimated one, respectively. The other metric is the generalization error (referred to as NMSE(G)) which is defined as $\| \mathcal{O}^{\mathsf{C}} * (\mathcal{Y} - \widehat{\mathcal{Y}}) \|_F / \| \mathcal{Y} \|_F$, where $\mathcal{O}^{\mathsf{C}}$ denotes the complement set of the observed set $\mathcal{O}$. The parameter λ for the LRTI is carefully selected as $\lambda = 0.3$ for the synthetic data set generated by the Tucker model and $\lambda = 0.2$ for the other data sets. The pre-defined multilinear rank for WTucker is set to be (12, 12, 12), (6, 8, 10), (5, 10, 10) and (10, 10, 10) for the CP, Tucker, Amino Acid and flow injection dataset, respectively. For our proposed method, we choose $\lambda_1 = 0.1$ for all data sets. Results are averaged over 100 independent runs. The rank or multilinear rank is estimated as the most frequently occurring rank or multilinear rank value. The standard deviation of the estimated rank is also reported as an indication of the stability of the inferred rank. Results are shown in Tables I– IV.

1) We observe that the proposed method presents the best recovery accuracy in most cases for the first three data sets. Also, from the synthetic data sets, we see that the proposed method can reliably estimate the true multilinear rank of the tensor.
2) Compared with the CP-decomposition based method LRTI, our proposed method presents a clear performance advantage over the LRTI when synthetic data are generated according to the Tucker model. More surprisingly, it also outperforms the LRTI when data are generated according to the CP model.
3) Our proposed method surpasses the tensor nuclear-norm based method HaLRTC by a big margin, particularly when the missing ratio is high (80% elements missing). This corroborates our claim that the proposed group log-sum functional is more effective than the tensor nuclear-norm in approximating the multilinear rank.
4) The WTucker method performs fairly well in the high SNR regime, but it is more sensitive to noise than our proposed method and suffers from considerable performance degradation in the low SNR regime.
5) We also observe that our proposed method has similar run times as the other three algorithms. As the number of missing entries increases, our proposed method might need a few more iterations to reach convergence, and thus the average run time increases with the number of missing entries.

B. Image Inpainting

The goal of image inpainting is to complete an image with missing pixels. For a two-dimensional RGB picture, we can treat it as a third-order tensor. Here we consider imputing an incomplete RGB image (of size $256 \times 256 \times 3$) via respective algorithms. The benchmark Lena image is used, with 50%, 80% and 90% missing entries considered in our simulations. The recovery accuracy is evaluated by the MSE metric which is defined as $\mathrm{MSE} = \frac{1}{M} \| \mathcal{O}^{\mathsf{C}} * (\mathcal{Y} - \widehat{\mathcal{Y}}) \|_F^2$, where $\mathcal{Y}$ and $\widehat{\mathcal{Y}}$ respectively denote the original normalized image and the estimated one, and M denotes the number of missing elements. For LRTI, the parameter λ is carefully selected to 5, 3 and 3 for 50%, 80% and 90% missing entries, respectively. The predefined rank for the WTucker is carefully set to (80, 80, 3), (40, 40, 3) and (15, 15, 3) for 50%, 80% and 90% missing entries, respectively. For our proposed method, λ_1 is set to 3, 0.5 and 0.3 for 50%, 80% and 90% missing entries, respectively. The observed and recovered images are shown in Fig. 2 and MSEs of

