




Rank minimization on tensor ring: an efficient approach for tensor decomposition and completion

Longhao Yuan^{1,2,3}  · Chao Li³ · Jianting Cao^{2,3,4} · Qibin Zhao^{1,3}

Received: 2 May 2019 / Revised: 22 July 2019 / Accepted: 16 September 2019 / Published online: 4 November 2019
© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2019

Abstract

In recent studies, tensor ring decomposition (TRD) has become a promising model for tensor completion. However, TRD suffers from the rank selection problem due to the undetermined multilinear rank. For tensor decomposition with missing entries, the sub-optimal rank selection of traditional methods leads to the overfitting/underfitting problem. In this paper, we first explore the latent space of the TRD and theoretically prove the relationship between the TR-rank and the rank of the tensor unfoldings. Then, we propose two tensor completion models by imposing the different low-rank regularizations on the TR-factors, by which the TR-rank of the underlying tensor is minimized and the low-rank structures of the underlying tensor are exploited. By employing the alternating direction method of multipliers scheme, our algorithms obtain the TR factors and the underlying tensor simultaneously. In experiments of tensor completion tasks, our algorithms show robustness to rank selection and high computation efficiency, in comparison to traditional low-rank approximation algorithms.

Keywords Tensor ring decomposition · Tensor completion · Structured nuclear norm · ADMM scheme

Editors: Kee-Eung Kim and Jun Zhu.

✉ Jianting Cao
cao@sit.ac.jp

✉ Qibin Zhao
qibin.zhao@riken.jp

Longhao Yuan
longhao.yuan@riken.jp

Chao Li
chao.li@riken.jp

¹ School of Automation, Guangdong University of Technology, Guangzhou, China

² Graduate School of Engineering, Saitama Institute of Technology, Fukaya, Japan

³ Tensor Learning Unit, RIKEN Center for Advanced Intelligence Project (AIP), Tokyo, Japan

⁴ School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China

1 Introduction

Tensors are the natural representations of higher-order data (Kolda and Bader 2009; Sidiropoulos et al. 2017) and have been successfully applied to machine learning (Chen et al. 2018; Novikov et al. 2015; Zhao et al. 2012), computer vision (Liu et al. 2013; Zhao et al. 2015), signal processing (Cichocki et al. 2015), remote sensing (Du et al. 2017), collaborative filtering (Hu et al. 2015) and so on. Most of the datasets in the applications are partially observed, which boosts the wide studies of the tensor completion problem (Long et al. 2018; Song et al. 2019). Tensor completion aims to recover the missing entries by sparse observations. The existing methods impose assumptions of various low-rank priors to discover the underlying tensor. According to the different types of low-rank assumption, tensor completion methods can be divided into two categories which are based on tensor decomposition and low-rank regularization, respectively.

Tensor decomposition is to find the decomposition factors (a.k.a., latent factors) of tensors, thereby casting tensors into a multilinear tensor latent space of low-dimensionality (very few degrees of freedom designated by the rank of the tensor decomposition). In recent years, tensor ring decomposition (TRD) has been proposed and shows the properties of super-linear compression ability and computational efficiency (Zhao et al. 2016). The most significant advantage of TRD is that the model complexity grows linearly in the tensor order, thus providing a natural solution for the “curse of dimensionality”. On the contrary, the number of model parameters of Tucker decomposition (TKD) increases exponentially in the tensor order. Although the CANDECOMP/PARAFAC decomposition (CPD) is a highly compact representation which has the desirable property of being linear in the tensor order, the CPD is difficult to optimize the latent factors. In recent years, based on the assumption that the underlying tensor is in TR structure, several tensor completion methods have been proposed and shown high performance and efficiency (Khoo et al. 2017; Wang et al. 2017; Yu et al. 2019; Yuan et al. 2018). One of the problems of TRD is that the performance of TR-based completion methods is very sensitive to model selection. Due to the interdependence among the optimal rank of the decomposition model, the different data structure and the missing patterns, it is rather challenging to select a proper rank for the data approximation. Moreover, finding the optimal TR-rank by cross-validation is not practical even for 3rd-order tensors, as TRD is defined in terms of multi-linear rank (i.e., the number of the undetermined TR-rank equals to the tensor order).

Low-rank-regularization-based tensor completion methods do not find the tensor decomposition factors directly. Instead, they make assumptions that the underlying tensor has a low-rank structure and impose convex surrogates of low-rankness on the tensor structure to minimize the rank. One of the most common such surrogates is named nuclear norm (a.k.a., trace norm, or Schatten norm), which is the sum of the singular values of the matrix. Various algorithms based on the minimization of CP-rank, Tucker-rank, TT-rank, and hierarchical-Tucker-rank have been proposed (Bengua et al. 2017; Liu et al. 2013, 2019; Yuan and Zhang 2016). The biggest advantage of this type of methods is that they are free from the rank selection since the rank of the underlying tensor will be automatically learned in the optimization process. However, the nuclear norm regularization of the tensor always requires multiple large-scale singular value decomposition (SVD) calculations on the tensor unfoldings, which in turn to require a high computational cost. Moreover, for TRD, the relationship between the TR-rank and the rank of tensor unfoldings has not yet been explored, so currently, there is no study about the tensor completion by low-TR-rank regularization.

The existing TR-based completion algorithms lack efficient solution regarding the model selection problem. Several studies (Khoo et al. 2017; Wang et al. 2017; Yuan et al. 2018) have proposed TR-based completion algorithms by gradient descent (GD) and alternative least squares (ALS) methods. The above algorithms have to manually tune the TR-rank to obtain the better solution which is time-consuming and inefficient. The algorithms in Liu et al. (2015, 2016) impose low-rank constraints on the CPD and TKD respectively to achieve fast computations. However, the strong constraints on the factors will lead to poor completion performance in many real-world data settings. For other tensor completion methods, Zhao et al. (2015) proposes a completion algorithm using Bayesian inference which can tune the CP-rank. Nevertheless, the multi-linear rank of TR makes it difficult to extend Bayesian methods to TRD. Moreover, the greedy rank-tuning algorithms based on CP decomposition (Yokota et al. 2016) and Tucker decomposition (Yokota et al. 2018) exhibit poor efficiency when facing large-scale tensor and multi-linear rank model.

