This is a repository copy of *Tensor Rank Estimation and Completion via CP-based Nuclear Norm*.

White Rose Research Online URL for this paper:
http://eprints.whiterose.ac.uk/120408/

Version: Accepted Version

**Proceedings Paper:**

https://doi.org/10.1145/3132847.3132945

---

**Reuse**
Unless indicated otherwise, fulltext items are protected by copyright with all rights reserved. The copyright exception in section 29 of the Copyright, Designs and Patents Act 1988 allows the making of a single copy solely for the purpose of non-commercial research or private study within the limits of fair dealing. The publisher or other rights-holder may allow further reproduction and re-use of this version - refer to the White Rose Research Online record for this item. Where records identify the publisher as the copyright holder, users can verify any specific terms of use on the publisher's website.

**Takedown**
If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.
Tensor Rank Estimation and Completion via CP-based Nuclear Norm

Qiguan Shi
Department of Computer Science,
Hong Kong Baptist University
Hong Kong, China
csqqs@comp.hkbu.edu.hk

Haiping Lu
Department of Computer Science,
University of Sheffield
Sheffield, U.K.
h.lu@sheffield.ac.uk

Yiu-ming Cheung
Department of Computer Science,
Hong Kong Baptist University
Hong Kong, China
ymc@comp.hkbu.edu.hk

ABSTRACT

Tensor completion (TC) is a challenging problem of recovering missing entries of a tensor from its partial observation. One main TC approach is based on CP/Tucker decomposition. However, this approach often requires the determination of a tensor rank a priori. This rank estimation problem is difficult in practice. Several Bayesian solutions have been proposed but they often under/overestimate the tensor rank while being slow. To address this problem of rank estimation with missing entries, we view the weight vector of the orthogonal CP decomposition of a tensor to be analogous to the vector of singular values of a matrix. Subsequently, we define a new CP-based tensor nuclear norm as the $L_1$-norm of this weight vector. We then propose Tensor Rank Estimation based on $L_1$-regularized orthogonal CP decomposition (TREL1) for both CP-rank and Tucker-rank. Specifically, we incorporate a regularization with CP-based tensor nuclear norm when minimizing the reconstruction error in TC to automatically determine the rank of an incomplete tensor. Experimental results on both synthetic and real data show that: 1) Given sufficient observed entries, TREL1 can estimate the true rank (both CP-rank and Tucker-rank) of incomplete tensors well; 2) The rank estimated by TREL1 can consistently improve recovery accuracy of decomposition-based TC methods; 3) TREL1 is not sensitive to its parameters in general and more efficient than existing rank estimation methods.

KEYWORDS

Tensor Rank Estimation; CP-based Tensor Nuclear Norm; CP Decomposition; Tensor Completion

1 INTRODUCTION

Tensors are generalization of vectors (first-order tensors) and matrices (second-order tensors). They are ubiquitous (e.g., multichannel EEGs, images, videos, and social networks) and attract increasing interests [16]. Tensor completion, a task of recovering the missing entries based on partially observed entries, has drawn much attention recently in many applications of machine learning [24, 28, 32, 34] and data mining [29–31, 36].

One popular approach to solving tensor completion problems is tensor nuclear norm minimization, which is extended from matrix [5] to tensor case as a convex surrogate for rank minimization [18]. Although the nuclear norm approximation leads to good tensor completion performance under typical conditions [7, 11, 19], there is no theoretical guarantee that it is the tightest convex envelope of a tensor rank. Moreover, this approach is not efficient on large-scale tensors due to the heavy computation of singular value decomposition (SVD).

Another popular approach is based on tensor decompositions including CANDECOMP/PARAFAC (CP) [6, 12]) and Tucker decomposition [35], which is more promising for large-scale data. These two main decompositions lead to two common definitions for tensor rank: CP-rank and Tucker-rank respectively. This approach often requires a tensor rank as input. For example, a CP decomposition with weighted optimization method (CP-WOPT) [1] and an alternating minimization algorithm for tensors with a (fixed) low-rank orthogonal CP decomposition (TenALS) [14] can obtain good completion results for data with missing values under typical conditions. However, they need to manually choose a CP-rank as input, which is quite challenging because estimating the CP-rank is NP-hard [13], especially given incomplete information. On the other hand, by enforcing orthogonality into Tucker model, a generalized higher-order orthogonal iteration method (gHOI) [20] is developed to efficiently solve tensor completion problem, where the Tucker-rank for their model is obtained via a heuristic rank-increasing scheme. Furthermore, a simple Tucker decomposition-based approach (Tucker-WOPT) [10] fails to recover missing data accurately if the pre-specified rank is smaller than the true rank. Most recently, a Riemannian manifold optimization method (FRTC) [15] achieves good recovery performance on large-scale tensors, while still requiring a good rank value to be pre-specified. Moreover, its time cost increases exponentially with increasing input Tucker-rank.

Some studies attempt to estimate the CP/Tucker-rank of incomplete tensors automatically. Several Bayesian models have been proposed to automatically determine the CP-rank [3, 27, 42, 43]. For example, the CP rank of an incomplete tensor can be inferred by employing a multiplicative gamma process prior in [27], where the inference is performed by Gibbs sampler with slow convergence. Most recently, a Bayesian robust tensor factorization (BRTF) [43] employs a fully Bayesian generative model for automatic CP-rank estimation. However, BRTF often under/over-estimates the true rank of incomplete tensors and has high computational cost.

To automatically estimate the Tucker-rank, an automatic relevance determination (ARD) algorithm is applied for sparse Tucker decomposition (ARD-Tucker) [23]. ARD is a hierarchical Bayesian
Tucker is not applicable to incomplete tensor data and its efficiency is quite low. Most recently, a robust Tucker-rank estimation method using Bayesian information criteria is proposed [39], while also only applicable to complete tensors.

