# Generalized Higher-Order Tensor Decomposition via Parallel ADMM

**Fanhua Shang[1], Yuanyuan Liu[2*], James Cheng[1]**

[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong
[2]Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong
{fhshang, jcheng}@cse.cuhk.edu.hk; yyliu@se.cuhk.edu.hk

## Abstract

Higher-order tensors are becoming prevalent in many scientific areas such as computer vision, social network analysis, data mining and neuroscience. Traditional tensor decomposition approaches face three major challenges: model selecting, gross corruptions and computational efficiency. To address these problems, we first propose a parallel trace norm regularized tensor decomposition method, and formulate it as a convex optimization problem. This method does not require the rank of each mode to be specified beforehand, and can automatically determine the number of factors in each mode through our optimization scheme. By considering the low-rank structure of the observed tensor, we analyze the equivalent relationship of the trace norm between a low-rank tensor and its core tensor. Then, we cast a non-convex tensor decomposition model into a weighted combination of multiple much smaller-scale matrix trace norm minimization. Finally, we develop two parallel alternating direction methods of multipliers (ADMM) to solve our problems. Experimental results verify that our regularized formulation is effective, and our methods are robust to noise or outliers.

## Introduction

The term tensor used in the context of this paper refers to a multi-dimensional array, also known as a multi-way or multi-mode array. For example, if $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, then we say $\mathcal{X}$ is a third-order tensor, where order is the number of ways or modes of the tensor. Thus, vectors and matrices are first-order and second-order tensors, respectively. Higher-order tensors arise in a wide variety of application areas, such as machine learning (Tomioka and Suzuki, 2013; Signoretto et al., 2014), computer vision (Liu et al., 2009), data mining (Yilmaz et al., 2011; Morup, 2011; Narita et al., 2012; Liu et al., 2014), numerical linear algebra (Lathauwer et al., 2000a; 2000b), and so on. Especially, with the rapid development of modern computer technology in recent years, tensors are becoming increasingly ubiquitous such as multi-channel images and videos, and have become increasingly popular (Kolda and Bader, 2009). When working with high-order tensor data, various new computational

challenges arise due to the exponential increase in time and memory space complexity when the number of orders increases. This is called the curse of dimensionality. In practice, the underlying tensor is often low-rank, even though the actual data may not be due to noise or arbitrary errors. Essentially, the major component contained in the given tensor is often governed by a relatively small number of latent factors.

One standard tool to alleviate the curse is tensor decomposition. Decomposition of high-order tensors into a small number of factors has been one of the main tasks in multi-way data analysis, and commonly takes two forms: Tucker decomposition (Tucker, 1966) and CANDE-COMP/PARAFAC (CP) (Harshman, 1970) decomposition. There are extensive studies in the literature for finding the Tucker decomposition and the CP decomposition for higher-order tensors (Kolda and Bader, 2009). In those tensor decomposition methods, their goal is to (approximately) reconstruct the input tensor as a sum of simpler components with the hope that these simpler components would reveal the latent structure of the data. However, existing tensor decomposition methods face three major challenges: rank selection, outliers and gross corruptions, and computational efficiency. Since the Tucker and CP decomposition methods are based on least-squares approximation, they are also very sensitive to outliers and gross corruptions (Goldfarb and Qin, 2014). In addition, the performance of those methods is usually sensitive to the given ranks of the involved tensor (Liu et al., 2009). To address the problems, we propose two robust and parallel higher-order tensor decomposition methods with trace norm regularization.

Recently, much attention has been drawn to the low-rank tensor recovery problem, which arises in a number of real-word applications, such as 3D image recovery, video inpainting, hyperspectral data recovery, and face reconstruction. Compared with matrix-based analysis methods, tensor-based multi-linear data analysis has shown that tensor models are capable of taking full advantage of the high-order structure to provide better understanding and more precision. The key idea of low-rank tensor completion and recovery methods is to employ matrix trace norm minimization (also known as the nuclear norm, which is the convex surrogate of the rank of the involved matrix). In addition, there are some theoretical developments that guarantee reconstruction

---

*Corresponding Author

of a low-rank tensor from partial measurements or grossly corrupted observations via solving the trace norm minimization problem under some reasonable conditions (Tomioka et al., 2011; Shi et al., 2013). Motivated by the recent progress in tensor completion and recovery, one goal of this paper is to extend the trace norm regularization to robust higher-order tensor decomposition.

Different from existing tensor decomposition methods, we first propose a parallel trace norm regularized tensor decomposition method, which can automatically determine the number of factors in each mode through our optimization scheme. In other words, this method does not require the rank of each mode to be specified beforehand. In addition, by considering the low-rank structure of the observed tensor and further improving the scalability of our convex method, we analyze the equivalent relationship of the trace norm between a low-rank tensor and its core tensor. Then, we cast the non-convex trace norm regularized higher-order orthogonal iteration model into a weighted combination of multiple much-smaller-scale matrix trace norm minimization. Moreover, we design two parallel alternating direction methods of multipliers (ADMM) to solve the proposed problems, which are shown to be fast, insensitive to initialization and robust to noise and/or outliers with extensive experiments.

## Notations and Related Work

We first introduce the notations, and more details can be seen in Kolda and Bader (2009). An $N$th-order tensor is denoted by a calligraphic letter, e.g., $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, and its entries are denoted by $t_{i_1 \cdots i_n \cdots i_N}$, where $i_n \in \{1, \cdots, I_n\}$ for $1 \leq n \leq N$. Fibers are the higher-order analogue of matrix rows and columns. The mode-$n$ fibers are vectors $t_{i_1 \cdots i_{n-1} i_{n+1} \cdots i_N}$ that are obtained by fixing the values of $\{i_1, \cdots, i_N\} \backslash i_n$.