TABLE I
SYNTHETIC DATA GENERATED ACCORDING TO THE CP MODEL

		10 dB		20 dB		30 dB	
		50%	80%	50%	80%	50%	80%
LRTI	NMSE(G)	0.0508	0.1361	0.0303	0.1083	0.0263	0.1049
	NMSE(T)	0.0704	0.1490	0.0415	0.1178	0.0360	0.1138
	Rank	6	6	6	6	6	6
	Std(R)	0.1	0.3	0.1	0.2	0	0.2
	runtime	1.3590	0.5944	1.3590	0.5755	1.3792	0.5854
HaLRTC	NMSE(G)	0.2206	0.6598	0.0808	0.5675	0.0271	0.5499
	NMSE(T)	0.3141	0.6748	0.1074	0.5693	0.0352	0.5501
	runtime	3.4337	3.4321	3.4332	3.4237	3.4354	3.4356
WTucker	NMSE(G)	0.1244	0.3537	0.0382	0.1062	0.0109	0.0309
	NMSE(T)	0.1636	0.3688	0.0505	0.1110	0.0145	0.0323
	runtime	188.7377	336.5523	202.6277	422.9503	163.3542	366.1216
IRTD	NMSE(G)	**0.0471**	**0.1004**	**0.0151**	**0.0341**	**0.0052**	**0.0175**
	NMSE(T)	**0.0655**	**0.1105**	**0.0209**	**0.0374**	**0.0071**	**0.0190**
	n-Rank	(6, 6, 6)	(6, 6, 6)	(6, 6, 6)	(6, 6, 6)	(6, 6, 6)	(6, 6, 6)
	Std(R)	(0, 0, 0)	(0.2, 0.1, 0.2)	(0, 0, 0)	(0, 0.2, 0.2)	(0.2, 0.2, 0.1)	(0.6, 0.5, 0.4)
	runtime	6.0015	51.4581	3.8603	30.0968	3.6856	23.7581

TABLE II
SYNTHETIC DATA GENERATED ACCORDING TO THE TUCKER DECOMPOSITION MODEL)

		10 dB		20 dB		30 dB	
		50%	80%	50%	80%	50%	80%
LRTI	NMSE(G)	0.0645	0.1140	0.0246	0.0660	0.0162	0.0605
	NMSE(T)	0.0893	0.1247	0.0338	0.0715	0.0220	0.0653
	Rank	13	9	7	7	7	7
	Std(R)	3.0	0.9	0.5	0.4	0.2	0.5
	runtime	39.9267	13.5612	16.0882	9.6229	15.7899	9.3247
HaLRTC	NMSE(G)	0.1572	0.4620	0.0542	0.2503	0.0176	0.1785
	NMSE(T)	0.2733	0.4833	0.0891	0.2543	0.0285	0.1791
	runtime	3.3098	3.3187	3.3113	3.3109	3.3128	3.3160
WTucker	NMSE(G)	0.0771	0.1803	0.0227	0.0543	0.0061	0.0157
	NMSE(T)	0.1052	0.1942	0.0311	0.0586	0.0083	**0.0169**
	runtime	21.4768	27.3009	20.9629	31.2424	19.4503	28.9243
IRTD	NMSE(G)	**0.0363**	**0.0743**	**0.0115**	**0.0269**	**0.0044**	**0.0155**
	NMSE(T)	**0.0507**	**0.0821**	**0.0160**	**0.0297**	**0.0061**	**0.0169**
	n-Rank	(3, 4, 5)	(3, 4, 5)	(3, 4, 5)	(3, 4, 5)	(3, 4, 5)	(3, 4, 5)
	Std(R)	(0, 0, 0.1)	(0, 0, 0.1)	(0, 0, 0)	(0.2, 0.2, 0.1)	(0.1, 0.1, 0.1)	(0.3, 0.2, 0.2)
	runtime	10.8999	37.2216	9.4080	31.8202	9.0746	31.3034

