Proceedings Paper:
Q, S., Lu, H. and Cheung, Y.M. (2017) Tensor Rank Estimation and Completion via CP-based Nuclear Norm. In: CIKM'17: ACM Conference on Information and Knowledge Management Proceedings. ACM Conference on Information and Knowledge Management, 06-10 Nov 2017, Pan Pacific Singapore. Association for Computing Machinery . ISBN 978-1-4503-4918-5

https://doi.org/10.1145/3132847.3132945

Tensor Rank Estimation and Completion via CP-based Nuclear Norm

Qiquan Shi
Department of Computer Science,
Hong Kong Baptist University
Hong Kong, China
csqqshi@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/over-estimate the tensor rank while being quite 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].

CIKM'17, , November 6–10, 2017, Singapore.

ACM ISBN ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/http://dx.doi.org/10.1145/XXXXXX.XXXXXX

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 approach widely used in many methods [26, 33, 37]. However, ARD-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 **T**ensor **R**ank **E**stimation based on $\mathbf{L_1}$-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 tensor 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 $\mathbf{x} \in \mathbb{R}^I$ and a matrix is denoted by a bold capital letter $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$. A higher-order ($N \geq 3$) tensor is denoted by a calligraphic letter $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$. The ith entry of a vector $\mathbf{a} \in \mathbb{R}^I$ is denoted by $\mathbf{a}(i)$, and the (i, j)th entry of a matrix $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ is denoted by $\mathbf{X}(i, j)$. The $(i_1, \cdots, i_N)$th entry of an Nth-order tensor $\mathcal{X}$ is denoted by $\mathcal{X}(i_1, \cdots, i_N)$, where $i_n \in \{1, \cdots, I_n\}$ and $n \in \{1, \cdots, N\}$. The Frobenius norm of a tensor $\mathcal{X}$ is defined by $\|\mathcal{X}\|_F = \langle \mathcal{X}, \mathcal{X} \rangle^{1/2}$. $\Omega \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is a binary index set: $\Omega(i_1, \cdots, i_N) = 1$ if $\mathcal{X}(i_1, \cdots, i_N)$ 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 Ω:

$$(\mathcal{P}_\Omega(\mathcal{X}))(i_1, \cdots, i_N) = \begin{cases} \mathcal{X}(i_1, \cdots, i_N), & \text{if}(i_1, \cdots, i_N) \in \Omega \\ 0, & \text{if}(i_1, \cdots, i_N) \in \Omega^c \end{cases}, \tag{1}$$

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

Definition 1. **Mode-n Product**. *A mode-n product between a tensor* $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *and a matrix/vector* $\mathbf{U} \in \mathbb{R}^{I_n \times J_n}$ *is denoted by* $\mathcal{Y} = \mathcal{X} \times_n \mathbf{U}^\top \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N}$*, with entries given by* $\mathcal{Y}_{i_1 \cdots i_{n-1} j_n i_{n+1} \cdots i_N} = \sum_{i_n} \mathcal{X}_{i_1 \cdots i_{n-1} i_n i_{n+1} \cdots i_N} \mathbf{U}_{i_n, j_n}$*, and we have* $\mathrm{Y}_{(n)} = \mathbf{U}^T \mathbf{X}_{(n)}$ *[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* $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ *is denoted as* $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times \Pi_{n^* \neq n} I_{n^*}}$.

2.2 CP and Tucker Decomposition

2.2.1 Tucker Decomposition and Tucker-rank. A tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is represented as a core tensor with factor matrices in Tucker decomposition model [16]:

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

where $\{\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}, n = 1, 2 \cdots N$, 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 Nth-order tensor $\mathcal{X}$ is an N-dimensional vector, denoted as $\mathbf{r} = (R_1, \cdots, R_N)$, whose n-th entry R_n is the rank of the mode-n unfolded matrix $\mathbf{X}_{(n)}$ of $\mathcal{X}$. R_n is the **mode-n rank**. Figure 1 illustrates this decomposition.

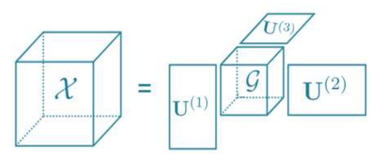


Figure 1: The Tucker decomposition of a third-order tensor.

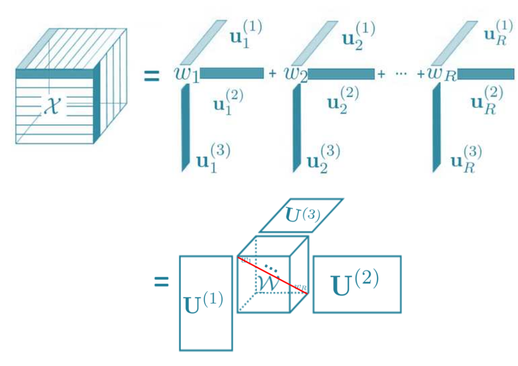


Figure 2: The CP decomposition of a third-order tensor $\mathcal{X}$, where the core tensor $\mathcal{W}$ is a super-diagonal tensor.

2.2.2 CP Decomposition and CP-rank. CP decomposition decomposes a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ as the weighted summation of a set of rank-one tensors:

$$
\begin{aligned}
\mathcal{X} &= \sum_{r=1}^{R} w_r \, \mathbf{u}_r^{(1)} \circ \cdots \circ \mathbf{u}_r^{(n)} \cdots \circ \mathbf{u}_r^{(N)} \\
&= \mathcal{W} \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \cdots \times_N \mathbf{U}^{(N)},
\end{aligned}
\tag{3}
$$

where each $\mathbf{u}_r^{(n)}, n = 1, \cdots, N$ is a unit vector with the weight absorbed into the weight vector $\mathbf{w} = [w_1, \cdots w_r, \cdots w_R]^\top \in \mathbb{R}^R$, 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 **T**ensor **R**ank **E**stimation methods based on $\mathbf{L_1}$-regularized orthogonal CP decomposition, namely, **TREL1**. For simpler notations, we consider third order tensors $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ 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 $\mathbf{u}_p^{(n)\top} \mathbf{u}_q^{(n)} = 0$ for $p \neq q$, and $\mathbf{u}_p^{(n)\top} \mathbf{u}_q^{(n)} = 1$ 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 $\mathcal{X}$ to be analogous to the ***vector of singular values*** of a matrix.