In this paper, in order to solve the rank selection problem of TRD and increase the computational efficiency, we propose a novel tensor completion approach which exploits the low-rankness of TR latent space by nuclear norm regularizations. Our main contributions are listed below:

- The relationship between the rank of the tensor unfoldings and the TRD factors is theoretically proved, based on which the low-rank surrogate on TR latent factors is imposed to minimize the TR-rank and explore more low-rank structure of the underlying tensor.
- Based on two different low-rank regularizations, we develop two tensor ring completion models termed as tensor ring low-rank factors (TRLRF) and tensor ring latent nuclear norm (TRLNN) which are suitable for different tensor completion tasks.
- The alternating direction method of multipliers (ADMM) solving scheme of the two models are developed. The experimental results of simulation data show that our algorithms are robust to rank selection. Moreover, the real-world data experiments show high performance and high efficiency of our algorithms in both low-order and high-order tensor completion tasks.

2 Preliminaries

2.1 Notations and tensor operations

We mainly adopt notations of Kolda and Bader (2009) in this paper. Scalars are denoted by standard lowercase letter or standard uppercase letter, e.g., x , X . Vectors, matrices and tensors are denoted by \mathbf{x} , \mathbf{X} and \mathcal{X} respectively. A sequence of tensor $\{\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \dots, \mathcal{X}^{(N)}\}$ is denoted by $\{\mathcal{X}^{(n)}\}_{n=1}^N$, or simply $[\mathcal{X}]$, in which $\mathcal{X}^{(n)}$ is the n th tensor of the sequence. The matrix sequences and vector sequences are defined in the same way. With index (i_1, i_2, \dots, i_N) , an element of a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is denoted by $\mathcal{X}(i_1, i_2, \dots, i_N)$. Moreover, the Frobenius norm of \mathcal{X} is defined by $\|\mathcal{X}\|_F = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$, where $\langle \cdot, \cdot \rangle$ is the inner product. The nuclear norm of a matrix is denoted by $\|\cdot\|_*$ which is the sum of the singular values of the matrix.

Moreover, we employ three types of tensor unfolding (matricization) operations in this paper. The standard mode- n unfolding (Kolda and Bader 2009) of tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is denoted by $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 \cdots I_{n-1} I_{n+1} \cdots I_N}$. The second mode- n unfolding operation of tensor \mathcal{X} which is often used in TR operations (Zhao et al. 2016) is denoted by $\mathbf{X}_{\langle n \rangle} \in \mathbb{R}^{I_n \times I_{n+1} \cdots I_N I_1 \cdots I_{n-1}}$. The third kind of mode- n unfolding of tensor \mathcal{X} is denoted by $\mathbf{X}_{[n]} \in$

$\mathbb{R}^{I_1 \cdots I_n \times I_{n+1} \cdots I_N}$ which is often applied in tensor train operations (Oseledets 2011). Furthermore, the inverse operation of tensor unfolding is matrix folding (tensorization), which transforms matrices to higher-order tensors. The folding operations of the three types of mode- n unfoldings are defined as $\text{fold}_{(n)}(\cdot)$, $\text{fold}_{\langle n \rangle}(\cdot)$ and $\text{fold}_{[n]}(\cdot)$ respectively, i.e., for a tensor \mathcal{X} , we have $\text{fold}_{(n)}(\mathbf{X}_{(n)}) = \mathcal{X}$.

2.2 Tensor completion by tensor ring decomposition

TRD decomposes a tensor into a sequence of 3rd-order latent tensors (a.k.a., TR factors). For $n = 1, \dots, N$, the TR factors are denoted by $\mathcal{G}^{(n)} \in \mathbb{R}^{R_n \times I_n \times R_{n+1}}$ with $R_1 = R_{N+1}$. For each TR factor, we define its 1st and 3rd modes as the “rank-modes” and its 2nd mode as the “dimension-mode” due to the form of TRD. Furthermore, we define the vector $[R_1, R_2, \dots, R_N]^T$ as TR-rank (Zhao et al. 2016). Note that in the context of TRD, the rank is defined as a N -dimensional vector rather than a scalar as its matrix counterpart. Given the TR factors, the (i_1, i_2, \dots, i_N) th element of the tensor \mathcal{X} can be given by:

$$\mathcal{X}(i_1, i_2, \dots, i_N) = \text{Trace}(\mathbf{G}_{i_1}^{(1)} \mathbf{G}_{i_2}^{(2)} \cdots \mathbf{G}_{i_N}^{(N)}), \tag{1}$$

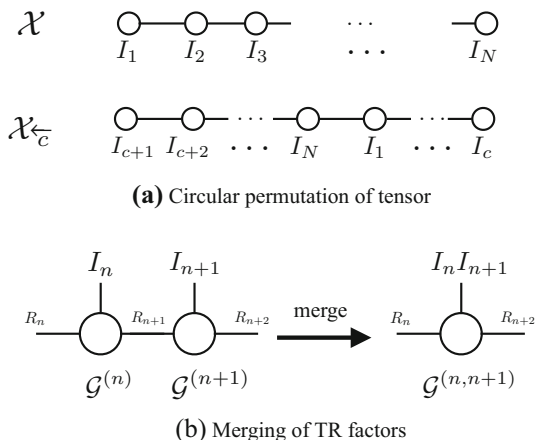
where $\text{Trace}(\cdot)$ denotes the trace operation which equals to the sum of the diagonal elements of a matrix and $\mathbf{G}_{i_n}^{(n)} \in \mathbb{R}^{R_n \times R_{n+1}}$ denotes the i_n th slice of $\mathcal{G}^{(n)}$ along the dimension-mode. To introduce the properties of TRD, we first define two tensor operations:

Definition 1 (Tensor circular permutation) The tensor circular permutation is to shift the tensor order by one direction. For example, if we anticlockwise-shift a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ by c steps, the output tensor is denoted by $\mathcal{X}_{\leftarrow c} \in \mathbb{R}^{I_{c+1} \times \cdots \times I_N \times I_1 \times \cdots \times I_c}$.

Definition 2 [Merging of TR factors (Zhao et al. 2019)] If a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is decomposed as TR factors $[\mathcal{G}]$, then the adjacent TR factors can be merged by reshaping and multiple matrix multiplication operations. For example, $\{\mathcal{G}^{(i)}, \mathcal{G}^{(i+1)}, \dots, \mathcal{G}^{(j)}\}$ can be merged as: $\mathcal{G}^{(i,i+1,\dots,j)} \in \mathbb{R}^{R_i \times \prod_{k=i}^j I_k \times R_{j+1}}$.