In this paper, we view the weight vector of the orthogonal CP decomposition of a tensor as analogous to the vector of singular values of the SVD of a matrix. We then define a simple CP-based tensor nuclear norm as the $L_1$ norm of this weight vector. Based on this new tensor norm, we propose Tensor Rank Estimation based on $L_1$-regularized orthogonal CP decomposition, denoted as TREL1. TREL1 can automatically determine both CP-rank and Tucker-rank accurately given sufficient observed entries by removing the zero entries of the weight vector after optimization. We solve the optimization problem by block coordinate descent, where we optimize a block of variables while fixing the other blocks and update one variable while fixing the other variables in each block. In a nutshell, our contributions are fourfold:

- We propose TREL1 to automatically estimate the CP-rank of an incomplete tensor via a newly defined CP-based tensor nuclear norm.
- We automatically estimate the Tucker-rank in each mode by degenerating TREL1 to matrix case and applying it on the unfolded matrices of an incomplete tensor.
- We develop an efficient block coordinate descent algorithm for model optimization.
- We carry out extensive experiments to show that TREL1 is not sensitive to its parameters in general and more efficient than existing tensor rank estimation methods, and using TREL1 for rank estimation can improve the recovery accuracy of the state-of-the-art decomposition-based tensor completion methods.

This paper is organized as follows. We review preliminaries and backgrounds in Section 2. In Section 3, we define a CP-based tensor nuclear norm and propose two rank estimation methods for both CP-rank and Tucker-rank estimation. We report empirical results in Section 4, and conclude this paper in Section 5.

## 2 PRELIMINARIES AND BACKGROUNDS

We first review the preliminaries and backgrounds [16, 22].

### 2.1 Notations and Operations

The number of dimensions of a tensor is the order and each dimension is a mode of it. A vector is denoted by a bold lower-case letter $\mathbf{x} \in \mathbb{R}^l$ and a matrix is denoted by a bold capital letter $\mathbf{X} \in \mathbb{R}^{I \times J}$. A higher-order ($N \geq 3$) tensor is denoted by a calligraphic letter $\mathbf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$. The $i$th entry of a vector $\mathbf{a} \in \mathbb{R}^l$ is denoted by $a(i)$, and the $(i,j)$th entry of a matrix $\mathbf{X} \in \mathbb{R}^{I \times J}$ is denoted by $X(i,j)$. The $(i_1, \cdots , i_N)$th entry of an $N$th-order tensor $\mathbf{X}$ is denoted by $X(i_1, \cdots , i_N)$, where $i_n \in \{1, \cdots , I_n\}$ and $n \in \{1, \cdots , N\}$. The Frobenius norm of a tensor $\mathbf{X}$ is defined by $\|X\|_F = (\sum_l X(l,l)^2)^{1/2}$. $\Omega \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is a binary index set: $\Omega(i_1, \cdots , i_N) = 1$ if $X(i_1, \cdots , i_N)$ is observed, and $\Omega(i_1, \cdots , i_N) = 0$ otherwise. $\mathcal{P}_\Omega$ is the associated sampling operator which acquires only the entries indexed by $\Omega$:

$$ (\mathcal{P}_\Omega(X))(i_1, \cdots , i_N) = \begin{cases} X(i_1, \cdots , i_N), & \text{if } (i_1, \cdots , i_N) \in \Omega \\ 0, & \text{if } (i_1, \cdots , i_N) \in \Omega^c \end{cases} $$

where $\Omega^c$ is the complement of $\Omega$. We have $\mathcal{P}_\Omega(X) + \mathcal{P}_{\Omega^c}(X) = X$.

**Definition 1. Mode-$n$ Product.** A mode-$n$ product between a tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and a matrix/vector $\mathbf{U} \in \mathbb{R}^{I_n \times J}$ is denoted by $\mathbf{Y} = \mathbf{X} \times_n \mathbf{U}^\top \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J \times I_{n+1} \times \cdots \times I_N}$, with entries given by $Y_{i_1, \cdots \i_{n-1}, j, i_{n+1}, \cdots , i_N} = \sum_{i_n} X_{i_1, \cdots , i_{n-1}, i_n, i_{n+1}, \cdots , i_N} U_{j, i_n}$, and we have $Y(n) = \mathbf{U}^\top \mathbf{X}(n)$ [22].

**Definition 2. Mode-$n$ Unfolding.** Unfolding, a.k.a., matricization or flattening, is the process of reordering the elements of a tensor into matrices along each mode [16]. A mode-$n$ unfolding matrix of a tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is denoted as $\mathbf{X}(n) \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times I_{n+1} \times \cdots \times I_N}$.

### 2.2 CP and Tucker Decomposition

#### 2.2.1 Tucker Decomposition and Tucker-rank.

A tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is represented as a core tensor with factor matrices in Tucker decomposition model [16]:

$$ \mathbf{X} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_N \mathbf{U}^{(N)}, $$

where $\{U^{(n)}\} \in \mathbb{R}^{I_n \times R_n}, n = 1, 2, \cdots , N$, and $R_n < I_n$ are factor matrices with orthonormal columns and $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ is the core tensor with smaller dimension. The Tucker-rank of an $N$th-order tensor $\mathbf{X}$ is an $N$-dimensional vector, denoted as $r = (R_1, \cdots , R_N)$, whose $n$-th entry $R_n$ is the rank of the mode-$n$ unfolded matrix $\mathbf{X}(n)$ of $\mathbf{X}$. $R_n$ is the mode-$n$ rank. Figure 1 illustrates this decomposition.

![Figure 1: The Tucker decomposition of a third-order tensor $\mathbf{X}$, where the core tensor $\mathcal{W}$ is a super-diagonal tensor.](image)

#### 2.2.2 Tucker Decomposition and Tucker-rank.

A tensor $\mathbf{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is represented as a core tensor with factor matrices in Tucker decomposition model [16]:

$$ \mathbf{X} = \mathcal{G} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_N \mathbf{U}^{(N)}, $$

where $\{U^{(n)}\} \in \mathbb{R}^{I_n \times R_n}, n = 1, 2, \cdots , N$, and $R_n < I_n$ are factor matrices with orthonormal columns and $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ is the core tensor with smaller dimension. The Tucker-rank of an $N$th-order tensor $\mathbf{X}$ is an $N$-dimensional vector, denoted as $r = (R_1, \cdots , R_N)$, whose $n$-th entry $R_n$ is the rank of the mode-$n$ unfolded matrix $\mathbf{X}(n)$ of $\mathbf{X}$. $R_n$ is the mode-$n$ rank. Figure 1 illustrates this decomposition.