Definition 3. *The **CP-based Tensor Nuclear Norm** of a tensor $\mathcal{X}$ is defined as the L_1 norm of the weight vector* $\mathbf{w}$ *of its orthogonal CP decomposition:* $\|\mathcal{X}\|_{CP} = \|\mathbf{w}\|_1$.[1]

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.

[1] For easy reading, we use $\|\mathbf{w}\|_1$ instead of $\|\mathcal{X}\|_{\text{CP}}$ below.

Thus, our objective function is:

$$
\begin{aligned}
\min_{\mathcal{X}, \mathbf{w}, \{\mathbf{u}_r^{(n)}\}, R} \quad & \lambda \|\mathbf{w}\|_1 + \frac{1}{2} \|\mathcal{X} - \sum_{r=1}^{R} w_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \mathbf{u}_r^{(3)}\|_F^2, \\
\text{s.t.} \quad & \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T}), \mathbf{u}_r^{(n)\top} \mathbf{u}_r^{(n)} = 1, n = 1 \cdots 3, \\
& \mathbf{u}_r^{(n)\top} \mathbf{u}_q^{(n)} = 0, q = 1 \cdots r-1, r = 1 \cdots R,
\end{aligned}
\tag{4}
$$

where $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is the given incomplete tensor with observed entries in $\mathbf{\Omega}$. $\mathbf{w} = [w_1, \cdots, w_r, \cdots, w_R]^\top$ is the *weight vector* and R is the CP-rank of $\mathcal{X}$. λ 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: $\{\{w_1, \mathbf{u}_1^{(1)}, \mathbf{u}_1^{(2)}, \mathbf{u}_1^{(3)}\}, \cdots, \{w_r, \mathbf{u}_r^{(1)}, \mathbf{u}_r^{(2)}, \mathbf{u}_r^{(3)}\}, \cdots, \{w_R, \mathbf{u}_R^{(1)}, \mathbf{u}_R^{(2)}, \mathbf{u}_R^{(3)}\}, \mathcal{X}\}$. 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 $\{w_r, \mathbf{u}_r^{(1)}, \mathbf{u}_r^{(2)}, \mathbf{u}_r^{(3)}\}$ is:

$$
\begin{aligned}
\mathcal{L}_{w_r, \mathbf{u}_r^{(n)}} &= \lambda |w_r| + \frac{1}{2} \|\mathcal{X}_r - w_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \mathbf{u}_r^{(3)}\|_F^2, \\
& \text{s.t. } \mathbf{u}_r^{(n)\top} \mathbf{u}_r^{(n)} = 1, n = 1 \cdots 3, \\
& \mathbf{u}_r^{(n)\top} \mathbf{u}_q^{(n)} = 0, q = 1 \cdots r-1, r = 1 \cdots R,
\end{aligned}
\tag{5}
$$

where $\mathcal{X}_r = \mathcal{X} - \sum_{q=1}^{r-1} w_q \mathbf{u}_q^{(1)} \circ \mathbf{u}_q^{(2)} \circ \mathbf{u}_q^{(3)}$ is the residual of the approximation. We use Lagrange multipliers to transform (5) to include all the constraints as:

$$
\begin{aligned}
\mathcal{L}_{w_r, \mathbf{u}_r^{(n)}} &= \lambda |w_r| + \frac{1}{2} \|\mathcal{X}_r - w_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \mathbf{u}_r^{(3)}\|_F^2 \\
& - \gamma (\mathbf{u}_r^{(n)\top} \mathbf{u}_r^{(n)} - 1) - \sum_{q=1}^{r-1} \mu_q \mathbf{u}_r^{(n)\top} \mathbf{u}_q^{(n)},
\end{aligned}
\tag{6}
$$

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

3.1.1 Update $\mathbf{u}_r^{(n)}$. The function (6) with respect to $\mathbf{u}_r^{(1)}$ is,

$$
\begin{aligned}
\mathcal{L}_{\mathbf{u}_r^{(1)}} &= \frac{1}{2} \|\mathcal{X}_r - w_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \mathbf{u}_r^{(3)}\|_F^2 \\
& - \gamma (\mathbf{u}_r^{(1)\top} \mathbf{u}_r^{(1)} - 1) - \sum_{q=1}^{r-1} \mu_q \mathbf{u}_r^{(1)} \mathbf{u}_q^{(1)},
\end{aligned}
\tag{7}
$$

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

$$
\begin{aligned}
\mathbf{u}_r^{(1)} = & (\mathcal{X}_r \times_2 \mathbf{u}_r^{(2)} \times_3 \mathbf{u}_r^{(3)}) / w_r \\
& - \Big(\sum_{q=1}^{r-1} \mathbf{u}_q^{(1)\top} (\mathcal{X}_r \times_2 \mathbf{u}_r^{(2)} \times_3 \mathbf{u}_r^{(3)}) \, \mathbf{u}_q^{(1)} \Big) / w_r,
\end{aligned}
\tag{8}
$$

Algorithm 1 CP-rank Estimation Based on L_1-Regularized Orthogonal CP Decomposition (**TREL1$_{\mathbf{CP}}$**)

Input: $\mathcal{P}_\Omega(\mathcal{T})$, Ω, λ, initial rank $\hat{R}$, maximum iterations K, and stopping tolerance *tol*.
Initialization: Set $\mathcal{Z} = \text{zeros}(I_1, I_2, I_3)$, $\mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T})$, $\mathcal{P}_{\Omega^c}(\mathcal{X}) = \mathbf{0}$; Initialize $\{\mathbf{u}_r^{(1)}, \mathbf{u}_r^{(2)}, \mathbf{u}_r^{(3)}, w_r\}_{r=1}^{\hat{R}}$ of $\mathcal{X}$ by RTPM [2].
for $k = 1, \ldots, K$ **do**
  $\mathcal{X}_1 = \mathcal{X}$;
  for $r = 1, \ldots, \hat{R}$ **do**
    if $w_r \neq 0$ **then**
      Update $\{\mathbf{u}_r^{(1)}, \mathbf{u}_r^{(2)}, \mathbf{u}_r^{(3)}\}$ by (8), (9), (10) respectively.
      Update w_r by (14).
    end if
    $\mathcal{X}_r = \mathcal{X}_r - w_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \mathbf{u}_r^{(3)}$.
  end for
  Update $\mathcal{X}$: Set $\mathcal{Z} = \mathcal{X} - \mathcal{X}_r$ and update the missing entries by: $\mathcal{P}_{\Omega^c}(\mathcal{X}) = \mathcal{P}_{\Omega^c}(\mathcal{Z})$.
  If $\|\mathbf{w}_{(k+1)} - \mathbf{w}_k\|_2 / \|\mathbf{w}_{(k+1)}\|_2 < tol$, break; otherwise, continue.
end for
CP-rank Estimation: Only keep $w_r > 0$ in w_r and then obtain the CP-rank $R = \text{length}(\mathbf{w})$.
output: R.