The diagrams to illustrate the two operations are shown in Fig. 1. Based on the two operations, we can introduce one of the most important properties of TRD which is circular

Fig. 1 **a** The tensor circular permutation of an N th order tensor by c steps of anticlockwise-shift. **b** The merging of the two adjacent TR factors $\mathcal{G}^{(n)}$ and $\mathcal{G}^{(n+1)}$



permutation invariance (Zhao et al. 2016). If tensor \mathcal{X} can be decomposed by $\{\mathcal{G}^{(n)}\}_{n=1}^N$, then its circular permutation satisfies:

$$\mathcal{X}_{\leftarrow c} = \Psi(\mathcal{G}^{(c+1)}, \dots, \mathcal{G}^{(N)}, \mathcal{G}^{(1)}, \dots, \mathcal{G}^{(c)}), \tag{2}$$

where $\Psi(\cdot)$ is the operator to calculate the tensor by using its TR factors. The following relationship of the tensor and the TR factors can be applied as the operator $\Psi(\cdot)$:

$$\mathbf{X}_{\langle n \rangle} = \mathbf{G}_{(2)}^{(n)} (\mathbf{G}_{\langle 2 \rangle}^{(\neq n)})^\top, \tag{3}$$

where $\mathcal{G}^{(\neq n)} \in \mathbb{R}^{R_{n+1} \times \prod_{i=1, i \neq n}^N I_i \times R_n}$ is a subchain tensor by merging operation of all TR factors except the n th core tensor, see more details in Zhao et al. (2019). We can see from (2) that for TRD, the circular permutation of a tensor is corresponding to the circular permutation of the TR factors.

Furthermore, as one of the applications of TRD, we can exploit TRD to predict missing entries from an incomplete tensor. In general, tensor ring completion can be done by solving the following problem:

$$\min_{\{\mathcal{G}\}, \mathcal{X}} \|\mathcal{X} - \Psi(\{\mathcal{G}\})\|_F^2 \quad s.t. \quad P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{T}), \tag{4}$$

where \mathcal{T} is the observed tensor and $P_\Omega(\mathcal{T})$ denotes all the observed entries w.r.t. the index set represented by Ω . The model is to optimize the TR factors $\{\mathcal{G}\}$ and obtain the completed tensor \mathcal{X} simultaneously. It is known from existing studies (Wang et al. 2017; Yuan et al. 2018) that the performance of tensor ring completion is largely effected by TR-rank. However, due to the undetermined multi-linear TR-rank and high-dimensionality of the incomplete data, determining suitable TR-rank for completion therefore becomes an extremely time-consuming work in practice.

2.3 Tensor completion by nuclear norm minimization

The tensor completion model based on nuclear norm minimization can be generally formulated by:

$$\min_{\mathcal{X}} \text{Rank}(\mathcal{X}) + \frac{\lambda}{2} \|P_\Omega(\mathcal{T} - \mathcal{X})\|_F^2, \tag{5}$$

where \mathcal{T} is the observed tensor and \mathcal{X} is the underlying tensor being recovered, $\text{Rank}(\cdot)$ is the rank regularizer and λ is the hyper-parameter. Most studies apply the overlapped nuclear norm of the tensor as the low-rank regularizer and employ nuclear norm as the low-rank surrogate which cast the problem into minimizing the summation of the nuclear norm (SNN) of the tensor unfoldings:

$$\min_{\mathcal{X}} \sum_{n=1}^N \|\mathbf{X}_{(n)}\|_* + \frac{\lambda}{2} \|P_\Omega(\mathcal{T} - \mathcal{X})\|_F^2. \tag{6}$$

Moreover, in consideration of increasing the flexibility of the low-rank constraint, a new type of low-rank regularizer for tensor, termed as latent nuclear norm (LNN), has been proposed and studied recently (Tomiooka and Suzuki 2013). The LNN equals the infimum of sum of a sequence of matrix nuclear norm. Given an N th order tensor \mathcal{X} , it is defined as

$$\|\mathcal{X}\|_{LNN} = \inf_{\mathcal{X} = \mathcal{W}^{(1)} + \dots + \mathcal{W}^{(N)}} \sum_{n=1}^N \|\mathbf{W}^{(n)}\|_*, \tag{7}$$

where $[\mathcal{W}] \in \mathbb{R}^{I_1 \times \dots \times I_N}$ are the latent components of the tensor \mathcal{X} . Compared to the conventional overlapped tensor nuclear norm (Liu et al. 2013), the LNN-based model is proved to provide more precise completion results especially in the unbalanced case of tensor rank (Tomioka and Suzuki 2013). The latent model considers the original tensor as a summation of several latent tensors and assumes that each latent tensor is low-rank in a specific mode:

$$\sum_{n=1}^N \|\mathbf{W}^{(n)}\|_* + \frac{\lambda}{2} \|P_{\Omega}(\mathcal{T} - \mathcal{X})\|_F^2, \tag{8}$$

$$s.t. \mathcal{X} = \sum_{n=1}^N \mathcal{W}^{(n)}.$$

The LNN model can fit the tensor better than overlapped nuclear norm model if the tensor is not low-rank in all modes, because of the flexible low-rank constraint of LNN. Tomioka and Suzuki (2013), it has been theoretically proved that the mean square error of a latent norm model scales no greater than the overlapped norm model. However, though the model (6) and model (8) can effectively regularize the rank of the underlying tensor in different situations, both of the two models have to process multiple SVD operations. When facing large-scale data, the huge computational cost of the nuclear-norm-minimization-based methods will be an intractable problem.

3 Low-rank regularizations on tensor ring latent space

3.1 The upper bounds for the tensor unfoldings

In consideration of imposing low-rank constraints on the TR-factors, we need to first deduce the relationship between the TR factors and the underlying tensor. In this subsection, we first prove that the ranks of the tensor unfoldings are upper bounded by the rank of TR factors. Then, we extend the theoretical result to a corollary that the rank of the unfoldings of the tensor under arbitrary circular permutations can be bounded by the TR-rank. This proves that the low-rank constraint on TR factors will impose a low-TR-rank constraint on the underlying tensor. Till now, works in Liu et al. (2013) and Bengua et al. (2017) have cast the low-rank tensor completion problem to minimizing the summation of the nuclear norm (SNN) as $\text{Rank}(\mathcal{X}) := \sum_{n=1}^N \|\mathbf{X}_{(n)}\|_*$ and $\text{Rank}(\mathcal{X}) := \sum_{n=1}^N \|\mathbf{X}_{[n]}\|_*$, respectively. These two low-rank regularizers minimize the Tucker-rank and the TT-rank of the underlying tensor respectively. However, currently, there is no study on the relation between the rank of TR factors and the rank of the underlying tensor. We provide the theoretical proof of the relationship below.