TABLE III
AMINO ACID DATA SET

		10 dB		20 dB		30 dB	
		50%	80%	50%	80%	50%	80%
LRTI	NMSE(G)	0.0512	0.0877	0.0505	0.0546	0.0432	0.0501
	NMSE(T)	0.0713	0.0964	0.0702	0.0594	0.0602	0.0543
	Rank	3	3	3	3	3	3
	Std(R)	0.5	0	0.3	0	0.3	0
	runtime	10.1082	7.1272	9.3665	6.2665	9.3229	6.0151
HaLRTC	NMSE(G)	0.1145	0.2472	0.0406	0.1056	0.0171	0.0496
	NMSE(T)	0.2513	0.2849	0.0816	0.1147	0.0282	0.0516
	runtime	6.3819	6.3534	6.3932	6.3461	6.3775	6.3365
WTucker	NMSE(G)	0.0973	0.2427	0.0308	0.0774	**0.0123**	0.0275
	NMSE(T)	0.1314	0.2580	0.0414	0.0822	**0.0163**	0.0289
	runtime	50.0902	265.9540	33.8282	73.4443	35.8945	73.3536
IRTD	NMSE(G)	**0.0420**	**0.0835**	**0.0209**	**0.0328**	0.0179	**0.0231**
	NMSE(T)	**0.0586**	**0.0920**	**0.0291**	**0.0361**	0.0250	**0.0254**
	n-Rank	(3, 3, 3)	(3, 3, 3)	(3, 3, 3)	(3, 3, 3)	(3, 3, 3)	(4, 3, 3)
	Std(R)	(0, 0, 0.1)	(0, 0.7, 0.4)	(0.5, 0, 0.1)	(0.9, 0, 0.1)	(0.3, 0, 0)	(0.3, 0.1, 0.1)
	runtime	7.9760	78.1650	9.2975	37.1556	8.5151	34.8738

TABLE IV
FLOW INJECTION DATA SET

		10 dB		20 dB		30 dB	
		50%	80%	50%	80%	50%	80%
LRTI	NMSE(G)	**0.0515**	0.1018	0.0401	0.0848	0.0343	0.0862
	NMSE(T)	**0.0721**	0.1127	0.0560	0.0938	0.0477	0.0953
	Rank	8	6	7	6	7	6
	Std(R)	1.1	0.6	0.7	0.3	0.1	0.5
	runtime	101.7574	25.0878	114.9761	18.9976	133.0485	19.2344
HaLRTC	NMSE(G)	0.1019	0.2239	0.0376	0.0890	0.0143	0.0377
	NMSE(T)	0.2457	0.2648	0.0801	0.0996	0.0265	0.0403
	runtime	7.4358	7.4215	7.4126	7.4033	7.4122	7.4139
WTucker	NMSE(G)	0.0657	0.1506	**0.0187**	**0.0427**	**0.0071**	**0.0134**
	NMSE(T)	0.0908	0.1641	**0.0259**	**0.0467**	**0.0098**	**0.0147**
	runtime	49.0418	100.8046	47.3599	88.4482	49.6637	92.7558
IRTD	NMSE(G)	0.0570	**0.0783**	0.0503	0.0645	0.0500	0.0625
	NMSE(T)	0.0800	**0.0869**	0.0705	0.0714	0.0700	0.0692
	n-Rank	(4, 5, 3)	(4, 6, 3)	(4, 5, 3)	(4, 5, 3)	(4, 5, 3)	(4, 5, 3)
	Std(R)	(0.1, 0.3, 0)	(0, 0.6, 0.7)	(0, 0.3, 0)	(0.2, 0.4, 0.3)	(0, 0.2, 0)	(0.1, 0.5, 0.2)
	runtime	21.8144	66.9538	19.5812	52.9545	18.5524	54.3140