and normalize $\mathbf{u}_r^{(1)} = \frac{\mathbf{u}_r^{(1)}}{\|\mathbf{u}_r^{(1)}\|_2}$. Note that we only update the blocks with non-zero weights. Similarly, we update $\mathbf{u}_r^{(2)}$ by

$$\begin{aligned}\mathbf{u}_r^{(2)} =& (\mathcal{X}_r \times_1 \mathbf{u}_r^{(1)} \times_3 \mathbf{u}_r^{(3)})/w_r \\ &- \Big(\sum_{q=1}^{r-1} \mathbf{u}_q^{(2)\top} (\mathcal{X}_r \times_1 \mathbf{u}_r^{(1)} \times_3 \mathbf{u}_r^{(3)})\ \mathbf{u}_q^{(2)}\Big)/w_r,\end{aligned} \tag{9}$$

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

$$\begin{aligned}\mathbf{u}_r^{(3)} =& (\mathcal{X}_r \times_1 \mathbf{u}_r^{(1)} \times_2 \mathbf{u}_r^{(2)})/w_r \\ &- \Big(\sum_{q=1}^{r-1} \mathbf{u}_q^{(3)\top} (\mathcal{X}_r \times_1 \mathbf{u}_r^{(1)} \times_2 \mathbf{u}_r^{(2)})\ \mathbf{u}_q^{(3)}\Big)/w_r,\end{aligned} \tag{10}$$

and normalize $\mathbf{u}_r^{(3)} = \frac{\mathbf{u}_r^{(3)}}{\|\mathbf{u}_r^{(3)}\|_2}$.

3.1.2 *Update w_r.*

The function (6) with respect to w_r is:

$$\mathcal{L}_{w_r} = \lambda |w_r| + \frac{1}{2} \|\mathcal{X}_r - w_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \mathbf{u}_r^{(3)}\|_F^2. \tag{11}$$

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

$$w_r = \mathcal{X}_r \times_1 \mathbf{u}_r^{(1)} \times_2 \mathbf{u}_r^{(2)} \times_3 \mathbf{u}_r^{(3)} - \frac{\lambda |w_r|}{\partial w_r}. \tag{12}$$

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

$$w_r = shrink_\lambda(\langle \mathcal{X}_r, \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \mathbf{u}_r^{(3)} \rangle), \tag{13}$$

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

Input: $\mathcal{P}_\Omega(\mathcal{T})$, Ω, λ, initial Tucker-rank $\hat{\mathbf{r}} = [\hat{R}_1, \hat{R}_2, \hat{R}_3]$, maximum iterations K, and stopping tolerance *tol*.
Set $\mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T})$, $\mathcal{P}_{\Omega^c}(\mathcal{X}) = \mathbf{0}$.
for $i = 1, \ldots, 3$ **do**
  $\mathbf{X}_{(i)}$= unfold $(\mathcal{X}, i)$.
  $\Omega_{(i)}$ =ones (size($\mathbf{X}_{(i)}$)), $\Omega_{(i)}(\mathbf{X}_{(i)} == 0) = 0$.
  Tucker-rank Estimation: Compute the Tucker-rank $\mathbf{r} = [R_1, R_2, R_3]$ via $R_i = \text{TREL1}_{\text{CP}}\ (\mathbf{X}_{(i)}, \Omega_{(i)}, \lambda, \hat{R}_i, K, tol)$.
end for
output: $\mathbf{r} = [R_1, R_2, R_3]$.

where *shrink* is the soft thresholding operator [25], and we denote $S = \langle \mathcal{X}_r, \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \mathbf{u}_r^{(3)} \rangle$:

$$w_r = shrink_\lambda(S) = \begin{cases} S - \lambda & (S > \lambda) \\ 0 & (|S| \leq \lambda) \\ S + \lambda & (S < -\lambda) \end{cases}. \tag{14}$$

3.1.3 *Update $\mathcal{X}$.*

The objective function (4) with respect to $\mathcal{X}$ is,

$$\begin{aligned}\min_{\mathcal{X}}\ & \frac{1}{2} \|\mathcal{X} - \sum_{r=1}^{R} w_r \mathbf{u}_r^{(1)} \circ \mathbf{u}_r^{(2)} \circ \mathbf{u}_r^{(3)}\|_F^2, \\ \text{s.t. } & \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T}),\end{aligned} \tag{15}$$

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

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$_{\text{CP}}$ less sensitive to parameter λ 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 $\mathbf{w}$, we only keep the weights greater than zero. The number of the remaining weights in $\mathbf{w}$ is the estimated CP-rank.

3.3 TREL1 for Tucker-rank Estimation

Since the Tucker-rank $\mathbf{r}$ consists of the mode ranks of unfolded matrices of $\mathcal{X}$ along each mode, we can compute the rank of each unfolded matrix $\mathbf{X}_{(i)}, i = 1, 2, 3$, by degenerating TREL1 to matrix case. For the mode-i unfolded matrix $\mathbf{X}_{(i)}$ of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, we have:

$$\min_{\mathbf{X}_{(i)}, \mathbf{w}, \{\mathbf{u}_r^{(n)}\}, R_i} \lambda\|\mathbf{w}\|_1 + \frac{1}{2}\|\mathbf{X}_{(i)} - \sum_{r=1}^{R_i} w_r\, \mathbf{u}_r^{(1)} \mathbf{u}_r^{(2)^\top}\|_F^2,$$
$$\text{s.t. } \mathcal{P}_\Omega(\mathcal{X}) = \mathcal{P}_\Omega(\mathcal{T}), \mathbf{u}_r^{(n)^\top} \mathbf{u}_r^{(n)} = 1, n = 1 \cdots 2, \tag{16}$$
$$\mathbf{u}_r^{(n)^\top} \mathbf{u}_q^{(n)} = 0, q = 1 \cdots r-1, r = 1 \cdots R_i,$$

where R_i is the rank (i-th entry of Tucker-rank) of mode-i unfolded matrix $\mathbf{X}_{(i)}$ of $\mathcal{X}$. Here, each unfolded matrix is approximated by an orthogonal CP decomposition, which is actually the SVD of the unfolded matrix as the orthogonal CP decomposition is a generalization of SVD from matrices to tensors.