Theorem 1 *Given an N th order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ which is in TR-format with the TR-rank $[R_1, R_2, \dots, R_N]^T$ and the TR factors are denoted by $\{\mathcal{G}^{(n)}\}_{n=1}^N \in \mathbb{R}^{R_n \times I_n \times R_{n+1}}$, then the following inequality holds for all $n = 1, \dots, N$:*

$$\text{Rank}(\mathbf{X}_{(n)}) \leq \text{Rank}(\mathbf{G}_{(2)}^{(n)}). \tag{9}$$

Proof For the n th TR factor $\mathcal{G}^{(n)}$, according to equation (3), the relationship between the rank of the tensor unfoldings and the rank of the TR factor unfoldings can be simply deduced by:

$$\begin{aligned} \text{Rank}(\mathbf{X}_{\langle n \rangle}) &\leq \min\{\text{Rank}(\mathbf{G}_{(2)}^{(n)}), \text{Rank}(\mathbf{G}_{\langle n \rangle}^{(\neq n)})\} \\ &\leq \text{Rank}(\mathbf{G}_{(2)}^{(n)}). \end{aligned} \tag{10}$$

The proof is completed by

$$\text{Rank}(\mathbf{X}_{<n>}) = \text{Rank}(\mathbf{X}_{(n)}) \leq \text{Rank}(\mathbf{G}_{(2)}^{(n)}). \tag{11}$$

□

This theorem proves the relationship between the rank of tensor unfoldings and the rank of the TR factors. The rank of mode- n unfolding of the tensor \mathcal{X} is upper bounded by the rank of the dimension-mode unfolding of the corresponding core tensor $\mathcal{G}^{(n)}$, which allows us to impose a low-rank constraint on $\{\mathcal{G}\}$ to explore the more low-rank structure of the underlying tensor. Furthermore, we extend Theorem 1 and provide Corollary 1, which reveals more rank relationships between the tensor and the TR factors.

Corollary 1 *If the tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \dots \times I_N}$ is in TR-format with the TR-rank $[R_1, R_2, \dots, R_N]^\top$ and the TR factors are denoted by $\{\mathcal{G}^{(n)}\}_{n=1}^N \in \mathbb{R}^{R_n \times I_n \times R_{n+1}}$, then for $\forall c$ and $n \in [1, N]$, the rank of $\mathbf{X}_{\leftarrow, [n]}^c$ is bounded by TR-rank as*

$$\text{Rank}(\mathbf{X}_{\leftarrow, [n]}^c) \leq R_{c+1} R_{t+1}. \tag{12}$$

where

$$t = \begin{cases} c + n, & n \leq N - c; \\ N - n + 1, & \text{otherwise.} \end{cases} \tag{13}$$

Proof By applying the property of circular permutation invariance of TR decomposition (Zhao et al. 2016) (Theorem 2.1), $\mathcal{X}_{\leftarrow}^c \in \mathbb{R}^{I_{c+1} \times \dots \times I_N \times I_1 \times \dots \times I_c}$ can be decomposed by the TR-factors $\{\mathcal{G}^{(c+1)}, \dots, \mathcal{G}^{(N)}, \mathcal{G}^{(1)}, \dots, \mathcal{G}^{(c)}\}$, or simply $\{\mathcal{G}_{\leftarrow, (1)}^c, \dots, \mathcal{G}_{\leftarrow, (N)}^c\}$. From equation (1), we can deduce the relationship of an arbitrary element of $\mathcal{X}_{\leftarrow}^c$ with index (i_1, \dots, i_N) and the TR factors as follow:

$$\mathcal{X}_{\leftarrow}^c(i_1, \dots, i_N) = \text{Trace} \left(\prod_{k=1}^n \mathbf{G}_{i_k}^{\leftarrow, (k)} \prod_{k=n+1}^N \mathbf{G}_{i_k}^{\leftarrow, (k)} \right). \tag{14}$$

For $n \in [1, N]$, we denote the merging operation of $\{\mathcal{G}_{\leftarrow, (1)}^c, \dots, \mathcal{G}_{\leftarrow, (n)}^c\}$ and $\{\mathcal{G}_{\leftarrow, (n+1)}^c, \dots, \mathcal{G}_{\leftarrow, (N)}^c\}$ as $\mathcal{G}_{\leftarrow, \leq n}^c$ and $\mathcal{G}_{\leftarrow, > n}^c$ respectively. Then, equation (3) applied in Theorem 1 can be rewritten as a more general form:

$$\mathbf{X}_{\leftarrow, [n]}^c = \mathbf{G}_{(2)}^{\leftarrow, \leq n} (\mathbf{G}_{> 2}^{\leftarrow, > n})^\top. \tag{15}$$

This indicates that there exists a matrix decomposition for $\mathbf{X}_{\leftarrow, [n]}^c$ of rank $R_{c+1} R_{t+1}$, so we have $\text{Rank}(\mathbf{X}_{\leftarrow, [n]}^c) \leq R_{c+1} R_{t+1}$. □

Corollary 1 proves that the unfoldings of the arbitrary circular permuted tensor have the rank upper bounds which are constrained by the TR-rank. Compared to Tucker-rank and TT-rank, which are the bounds of the other kinds of tensor unfoldings, the TR-rank can bound more tensor unfolding structures, thus exploiting more low-rank structures of the underlying tensor. From Theorem 1, we know that the rank of the unfolded tensor is an “under-estimator” of the product of TR-rank. Meanwhile, we can also infer that the regularization on the TR-rank is equivalent to minimizing the rank of the tensor unfoldings.