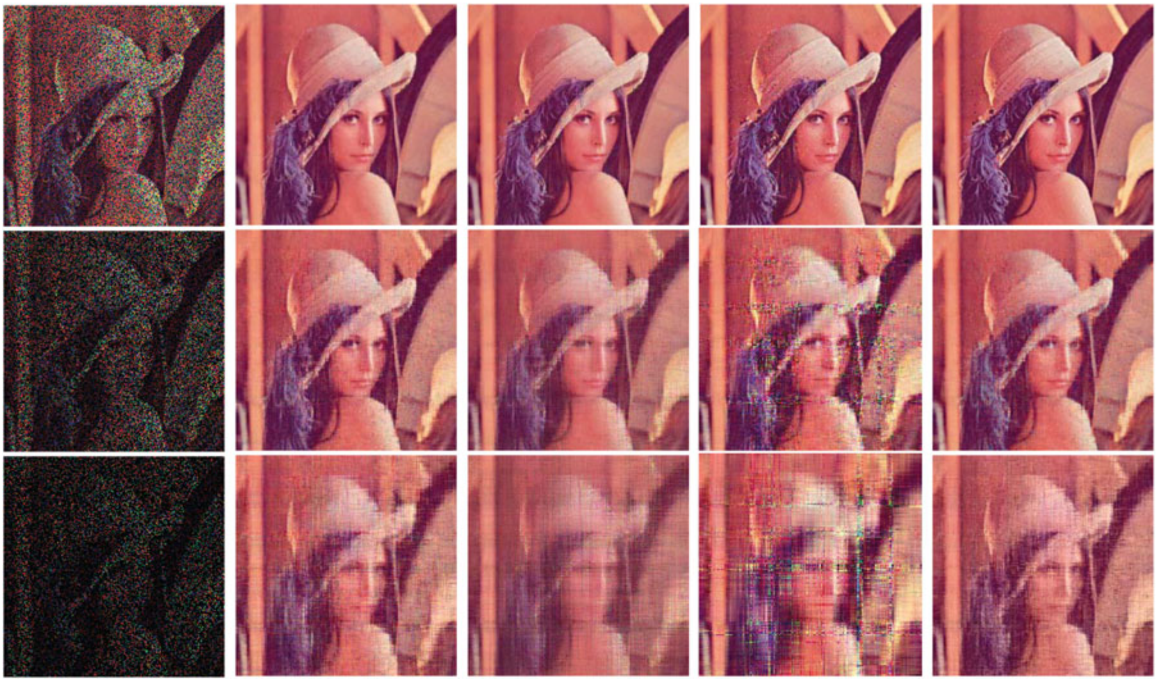


Fig. 2. RGB image inpainting. From left to right: Observed image, images reconstructed by LRTI, HaLRTC, WTucker, and IRTD. Top row: 50% missing. Middle row: 80% missing. Bottom row: 90% missing.

respective algorithms are shown in Table V. From Table V, we see that the proposed method renders a reliable recovery even with 90% missing entries, while the other two Tucker model-based methods WTucker and HaLRTC incur a considerable performance degradation when the missing ratio is high. It should be noted that our proposed method and other competing methods being considered in our paper are not specifically designed for inpainting. Those specialized inpainting methods (e.g., [21], [51]) which utilize both global and local structures inherent in image data may achieve better performance.

TABLE V
IMAGE INPAINTING

		50%	80%	90%
LRTI	MSE	0.0023	**0.0046**	0.0098
	Rank	89	73	49
HaLRTC	MSE	0.0019	0.0066	0.0134
WTucker	MSE	0.0032	0.0188	0.0334
IRTD	MSE	**0.0015**	**0.0046**	**0.0082**
	Rank	(84, 94, 3)	(37, 40, 3)	(23, 26, 3)