3.3.1 Estimate Tucker-rank. We degenerate the TREL1 to matrix case to estimate the mode ranks $\{R_i\}_{i=1}^3$ of unfolded matrices along each mode, and finally determine the Tucker-rank $\mathbf{r} = [R_1, R_2, R_3]$. We denote this TREL1 for Tucker-rank estimation as $\mathbf{TREL1_{Tucker}}$ and summarize it in **Algorithm** 2. Here, we use random initialization for weights and factors of $\mathbf{X}_{(i)}$ because RTPM is only for third-order tensors.

Remark: This mode-wise estimation in $\mathbf{TREL1_{Tucker}}$ 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 $\{\mathbf{u}_r^{(n)}\}_{n=1}^3$ and (15). The overall time complexity is $O(KR(\prod_{j=1}^3 I_j)$.

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

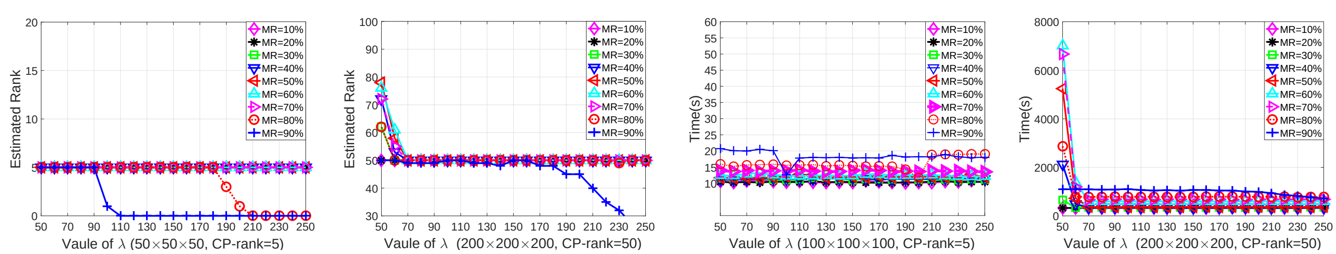
4.1.1 Compared Methods. We mainly consider decomposition-based methods with two steps: (i) rank estimation, and (ii) tensor completion with the rank estimated in (i). In addition, we tested three popular baseline methods (SiLRTC, FaLRTC and HaLRTC) in [18, 19]. FaLRTC performs the best among the three, but inferior to gHOI 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.

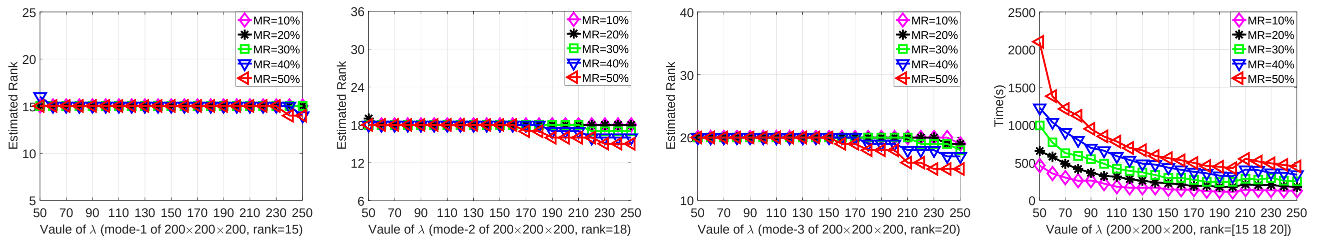
(ii) *Tensor completion.* We study *two* representative CP decomposition-based methods and *three* representative Tucker decomposition-based methods:
- **CP-WOPT** [1]: CP decomposition with missing data is formulated as a weighted least squares problem and solved by a gradient descent optimization approach.
- **TenALS** [14]: decomposing of incomplete tensors is formulated as a low-rank *orthogonal* CP decomposition problem, solved by an alternating minimization algorithm.
- **gHOI** [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



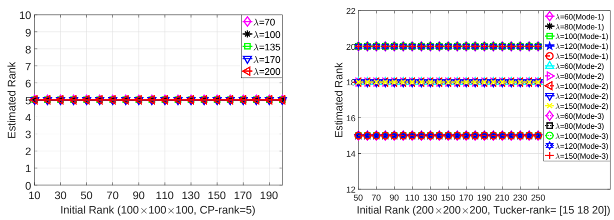
(a) Estimated ranks: 100×100×100 ($R = 5$) (b) Estimated ranks: 200 × 200 × 200 ($R = 50$) (c) Time costs: 100 × 100 × 100 ($R = 5$) (d) Time costs: 200 × 200 × 200 ($R = 50$)

Figure 3: (a-b) Estimated *CP-ranks* of two tensors ($100 \times 100 \times 100$ with $R = 5$ and $200 \times 200 \times 200$ with $R = 50$) by $\mathbf{TREL1_{CP}}$ with $\lambda \in [50 : 10 : 250]$; (c-d) the corresponding time costs.



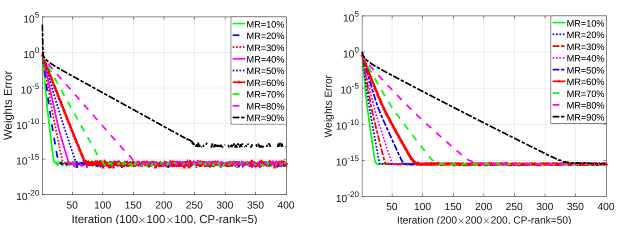
(a) Estimated ranks on **mode-1**: $R_1 = 15$ (b) Estimated ranks on **mode-2**: $R_1 = 18$ (c) Estimated ranks on **mode-3**: $R_3 = 20$ (d) Time costs: $200 \times 200 \times 200$ ($\mathbf{r} = \{15\ 18\ 20\}$)

Figure 4: (a-c) Estimated *Tucker-ranks in each mode* of a $200 \times 200 \times 200$ ($\mathbf{r} = [15, 18, 20]$) **tensor by TREL1$_{\text{Tucker}}$ with** $\lambda \in [50 : 10 : 250]$; **(d) The corresponding time costs.**



(a) $100 \times 100 \times 100$ ($R = 5$) (b) $200 \times 200 \times 200$ ($\mathbf{r} = \{15\ 18\ 20\}$)

Figure 5: (a) Estimated *CP-ranks* of the $100 \times 100 \times 100$ ($R = 5$) **tensor with** 50% **missing entries by TREL1$_{\text{CP}}$ given different initial ranks** $\hat{R} \in [10 : 10 : 200]$ **(b) Estimated *Tucker-ranks* of the** $200 \times 200 \times 200$ ($\mathbf{r} = [15, 18, 20]$) **with** 50% **missing entries by TREL1$_{\text{Tucker}}$ given different initial rank** $\hat{R} \in [50 : 10 : 250]$.