![Figure 2: The CP decomposition of a third-order tensor $\mathbf{X}$, where the core tensor $\mathcal{W}$ is a super-diagonal tensor.](image)
2.2.2 CP Decomposition and CP-rank. CP decomposition decomposes a tensor $X \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ as the weighted summation of a set of rank-one tensors:

$$X = \sum_{r=1}^{R} w_r u_r^{(1)} \circ \cdots \circ u_r^{(N)} = \mathcal{W} \cdot \mathcal{U}^T,$$

where each $u_r^{(n)}$, $n = 1, \ldots, N$ is a unit vector with the weight absorbed into the weight vector $w = [w_1, \cdots, w_r, \cdots, w_R]^T \in \mathbb{R}^R$, and $\circ$ denotes the outer product [16]. Figure 2 shows that CP decomposition is also could be reformulated as the Tucker decomposition where the core tensor $\mathcal{W}$ is a super-diagonal tensor, i.e., $\mathcal{W}(r, \cdots, r) = w_r$. $R$ is CP-rank as the minimum number of rank-one components.

3 PROPOSED TENSOR RANK ESTIMATION METHODS

This section presents new Tensor Rank Estimation methods based on $L_1$-regularized orthogonal CP decomposition, namely, TREL1. For simpler notations, we consider third order tensors $X \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, only while our methods generalize easily to higher-order tensors.

Orthogonal CP decomposition. In this paper, we consider the orthogonal CP decomposition, i.e., we enforce $u_r^{(n)^T} u_q^{(n)} = 0$ for $p \neq q$, and $u_p^{(n)^T} u_q^{(n)} = 1$ otherwise. There are two motivations:

1. CP decomposition can be viewed as a generalization of SVD to tensors [8]. It seems natural to inherit the orthogonality of SVD in CP decomposition.

2. Although orthogonality is considered unnecessary in general or even impossible in certain cases in exact CP decomposition [4, 9, 40], some recent studies show that imposing orthogonality in the CP model can turn non-unique tensor decomposition into a unique one with guaranteed optimality [2, 14, 17, 40].

Tensor decomposition with missing data is more challenging than that with complete data in traditional problems. Furthermore, it is important to estimate a good rank from an incomplete tensor for accurate tensor completion. Therefore, we believe incorporating orthogonality into the CP model can help us determine the rank and further recover the tensor in the context of tensor completion. Our empirical studies to be presented later will show that the orthogonality constraint indeed gives better tensor rank estimation and completion results. Furthermore, we view the weight vector $w$ of the orthogonal CP decomposition of a tensor $X$ to be analogous to the vector of singular values of a matrix.

**Definition 3.** The CP-based Tensor Nuclear Norm of a tensor $X$ is defined as the $L_1$ norm of the weight vector $w$ of its orthogonal CP decomposition: $\|X\|_{CP} = \|w\|_1$.\footnote{For easy reading, we use $\|w\|_1$ instead of $\|X\|_{CP}$ below.}

In TREL1, we incorporate a regularization of CP-based tensor nuclear norm while minimizing the reconstruction error to obtain the estimated rank of an incomplete tensor and a low-rank recovery.

Thus, our objective function is:

$$\min_{X, w, \{u_r^{(n)}\}_R} \lambda \|w\|_1 + \frac{1}{2} \|X - \sum_{r=1}^{R} w_r u_r^{(1)} \circ \cdots \circ u_r^{(N)}\|_F^2,$$

s.t. $\mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(T), u_r^{(n)^T} u_q^{(n)} = 0, q = 1 \cdots r - 1, r = 1 \cdots R,$

where $T \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is the given incomplete tensor with observed entries in $\Omega, w = [w_1, \cdots, w_r, \cdots, w_R]^T$ is the weight vector and $R$ is the CP-rank of $X$. $\lambda$ is a regularization parameter.

3.1 Derivation of TREL1 by BCD

We employ the Block Coordinate Descent (BCD) (a.k.a., alternating minimization [14]) method for optimization. We divide the target variables into $R + 1$ blocks: $\{w, u_r^{(1)}, u_r^{(2)}, u_r^{(3)}, \cdots, w_r, u_r^{(1)}, u_r^{(2)}, u_r^{(3)}\}, \cdots, \{w_R, u_R^{(1)}, u_R^{(2)}, u_R^{(3)}\}, X$. We optimize a block of variables while fixing the other blocks, and update one variable while fixing the other variables in each group. After updating the $R + 1$ blocks, we finally determine the tensor rank.

The Lagrangian function of (4) with respect to the $r$-th block $\{w_r, u_r^{(1)}, u_r^{(2)}, u_r^{(3)}\}$ is:

$$\mathcal{L}_{w_r, \{u_r^{(n)}\}} = \lambda \|w_r\|_1 + \frac{1}{2} \|X_r - w_r u_r^{(1)} \circ \cdots \circ u_r^{(N)}\|_F^2,$$

s.t. $u_r^{(n)^T} u_q^{(n)} = 0, q = 1 \cdots r - 1, r = 1 \cdots R,$

where $X_r = X - \sum_{q=1}^{r-1} w_q u_q^{(1)} \circ \cdots \circ u_q^{(N)}$ is the residual of the approximation. We use Lagrange multipliers to transform (5) to include all the constraints as:

$$\mathcal{L}_{w_r, \{u_r^{(n)}\}} = \lambda \|w_r\|_1 + \frac{1}{2} \|X_r - w_r u_r^{(1)} \circ \cdots \circ u_r^{(N)}\|_F^2,$$

- $\gamma (u_r^{(n)^T} u_r^{(n)} - 1) - \sum_{q=1}^{r-1} \mu_q u_r^{(n)^T} u_q^{(n)},$

where $\gamma$ and $\{|\mu_q|\}_{q=1}^{r-1}$ are the Lagrange multipliers.