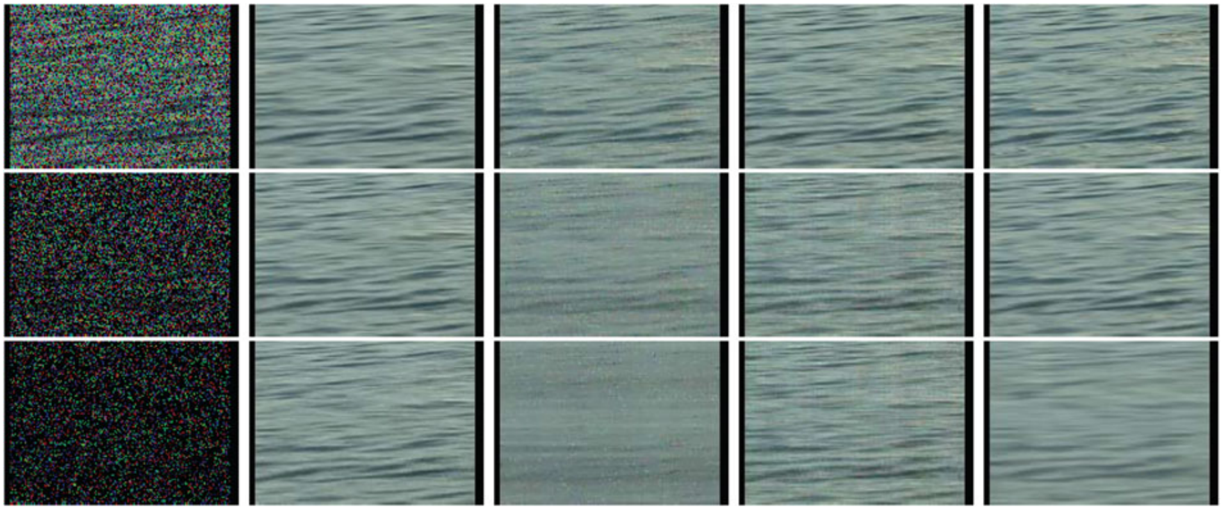


Fig. 3. Video recovery. From left to right: The first image of the video sequence, images reconstructed by LRTI, HaLRTC, WTucker, and IRTD. Top row: 50% missing. Middle row: 80% missing. Bottom row: 90% missing.

TABLE VI
VIDEO RECOVERY

		50%	80%	90%
LRTI	MSE	8.55E-04	0.0011	**0.0013**
	Rank	152	148	49
HaLRTC	MSE	0.0016	0.0053	0.0085
WTucker	MSE	5.96E-04	0.0020	0.0036
IRTD	MSE	**2.16E-04**	**8.60E-04**	0.0026
	Rank	(112, 62, 3, 32)	(80, 30, 3, 26)	(47, 35, 3, 20)

C. Video Recovery

In this subsection, we consider the problem of recovering a video sequence with missing pixels. A video sequence consisting of a number of consecutive color images can be viewed as a fourth-order tensor. The ocean video sequence of size $112 \times 160 \times 3 \times 32$ [17] is used in our experiments. Again, 50%, 80% and 90% missing ratios are considered and the recovery accuracy is evaluated by the MSE defined in the above subsection. For LRTI, the parameter λ is set to 5, 3 and 3 for 50%, 80% and 90% missing ratios, respectively. The pre-defined rank for the WTucker is set to (100, 100, 3, 32), (100, 100, 3, 32) and (80, 80, 3, 32) for 50%, 80% and 90% missing ratios, respectively. For our proposed method, λ_1 is set to 1, 0.2 and 0.2 for 50%, 80% and 90% missing ratios, respectively. The MSEs of respective algorithms were reported in Table VI, along with the ranks estimated by our proposed method and the LRTI. We see that all methods yield quite decent recovery results and the proposed method achieves the best performance for 50% and 80% missing ratios. The first image of the video sequence (with missing elements) and images recovered by respective algorithms are shown in Fig. 3.

VII. Conclusions and Future Work

In this paper, we proposed an iterative reweighted algorithm to decompose an incomplete tensor into a concise Tucker decomposition. To automatically determine the model complexity, we introduced a new notion called order-$(N-1)$ sub-tensor and introduced a group log-sum penalty on every order-$(N-1)$ sub-tensors to achieve a structural sparse core tensor. By shrinking the zero order-$(N-1)$ sub-tensors, the core tensor becomes a smaller one and a compact Tucker decomposition can be obtained. By resorting to the majorization-minimization approach, an iterative reweight algorithm was developed. Also, the over-relaxed monotone fast iterative shrinkage-thresholding technique is adapted and embedded in the iterative reweighted process to reduce the computational complexity. The performance of the proposed method is evaluated using synthetic data and real data. Simulation results show that the proposed method offers competitive performance compared with existing methods. Possible future directions along the line of the current work include the following two aspects. One meaningful research direction, we believe, is to develop an online algorithm that allows to deal with massive tensors, for which batch methods may be infeasible. Another direction is to study side information-aided tensor decomposition. In some practical applications such as recommender systems, we may have side information about the mode entities in form of their features and/or their adjacency network. This side information can be exploited to better extract the latent structure of the tensor.

References

[1] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, “Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering,” in *Proc. 4th ACM Conf. Recommender Syst.*, Barcelona, Spain, Sept. 26–30, 2010, pp. 79–86.

[2] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell, “Temporal collaborative filtering with Bayesian probabilistic tensor factorization,” in *Proc. SIAM Int. Conf. Data Mini.*, 2010, vol. 10, pp. 211–222.

[3] G. Adomavicius and A. Tuzhilin, “Context-aware recommender systems,” *Recommender Systems Handbook*. 2011, pp. 217–253, Springer US.