(a) $100 \times 100 \times 100$ ($R = 5$) (b) $200 \times 200 \times 200$ ($R = 50$)

Figure 6: Convergence curves of TREL1$_{\text{CP}}$ in terms of *weights error*: $\|\mathbf{w}_{(k+1)} - \mathbf{w}_k\|_2 / \|\mathbf{w}_{(k+1)}\|_2$ **on two tensors.**

CP-ranks $R = \{5, 5, 25, 50\}$, respectively, following [43]. We also generated three synthetic tensors of size $\{50\times50\times50, 80\times100\times120, 200\times 200 \times 200\}$ with Tucker-ranks $\mathbf{r} = \{[5, 5, 5], [8, 10, 12], [15, 18, 20]\}$, respectively, following [15].

4.1.3 Real Data. We evaluate tensor rank estimation and completion on *six* **real** tensors: Hall sequence ($144 \times 176 \times 300$) [43], Knix medical images ($256 \times 256 \times 24$) [38] and Basketball video ($144 \times 256 \times 40$) [41] for CP-rank estimation; Amino Acid data ($5 \times 61 \times 201$) [27], Flow Injection data ($12 \times 100 \times 89$) and Ocean video ($160\times112\times32$) [19] for Tucker-rank estimation. We uniformly select 10% − 90% entries of each tensor at random as missing data and use **"MR"** to denote **M**issing **R**atio.

4.1.4 Parameter Settings. We set the maximum iterations $K = 500$, $tol = 1e-5$ for all methods and the initial rank $\hat{R} = \text{round}(1/2 \times \text{mean}(I_1, I_2, I_3))$ 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]: $\|\mathcal{Z} - \mathcal{T}\|_F / \|\mathcal{T}\|_F$, where $\mathcal{Z}$ is the recovered tensor given a few entries from the (ground truth) $\mathcal{T}$. We repeat each run 10 times and report the average results. We report the running time as well, in seconds. The setting of λ in TREL1 will be discussed in the following sensitivity study.

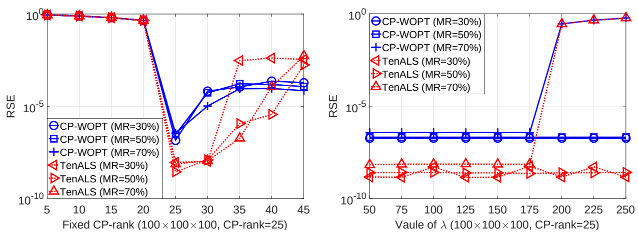
4.2 Parameter Sensitivity

4.2.1 Rank Estimation Sensitivity on λ. Figures 3 and 4 show the rank estimation results on various synthetic tensors by TREL1 with different λs. 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 λ in most cases. Only for very high missing ratios (e.g., MR = 90%), a large λ (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 λ 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 λ either on tensors with no more than half of data missing (i.e., MR $\leq$ 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 λ. In addition, it is interesting to show that the time cost with a lager λ is lower, as shown in Fig. 4(d).

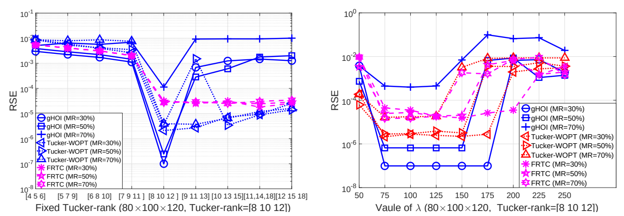
In summary, CP/Tucker rank estimation performance does not require careful tuning of λ. The CP-rank estimation is accurate even for some challenging high MRs. Furthermore, the selection of λ is largely insensitive to data. For example, good λ values for synthetic tensors are good values for real tensors 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 λ and there is no need to set separate λ values for each mode. Therefore, setting λ 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 of λ values are not sensitive to $\hat{R}$ for both CP and Tucker models. Thus, we set $\hat{R} = \text{round}(1/2 \times \text{mean}(I_1, I_2, I_3))$ for all tests.



(a) Using manually fixed CP-ranks (b) Using estimated CP-ranks via λs

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 λ.



(a) Using manually fixed Tucker-ranks (b) Using estimated Tucker-ranks via λs

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

4.3 Convergence Study

Since $\text{TREL1}_{\text{Tucker}}$ can be viewed as performing TREL1_{CP} multiple times on unfolded matrices, we only study the convergence of TREL1 for CP-rank estimation in terms of *weights error*: $\|\mathbf{w}_{(k+1)} - \mathbf{w}_k\|_2 / \|\mathbf{w}_{(k+1)}\|_2$. Figure 6 shows that TREL1_{CP} 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 λ 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 λ to a limited number of values only. We can see a wide range of λ 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 λ 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 <u>second smallest RSE</u> in <u>underline</u>. Here, the corresponding CP-rank estimated by TREL1_{CP} and BRTF are denoted as **TREL1-*R*** and **BRTF-*R*** respectively, and the corresponding Tucker-rank estimated by $\text{TREL1}_{\text{Tucker}}$ and ARD-Tucker 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_{CP} 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, $\text{TREL1}_{\text{Tucker}}$ 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.