3.2 Model formulation

Traditional rank minimization based tensor completion methods perform nuclear norm regularization of multiple matrices generated by tensor unfoldings, and thus suffering from high computational cost of large-scale SVD operations in every iteration. To reduce the computational cost, we impose low-rank regularizations on each of the TR factors. By imposing the nuclear norm regularizations on the TR factors, we can largely decrease the computational complexity of our model compared to tensor completion models based on (6) which impose low-rank regularization on the whole tensor. Our basic tensor completion model is formulated as follow:

$$\begin{aligned} \min_{[\mathcal{G}], \mathcal{X}} \quad & \sum_{n=1}^N \text{Rank}(\mathcal{G}^{(n)}) + \frac{\lambda}{2} \|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2, \\ \text{s.t.} \quad & P_{\Omega}(\mathcal{X}) = P_{\Omega}(\mathcal{T}). \end{aligned} \quad (16)$$

Based on Theorem 1, we impose nuclear norm regularization on the dimension-mode of the TR factors (i.e., $\|\mathbf{G}_{(2)}^{(n)}\|_*$) to explore more low-rank structure of the underlying tensor. Moreover, according to Corollary 1, we consider to impose nuclear norm regularizations on the two rank-modes of the TR factors, i.e., the unfoldings of the TR factors along mode-1 and mode-3, which can be expressed by $\sum_{n=1}^N \|\mathbf{G}_{(1)}^{(n)}\|_* + \sum_{n=1}^N \|\mathbf{G}_{(3)}^{(n)}\|_*$. When the model is optimized, nuclear norms of the rank-mode unfoldings and the fitting error of the approximated tensor are minimized simultaneously, resulting in the initial TR-rank becoming the upper bound of the real TR-rank of the tensor, thus equipping our model with robustness to rank selection. Finally, the tensor ring low-rank factors (TRLRF) model can be expressed as:

$$\begin{aligned} \min_{[\mathcal{G}], \mathcal{X}} \quad & \sum_{n=1}^N \sum_{i=1}^3 \|\mathbf{G}_{(i)}^{(n)}\|_* + \frac{\lambda}{2} \|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2 \\ \text{s.t.} \quad & P_{\Omega}(\mathcal{X}) = P_{\Omega}(\mathcal{T}), \end{aligned} \quad (17)$$

where the optimization objectives are the recovered underlying tensor \mathcal{X} and the TR factors $[\mathcal{G}]$, $\lambda > 0$ is the tuning parameter and \mathcal{T} is the input incomplete tensor. Our TRLRF model has two distinctive advantages. Firstly, the low-rank assumption (i.e., nuclear norm) is placed on tensor factors instead of on the original tensor, this greatly reduces the computational complexity of the SVD operation. Secondly, low-rankness of tensor factors can enhance the robustness to rank selection, which can alleviate the burden of searching for better TR-rank and reduce the computational cost in the implementation.

By imposing overlapped nuclear norms on the TRD latent factors, TRLRF can minimize the TR-rank of each mode of the tensor and explore more low-rank structures. However, when the low-rank property in each mode of the underlying tensor is unbalanced (which is usually the case in real-world data), the equal low-rank constraint on each TR factor will become less efficient. Inspired by the previous study of latent nuclear norm (Tomioka and Suzuki 2013), which is a more flexible low-rank constraint than overlapped nuclear norm, we employ model (7) to the TR factors. In this respect, we further decompose each TR factor into a sum of latent components. Under the low-rank regularization of the latent model, the underlying tensor does not need to be low-rank at every mode. Our tensor ring latent nuclear norm (TRLNN) model is formulated as:

$$\begin{aligned}
 & \min_{[\mathcal{G}], \mathcal{X}} \sum_{n=1}^N \sum_{i=1}^3 \|\mathbf{W}_{(i)}^{(n,i)}\|_* + \frac{\lambda}{2} \|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2 \\
 & \text{s.t. } P_{\Omega}(\mathcal{X}) = P_{\Omega}(\mathcal{T}), \mathcal{G}^{(n)} = \sum_{i=1}^3 \mathbf{W}^{(n,i)}, \\
 & n = 1, \dots, N.
 \end{aligned} \tag{18}$$

Similar to the TRLRF model, the TRLNN is also to optimize the TR factors and the underlying tensor simultaneously. From model (18), we can see that the rank along different modes for each TR factor $\mathcal{G}^{(n)}$ is independently regularized by different components $\mathbf{W}^{(n,i)}$, which is well suited to the unbalanced rank scenario. The TRLNN model is considered to be a better setting than TRLRF for a tensor that has unbalanced low-rank structure (Tomioka and Suzuki 2013).

3.3 ADMM solving scheme

The alternating direction method of multipliers (ADMM) (Boyd et al. 2011) is the most commonly-used and efficient algorithm to solve constrained optimization problems. Due to the non-smoothness property of the nuclear norm regularizers of our models, the conventional gradient-descent-based algorithms usually lead to slow convergence rate (sub-linearly), so we apply the ADMM to solve our models. As shown in existing studies (Liu et al. 2013, 2015), we can utilize ADMM to achieve more efficient solving schemes of our models.

3.3.1 TRLRF

To solve the model in (17) by ADMM scheme, because the variables of TRLRF model are inter-dependent, we impose auxiliary variables to simplify the optimization. Thus, the TRLRF model can be rewritten as

$$\begin{aligned}
 & \min_{[\mathcal{M}], [\mathcal{G}], \mathcal{X}} \sum_{n=1}^N \sum_{i=1}^3 \|\mathbf{M}_{(i)}^{(n,i)}\|_* + \frac{\lambda}{2} \|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2, \\
 & \text{s.t. } \mathbf{M}_{(i)}^{(n,i)} = \mathbf{G}_{(i)}^{(n)}, n = 1, \dots, N, i = 1, 2, 3, \\
 & P_{\Omega}(\mathcal{X}) = P_{\Omega}(\mathcal{T}),
 \end{aligned} \tag{19}$$

where $[\mathcal{M}] := \{\mathcal{M}_{(i)}^{(n,i)}\}_{n=1, i=1}^{N,3}$ are the auxiliary variables of $[\mathcal{G}]$. By merging the additional equal constraints of the auxiliary variables into the Lagrangian equation, the augmented Lagrangian function of TRLRF model becomes

$$\begin{aligned}
 & L([\mathcal{G}], \mathcal{X}, [\mathcal{M}], [\mathcal{Y}]) \\
 & = \sum_{n=1}^N \sum_{i=1}^3 (\|\mathbf{M}_{(i)}^{(n,i)}\|_* + \langle \mathcal{Y}^{(n,i)}, \mathcal{M}^{(n,i)} - \mathcal{G}^{(n)} \rangle \\
 & \quad + \frac{\mu}{2} \|\mathcal{M}^{(n,i)} - \mathcal{G}^{(n)}\|_F^2) + \frac{\lambda}{2} \|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2, \\
 & \text{s.t. } P_{\Omega}(\mathcal{X}) = P_{\Omega}(\mathcal{T}),
 \end{aligned} \tag{20}$$