[4] P. Rai, Y. Wang, and L. Carin, “Leveraging features and networks for probabilistic tensor decomposition,” in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 2942–2948.

[5] I. Sutskever, J. B. Tenenbaum, and R. R. Salakhutdinov, “Modelling relational data using Bayesian clustered tensor factorization,” *Adv. Neural Inf. Process. Syst.*, pp. 1821–1828, 2009.

[6] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi, "Tensor decomposition of EEG signals: A brief review," *J. Neurosci. Methods*, vol. 248, pp. 59–69, 2015.

[7] A. Cichocki, R. Zdunek, A. H. Phan, and S. I. Amari, *Nonnegative Matrix and Tensor Factorizations*. New York, NY, USA: Wiley, 2009.

[8] M. Haardt, F. Roemer, and G. D. Galdo, "Higher-order SVD-based subspace estimation to improve the parameter estimation accuracy in multidimensional harmonic retrieval problems," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3198–3213, Jul. 2008.

[9] W. Guo and W. Yu, "Variational Bayesian PARAFAC decomposition for multidimensional harmonic retrieval," in *Proc. IEEE CIE Int. Conf. Radar (Radar)*, 2011, vol. 2, pp. 1864–1867.

[10] E. Benetos and C. Kotropoulos, "A tensor-based approach for automatic music genre classification," in *Proc. 16th Eur. Signal Process. Conf.*, 2008, pp. 1–4.

[11] G. Zhou, A. Cichocki, Q. Zhao, and S. Xie, "Nonnegative matrix and tensor factorizations: An algorithmic perspective," *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 54–65, May 2014.

[12] Q. Li and D. Schonfeld, "Multilinear discriminant analysis for higher-order tensor data classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2524–2537, Dec. 2014.

[13] P. Rai, Y. Wang, S. Guo, G. Chen, D. Dunson, and L. Carin, "Scalable Bayesian low-rank decomposition of incomplete multiway tensors," in *Proc. 31st Int. Conf. Mach. Learn. (ICML-14)*, Beijing, China, 2014, vol. 32, pp. 1800–1808.

[14] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[15] R. Bro, "PARAFAC. Tutorial and applications," *Chemometr. Intell. Lab. Syst.*, vol. 38, no. 2, pp. 149–171, 1997.

[16] A. Cichocki, D. P. Mandic, H. A. Phan, C. F. Caiafa, G. Zhou, Q. Zhao, and L. D. Lathauwer, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 145–163, Mar. 2015.

[17] 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.*, 2009, pp. 2114–2121.

[18] J. A. G. Mateos and G. B. Giannakis, "Rank regularization and Bayesian inference for tensor completion and extrapolation," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5689–5703, Nov. 2013.

[19] M. Mardani, G. Mateos, and G. B. Giannakis, "Subspace learning and imputation for streaming big data matrices and tensors," *IEEE Trans. Signal Process.*, vol. 63, no. 10, pp. 2663–2677, May 2015.

[20] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometr. Intell. Lab. Syst.*, vol. 106, no. 1, pp. 41–56, 2011.

[21] 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.

[22] M. F. A. Jukić, "Tucker factorization with missing data with application to low-n-rank tensor completion," *Multidimen. Syst. Signal Process.*, vol. 26, pp. 677–692, 2015.

[23] 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.

[24] Y. Liu, F. Shang, W. Fan, J. Cheng, and H. Cheng, "Generalized higher-order orthogonal iteration for tensor decomposition and completion," in *Adv. Neural Inf. Process. Syst.*, 2014, pp. 1763–1771.

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

[26] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian CP factorization of incomplete tensors with automatic rank determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1751–1763, Sept. 2015.

[27] Z. Xu, F. Yan, and Y. Qi, "Bayesian nonparametric models for multiway data analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 2, pp. 475–487, Feb. 2015.

[28] L. D. Lathauwer, B. D. 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.

[29] X. Fu, K. Huang, W.-K. Ma, N. D. Sidiropoulos, and R. Bro, "Joint tensor factorization and outlying slab suppression with applications," *IEEE Trans. Signal Process.*, vol. 63, no. 23, pp. 6315–6328, Dec. 2015.