3.1.1 Update $u_r^{(n)}$. The function (6) with respect to $u_r^{(1)}$ is,

$$\mathcal{L}_{u_r^{(1)}} = \frac{1}{2} \|X_r - w_r u_r^{(1)} \circ \cdots \circ u_r^{(N)}\|_F^2,$$

- $\gamma (u_r^{(1)^T} u_r^{(1)} - 1) - \sum_{q=1}^{r-1} \mu_q u_r^{(1)^T} u_q^{(1)},$

Then we set the partial derivative of $\mathcal{L}_{u_r^{(1)}}$ with respect to $u_r^{(1)}$ to zero and eliminate the Lagrange multipliers, and get:

$$u_r^{(1)} = \frac{X_r \cdot u_r^{(2)} \cdot u_r^{(3)} / \gamma}{w_r},$$

- $\left(\sum_{q=1}^{r-1} u_q^{(1)^T} (X_r \cdot u_r^{(2)} \cdot u_r^{(3)} / \gamma) / w_r\right),$ \footnote{For easy reading, we use $\|w\|_1$ instead of $\|X\|_{CP}$ below.}
and normalize $u_r^{(3)} = \frac{u_r^{(3)}}{|u_r^{(3)}|_2}$. Note that we only update the blocks with non-zero weights. Similarly, we update $u_r^{(2)}$ by

$$u_r^{(2)} = (X_r \times_1 u_r^{(1)} \times_3 u_r^{(3)})/w_r - \left( \sum_{q=1}^{r-1} u_q^{(2)} \times_1 (X_r \times_1 u_r^{(1)} \times_3 u_r^{(3)}) u_q^{(2)} \right)/w_r,$$

and normalize $u_r^{(2)} = \frac{u_r^{(2)}}{|u_r^{(2)}|_2}$, and update $u_r^{(3)}$ by

$$u_r^{(3)} = (X_r \times_1 u_r^{(1)} \times_2 u_r^{(2)})/w_r - \left( \sum_{q=1}^{r-1} u_q^{(3)} \times_1 (X_r \times_1 u_r^{(1)} \times_2 u_r^{(2)}) u_q^{(3)} \right)/w_r,$$

and normalize $u_r^{(3)} = \frac{u_r^{(3)}}{|u_r^{(3)}|_2}$.

### 3.1.2 Update $w_r$. The function (6) with respect to $w_r$ is:

$$L_{w_r} = \lambda |w_r|_1 + \frac{1}{2} \|X_r - w_r u_r^{(1)} \circ u_r^{(2)} \circ u_r^{(3)}\|_F^2.$$  

Then we set the partial derivative $\frac{\partial L_{w_r}}{\partial w_r}$ to zero and obtain,

$$w_r = X_r \times_1 u_r^{(1)} \times_2 u_r^{(2)} \times_3 u_r^{(3)} - \frac{\lambda |w_r|_1}{|w_r|}.$$

Based on the soft thresholding algorithm [25] for $L_1$ regularization, we update $w_r$ in (12) by:

$$w_r = shrink_\lambda((X_r \times_1 u_r^{(1)} \circ u_r^{(2)} \circ u_r^{(3)}),$$.  

Algorithm 2 Tucker-rank Estimation Based on $L_1$-Regularized Orthogonal CP Decomposition (TREL1$_{Tucker}$)

1: **Input:** $\mathcal{P}_\Omega(T)$, $\Omega$, $\lambda$, initial Tucker-rank $\hat{R}$, maximum iterations $K$, and stopping tolerance tol.
2: **Initialization:** Set $Z = \text{zeros}(I_1, I_2, I_3)$, $\mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(T)$, $\mathcal{P}_\Omega(X) = 0$; Initialize $[u_r^{(1)}; u_r^{(2)}; u_r^{(3)}]_{r=1}^\hat{R} = \hat{X}$ by RTPM [2].
3: for $i = 1, \ldots, 3$ do
4:  $X_i = \hat{X}$;
5:  for $r = 1, \ldots, \hat{R}$ do
6:   if $w_r \neq 0$ then
7:     $(X_r \times_1 u_r^{(1)} \times_3 u_r^{(3)})/w_r$
8:   end if
9:  end for
10: Update $X$: Set $Z = X - X_r$ and update the missing entries by $\mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(Z)$.
11: If $|w_{k+1} - w_k|_2/\|w_{k+1}\|_2 < tol$, break; otherwise, continue.
12: end for
13: **Output:** $r = [R_1, R_2, R_3]$.

where $shrink$ is the soft thresholding operator [25], and we denote $S = \langle X_r, u_r^{(1)} \circ u_r^{(2)} \circ u_r^{(3)} \rangle$.

$$min_X \frac{1}{2} \|X - \sum_{r=1}^R w_r u_r^{(1)} \circ u_r^{(2)} \circ u_r^{(3)}\|_F^2,$$

s.t. $\mathcal{P}_\Omega(X) = \mathcal{P}_\Omega(T)$.

By deriving the Karush-Kuhn-Tucker (KKT) conditions for function (15) [20], we can update $X$ by $X = \mathcal{P}_\Omega(X) + \mathcal{P}_\Omega(Z)$, where $Z = \sum_{r=1}^R w_r u_r^{(1)} \circ u_r^{(2)} \circ u_r^{(3)}$.

### 3.2 TREL1 for CP-rank Estimation

Applying TREL1 directly for CP-rank estimation, we obtain a new CP-rank estimation method, namely, TREL1$_{CP}$, summarized in Algorithm 1. Here we specify an initial medium rank value $\hat{R}$ for efficiency though we could also automatically set it to some high rank value, e.g., min$(I_1, I_2, I_3)$. Besides, to obtain a good initialization, we initialize the CP decomposition of an incomplete tensor using Robust Tensor Power Method (RTPM) [2] following [14]. RTPM makes TREL1$_{CP}$ less sensitive to parameter $\lambda$ than using random initialization.

3.2.1 Estimate CP-rank. In Algorithm 1, after iteratively updating all the $R + 1$ blocks till convergence or reaching the maximum iterations, we finally determine the CP-rank: checking the weight vector $w$, we only keep the weights greater than zero. The number of the remaining weights in $w$ is the estimated CP-rank.