The mode-$n$ unfolding, also known as matricization, of an $N$th-order tensor $\mathcal{T}$ is denoted by $\mathcal{T}_{(n)} \in \mathbb{R}^{I_n \times \Pi_{j \neq n} I_j}$ and arranges the mode-$n$ fibers to be the columns of the resulting matrix $\mathcal{T}_{(n)}$ such that the mode-$n$ fiber becomes the row index and all other $(N-1)$ modes become the column indices. The tensor element $(i_1, i_2, \cdots, i_N)$ is mapped to the matrix element $(i_n, j)$, where

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

The inner product of two same-sized tensors $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $\mathcal{B} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is the sum of the product of their entries, $< \mathcal{A}, \mathcal{B} >= \sum_{i_1, \cdots i_N} a_{i_1 \cdots i_N} b_{i_1 \cdots i_N}$. The Frobenius norm of an $N$th-order $\mathcal{T}$ is defined as:

$$\|\mathcal{T}\|_F := \sqrt{\sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} t_{i_1 \cdots i_N}^2}.$$

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

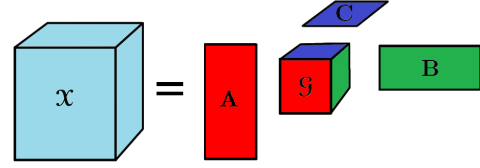$$A \otimes B = [a_{ij} B]_{mp \times nq}.$$



Figure 1: Illustration of the Tucker decomposition of a third-order tensor.

The $n$-mode matrix product of a tensor $\mathcal{T}$ with a matrix $U \in \mathbb{R}^{J \times I_n}$, denoted by $\mathcal{T} \times_n U \in \mathbb{R}^{I_1 \times \cdots I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$, is defined as:

$$(\mathcal{T} \times_n U)_{i_1 \cdots i_{n-1} j i_{n+1} \cdots i_N} = \sum_{i_n=1}^{I_n} t_{i_1 i_2 \cdots i_N} u_{j i_n}.$$

## Tensor Decomposition

We will review two popular models for tensor decomposition, i.e., the Tucker decomposition and the CANDE-COMP/PARAFAC (CP) decomposition. It is well known that finding the CP decomposition with the minimum tensor rank is a hard problem, and there is no straightforward algorithm for computing the rank for higher-order tensors (Hillar and Lim, 2013). The Tucker decomposition decomposes a given tensor $\mathcal{T}$ into a core tensor multiplied by a factor matrix along each mode as follows:

$$\mathcal{T} = \mathcal{G} \times_1 U_1 \times_2 \cdots \times_N U_N, \tag{1}$$

where $U_n \in \mathbb{R}^{I_n \times R_n}$ are the factor matrices, which can be thought of as the principal components in each mode, and the entries of the core tensor $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \cdots \times R_N}$ show the level of interaction between the different components. Since the decomposition rank $R_n$ $(n = 1, \cdots, N)$ is in general much smaller than $I_n$ $(n = 1, \cdots, N)$, the core tensor $\mathcal{G}$ can be thought of as a low-rank version of $\mathcal{T}$ (e.g., the Tucker decomposition of a third-order tensor is illustrated in Figure 1). In this sense, the storage of the Tucker decomposition form can be significantly smaller than that of the original tensor. Moreover, unlike the rank of the tensor, $R_n$, i.e., the mode-$n$ rank $(n = 1, \cdots, N)$, is the rank of the mode-$n$ unfolding, and is clearly computable. Hence, we are particularly interested in extending the Tucker decomposition for tensor analysis.

If the factor matrices of the Tucker decomposition are constrained orthogonal, the decomposition form is referred to as the higher-order singular value decomposition (HOSVD, Lathauwer et al., 2000a) or higher-order orthogonal iteration (HOOI, Lathauwer et al., 2000b), where the latter leads to the estimation of best rank-$(R_1, R_2, \ldots, R_N)$ approximations while the truncation of HOSVD may achieve a good rank-$(R_1, R_2, \ldots, R_N)$ approximation but in general not the best possible one (Lathauwer et al., 2000b). Therefore, HOOI is the most widely used tensor decomposition method (Kolda and Bader, 2009), and takes the following form

$$\min_{\mathcal{G}, U_n} \|\mathcal{T} - \mathcal{G} \times_1 U_1 \times_2 \cdots \times_N U_N\|_F^2,$$
$$\text{s.t., } U_n^T U_n = I_{R_n}, n = 1, \cdots, N. \tag{2}$$

# Convex Tensor Decomposition Method

## Convex Tensor Decomposition Model

Given a tensor $\mathcal{T}$, our goal is to find a low-rank tensor $\mathcal{X}$, in order to minimize the Frobenius norm of their difference as follows:

$$\min_{\mathcal{X}} \frac{1}{2}\|\mathcal{X} - \mathcal{T}\|_F^2. \tag{3}$$

Different from the matrix case, the low-rank tensor estimation problem (3) is in general hard to solve (Narita et al., 2012). Following the progress in tensor completion, we cast it into a (weighted) trace norm minimization problem:

$$\min_{\mathcal{X}} \sum_{n=1}^{N} \|\mathcal{X}_{(n)}\|_{\text{tr}} + \frac{\lambda}{2}\|\mathcal{X} - \mathcal{T}\|_F^2, \tag{4}$$

where $\|\mathcal{X}_{(n)}\|_{\text{tr}}$ denotes the trace norm of the unfolded matrix $\mathcal{X}_{(n)}$, i.e., the sum of its singular values, $\lambda > 0$ is a regularization parameter. For handling the unbalanced target tensor, we briefly introduce the preselected weights $\alpha_n \geq 0$ (satisfying $\sum_n \alpha_n = 1$) to the trace norm term in (4).