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

[31] J. B. Kruskal, "Rank, decomposition, and uniqueness for 3-way and n-way arrays," *Multiway Data Anal.*, vol. 33, pp. 7–18, 1989.

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

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

[34] H. Tan, B. Cheng, W. Wang, Y.-J. Zhang, and B. Ran, "Tensor completion via a multi-linear low-n-rank factorization model," *Neurocomput.*, vol. 133, pp. 161–169, 2014.

[35] F. Jiang, M. Oskarsson, and K. Astrom, "On the minimal problems of low-rank matrix factorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2549–2557.

[36] M. Yamagishi and I. Yamada, "Over-relaxation of the fast iterative shrinkage-thresholding algorithm with variable stepsize," *Inverse Problems*, vol. 27, no. 10, pp. 105008, Sept. 2011.

[37] T. G. Kolda, "Multilinear operators for higher-order decompositions," U.S. Dept. of Energy, 2006.

[38] E. J. Candés, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, pp. 877–905, 2008.

[39] R. Chartrand and W. Yin, "Iteratively reweighted algorithms for compressive sensing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2008, pp. 3869–3872.

[40] D. Wipf and S. Nagarajan, "Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 317–329, Apr. 2010.

[41] Y. Shen, J. Fang, and H. Li, "Exact reconstruction analysis of log-sum minimization for compressed sensing," *IEEE Signal Process. Lett.*, vol. 20, no. 12, pp. 1223–1226, Dec. 2013.

[42] L. Yuan, J. Liu, and J. Ye, "Efficient methods for overlapping group lasso," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2104–2116, Sep. 2013.

[43] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Statist. Soc. B (Statist. Methodol.)*, vol. 68, no. 1, pp. 49–67, 2006.

[44] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *Amer. Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[45] R. A. Harshman, "Foundations of the PARAFAC procedure: Model and conditions for an "explanatory" multi-mode factor analysis," in *UCLA Working Papers in Phonetics*, 1969, pp. 1–84.

[46] P. Comon, X. Luciani, and A. L. F. D. Almeida, "Tensor decompositions, alternating least squares and other tales," *J. Chemometr.*, vol. 23, no. 7–8, pp. 393–405, 2009.

[47] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

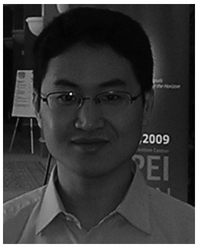
[48] W. Rudin, *Principles of Mathematical Analysis*. New York, NY, USA: McGraw-Hill, 1964, vol. 3.

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

[50] L. Nørgaard and C. Ridder, "Rank annihilation factor analysis applied to flow injection analysis with photodiode-array detection," *Chemometr. Intell. Lab. Syst.*, vol. 23, no. 1, pp. 107–114, 1994.

[51] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. 27th Annu. Conf. Comput. Graph. Interact. Tech.*, 2000, pp. 417–424, ACM Press/Addison-Wesley.

Linxiao Yang received the B.S. degree from the Southwest Jiaotong University, Chengdu, China in 2013. Since August 2015, he has been working towards his Ph.D. degree with the University of Electronic of Science and Technology of China. His current research interests include compressed sensing, sparse theory, tensor analysis and machine learning.



Jun Fang (M'08) received the B.S. and M.S. degrees from the Xidian University, Xi'an, China in 1998 and 2001, respectively, and the Ph.D. degree from the National University of Singapore, Singapore, in 2006, all in electrical engineering.

During 2006, he was a postdoctoral research associate in the Department of Electrical and Computer Engineering, Duke University. From January 2007 to December 2010, he was a research associate with the Department of Electrical and Computer Engineering, Stevens Institute of Technology. Since January 2011, he has been with the University of Electronic of Science and Technology of China. His current research interests include sparse theory and compressed sensing, and Bayesian inference for data analysis.

Dr. Fang received the IEEE Jack Neubauer Memorial Award in 2013 for the best systems paper published in the IEEE Transactions on Vehicular Technology. He is an Associate Technical Editor for IEEE Communications Magazine, and an Associate Editor for IEEE Signal Processing Letters.