	Data	Synthetic $50 \times 50 \times 50$ $R = \mathbf{5}$			Synthetic $100 \times 100 \times 100$ $R = \mathbf{25}$			Synthetic $200 \times 200 \times 200$ $R = \mathbf{50}$			Real $144 \times 256 \times 40$ Hall Sequence			Real $256 \times 256 \times 24$ Knix Medical Images			Real $144 \times 176 \times 300$ Basketball Video		
	MR	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%
TREL1$_{\mathbf{CP}}$	Est.R	***5***	***5***	***5***	***25***	***25***	***25***	***50***	***50***	***50***	4	3	3	5	3	2	7	7	5
	Time	1.59	1.61	2.41	29.17	33.04	43.74	487.60	526.28	568.47	399.30	366.20	363.08	98.83	76.70	79.87	74.13	70.34	69.34
BRTF[43]	Est.R	9	6	***5***	36	42	32	85	79	73	2	1	1	1	1	1	2	2	1
	Time	165.57	100.35	88.47	4178.78	5901.48	3462.21	1.80E+05	1.87E+05	1.30E+05	2944.40	2239.89	2089.96	960.78	784.26	766.29	280.61	265.87	226.36

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 refer to the corresponding CP-ranks estimated by TREL1$_{\mathbf{CP}}$ and BRTF.

	Data	Synthetic $50 \times 50 \times 50$ $R = \mathbf{5}$			Synthetic $100 \times 100 \times 100$ $R = \mathbf{25}$			Synthetic $200 \times 200 \times 200$ $R = \mathbf{50}$			Real $144 \times 256 \times 40$ Hall Sequence			Real $256 \times 256 \times 24$ Knix Medical Images			Real $144 \times 176 \times 300$ Basketball Video		
	MR	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%
CP-WOPT[1]	RSE	1.20E-07	1.43E-07	6.12E-07	1.94E-07	3.42E-07	4.79E-07	7.27E-08	9.56E-08	1.37E-07	**1.97E-01**	2.09E-01	**2.11E-01**	3.45E-01	**3.80E-01**	**4.22E-01**	**1.87E-01**	**1.88E-01**	2.05E-01
with **TREL1**-R	Time	6.16	3.12	1.65	1015.99	636.24	629.92	3.46E+04	1.53E+04	1.19E+04	5118.82	1779.43	1114.47	1746.97	207.79	57.33	1060.02	971.85	299.95
CP-WOPT[1]	RSE	9.96E-05	3.82E-05	6.12E-07	1.94E-04	1.87E-04	6.49E-05	3.39E-04	1.86E-04	1.04E-04	2.35E-01	2.62E-01	2.62E-01	4.72E-01	4.72E-01	4.73E-01	2.49E-01	2.49E-01	3.06E-01
with BRTF-R	Time	23.62	9.69	1.66	6149.26	4631.69	1725.27	1.72E+05	1.15E+05	5.33E+04	751.29	44.66	24.08	11.66	6.90	3.81	73.39	44.35	2.95
TenALS [14]	RSE	**7.11E-09**	**1.65E-09**	**9.74E-09**	**1.36E-09**	**2.81E-09**	**8.26E-09**	**3.26E-09**	**5.96E-09**	**1.52E-09**	**1.97E-01**	**2.08E-01**	**2.11E-01**	**3.43E-01**	**3.80E-01**	**4.22E-01**	**1.87E-01**	1.89E-01	**2.04E-01**
with **TREL1**-R	Time	1.19	1.22	1.26	97.59	104.84	108.93	1493.04	1829.04	2178.05	1755.47	1454.21	1381.88	541.85	280.75	182.67	736.83	712.95	430.92
TenALS[14]	RSE	1.61E-08	1.32E-08	**9.74E-09**	3.32E-07	1.48E-08	9.06E-04	1.13E-05	5.50E-06	9.58E-05	2.35E-01	2.62E-01	2.62E-01	4.72E-01	4.72E-01	4.73E-01	2.48E-01	2.49E-01	3.06E-01
with BRTF-R	Time	24.02	14.78	1.23	2290.32	1876.08	1805.51	5.18E+04	5.49E+04	5.46E+04	1002.08	424.07	419.32	103.92	103.95	98.67	124.50	124.66	66.59

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, because 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 Amino Acid in all cases, while ARD-Tucker fails to get the exact mode ranks for these real tensors. In addition, the results shown in the right half of Table 4 demonstrate that: with TREL1-**r**, the three Tucker decomposition-based tensor completion methods improves 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 rank estimation: As seen from Table 1: **TREL1$_{\mathbf{CP}}$** 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$_{\text{CP}}$ is about 300 times faster than BRTF on average on the $200 \times 200 \times 200$ tensor with $R = 50$. Table 3 shows that **TREL1$_{\mathbf{Tucker}}$** 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 lager 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.

	Data	Synthetic $50 \times 50 \times 50$ $\mathbf{r} = [5,\ 5,\ 5]$			Synthetic $80 \times 100 \times 120$ $\mathbf{r} = [8,\ 10,\ 12]$			Synthetic $200 \times 200 \times 200$ $\mathbf{r} = [15, 18, 20]$			Real $5 \times 61 \times \mathbf{201}$ Amino Acid ($R_3 = \mathbf{4}$)			Real $12 \times \mathbf{100} \times 89$ Flow Injection ($R_2 = \mathbf{7}$)			Real $\mathbf{160} \times 112 \times 32$ Ocean Video ($R_1 = \mathbf{22}$)		
	MR(%)	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%
L1TRE$_{\text{Tucker}}$	Est.R_1	***5***	***5***	***5***	***8***	***8***	***8***	***15***	***15***	***15***	5	5	5	11	11	11	***22***	***22***	***22***
	Est.R_2	***5***	***5***	***5***	***10***	***10***	***10***	***18***	***18***	***18***	7	6	4	***7***	***7***	***7***	51	51	51
	Est. R_3	***5***	***5***	***5***	***12***	***12***	***12***	***20***	***20***	***20***	***4***	***4***	***4***	23	22	31	32	32	32
	Time	1.86	3.10	6.87	32.14	56.87	99.73	430.95	788.72	1442.56	1.63	2.82	5.31	3.83	5.87	19.71	35.37	54.46	101.10
ARD-Tucker [23]	Est.R_1	6	23	25	2	41	48	26	87	94	5	5	5	12	12	11	11	23	27
	Est.R_2	6	25	25	3	50	50	78	100	100	25	18	30	12	29	34	15	35	41
	Est.R_3	6	25	25	3	50	50	87	100	100	25	17	18	20	34	34	19	32	34
	Time	50.58	90.27	93.08	127.82	310.78	260.07	1021.62	1398.99	1584.68	19.23	36.27	55.35	66.93	83.17	80.16	86.44	161.09	201.89

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$_{\text{Tucker}}$ and ARD-Tucker.