## Parallel Optimization Algorithm

Due to the interdependent matrix trace norm terms, the proposed tensor decomposition model (4) is very difficult to solve. Thus, we introduce some auxiliary variables $\mathcal{M}_n$ into the model (4) and reformulate it into the following equivalent form:

$$\min_{\mathcal{X},\{\mathcal{M}_n\}} \sum_{n=1}^{N} \|\mathcal{M}_{n,(n)}\|_{\text{tr}} + \frac{\lambda}{2}\|\mathcal{X} - \mathcal{T}\|_F^2, \tag{5}$$
$$\text{s.t., } \mathcal{M}_n = \mathcal{X}, \ n = 1, \cdots, N.$$

In the following, we will design a parallel alternating direction method of multipliers (ADMM) for solving the problem (5). The algorithm decomposes the original problem into smaller subproblems and solves them individually in parallel at each iteration. The parallel ADMM for the problem (5) is derived by minimizing the augmented Lagrange function $\mathcal{L}_\mu$ with respect to $(\mathcal{X}, \{\mathcal{M}_n\})$ in parallel, and then updating the multiplier tensor $\mathcal{Y}_n$. But the parallelization is likely to diverge, therefore, it is necessary to modify the common algorithm in a certain way to guarantee its convergence. Several variants of parallel ADMM have been proposed in He (2009) and Deng et al., (2013) by taking additional correction steps at every iteration. Similar to Deng et al., (2013), we add some proximal terms to each subproblems and attach a relaxation parameter $\gamma > 0$ for the update of the penalty parameter $\mu > 0$.

**Computing** $\{\mathcal{M}_n^{k+1}, \ n = 1, \cdots, N\}$**:** By keeping all other variables fixed, the optimal $\mathcal{M}_n^k$ is the solution to the following problem:

$$\min_{\mathcal{M}_n} \|\mathcal{M}_{n,(n)}\|_{\text{tr}} + \frac{\mu}{2}\|\mathcal{M}_n - \mathcal{X}^k + \mathcal{Y}_n^k/\mu\|_F^2$$
$$+ \frac{\tau_n}{2}\|\mathcal{M}_n - \mathcal{M}_n^k\|_F^2, \tag{6}$$

---

**Algorithm 1** Solving problem (5) via parallel ADMM

**Input:** Given $\mathcal{T}$, $\lambda$, and $\mu$.
**Initialize:** $\mathcal{M}_n^0 = \mathcal{X}^0 = \mathcal{Y}_n^0 = 0, \forall n \in \{1, \cdots, N\}$.
1: **while** not converged **do**
2:   **for** $n = 1, \cdots, N$ **do**
3:     Update $\mathcal{M}_n^{k+1}$ by (7),
      where $U_n S_n V_n^T = \frac{\mu\mathcal{X}_{(n)}^k - \mathcal{Y}_{(n)}^k + \tau_n\mathcal{M}_{(n)}^k}{\mu + \tau_n}$.
4:   **end for**
5:   Update $\mathcal{X}^{k+1}$ by (9).
6:   **for** $n = 1, \cdots, N$ **do**
7:     $\mathcal{Y}_n^{k+1} = \mathcal{Y}_n^k + \gamma\mu(\mathcal{M}_n^{k+1} - \mathcal{X}^{k+1})$.
8:   **end for**
9:   Check the convergence condition,
    $\|\mathcal{X}^{k+1} - \mathcal{T}\|_F < \text{Tol}$.
10: **end while**
11: $\mathcal{G} = \mathcal{X}^{k+1} \times_1 U_1^T \cdots \times_N U_N^T$.
**Output:** $\mathcal{X}^k$, $\mathcal{G}$, and $U_n$, $n = 1, \cdots, N$.

---

where $\tau_n > 0$ is a constant for the proximal term $\|\mathcal{M}_n - \mathcal{M}_n^k\|_F^2$. Following (Cai et al., 2010), we obtain a closed-form solution to the problem (6) via the following proximal operator of the trace norm:

$$\mathcal{M}_n^{k+1} = \text{refold}(\text{prox}_{\frac{1}{\mu + \tau_n}}(\frac{\mu\mathcal{X}_{(n)}^k - \mathcal{Y}_{(n)}^k + \tau_n\mathcal{M}_{(n)}^k}{\mu + \tau_n})), \tag{7}$$

where $\text{refold}(\cdot)$ denotes the refolding of the matrix into a tensor, i.e., the reverse process of unfolding.