### 3.3 TREL1 for Tucker-rank Estimation

Since the Tucker-rank $r$ consists of the mode ranks of unfolded matrices of $X$ along each mode, we can compute the rank of each unfolded matrix $X_{(i)}$, $i = 1, 2, 3$, by degenerating TREL1 to matrix case. For the mode-$i$ unfolded matrix $X_{(i)}$ of a tensor $X \in \mathbb{R}^{I_1 \times I_2 \times I_3}$,
We implemented TREL1 in MATLAB to evaluate the rank estimation. We analyze the complexity of TREL1 following [21], which mainly considers multiplication for weights and factors of Tucker-based works. However, the key difference is that our TREL1 degenerates the TREL1 to matrix Tucker-rank estimation as TREL1_Tucker and summarize it in Algorithm 2. Here, we use random initialization for weights and factors of X(i) because RTPM is only for third-order tensors.

Remark: This mode-wise estimation in TREL1_Tucker shares the same spirit as the Tucker-based nuclear norm (i.e., sum of the nuclear-norms of all the matricizations) and many other existing Tucker-based works. However, the key difference is that our TREL1 objective is to estimate the true Tucker-rank while Tucker-based nuclear norm is used to minimize the Tucker-rank. As to be shown in the empirical studies (e.g., Figs. 7 and 8), a smaller rank is not necessarily better and a rank smaller than the true rank often deteriorates the recovery performance.

3.4 Complexity Analysis

We analyze the complexity of TREL1 following [21], which mainly includes the shrinkage operator and some multiplications. At each iteration, the time complexity of performing the shrinkage operator in (13) is $O(R\prod_{j=1}^3 I_j)$. This is also the time complexity of computing $(u_r^{(n)}\prod_{n=1}^3$ and (15). The overall time complexity is $O(RR\prod_{j=1}^3 I_j)$.

4 EXPERIMENTAL RESULTS

We implemented TREL1 in MATLAB to evaluate the rank estimation and tensor completion/recovery performance. All experiments were performed on a PC (Intel Xeon(R) 4.0GHz, 64GB).

4.1 Experimental Setup

4.1.1 Compared Methods. We mainly consider decomposition-based methods with two steps: (i) rank estimation, and (ii) tensor completion with the rank estimated in (i). In addition, we tested three popular baseline methods (SiLRTC, FaLRTC and HaLRTC) in [18, 19]. FaLRTC performs the best among the three, but inferior to gHOU with TREL1 on the whole. Thus, their results are not included below for more compact presentation.

(i) Rank estimation. We study four existing methods:

- MGP-CP [27]: a Bayesian method for low-rank CP decomposition of incomplete tensors, which infers the CP-rank using a multiplicative gamma process.
- BRTF [43]: a Bayesian robust tensor factorization which employs a fully Bayesian generative model for automatic CP-rank estimation.
- ARD-Tucker [23]: an automatic relevance determination algorithm for sparse Tucker decomposition using the gradient based sparse coding algorithm.
- SCORE [39]: a robust Tucker-rank estimation method using Bayesian information criteria for complete tensors. Among the four methods, BRTF and ARD-Tucker performed much better than MGP-CP and SCORE, respectively. Thus, we only report the comparison of TREL1 against BRTF and ARD-Tucker to save space.

(ii) Tensor completion. We study two representative CP decomposition-based methods and three representative Tucker decomposition-based methods:

- CP-WOPT [1]: CP decomposition with missing data is formulated as a weighted least squares problem and solved by a gradient descent optimization approach.
- TenALS [14]: decomposing of incomplete tensors is formulated as a low-rank orthogonal CP decomposition problem, solved by an alternating minimization algorithm.
- gHOU [20]: a generalized higher-order orthogonal iteration tensor completion method, based on orthogonal Tucker decomposition.
- Tucker-WOPT [10]: a nonlinear conjugate gradient method that solves Tucker decomposition with missing data in a similar way as CP-WOPT.
- FRTC [15]: a Riemannian manifold preconditioning approach for tensor completion with rank constraint.

4.1.2 Synthetic Data. We generated four synthetic tensors of size [50×50×50, 100×100×100, 100×100×100, 200×200×200] with...
4.1.3 Real Data. We evaluate tensor rank estimation and completion on six real tensors: Hall sequence (144 × 176 × 300) [43], Knx medical images (256 × 256 × 24) [38] and Basketball video (144 × 256 × 40) [41] for CP-rank estimation; Amino Acid data (5 × 61 × 201) [27], Flow Injection data (12 × 100 × 89) and Ocean video (160 × 112 × 32) [19] for Tucker-rank estimation. We uniformly select 10% - 90% entries of each tensor at random as missing data and use "MR" to denote Missing Ratio.

4.1.4 Parameter Settings. We set the maximum iterations $K = 500$, $tol = 1e-5$ for all methods and the initial rank $\hat{R} = \text{round}(1/2 \times \min(I_1, I_2, I_3))$ for TREL1. We set other parameters of the compared methods following guidance from the original papers. We compare the estimated rank against the true rank to evaluate the rank estimation error. We measure tensor completion performance by Relative Square Error (RSE) [18]; $\|\mathbf{Z} - \mathbf{T}\|_F/\|\mathbf{T}\|_F$, where $\mathbf{Z}$ is the recovered tensor given a few entries from the (ground truth) $\mathbf{T}$. We repeat each run 10 times and report the average results. We report the running time as well, in seconds. The setting of $\lambda$ in TREL1 will be discussed in the following sensitivity study.

4.2 Parameter Sensitivity

4.2.1 Rank Estimation Sensitivity on $\lambda$. Figures 3 and 4 show the rank estimation results on various synthetic tensors by TREL1 with different $\lambda$s. As seen from Figs. 3(a) and 3(b), the rank estimation performance of TREL1$_{\text{CP}}$ is stable and not sensitive to the values of $\lambda$ in most cases. Only for very high missing ratios (e.g., MR = 90%), a large $\lambda$ (e.g., $\lambda = 110$) can make the $L_1$ regularization dominate the whole objective function (4) and result in zero rank. In addition, the time costs of TREL1$_{\text{CP}}$ are stable with most of $\lambda$ values (e.g., $\lambda \in [70, 200]$), as shown in Figs. 3(c) and 3(d).