where $[\mathcal{Y}] := \{\mathcal{Y}^{(n,i)}\}_{n=1, i=1}^{N,3}$ are the Lagrangian multipliers, and $\mu > 0$ is a penalty parameter. For $n = 1, \dots, N, i = 1, 2, 3, \mathcal{G}^{(n)}, \mathcal{M}^{(n,i)}$ and $\mathcal{Y}^{(n,i)}$ are each independent, so we

can update them by the updating scheme below. By using (20), the augmented Lagrangian function w.r.t. $\mathcal{G}^{(n)}$ can be simplified as

$$L(\mathcal{G}^{(n)}) = \sum_{i=1}^3 \frac{\mu}{2} \left\| \mathcal{M}^{(n,i)} - \mathcal{G}^{(n)} + \frac{1}{\mu} \mathbf{Y}^{(n,i)} \right\|_F^2 + \frac{\lambda}{2} \left\| \mathcal{X} - \Psi([\mathcal{G}]) \right\|_F^2 + C_{\mathcal{G}}, \tag{21}$$

where the constant $C_{\mathcal{G}}$ consists of other parts of the Lagrangian function which is irrelevant to updating $\mathcal{G}^{(n)}$. This is a least squares problem, so for $n = 1, \dots, N$, $\mathcal{G}^{(n)}$ can be updated by

$$\mathcal{G}_+^{(n)} = \text{fold}_{(2)} \left(\left(\sum_{i=1}^3 (\mu \mathbf{M}_{(2)}^{(n,i)} + \mathbf{Y}_{(2)}^{(n,i)}) \right) + \lambda \mathbf{X}_{\langle n \rangle} \mathbf{G}_{\langle 2 \rangle}^{(\neq n)} (\lambda \mathbf{G}_{\langle 2 \rangle}^{(\neq n)\top} \mathbf{G}_{\langle 2 \rangle}^{(\neq n)} + 3\mu \mathbf{I})^{-1} \right), \tag{22}$$

where $\mathbf{I} \in \mathbb{R}^{R_n^2 \times R_n^2}$ denotes the identity matrix. For $i = 1, 2, 3$, the augmented Lagrangian functions w.r.t. $[\mathcal{M}]$ is expressed as

$$L(\mathcal{M}^{(n,i)}) = \frac{\mu}{2} \left\| \mathcal{M}^{(n,i)} - \mathcal{G}^{(n)} + \frac{1}{\mu} \mathbf{Y}^{(n,i)} \right\|_F^2 + \left\| \mathbf{M}_{(i)}^{(n,i)} \right\|_* + C_{\mathcal{M}}, \tag{23}$$

where $C_{\mathcal{M}}$ is considered as a constant. The above formulation has a closed-form (Cai et al. 2010), which is given by

$$\mathcal{M}_+^{(n,i)} = \text{fold}_{(i)} \left(D_{\frac{1}{\mu}} \left(\mathbf{G}_{(i)}^{(n)} - \frac{1}{\mu} \mathbf{Y}_{(i)}^{(n,i)} \right) \right), \tag{24}$$

where $D_{\beta}(\cdot)$ is the singular value thresholding (SVT) operation, e.g., if \mathbf{USV}^{\top} is the singular value decomposition of matrix \mathbf{A} , then $D_{\beta}(\mathbf{A}) = \mathbf{U} \max\{\mathbf{S} - \beta \mathbf{I}, \mathbf{0}\} \mathbf{V}^{\top}$. The augmented Lagrangian functions w.r.t. \mathcal{X} is given by

$$L(\mathcal{X}) = \frac{\lambda}{2} \left\| \mathcal{X} - \Psi([\mathcal{G}]) \right\|_F^2 + C_{\mathcal{X}}, \tag{25}$$

s.t. $P_{\Omega}(\mathcal{X}) = P_{\Omega}(\mathcal{T})$,

which is equivalent to the tensor decomposition-based model in (4). The expression for \mathcal{X} is updated by inputting the observed values in the corresponding entries, and by approximating the missing entries by the updated TR factors $[\mathcal{G}]$ for every iteration, i.e.,

$$\mathcal{X}_+ = P_{\Omega}(\mathcal{T}) + P_{\bar{\Omega}}(\Psi([\mathcal{G}])), \tag{26}$$

where $\bar{\Omega}$ is the set of indices of missing entries which is a complement to Ω . For $n = 1, \dots, N$ and $i = 1, 2, 3$, the Lagrangian multiplier $\mathbf{Y}^{(n,i)}$ is updated as

$$\mathbf{Y}_+^{(n,i)} = \mathbf{Y}^{(n,i)} + \mu (\mathcal{M}^{(n,i)} - \mathcal{G}^{(n)}). \tag{27}$$

In addition, the penalty term of the Lagrangian functions L is restricted by μ which is also updated for every iteration by $\mu_+ = \max\{\rho\mu, \mu_{max}\}$, where $1 < \rho < 1.5$ is a tuning hyper parameter.

3.3.2 TRLNN

Different from the solving scheme of TRLRF, the TRLNN model dose not need auxiliary variables. We first merge the equal constraint and formulate the augmented Lagrangian function as:

$$\begin{aligned}
 L([\mathcal{G}], \mathcal{X}, [\mathcal{W}], [\mathcal{Y}]) &= \sum_{n=1}^N \left(\sum_{i=1}^3 \|\mathbf{W}_{(i)}^{(n,i)}\|_* + \langle \mathcal{Y}^{(n)}, \sum_{i=1}^3 \mathcal{W}^{(n,i)} - \mathcal{G}^{(n)} \rangle \right. \\
 &\quad \left. + \frac{\mu}{2} \left\| \sum_{i=1}^3 \mathcal{W}^{(n,i)} - \mathcal{G}^{(n)} \right\|_F^2 \right) + \frac{\lambda}{2} \|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2, \\
 \text{s.t. } P_{\Omega}(\mathcal{X}) &= P_{\Omega}(\mathcal{T}).
 \end{aligned} \tag{28}$$