**Computing** $\mathcal{X}^{k+1}$**:** The optimal $\mathcal{X}^{k+1}$ with all other variables fixed is the solution to the following problem:

$$\min_{\mathcal{X}} \frac{\lambda}{2}\|\mathcal{X} - \mathcal{T}\|_F^2 + \sum_{n=1}^{N} \frac{\mu}{2}\|\mathcal{M}_n^k - \mathcal{X} + \mathcal{Y}_n^k/\mu\|_F^2$$
$$+ \frac{\tau_{N+1}}{2}\|\mathcal{X} - \mathcal{X}^k\|_F^2, \tag{8}$$

where $\tau_{N+1} > 0$ is a constant for the proximal term $\|\mathcal{X} - \mathcal{X}^k\|_F^2$. Since the problem (8) is a smooth convex optimization problem, it is easy to show that the optimal solution to (8) is given by

$$\mathcal{X}^{k+1} = \frac{\sum_{n=1}^{N}(\mu\mathcal{M}_n^k + \mathcal{Y}_n^k) + \lambda\mathcal{T} + \tau_{N+1}\mathcal{X}^k}{N\mu + \lambda + \tau_{N+1}}. \tag{9}$$

Based on the description above, we develop a parallel ADMM algorithm for the convex tensor decomposition (CTD) problem (5), as outlined in **Algorithm 1**. In Algorithm 1, we use a Jacobi-type scheme to update $(N + 1)$ blocks of variables, $\{\mathcal{M}_1, \cdots, \mathcal{M}_N\}$ and $\mathcal{X}$. By the definition $f(\mathcal{M}_1, \cdots, \mathcal{M}_N) := \sum_{n=1}^{N} \|\mathcal{M}_{n,(n)}\|_{\text{tr}}$ and $g(\mathcal{X}) := \frac{\lambda}{2}\|\mathcal{X} - \mathcal{T}\|_F^2$, it is easy to verify that the problem (5) and Algorithm 1 satisfy the convergence conditions in Deng et al., (2013).

**Theorem 1** *Let* $\tau_i > \mu(\frac{N}{2-\gamma} - 1)$, $i = 1, \ldots, N + 1$. *Then the sequence* $\{\mathcal{M}_1^k, \cdots, \mathcal{M}_N^k, \mathcal{X}^k\}$ *generated by Algorithm 1 converges to an optimal solution* $\{\mathcal{M}_1^*, \cdots, \mathcal{M}_N^*, \mathcal{X}^*\}$ *of*

the problem (5). Hence, the sequence $\{\mathcal{X}^k\}$ converges to an optimal solution to the low-rank tensor decomposition problem (4).

Our convex Tucker decomposition method employs matrix trace norm regularization and uses the singular value decomposition (SVD) in Algorithm 1, which becomes a little slow or even not applicable for large-scale problems. Motivated by this, we will propose a scalable non-convex low-rank tensor decomposition method.

## Non-Convex Tensor Decomposition Method

Several researchers (Keshavan et al., 2010; Wen et al., 2012) have provided some matrix rank estimation strategies to compute some good values $(r_1, r_2, \ldots, r_N)$ for the $N$-rank of the involved tensor. Thus, we only set some relatively large integers $(R_1, R_2, \ldots, R_N)$ such that $R_n \geq r_n$, $n = 1, \cdots, N$.

**Theorem 2** *Let* $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ *with* $N$-*rank*= $(r_1, r_2, \cdots, r_N)$ *and* $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ *satisfy* $\mathcal{X} = \mathcal{G} \times_1 U_1 \cdots \times_N U_N$, *and* $U_n^T U_n = I_{R_n}$, $n = 1, 2, \cdots, N$, *then*

$$\|\mathcal{X}_{(n)}\|_{\text{tr}} = \|\mathcal{G}_{(n)}\|_{\text{tr}}, \ n = 1, 2, \cdots, N. \tag{10}$$

The proof of Theorem 2 can be found in the supplemental material. According to the theorem, it is cleat that although the core tensor $\mathcal{G}$ of size $(R_1, R_2, \cdots, R_N)$ has much smaller sizes than the original tensor $\mathcal{X}$ (usually, $R_n \ll I_n, n = 1, 2, \cdots, N$), their trace norm is identical. In other words, each unfolding $\mathcal{G}_{(n)} \in \mathbb{R}^{R_n \times \Pi_{j \neq n} R_j}$ of the core tensor $\mathcal{G}$ has much smaller sizes than that of the original tensor, $\mathcal{X}_{(n)} \in \mathbb{R}^{I_n \times \Pi_{j \neq n} I_j}$. Therefore, we use the trace norm of each unfolding of the core tensor to replace that of the original tensor.

## Generalized HOOI Model with Trace Norm Penalties

According to the analysis above, our trace norm regularized HOOI model is formulated into the following form:

$$\min_{\mathcal{G},\{U_n\}} \sum_{n=1}^{N} \|\mathcal{G}_{(n)}\|_{\text{tr}} + \frac{\lambda}{2}\|\mathcal{T} - \mathcal{G} \times_1 U_1 \cdots \times_N U_N\|_F^2,$$
$$\text{s.t., } U_n^T U_n = I_{R_n}, n = 1, 2, \cdots, N. \tag{11}$$

The core tensor trace norm regularized HOOI model (11), also called Generalized HOOI, can alleviate the SVD computational burden of large unfolded matrices involved in the convex Tucker decomposition problem (4). Furthermore, we use the trace norm regularization term in (11) to promote the robustness of the rank selection, while the original Tucker decomposition method is usually sensitive to the given ranks $(R_1, R_2, \cdots, R_N)$ (Liu et al., 2009). Due to the interdependent matrix trace norm terms, we apply the variable-splitting technique and introduce some auxiliary variables $G_n \in \mathbb{R}^{R_n \times \Pi_{j \neq n} R_j}$ into our model (11), and then reformu-

late the model (11) into the following equivalent form:

$$\min_{\mathcal{G},\{U_n, G_n\}} \sum_{n=1}^{N} \|G_n\|_{\text{tr}} + \frac{\lambda}{2}\|\mathcal{T} - \mathcal{G} \times_1 U_1 \cdots \times_N U_N\|_F^2,$$
$$\text{s.t., } G_n = \mathcal{G}_{(n)}, U_n^T U_n = I_{R_n}, n = 1, 2, \cdots, N. \tag{12}$$

## Parallel Optimization Procedure

In this part, we will also propose a parallel ADMM algorithm to solve the problem (12).