Figure 4 shows that TREL1$_{\text{Tucker}}$ is not sensitive to $\lambda$ either on tensors with no more than half of data missing (i.e., MR ≤ 50%). However, for larger MRs, the rank estimation performance will deteriorate, which is not shown in the figures for clarity. Nevertheless, this is not surprising by noting that TREL1 is formulated based on CP decomposition so it suits the CP model better than the Tucker model. Nevertheless, for small to medium MRs, TREL1 can mostly produce an accurate estimate of the Tucker-rank for a wide range of $\lambda$. In addition, it is interesting to show that the time cost with a larger $\lambda$ is lower, as shown in Fig. 4(d).

In summary, CP/Tucker rank estimation performance does not require careful tuning of $\lambda$. The CP-rank estimation is accurate even for some challenging high MRs. Furthermore, the selection of $\lambda$ is largely insensitive to data. For example, good $\lambda$ values for synthetic tensors are good values for real tensors as well (to be shown in the following). Thus, we can fix $\lambda = 100$ in both CP/Tucker-rank estimation for both synthetic and real tensors. Note that in Tucker-rank estimation, we only need to set a single $\lambda$ and there is no need to set separate $\lambda$ values for each mode. Therefore, setting $\lambda$ is much more user-friendly than manually setting the rank values.

4.2.2 Rank Estimation Sensitivity on Initial Rank $\hat{R}$. Figures 5(a) and 5(b) study the sensitivity of rank estimation on $\hat{R}$. We can see the rank estimation results by TREL1 with different $\lambda$ values are not sensitive to $\hat{R}$ for both CP and Tucker models. Thus, we set $\hat{R} = \text{round}(1/2 \times \min(I_1, I_2, I_3))$ for all tests.
Figure 7: RSE of recovering a tensor (true CP-rank $R = 25$) via CP decomposition-based methods given (a) manually fixed ranks and (b) rank estimated via TREL1 with different $\lambda$.

Figure 8: RSE of recovering a tensor (true Tucker-rank $r = [8, 10, 12]$) via Tucker decomposition-based methods given (a) manually fixed ranks and (b) rank estimated via TREL1 with different $\lambda$.

4.3 Convergence Study

Since TREL1 can be viewed as performing TREL1 multiple times on unfolded matrices, we only study the convergence of TREL1 for CP-rank estimation in terms of weights error: $\|w_{(k+1)} - w_k\|_2/\|w_{(k+1)}\|_2$. Figure 6 shows that TREL1 converges within 100 iterations except for a large MR (> 70%), which needs more iterations to converge.

4.4 Effects of Rank Value on Completion Performance

Here, we present studies that investigate the effects of rank estimation accuracy on tensor completion performance of decomposition-based methods. All five decomposition-based tensor completion methods (i.e., CP-based CP-WOPT and TenALS, and Tucker-based gHOI, Tucker-WOPT and FRTC) listed in Sec. 4.1.1 (ii) are studied. We compare tensor completion performance with two ways of rank determination: (i) setting the rank manually; (ii) setting $\lambda$ in TREL1 to estimate the rank. We show the results of recovering two synthetic tensors with MR = [30%, 50%, 70%] in Figs. 7 and 8.

- As seen from Figs. 7(a) and 8(a), the completion performance (in RSE) of all five methods is highly sensitive to the manually set rank value. Even a slight error in the rank value (particularly lower-than-true ranks) can lead to serious performance degradation. Only given the true tensor ranks, all the five methods can achieve their best completion results in all cases. For CP-WOPT, TenALS and gHOI, setting any rank value different from the true rank gives much worse performance than their best results. Tucker-WOPT and FRTC can achieve good results given a higher-than-true rank although still worse than their best results.
- In contrast, Figs. 7(b) and 8(b) show the corresponding results with TREL1 rank estimation by setting $\lambda$ to a limited number of values only. We can see a wide range of $\lambda$ values lead to the best performance of all methods. Such range is particular wide for CP-based methods and narrower for Tucker-based methods, which is not surprising since TREL1 is designed based on a CP model.

This study shows the advantage of TREL1 in rank estimation, compared to manually specifying the rank. TREL1 greatly simplifies parameter tuning where a simple setting of $\lambda$ from a wide range of feasible values works for a wide range of methods and data. This not only improves the completion performance but also reduces the time cost in parameter tuning.

4.5 Tensor Rank Estimation and Completion Performance

Here, we report the results for MR = [30%, 50%, 70%] in four tables. We highlight the correctly estimated rank in italic and bold fonts, smallest RSE results in bold fonts and the second smallest RSE in bold fonts. Here, the corresponding CP-rank estimated by TREL1 and ARD are denoted as TREL1-R and ARD-R respectively, and the corresponding Tucker-rank estimated by TREL1 and ARD are denoted as TREL1-r and ARD-r respectively. Furthermore, the estimated tensor ranks are fed into decomposition-based tensor completion methods to compare the recovery performance.

4.5.1 CP-rank Estimation and Tensor Completion.

On synthetic tensors: As shown in the left half of Table 1, TREL1 correctly determines the true CP-ranks of the synthetic tensors in all cases, while BRTF over-estimates the ranks (it only succeeds in one case). More importantly, with TREL1-R, both CP-WOPT and TenALS achieve their best recovery results, as seen from the left half of Table 2. Moreover, TenALS outperforms CP-WOPT both given the true ranks (TREL1-R), which demonstrates the benefits of orthogonal CP decomposition for tensor completion.

On real tensors: We cannot directly evaluate the estimated CP-ranks since we do not know the ground-truth of CP-ranks for real tensors. Thus, we alternatively compare the tensor completion results affected by the estimated CP-ranks. As seen from the right half of Table 1, TREL1-R improves the completion performance of CP-WOPT and TenALS with around 25% than that of using BRTF-R. Moreover, with TREL1-R, TenALS still achieves better results than CP-WOPT on the whole.