	Data	Synthetic $50 \times 50 \times 50$ $\mathbf{r} = [5,\ 5,\ 5]$			Synthetic $80 \times 100 \times 120$ $\mathbf{r} = [8,\ 10,\ 12]$			Synthetic $200 \times 200 \times 200$ $\mathbf{r} = [15, 18, 20]$			Real $5 \times 61 \times \mathbf{201}$ Amino Acid ($R_3 = \mathbf{4}$)			Real $12 \times \mathbf{100} \times 89$ Flow Injection ($R_2 = \mathbf{7}$)			Real $\mathbf{160} \times 112 \times 32$ Ocean Video ($R_1 = \mathbf{22}$)		
	MR(%)	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%	30%	50%	70%
gHOI [20]	RSE	**1.50E-06**	4.50E-06	3.51E-05	**9.87E-08**	**1.82E-07**	1.12E-04	**3.61E-07**	2.09E-04	1.98E-03	3.34E-03	3.38E-02	5.84E-02	**1.07E-02**	4.63E-02	1.23E-01	**3.56E-02**	**4.78E-02**	**6.40E-02**
with **TREL1-r**	Time	1.17	1.23	1.30	2.33	2.58	5.37	1271.58	1301.92	1128.82	0.66	1.08	0.83	2.26	2.34	3.03	8.33	9.73	12.03
gHOI [20]	RSE	1.85E-02	8.69E-02	2.02E-01	3.95E-03	7.07E-02	1.94E-01	1.01E-02	5.18E-02	1.24E-01	1.15E-01	1.47E-01	2.58E-01	1.76E-02	1.78E-01	3.45E-01	7.26E-02	6.73E-02	7.68E-02
with ARD-**r**	Time	1.77	2.98	5.12	2.65	19.72	16.06	27.30	63.27	107.69	1.97	1.95	3.03	2.80	3.48	3.64	5.14	9.38	12.41
Tuker-WOPT	RSE	3.22E-06	**3.43E-06**	**4.22E-06**	2.49E-06	2.97E-06	1.99E-05	6.81E-05	3.24E-04	5.86E-04	3.30E-03	5.56E-03	9.28E-03	1.38E-02	**2.16E-02**	**2.57E-02**	6.09E-02	6.30E-02	6.74E-02
[10] with **TREL1-r**	Time	399.44	184.96	154.48	681.72	1187.41	300.53	1271.58	1301.92	1128.82	81.91	87.30	83.69	183.23	207.14	107.88	221.40	301.67	265.43
Tuker-WOPT	RSE	1.38E-02	2.54E-04	2.96E-04	8.68E-03	2.01E-04	2.95E-04	3.13E-03	3.34E-03	3.90E-03	1.15E-02	1.03E-02	1.12E-02	1.47E-02	2.18E-02	5.20E-02	1.28E-01	8.51E-02	7.90E-02
[10] with ARD-**r**	Time	79.83	105.96	118.69	184.78	493.82	438.93	3157.47	3409.49	4741.76	203.81	227.49	469.54	175.08	56.91	76.29	141.98	218.34	259.79
FRTC [15]	RSE	2.05E-04	1.98E-04	2.26E-04	2.12E-05	1.49E-05	**1.60E-05**	5.56E-06	**5.06E-06**	**6.09E-06**	**3.01E-03**	**5.31E-03**	**8.95E-03**	5.22E-02	3.36E-01	6.60E+00	5.51E-01	3.54E+00	2.14E+01
with **TREL1-r**	Time	2.12	2.00	1.77	71.34	55.46	39.98	2624.47	2172.52	1691.71	3.80	10.91	5.30	12.54	34.14	545.78	4.26E+04	3.59E+04	2.92E+04
FRTC [15]	RSE	1.10E-02	7.67E-01	4.44E+00	7.22E-03	2.17E-01	2.65E+00	–	–	–	3.05E-02	3.75E-02	1.16E-01	1.78E-01	5.34E+01	2.98E+02	1.77E+00	1.10E+01	2.22E+01
with ARD-**r**	Time	6.45	3517.56	3053.61	24.00	1.78E+05	1.43E+05	–	–	–	1341.69	604.89	471.05	28.78	2486.80	2205.16	1.58E+04	2.87E+04	2.63E+04

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, though 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$_{\text{CP}}$ and TREL1$_{\text{Tucker}}$, 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: JCYJ20160531194006833, the HKBU KTO grant (MPCF-004-2017/18), and Hong Kong PhD Fellowship Scheme. We thank Prof. Piyush Rai, Dr. Bamdev Mishra, and Dr. Hiroyuki KASAI for their code sharing and helpful discussion.

REFERENCES

[1] Evrim Acar, Daniel M Dunlavy, Tamara G Kolda, and Morten Mørup. 2011. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems* 106, 1 (2011), 41–56.

[2] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. 2014. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research* 15, 1 (2014), 2773–2832.

[3] Juan Andrés Bazerque, Gonzalo Mateos, and Georgios B Giannakis. 2013. Rank regularization and Bayesian inference for tensor completion and extrapolation. *IEEE Transactions on Signal Processing* 61, 22 (2013), 5689–5703.

[4] Göran Bergqvist and Erik G Larsson. 2010. The higher-order singular value decomposition: Theory and an application. *IEEE Signal Processing Magazine* 27, 3 (2010), 151–154.

[5] Emmanuel J Candès and Benjamin Recht. 2009. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics* 9, 6 (2009), 717–772.

[6] J Douglas Carroll and Jih-Jie Chang. 1970. Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika* 35, 3 (1970), 283–319.

[7] Yi-Lei Chen, Chiou-Ting Hsu, and Hong-Yuan Mark Liao. 2014. Simultaneous tensor decomposition and completion using factor priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 3 (2014), 577–591.

[8] Wei Chu and Zoubin Ghahramani. 2009. Probabilistic models for incomplete multi-dimensional arrays. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*. Citeseer.

[9] Jean Baptiste Denis and Thierry Dhorne. 1989. Orthogonal tensor decomposition of 3-way tables. In *Multiway Data Analysis*. 31–37.

[10] Marko Filipović and Ante Jukić. 2015. Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidimensional Systems and Signal Processing* 26, 3 (2015), 677–692.

[11] Silvia Gandy, Benjamin Recht, and Isao Yamada. 2011. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems* 27, 2 (2011), 25010–25028.

[12] Richard A Harshman. 1970. Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics* (1970), 1–84.

[13] Christopher J Hillar and Lek-Heng Lim. 2013. Most tensor problems are NP-hard. *J. ACM* 60, 6 (2013), 45.