**Updating** $\{\mathcal{G}^{k+1}, U_1^{k+1}, \cdots, U_N^{k+1}\}$**:** The optimization problem with respect to $\mathcal{G}$ and $\{U_1, \cdots, U_N\}$ is formulated as follows:

$$\min_{\mathcal{G},\{U_n\}} \sum_{n=1}^{N} \frac{\mu}{2}\|\mathcal{G}_{(n)} - G_n^k + Y_n^k/\mu\|_F^2$$
$$+ \frac{\lambda}{2}\|\mathcal{T} - \mathcal{G} \times_1 U_1 \cdots \times_N U_N\|_F^2. \tag{13}$$

Following (Lathauwer et al., 2000b), it is sufficient to determine the matrices $\{U_1, \cdots, U_N\}$ for the optimization of the problem (13). For any estimate of these matrices, the optimal solution with respect to $\mathcal{G}$ is given by the following theorem.

**Theorem 3** *For given matrices* $\{U_1, \cdots, U_N\}$, *the optimal core tensor* $\mathcal{G}$ *to the optimization problem (13) is given by*

$$\mathcal{G}^{k+1} = \frac{\lambda}{\lambda + N\mu} \mathcal{T} \times_1 (U_1^k)^T \cdots \times_N (U_N^k)^T$$
$$+ \frac{\mu}{\lambda + N\mu} \sum_{n=1}^{N} \text{refold}(G_n^k - Y_n^k/\mu). \tag{14}$$

The proof of Theorem 3 can be found in the supplemental material. In the following, we design a generalized higher order orthogonal iteration scheme for solving $\{U_1, \cdots, U_N\}$ that is an alternating least squares (ALS) approach to solve the rank-$(R_1, \ldots, R_N)$ problem. Analogous with Theorem 4.2 in Lathauwer et al., (2000b), we first state that the minimization problem (13) can be formulated as follows.

**Theorem 4** *Assume a real* $N$th-order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \ldots I_N}$. *Then the minimization of the problem (13) over the matrices* $U_1, \ldots, U_N$ *having orthonormal columns is equivalent to the maximization of the following problem*

$$h(U_1, \ldots, U_N) = \langle \mathcal{T}, \mathcal{G} \times_1 U_1 \cdots \times_N U_N \rangle. \tag{15}$$

The proof of Theorem 4 can be found in the supplemental material. According to the theorem, a generalized higher-order orthogonal iteration scheme is proposed to solve the problem (15) that solves $U_n$, $n = 1, \ldots, N$ by fixing other variables $U_j, j \neq n$ in parallel. Imagine that the matrices $\{U_1, \ldots, U_{n-1}, U_{n+1}, \ldots, U_N\}$ are fixed and that the optimization problem (13) is thought of as a quadratic expression in the components of the matrix $U_n$ that is being optimized. Considering that the matrix having orthonormal columns, we have

$$h = \text{trace}((U_n^{k+1})^T \mathcal{T}_{(n)} W_n^T), \tag{16}$$

where trace($A$) denotes the trace of the matrix $A$, and

$$W_n = \mathcal{G}_{(n)}[(U_1^k)^T \cdots \otimes (U_{n-1}^k)^T \\ \otimes (U_{n+1}^k)^T \cdots \otimes (U_N^k)^T]. \tag{17}$$

This is actually the well-known orthogonal procrustes problem (Nick, 1995), whose global optimal solution is given by the singular value decomposition of $\mathcal{T}_{(n)}W_n^T$, i.e.,

$$U_n^{k+1} = \widehat{U_n}(\widehat{V_n})^T, \tag{18}$$

where $\widehat{U_n}$ and $\widehat{V_n}$ are obtained by SVD of $\mathcal{T}_{(n)}W_n^T$. Repeating the procedure above in parallel for different modes leads to an alternating least squares scheme for solving the maximization of the problem (15).

**Updating** $\{G_1^{k+1}, \cdots, G_N^{k+1}\}$**:** By keeping all other variables fixed, $G_n^{k+1}$ is updated by solving the following problem:

$$\min_{G_n} \|G_n\|_{\mathrm{tr}} + \frac{\mu}{2}\|G_n - \mathcal{G}_{(n)}^k - Y_n^k/\mu\|_F^2 + \frac{\tau_n}{2}\|G_n - G_n^k\|_F^2. \tag{19}$$

Similar to the problem (6), it is easy to obtain a closed-form solution to the problem (19):

$$G_n^{k+1} = \mathrm{prox}_{1/(\mu+\tau_n)}\left(\frac{\mu\mathcal{G}_{(n)}^k + Y_n^k + \tau_n G_n^k}{\mu + \tau_n}\right). \tag{20}$$

From (20), it is clear that only some smaller sized matrices $(\mu\mathcal{G}_{(n)}^k + Y_n^k + \tau_n G_n^k)/(\mu + \tau_n) \in \mathbb{R}^{R_n \times \Pi_{j \neq n} R_j}$ need to perform SVD. Thus, our non-convex trace norm regularized method has a significantly lower computational complexity with $O(\sum_n R_n^2 \times \Pi_{j \neq n} R_j)$, while the computational complexity of our convex algorithm for the problem (4) is $O(\sum_n I_n^2 \times \Pi_{j \neq n} I_j)$ at each iteration.