4.5.2 Tucker-rank Estimation and Tensor Completion.

On synthetic tensors: As reported in the left half of Table 3, TREL1 can correctly determine the true Tucker-ranks of the synthetic tensors in all cases, while ARD-Tucker fails (over-estimates or under-estimates) in these cases. Furthermore, with our estimated true ranks (TREL1-r), the Tucker decomposition-based tensor completion methods (gHOI, Tucker-WOPT and FRTC) outperform the cases of using Tucker-ranks estimated by ARD-Tucker (ARD-r) by several orders, as shown in the left half of Table 4. Besides, FRTC fails to recover the tensors with more than 39
Table 1: CP-rank estimation on synthetic and real tensors with MR = {30%, 50%, 70%} missing entries. Est. \( R \) is the estimated CP-ranks and Time in seconds.

<table>
<thead>
<tr>
<th>Data</th>
<th>Synthetic</th>
<th>Synthetic</th>
<th>Synthetic</th>
<th>Real</th>
<th>Real</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50 x 50</td>
<td>100 x 100</td>
<td>200 x 200</td>
<td>144 x 256</td>
<td>256 x 256</td>
<td>144 x 176</td>
</tr>
<tr>
<td>R = 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30%</td>
<td>50%</td>
<td>70%</td>
<td>30%</td>
<td>50%</td>
<td>70%</td>
<td>30%</td>
</tr>
<tr>
<td>TREL1(_CP)</td>
<td>5 5 5</td>
<td>25 25 25</td>
<td>50 50 50</td>
<td>4 3 3</td>
<td>5 3 2</td>
<td>7 7 5</td>
</tr>
<tr>
<td>Time</td>
<td>1.59</td>
<td>1.61</td>
<td>2.41</td>
<td>482.56</td>
<td>529.28</td>
<td>658.47</td>
</tr>
<tr>
<td>BRTF[43]</td>
<td>9 6 5</td>
<td>36 42 32</td>
<td>85 79 73</td>
<td>2 1 1</td>
<td>1 1 1</td>
<td>2 2 1</td>
</tr>
<tr>
<td>Time</td>
<td>165.57</td>
<td>100.35</td>
<td>88.47</td>
<td>1829.04</td>
<td>1878.04</td>
<td>1978.05</td>
</tr>
</tbody>
</table>

Table 2: Tensor completion results by CP-based methods given estimated CP-ranks on synthetic and real tensors with MR = {30%, 50%, 70%} missing entries. TREL1\(_R\) and BRTF\(_R\) correspond to the estimated CP-ranks estimated by TREL1\(_CP\) and BRTF.

<table>
<thead>
<tr>
<th>Data</th>
<th>Synthetic</th>
<th>Synthetic</th>
<th>Synthetic</th>
<th>Real</th>
<th>Real</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50 x 50</td>
<td>100 x 100</td>
<td>200 x 200</td>
<td>144 x 256</td>
<td>256 x 256</td>
<td>144 x 176</td>
</tr>
<tr>
<td>R = 5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30%</td>
<td>50%</td>
<td>70%</td>
<td>30%</td>
<td>50%</td>
<td>70%</td>
<td>30%</td>
</tr>
<tr>
<td>with TREL1(_R)</td>
<td>Time 6.16</td>
<td>3.12</td>
<td>1.65</td>
<td>1015.99</td>
<td>663.24</td>
<td>629.92</td>
</tr>
<tr>
<td>BRTF(_R)</td>
<td>Time 23.62</td>
<td>9.69</td>
<td>1.66</td>
<td>6149.26</td>
<td>4431.69</td>
<td>1725.27</td>
</tr>
<tr>
<td>TenALS[14]</td>
<td>RSE 7.11E-09</td>
<td>1.65E-09</td>
<td>9.74E-09</td>
<td>1.36E-09</td>
<td>2.81E-09</td>
<td>8.26E-09</td>
</tr>
<tr>
<td>with TREL1(_R)</td>
<td>Time 1.19</td>
<td>1.22</td>
<td>1.26</td>
<td>97.59</td>
<td>104.84</td>
<td>108.93</td>
</tr>
<tr>
<td>TenALS[14]</td>
<td>RSE 1.61E-08</td>
<td>1.12E-08</td>
<td>9.74E-09</td>
<td>3.32E-07</td>
<td>1.48E-08</td>
<td>9.06E-04</td>
</tr>
<tr>
<td>with BRTF(_R)</td>
<td>Time 24.02</td>
<td>14.78</td>
<td>1.23</td>
<td>2290.32</td>
<td>1875.08</td>
<td>1805.51</td>
</tr>
</tbody>
</table>

hours time costs in five cases as its computational cost increases exponentially given over-estimated Tucker-ranks (ARD-r).

On real tensors: Unlike synthetic data with true Tucker-rank because we constructed them via QR decomposition and can control the dimensions of its core tensor (Tucker-rank), the real tensors naturally do not have exact low Tucker-ranks. We can unfold a real tensor along each mode and then truncate its mode rank \( R_1, R_2 \) and \( R_3 \) to be exact low-rank. However, if the unfolded matrices are interdependent, we can only control the low-rank in one mode exactly. Therefore, we studied the cases of truncating the unfolded matrices of a tensor into exact low-rank in one of the three modes, and report the results for the mode with the highest dimension. In this way, we can directly evaluate the estimated Tucker-rank exactly in one mode at least. As shown in the right half of Table 3, our method can correctly estimate the mode-1 rank \( R_1 = 22 \) of Ocean video, the mode-2 rank \( R_2 = 7 \) of Flow Injection and the mode-3 rank \( R_3 = 4 \) of Ammino Acid in all cases, while ARD-Tucker fails to get the exact mode ranks for these real tensors. The results shown in the right half of Table 4 demonstrate that: with TREL1\(_R\), the three Tucker decomposition-based tensor completion methods improve recovery performance than that of given ARD-r. Nevertheless, with ARD-r, Tucker-WOPT achieves the second best recovery results in two cases because it assumes the true ranks can be over-estimated.