Due to the interdependence of $[\mathcal{G}]$, $[\mathcal{W}]$ and $[\mathcal{Y}]$, we provide the updating scheme of these variables as below. To update $[\mathcal{G}]$, for $n = 1, \dots, N$, model (28) can be rewritten by:

$$\begin{aligned}
 L(\mathcal{G}^{(n)}) &= \frac{\mu}{2} \left\| \sum_{i=1}^3 \mathcal{W}^{(n,i)} - \mathcal{G}^{(n)} + \frac{1}{\mu} \mathcal{Y}^{(n)} \right\|_F^2 \\
 &\quad + \frac{\lambda}{2} \|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2 + C_{\mathcal{G}},
 \end{aligned} \tag{29}$$

where $C_{\mathcal{G}}$ is the irrelevant part of the augmented Lagrangian function to update $\mathcal{G}^{(n)}$ and can be considered as a constant value. In this way, updating $\mathcal{G}^{(n)}$ equals to solving a least squares problem, so $\mathcal{G}^{(n)}$ can be updated by:

$$\begin{aligned}
 \mathcal{G}_+^{(n)} &= \text{fold}_{(2)} \left((\lambda \mathbf{X}_{\langle 2 \rangle} + \mu \sum_{i=1}^3 \mathbf{W}_{(2)}^{(n,i)} \right. \\
 &\quad \left. + \mathbf{Y}_{(2)}^{(n)}) (\lambda \mathbf{G}_{\langle 2 \rangle}^{(\neq n, \top)} + \mu \mathbf{I})^{-1} \right),
 \end{aligned} \tag{30}$$

where $\mathbf{I} \in \mathbb{R}^{R_n^2 \times R_n^2}$ is an identity matrix. Similarly, for $n = 1, \dots, N, i = 1, 2, 3$, in order to update $[\mathcal{W}]$, function (28) can be rewritten by:

$$\begin{aligned}
 L(\mathcal{W}^{(n,i)}) &= \frac{\mu}{2} \|\mathcal{W}^{(n,i)} + \sum_{j=1, j \neq i}^3 \mathcal{W}^{(n,i)} - \mathcal{G}^{(n)}\|_F^2 \\
 &\quad + \frac{1}{\mu} \|\mathcal{Y}^{(n,i)}\|_F^2 + \|\mathbf{W}_{(i)}^{(n,i)}\|_* + C_{\mathcal{W}},
 \end{aligned} \tag{31}$$

where $C_{\mathcal{W}}$ is the variable which is not related to $\mathcal{W}^{(n,i)}$. The formulation has a closed-form solution, which is given by:

$$\mathcal{W}_+^{(n,i)} = \text{fold}_{(i)} \left(D_{\frac{1}{\mu}}(\mathbf{G}_{(i)}^{(n)} - \frac{1}{\mu} \mathbf{Y}_{(i)}^{(n,i)} - \sum_{j=1, j \neq i}^3 \mathcal{W}^{(n,i)}) \right). \tag{32}$$

Next, to update \mathcal{X} , the augmented Lagrangian function (20) can be rewritten by:

$$\begin{aligned}
 L(\mathcal{X}) &= \frac{\lambda}{2} \|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2 + C_{\mathcal{X}}, \\
 \text{s.t. } P_{\Omega}(\mathcal{X}) &= P_{\Omega}(\mathcal{T}),
 \end{aligned} \tag{33}$$

which is a standard model for TR-based completion, and \mathcal{X} can be updated by:

$$\mathcal{X}_+ = P_{\Omega}(\mathcal{T}) + P_{\bar{\Omega}}(\Psi([\mathcal{G}])). \quad (34)$$

Finally, for $n = 1, \dots, N$, the closed-form solution for Lagrangian multipliers $\mathcal{Y}^{(n)}$ is updated by

$$\mathcal{Y}_+^{(n)} = \mathcal{Y}^{(n)} + \mu \left(\sum_{i=1}^3 \mathcal{W}^{(n,i)} - \mathcal{G}^{(n)} \right). \quad (35)$$

The parameter settings and the algorithm details will be provided in the experiment section.

3.4 Complexity analysis and convergence analysis

We assume to recover a tensor $\mathcal{X} \in \mathbb{R}^{I \times I \times \dots \times I}$ by our models with TR-rank $R_1 = R_2 = \dots = R_N = R$ for simplicity. The computational complexity of updating $[\mathcal{M}]$ for TRLRF and $[\mathcal{W}]$ for TRLNN are mainly spent on the SVD calculation, which are $\mathcal{O}(NIR^3 + I^2R^2)$ equally. The complexity of HaLRTC (Liu et al. 2013) is $\mathcal{O}(NI^{N+1})$ which is much higher than our models as it conducts the SVD on the whole tensor.

Moreover, the main computational complexity of our algorithms is cost by updating $[\mathcal{G}]$. In (22) and (30), the multiplication $\mathbf{X}_{<n>} \mathbf{G}_{<2>}^{(\neq n)}$ costs $\mathcal{O}(R^2I^N)$ complexity, and calculating the inverse of the matrix of size $R^2 \times R^2$ costs $\mathcal{O}(R^6)$ complexity. The calculation repeats N times in every iteration of the algorithm, so the update of $[\mathcal{G}]$ in the two models both cost $\mathcal{O}(NR^2I^N + NR^6)$ complexity. It is comparable to the computational complexity of TRALS (Wang et al. 2017) which is $\mathcal{O}(PNR^4I^N + NR^6)$ where P denotes the observation rate. However, the TRALS applies slice-wise update scheme, and our algorithms apply the factor-wise update scheme which needs much fewer loops to update all the TR factors. Because of the representation capability of TRD, the high power in R is not an issue for the complexity of TR-based algorithms. The TR-rank of TR-based algorithms can always be set as a small value. Another desirable property of TR-rank regularization of our algorithms is that it can speed up the model selection process in practice, and thus the computational cost of our algorithm can be greatly reduced.

It should be noted that our TRLRF and TRLNN models are non-convex, so the convergence to the global minimum cannot be theoretically guaranteed. However, the convergence of our algorithms can be verified empirically (see experiment details in Fig. 2). By applying the synthetic tensor which has the TR structure, we conduct the completion experiment by our algorithms in different TR-rank and different hyper-parameter λ . Each independent experiment is conducted 100 times and the average results are shown in the graphs. From the figure, we can see that the convergence of our algorithms is fast and stable. Moreover, the extensive experimental results in the next section also illustrate the stability and effectiveness of our algorithms.