Based on the analysis above, we develop a parallel ADMM algorithm for solving the low-rank non-convex tensor decomposition (NCTD) problem (11), as outlined in **Algorithm 2**. Our algorithm is essentially a Jacobi-type scheme of ADMM, and is well suited for parallel and distributed computing and hence is particularly attractive for solving certain large-scale problems. Moreover, the update strategy of Gauss-Seidel version of ADMM is easily implemented. This algorithm can be accelerated by adaptively changing $\mu$. An efficient strategy (Lin et al., 2011) is to let $\mu^0 = \mu$ (initialized in Algorithm 1 and Algorithm 2) and increase $\mu^k$ iteratively by $\mu^{k+1} = \rho\mu^k$, where $\rho \in (1.0, 1.1]$ in general and $\mu^0$ is a small constant. Moreover, the stability and efficiency of our NCTD method can be validated in the experimental section.

## Complexity Analysis

We now discuss the time complexity of our NCTD algorithm. For the problem (11), the main running time of our NCTD algorithm is consumed by performing SVD for the proximal operator and some multiplications. The time complexity of performing the proximal operator in (20) is $O_1 := O(\sum R_n^2 \Pi_{j \neq n} R_j)$. The time complexity of some multiplication operators is $O_2 := O(\sum I_n R_n \Pi_{j \neq n} I_j + \sum R_n \Pi_{j \neq n} I_j R_j)$ and $O_3 := O(\sum R_n^2 I_n)$. Thus, the total time complexity of our NCTD method is $O(T(O_1 + O_2 + O_3))$, where $T$ is the number of iterations.

---

**Algorithm 2** Solving problem (11) via Parallel ADMM

---

**Input:** $\mathcal{T}$, the tensor ranks $(R_1, R_2, \cdots, R_N)$, and $\lambda$.
**Initialize:** $Y_n^0 = 0$, $G_n^0 = 0$, $U_n^0 = \mathrm{rand}(I_n, R_n)$, $\mu^0 = 10^{-4}$, $\mu_{max} = 10^{10}$, $\rho = 1.05$, and $\mathrm{Tol} = 10^{-5}$.
1: **while** not converged **do**
2:   Update $\mathcal{G}^{k+1}$ by (14).
3:   **for** $n = 1, \cdots, N$ **do**
4:     Update $U_n^{k+1}$ by (18);
5:     Update $G_n^{k+1}$ by (20);
6:     Update the multiplier $Y_n^{k+1}$ by
        $Y_n^{k+1} = Y_n^k + \gamma\mu^k(\mathcal{G}_{(n)}^{k+1} - G_n^{k+1})$.
7:   **end for**
8:   Update $\mu^{k+1}$ by $\mu^{k+1} = \min(\rho\mu^k, \mu_{max})$.
9:   Check the convergence condition,
      $\max\{\|\mathcal{G}_{(n)}^{k+1} - G_n^{k+1}\|_F, \ n = 1, \cdots, N\} < \mathrm{Tol}$.
10: **end while**
**Output:** $\mathcal{G}$ and $U_n$, $n = 1, \cdots, N$.

---

## Experimental Results

In this section, we evaluate both the effectiveness and efficiency of our methods for solving tensor decomposition problems using both synthetic and real-world data. All experiments were performed on an Intel(R) Core (TM) i5-4570 (3.20 GHz) PC running Windows 7 with 8GB main memory.

### Synthetic Data

In this part, we generated a low-rank $N$th-order tensor $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \cdots \times I_N}$, which was used as the ground truth data. The tensor data followed the Tucker model, i.e., $\mathcal{T} = \mathcal{G} \times_1 U_1 \cdots \times_N U_N$, where the core tensor $\mathcal{G} \in \mathbb{R}^{r \times r \cdots \times r}$ and the factor matrices $U_n$ were generated with i.i.d. standard Gaussian entries. The order of the tensors varied from three to four, and the rank $r$ was set to $\{5, 10, 20\}$. Finally, we decomposed the input tensor $\mathcal{T} + \delta\Delta$ by our CTD and NCTD methods and the state-of-the-art algorithms including HOSVD (Vannieuwenhoven et al., 2012), HOOI (Lathauwer et al., 2000b), Mixture (Tomioka et al., 2013) and ADMM (Gandy et al., 2011), where $\Delta$ is a noise tensor with independent normally distributed entries, and $\delta$ is set to 0.02.

We set the tensor ranks $R_n = \lfloor 1.2r \rfloor$, $n = 1, \ldots, N$ and $\mathrm{Tol} = 10^{-5}$ for all these algorithms. We set the regularization parameter $\lambda = 100$ for our methods. Other parameters of Mixture, ADMM, HOSVD and HOOI are set to their default values. The relative square error (RSE) of the estimated tensor $\mathcal{X}$ is given by $\mathrm{RSE} = \|\mathcal{X} - \mathcal{T}\|_F / \|\mathcal{T}\|_F$. The average experimental results (RSE and time cost) of 50 independent runs are shown in Table 1, where the order of tensor data varies from three to four. From the results shown in Table 1, we can see that our methods can yield much more accurate solutions, and outperform their individual competitors, HOSVD, HOOI, Mixture and ADMM, in terms of both RSE and efficiency.