[14] Prateek Jain and Sewoong Oh. 2014. Provable tensor factorization with missing data. In *Advances in Neural Information Processing Systems*. 1431–1439.

[15] Hiroyuki Kasai and Bamdev Mishra. 2016. Low-rank tensor completion: a Riemannian manifold preconditioning approach. In *Proceedings of The 33rd International Conference on Machine Learning*. 1012–1021.

[16] Tamara G Kolda and Brett W Bader. 2009. Tensor decompositions and applications. *SIAM Rev.* 51, 3 (2009), 455–500.

[17] Wim P Krijnen, Theo K Dijkstra, and Alwin Stegeman. 2008. On the non-existence of optimal solutions and the occurrence of "degeneracy" in the CANDECOMP/PARAFAC Model. *Psychometrika* 73, 3 (2008), 431–439.

[18] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. 2009. Tensor completion for estimating missing values in visual data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2114–2121.

[19] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. 2013. Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 1 (2013), 208–220.

[20] Yuanyuan Liu, Fanhua Shang, Wei Fan, James Cheng, and Hong Cheng. 2014. Generalized higher-order orthogonal iteration for tensor decomposition and completion. In *Advances in Neural Information Processing Systems*. 1763–1771.

[21] Yuanyuan Liu, Fanhua Shang, Licheng Jiao, James Cheng, and Hong Cheng. 2015. Trace norm regularized CANDECOMP/PARAFAC decomposition with missing data. *IEEE Transactions on Cybernetics* 45, 11 (2015), 2437–2448.

[22] Haiping Lu, Konstantinos N Plataniotis, and Anastasios Venetsanopoulos. 2013. *Multilinear Subspace Learning: Dimensionality Reduction of Multidimensional Data*. CRC press.

[23] Morten Mørup and Lars Kai Hansen. 2009. Automatic relevance determination for multi-way models. *Journal of Chemometrics* 23, 7-8 (2009), 352–363.

[24] Cun Mu, Bo Huang, John Wright, and Donald Goldfarb. 2014. Square Deal: Lower Bounds and Improved Relaxations for Tensor Recovery. In *Proceedings of the 31st International Conference on Machine Learning*. 73–81.

[25] Stanley Osher, Yu Mao, Bin Dong, and Wotao Yin. 2011. Fast linearized Bregman iteration for compressive sensing and sparse denoising. *arXiv preprint arXiv:1104.0262* (2011).

[26] Yuan Alan Qi, Thomas P Minka, Rosalind W Picard, and Zoubin Ghahramani. 2004. Predictive automatic relevance determination by expectation propagation. In *Proceedings of the 21st International Conference on Machine Learning*. ACM, 85.

[27] Piyush Rai, Yingjian Wang, Shengbo Guo, Gary Chen, David Dunson, and Lawrence Carin. 2014. Scalable Bayesian low-rank decomposition of incomplete multiway tensors. In *Proceedings of the 31st International Conference on Machine Learning*. 1800–1808.

[28] Wenjie Ruan, Peipei Xu, Quan Z Sheng, Nguyen Khoi Tran, Nickolas JG Falkner, Xue Li, and Wei Emma Zhang. 2016. When Sensor Meets Tensor: Filling Missing Sensor Values Through a Tensor Approach. In *Proceedings of the 25th ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2025–2028.

[29] Fanhua Shang, Yuanyuan Liu, James Cheng, and Hong Cheng. 2014. Robust principal component analysis with missing data. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 1149–1158.

[30] Qiquan Shi, Yiu-ming Cheung, and Qibin Zhao. 2017. Feature Extraction for Incomplete Data via Low-rank Tucker Decomposition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (Accepted)*.

[31] Kijung Shin, Lee Sael, and U Kang. 2017. Fully scalable methods for distributed tensor factorization. *IEEE Transactions on Knowledge and Data Engineering* 29, 1 (2017), 100–113.

[32] Marco Signoretto, Quoc Tran Dinh, Lieven De Lathauwer, and Johan AK Suykens. 2014. Learning with tensors: a framework based on convex optimization and spectral regularization. *Machine Learning* 94, 3 (2014), 303–351.

[33] Vincent YF Tan and Cédric Févotte. 2013. Automatic relevance determination in nonnegative matrix factorization with the β-divergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 7 (2013), 1592–1605.

[34] Giorgio Tomasi and Rasmus Bro. 2005. PARAFAC and missing values. *Chemometrics and Intelligent Laboratory Systems* 75, 2 (2005), 163–180.

[35] Ledyard R Tucker. 1963. Implications of factor analysis of three-way matrices for measurement of change. *Problems in Measuring Change* 15 (1963), 122–137.

[36] Yichen Wang, Robert Chen, Joydeep Ghosh, Joshua C Denny, Abel Kho, You Chen, Bradley A Malin, and Jimeng Sun. Rubik: Knowledge guided tensor factorization and completion for health data analytics. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[37] David P Wipf and Srikantan S Nagarajan. 2008. A new view of automatic relevance determination. In *Advances in Neural Information Processing Systems*. 1625–1632.

[38] Yuning Yang, Yunlong Feng, and Johan AK Suykens. 2016. Robust low-rank tensor recovery with regularized redescending M-estimator. *IEEE Transactions on Neural Networks and Learning Systems* 27, 9 (2016), 1933–1946.

[39] Tatsuya Yokota, Namgil Lee, and Andrzej Cichocki. 2017. Robust multilinear tensor rank estimation using higher order singular value decomposition and information criteria. *IEEE Transactions on Signal Processing* 65, 5 (2017), 1196–1206.

[40] Tong Zhang and Gene H Golub. 2001. Rank-one approximation to high order tensors. *SIAM J. Matrix Anal. Appl.* 23, 2 (2001), 534–550.

[41] Zemin Zhang, Gregory Ely, Shuchin Aeron, Ning Hao, and Misha Kilmer. 2014. Novel methods for multilinear data completion and de-noising based on tensor-SVD. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 3842–3849.

[42] Qibin Zhao, Liqing Zhang, and Andrzej Cichocki. 2015. Bayesian CP Factorization of Incomplete Tensors with Automatic Rank Determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 9 (2015), 1751–1763.

[43] Qibin Zhao, Guoxu Zhou, Liqing Zhang, Andrzej Cichocki, and Shun-Ichi Amari. 2016. Bayesian robust tensor factorization for incomplete multiway data. *IEEE Transactions on Neural Networks and Learning Systems* 27, 4 (2016), 736–748.