Hongbin Li (M'99–SM'08) received the B.S. and M.S. degrees from the University of Electronic Science and Technology of China, in 1991 and 1994, respectively, and the Ph.D. degree from the University of Florida, Gainesville, FL, in 1999, all in electrical engineering.

From July 1996 to May 1999, he was a Research Assistant in the Department of Electrical and Computer Engineering at the University of Florida. Since July 1999, he has been with the Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ, where he is a Professor. He was a Summer Visiting Faculty Member at the Air Force Research Laboratory in the summers of 2003, 2004 and 2009. His general research interests include statistical signal processing, wireless communications, and radars.

Dr. Li received the IEEE Jack Neubauer Memorial Award in 2013 for the best systems paper published in the IEEE Transactions on Vehicular Technology, the Outstanding Paper Award from the IEEE AFICON Conference in 2011, the Harvey N. Davis Teaching Award in 2003 and the Jess H. Davis Memorial Award for excellence in research in 2001 from Stevens Institute of Technology, and the Sigma Xi Graduate Research Award from the University of Florida in 1999. He has been a member of the IEEE SPS Signal Processing Theory and Methods (2011 to now) Technical Committee (TC) and the IEEE SPS Sensor Array and Multichannel TC (2006 to 2012). He has been an Associate Editor for *Signal Processing* (Elsevier) (since 2013), IEEE Transactions on Signal Processing (2006 to 2009 and since 2014), IEEE Signal Processing Letters (2005 to 2006), and IEEE Transactions on Wireless Communications (2003 to 2006), as well as a Guest Editor for IEEE Journal of Selected Topics in Signal Processing and *EURASIP Journal on Applied Signal Processing*. He has been involved in various conference organization activities, including serving as a General Co-Chair for the 7th IEEE Sensor Array and Multichannel Signal Processing (SAM) Workshop, Hoboken, NJ, June 17–20, 2012. Dr. Li is a member of Tau Beta Pi and Phi Kappa Phi.

Bing Zeng (M'91–SM'13–F'15) received his B.S. and M.S. degrees in electronic engineering from University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1983 and 1986, respectively, and his Ph.D. degree in electrical engineering from Tampere University of Technology, Tampere, Finland, in 1991.

He worked as a postdoctoral fellow at University of Toronto from September 1991 to July 1992 and as a Researcher at Concordia University from August 1992 to January 1993. He then joined the Hong Kong University of Science and Technology (HKUST). After 20 years of service at HKUST, he returned to UESTC in the summer of 2013, through China "1000-Talent-Scheme". At UESTC, he leads the Institute of Image Processing to work on image and video processing, 3D and multi-view video technology, and visual big data.

During his tenure at HKUST and UESTC, he graduated more than 30 Master and Ph.D. students, received 25 research grants, filed 8 international patents, and published more than 200 papers. Three representing works are as follows: one paper on fast block motion estimation, published in IEEE Transactions on Circuits and Systems for Video Technology (TCSVT) in 1994, has so far been SCI-cited more than 880 times (Google-cited more than 1980 times) and currently stands at the seventh position among all papers published in this Transactions; one paper on smart padding for arbitrarily-shaped image blocks, published in IEEE TCSVT in 2001, leads to a patent that has been successfully licensed to companies; and one paper on directional discrete cosine transform (DDCT), published in IEEE TCSVT in 2008, receives the 2011 IEEE TCSVT Transactions Best Paper Award. He also received the best paper award at China-Com three times (2009 Xian, 2010 Beijing, and 2012 Kunming).

He served as an Associate Editor of IEEE TCSVT for eight years and received the Best Associate Editor Award in 2011. He is currently on the Editorial Board of the *Journal of Visual Communication and Image Representation* and serves as the General Co-Chair of IEEE VCIP-2016 to be held in Chengdu, China, in November 2016. He received a 2nd Class Natural Science Award (the first recipient) from Chinese Ministry of Education in 2014 and was elected as an IEEE Fellow in 2015 for contributions to image and video coding.