A phase transition plot uses greyscale colors to depict how likely a certain kind of low-rank tensors can be recovered by these algorithms for a range of different given ranks and noise variances $\delta$. Phase transition plots are important means

Table 1: Performance comparison of estimation accuracy (RSE) and running time (seconds) on the synthetic data:

(a) Tensor size: $200 \times 200 \times 200$

| | HOSVD | | HOOI | | Mixture | | ADMM | | CTD | | NCTD | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rank | RSE | Time | RSE | Time | RSE | Time | RSE | Time | RSE | Time | RSE | Time |
| 5 | 3.27e-02 | 19.46 | 3.27e-02 | 570.85 | 1.46e-02 | 4792.74 | 1.47e-02 | 2946.27 | 6.52e-03 | 88.74 | 6.52e-03 | 16.50 |
| 10 | 3.33e-02 | 20.36 | 3.32e-02 | 468.65 | 1.47e-02 | 4567.61 | 1.48e-02 | 2512.85 | 6.59e-03 | 85.62 | 6.60e-03 | 15.25 |
| 20 | 3.34e-02 | 21.02 | 3.33e-02 | 380.87 | 1.47e-02 | 4235.65 | 1.48e-02 | 2230.64 | 6.61e-03 | 72.69 | 6.62e-03 | 13.74 |

(b) Tensor size: $60 \times 60 \times 60 \times 60$

| | HOSVD | | HOOI | | Mixture | | ADMM | | CTD | | NCTD | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Rank | RSE | Time | RSE | Time | RSE | Time | RSE | Time | RSE | Time | RSE | Time |
| 5 | 3.89e-02 | 25.14 | 3.87e-02 | 964.26 | 1.63e-02 | 5175.79 | 1.64e-02 | 3378.06 | 7.16e-03 | 105.01 | 7.17e-03 | 20.35 |
| 10 | 3.91e-02 | 23.94 | 3.90e-02 | 607.17 | 1.64e-02 | 4971.15 | 1.64e-02 | 3280.32 | 6.30e-03 | 101.30 | 6.30e-03 | 18.89 |
| 20 | 3.92e-02 | 22.81 | 3.91e-02 | 415.43 | 1.65e-02 | 4773.66 | 1.65e-02 | 3031.54 | 5.67e-03 | 98.45 | 5.67e-03 | 17.63 |

to compare the performance of different tensor estimation methods. If the relative error RSE $\leq 10^{-2}$, the estimation is regarded as successful. Figure 2 depicts the phase transition plots of HOSVD, HOOI, CTD and NCTD on the third-order tensor data with the rank $r = 10$, where the given tensor ranks $R_n$, $n = 1, 2, 3$ varied from 10 to 50 with increment 4 and $\delta$ from 0 to 0.05 with increment 0.005. For each setting, 50 independent trials were run. From the experimental results shown in Figure 2, we can see that CTD and NCTD perform significantly better than HOSVD and HOOI.



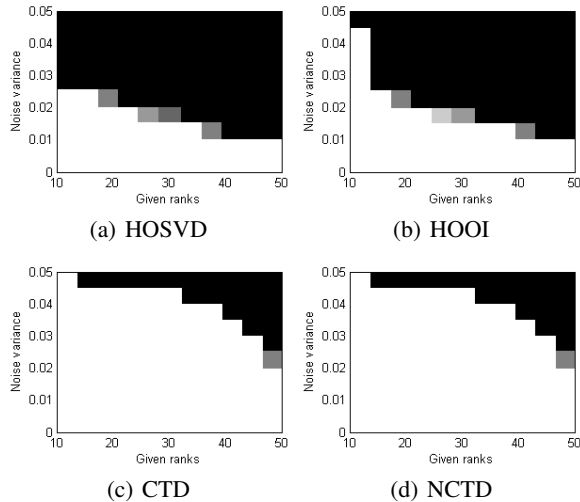(a) HOSVD    (b) HOOI

(c) CTD    (d) NCTD

Figure 2: Phase transition plots for different methods on third-order low-rank tensors with Gaussian random noise, where white denotes perfect estimation in all experiments, and black denotes failure for all experiments.

## MRI Data

This part compares our CTD and NCTD methods, HOSVD and HOOI on a $181 \times 217 \times 181$ brain MRI data used in (Liu et al., 2009). This data set is approximately low-rank: for the three mode unfoldings, the numbers of singular values larger

than 1% of the largest one are 17, 21, and 17, respectively. Figure 3 shows the average relative errors and running times of ten independent trials for each setting of the given ranks. From the results, we see that our CTD and NCTD methods consistently attain much lower relative errors than those by HOSVD and HOOI. Moreover, our NCTD method is usually faster than the other methods.
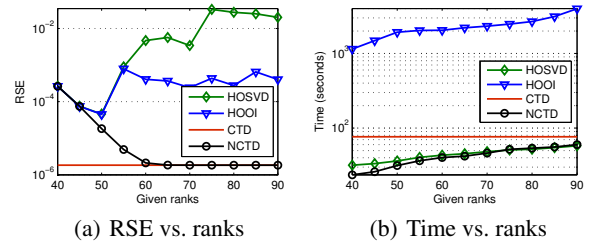


(a) RSE vs. ranks    (b) Time vs. ranks

Figure 3: Comparison of HOSVD, HOOI, CTD and NCTD in terms of estimation accuracy (RSE) and time cost (in the logarithmic scale) on the brain MRI data set.