4.5.3 Time Cost of Rank Estimation and Tensor Completion.

Time cost of TREL1\(_R\) estimation: As seen from Table 1: TREL1\(_CP\) is much faster than BRTF and only needs less than 1% and 18% of BRTF’s time cost on synthetic and real tensors on average respectively. Especially on the larger tensors with higher ranks, for example, TREL1\(_CP\) is about 30 times faster than BRTF on average on the 200 × 200 × 200 tensor with \( R = 50 \). Table 3 shows that TREL1\(_Tucker\) is about 9 times faster than ARD-Tucker on average on the synthetic and real tensors. Thus, due to the expensive time costs of the compared methods, it is not feasible to report results of the larger tensors here.
Table 3: Tucker-rank estimation results on synthetic and real tensors with MR = \{30\%, 50\%, 70\%\} missing entries. Time in seconds. Est.\(R_1\), Est.\(R_2\) and Est.\(R_3\) are the estimated Tucker-ranks in mode-1, mode-2 and mode-3, respectively.

<table>
<thead>
<tr>
<th>Data</th>
<th>Synthetic</th>
<th>Synthetic</th>
<th>Synthetic</th>
<th>Real</th>
<th>Real</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50 × 50 × 50</td>
<td>80 × 100 × 120</td>
<td>200 × 200 × 200</td>
<td>5 × 61 × 201</td>
<td>12 × 100 × 89</td>
<td>160 × 112 × 32</td>
</tr>
<tr>
<td>(r)</td>
<td>([5, 5, 5])</td>
<td>([8, 10, 12])</td>
<td>([15, 18, 20])</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>1.86</td>
<td>1.23</td>
<td>2.41</td>
<td>26</td>
<td>87</td>
<td>94</td>
</tr>
<tr>
<td>(RSE)</td>
<td>11.19</td>
<td>10.87</td>
<td>9.53</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\text{ARD-Tucker})</td>
<td>Est.(R_1)</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Est.(R_2)</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Est.(R_3)</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td>30.58</td>
<td>29.07</td>
<td>23.05</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td></td>
<td>3.67</td>
<td>3.63</td>
<td>3.64</td>
<td>6.18</td>
<td>6.18</td>
<td>6.18</td>
</tr>
</tbody>
</table>
| Table 4: Tensor completion results by Tucker-based methods given estimated ranks on synthetic and real tensors with MR = \{30\%, 50\%, 70\%\} missing entries. Time in seconds and "-" indicates that the results cost more than 50 hours. TREL1-\(r\) and ARD-\(r\) refer to the corresponding Tucker-ranks estimated by TREL1\_Tucker and ARD-Tucker.

<table>
<thead>
<tr>
<th>Data</th>
<th>Synthetic</th>
<th>Synthetic</th>
<th>Synthetic</th>
<th>Real</th>
<th>Real</th>
<th>Real</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>50 × 50 × 50</td>
<td>80 × 100 × 120</td>
<td>200 × 200 × 200</td>
<td>5 × 61 × 201</td>
<td>12 × 100 × 89</td>
<td>160 × 112 × 32</td>
</tr>
<tr>
<td>(r)</td>
<td>([5, 5, 5])</td>
<td>([8, 10, 12])</td>
<td>([15, 18, 20])</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>399.44</td>
<td>262.44</td>
<td>315.74</td>
<td>27.30</td>
<td>63.27</td>
<td>107.69</td>
</tr>
<tr>
<td>(RSE)</td>
<td>8.07</td>
<td>7.96</td>
<td>8.05</td>
<td>6.18</td>
<td>6.18</td>
<td>6.18</td>
</tr>
<tr>
<td>(\text{ARD-Tucker})</td>
<td>Est.(R_1)</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Est.(R_2)</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Est.(R_3)</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>Time</td>
<td>50.58</td>
<td>40.97</td>
<td>29.38</td>
<td>26</td>
<td>87</td>
</tr>
<tr>
<td></td>
<td>28.78</td>
<td>28.78</td>
<td>28.78</td>
<td>30.97</td>
<td>30.97</td>
<td>30.97</td>
</tr>
</tbody>
</table>

**Time cost of tensor completion using TREL1:** As shown in Table 1: unlike on synthetic tensors, CP-WOPT and TenALS with TREL1-\(r\) cost more time than those with BRTF-\(r\) on real tensors because TREL1-\(rs\) are larger than BRTF-\(Rs\), thus leading to better accuracy. This increased time cost is inherent for the tensor completion algorithms rather than TREL1. On Tucker-rank estimation, FRTC with TREL1-\(r\) is much faster than FRTC given ARD-\(r\) in most cases, as observed in Table 4.

5 CONCLUSION

In this paper, we defined a simple CP-based tensor nuclear norm and proposed two novel tensor rank (both CP-rank and Tucker-rank) estimation methods, TREL1\_CP and TREL1\_Tucker, based on orthogonal CP decomposition. In the proposed methods, we impose an \(L_1\) regularization on the weight vector of the orthogonal CP decomposition, served as the CP-based tensor nuclear norm, while minimizing the reconstruction error. This leads to automatic
rank determination for incomplete tensors. As demonstrated in our experimental results, TREL1 can correctly determine the true tensor ranks (both CP-rank and Tucker-rank) of synthetic tensors, and also can estimate the rank of real tensors well given sufficient observed entries. More importantly, our estimated tensor ranks consistently improved the recovery performance of decomposition-based tensor completion methods. Besides, TREL1 is not sensitive to its parameters in general and much more efficient than existing tensor rank estimation methods.

ACKNOWLEDGMENT

This work is supported by the NSFC Grant: 61672444 and 61272366, HKBU Faculty Research Grant: FRG2/16-17/051, the SZSTI Grant: JCY201605311900683, the HKBU KTO grant (MPC-004-2017/18), and Hong Kong PhD Fellowship Scheme. We thank Prof. Pyiush Rai, Dr. Bamdev Mishra, and Dr. Hiroyuki KASAI for their code sharing and helpful discussion.

REFERENCES