## Conclusions

In this paper we first proposed a convex trace norm regularized tensor decomposition method, which can automatically determine the number of factors in each mode through our optimization scheme. In addition, by considering the low-rank structure of input tensors, we analyzed the equivalence relationship of the trace norm between a low-rank tensor and its core tensor. Then, we cast the non-convex tensor decomposition model into a weighted combination of multiple much-smaller-scale matrix trace norm minimization. Finally, we developed two efficient parallel ADMM algorithms for solving the proposed problems. Convincing experimental results demonstrate that our regularized formulation is reasonable, and our methods are robust to noise or outliers. Moreover, our tensor decomposition methods can handle some tensor recovery problems, such as tensor completion, and low-rank and sparse tensor decomposition.

## References

Cai, J.; Candes, E.; and Shen, Z. 2010. A singular value thresholding algorithm for matrix completion. *SIAM J. Optim.* 20(4): 1956–1982.

De Lathauwer, L., and Vandewalle, J. 2004. Dimensionality reduction in higher-order signal processing and rank-(r1, r2, . . . , rn) reduction in multilinear algebra. *Linear. Algebra. Appl.* 391: 31–55.

Deng, W.; Lai, M.; and Yin, W. 2013. On the $o(1/k)$ convergence and parallelization of the alternating direction method of multipliers. ArXiv:1312.3040.

Gandy, S.; Recht, B.; and Yamada. I. 2011. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problem*, 27(2) 025010.

Goldfarb, D., and Qin, Z. 2014. Robust low-rank tensor recovery: models and algorithms. *SIAM J. Matrix Anal. Appl.*

Harshman, R.A. 1970. Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multimodal factor analysis. *UCLA working papers in phonetics* 16: 1–84.

He, B. 2009. Parallel splitting augmented Lagrangian methods for monotone structured variational inequalities. *Comput. Optim. Appl.* 42: 195–212.

Hillar, C.J., and Lim, L.H. 2013. Most tensor problems are NP hard. *J. ACM*.

Keshavan, R.H.; Montanari, A.; and Oh, S. 2010. Matrix completion from a few entries. *IEEE Trans. Inform. Theory* 56(6): 2980–2998.

Kolda, T.G., and Bader, B.W. 2009. Tensor decompositions and applications. *SIAM Review* 51(3): 455–500.

Lathauwer, L.; Moor, B.; and Vandewalle, J. 2000a. A multilinear singular value decomposition. *SIAM J. Matrix. Anal. Appl.* 21(4): 1253–1278.

Lathauwer, L.; Moor, B.; and Vandewalle, J. 2000b. On the best rank-1 and rank-$(R_1, R_2, \cdots, R_N)$ approximation of high-order tensors. *SIAM J. Matrix. Anal. Appl.* 21(4): 1324–1342.

Lin, Z.; Liu, R.; and Su, Z. 2011. Linearized alternating direction method with adaptive penalty for low-rank representation. In *NIPS*, 612–620.

Liu, J.; Musialski, P.; Wonka, P.; and Ye, J. 2009. Tensor completion for estimating missing values in visual data. In *ICCV* 2114–2121.

Liu, Y.; Shang, F.; Cheng, H.; Cheng, J.; and Tong, H. 2014. Factor matrix trace norm minimization for low-rank tensor completion. In *SDM*.

Morup, M. 2011. Applications of tensors (multiway array) factorizations and decompositions in data mining. *WIREs: Data Min. Knowl. Disc.* 1(1): 24–40.

Narita, A.; Hayashi, K.; Tomioka, R.; and Kashima, H. 2012. Tensor factorization using auxiliary information. *Data Min. Knowl. Disc.* 25(2): 298–324.

Nick, H. 1995. Matrix procrustes problems.

Shi, Z.; Han, J.; Zheng, T.; and Li, J. 2013. Guarantees of augmented trace norm models in tensor recovery. In *IJCAI* 1670–1676.

Shang, F.; Jiao, L.C.; Shi, J.; and Chai, J. 2011. Robust positive semidefinite L-Isomap ensemble. *Pattern Recogn. Lett.* 32(4): 640–649.

Signoretto, M.; Dinh, Q.T.; Lathauwer, L.D.; and Suykens, J. 2014. Learning with tensors: a framework based on convex optimization and spectral regulation. *Mach. Learn.*

Tomioka, R.; Suzuki, T.; Hayashi, K.; and Kashima, H. 2011. Statistical Performance of Convex Tensor Decomposition. In *NIPS* 972–980.

Tomioka, R., and Suzuki, T. 2013. Convex tensor decomposition via structured Schatten norm regularization. In *NIPS* 1331–1339.

Tucker, L.R. 1966. Some mathematical notes on three-mode factor analysis. *Psychometrika* 31: 279–311.

Vannieuwenhoven, N.; Vandebril, R.; and Meerbergen, K. 2012. A new truncation strategy for the higher-order singular value decomposition. *SIAM J. Sci. Comput.* 34(2): 1027–1052.

Wen, Z.; Yin, W.; and Zhang, Y. 2012. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Math. Prog. Comp.* 4(4): 333–361.

Yilmaz, Y.K.; Cemgil, A.T.; and Simsekli, U. 2011. Generalized coupled tensor factorization. In *NIPS* 2151–2159.