4 Experimental results

In the experiment section, we firstly testify the rank robustness of our algorithms and the difference of our algorithms by the simulation experiment. Then by numerous benchmark and real-world data, we testify the performance of our algorithms in many situations and compare with the other low-rank approximation algorithms. Moreover, we consider to set

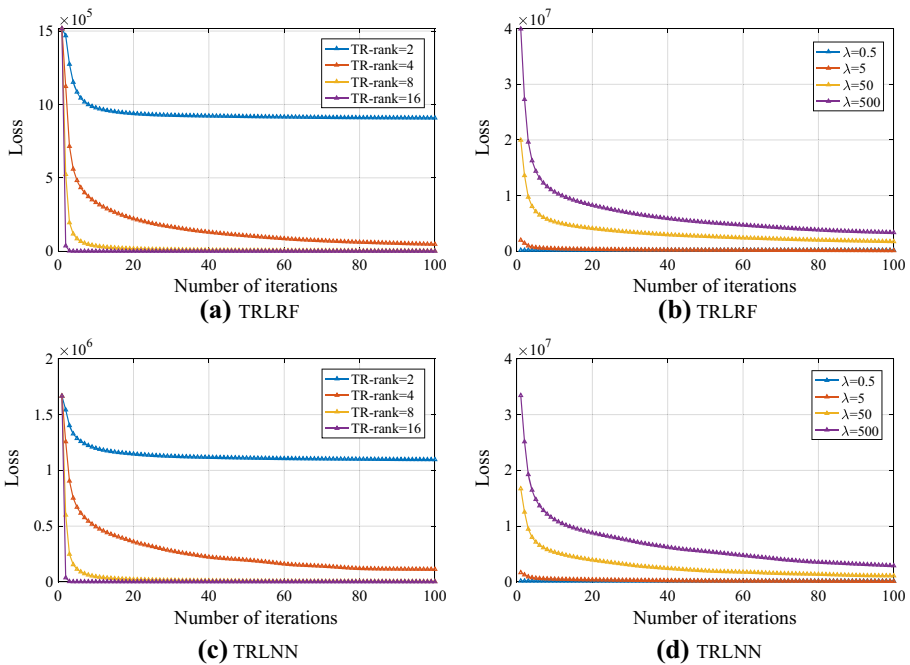


Fig. 2 Illustration of convergence for TRLRF and TRLNN under different hyper-parameter choices. A synthetic tensor with TR structure (size $7 \times 8 \times 7 \times 8$ with TR-rank $\{4,4,4,4\}$, missing rate 0.5) is tested. The experiment records the change of the objective function values along the number of iterations. **a** and **b** show the convergence curves of TRLRF under different TR-rank and λ respectively. The convergence curves of TRLNN are presented in **(c)** and **(d)**

two optimization stopping conditions: (i) maximum number of iterations k_{max} and (ii) the difference between two iterations (i.e., $\|\mathcal{X} - \mathcal{X}_{last}\|_F / \|\mathcal{X}\|_F$) which is thresholded by the tolerance tol . The implementation process and hyper-parameter selection of TRLRF and TRLNN are summarized in Algorithm 1.

Algorithm 1. Solving scheme and parameter settings of TRLRF and TRLNN.

- 1: **Input:** incomplete observation $P_{\Omega}(\mathcal{T})$, initial TR-rank $\{R_n\}_{n=1}^N$.
- 2: **Initialization:** For $n = 1, \dots, N, i = 1, 2, 3$, random sample $\mathcal{G}^{(n)}$ by Gaussian distribution $N \sim (0, 1)$, $[\mathcal{Y}] = 0, [\mathcal{M}] = 0, [\mathcal{W}] = 0, \lambda = 5, \mu^0 = 1, \mu_{max} = 10^2, \rho = 1.01, tol = 10^{-6}, k = 0, k_{max} = 300$.
- 3: **For** $k = 1$ to k_{max} **do**
- 4: $\mathcal{X}_{last} = \mathcal{X}$.
- 5: **For** TRLRF, update the variables by (22), (24), (26), (27).
For TRLNN, update the variables by (30), (32), (34), (35).
- 6: $\mu = \max(\rho\mu, \mu_{max})$
- 7: **If** $\|\mathcal{X} - \mathcal{X}_{last}\|_F / \|\mathcal{X}\|_F < tol$, **break**
- 8: **End for**
- 9: **Output:** completed tensor \mathcal{X} and TR factors $[\mathcal{G}]$.

Moreover, for performance evaluation, the relative square error (RSE) and peak signal-to-noise ratio (PSNR) are adopted for the evaluation of the completion results. RSE is calculated

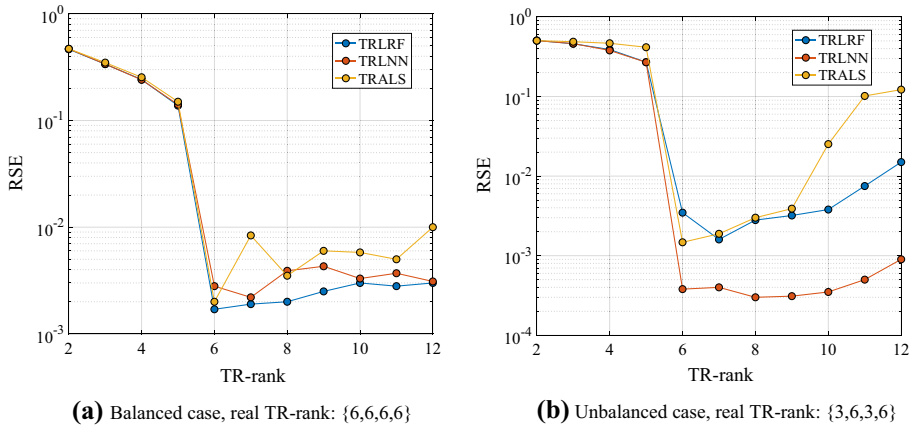


Fig. 3 Completion results of three TR-based algorithms with the increase of the selected TR-rank. Each element of the prescribed TR-rank is set identically in the algorithms, and the real TR-rank of (a) and (b) are balance and unbalance, respectively

by $RSE = \|\mathcal{T}_{real} - \mathcal{Y}\|_F / \|\mathcal{T}_{real}\|_F$, where \mathcal{T}_{real} is the real tensor with full observations, \mathcal{Y} is the completed tensor. PSNR is obtained by $PSNR = 10 \log_{10}(255^2 / MSE)$, where MSE is deduced by $MSE = \|\mathcal{X} - \mathcal{Y}\|_F^2 / \text{num}(\mathcal{X})$, and $\text{num}(\cdot)$ denotes the total number of the elements of the tensor. The algorithms are implemented on Matlab software and all the computations are conducted by using a Mac computer with Intel Core i7 and 64GB DDR3 memory.¹

4.1 Synthetic data

In the synthetic data experiment, we mainly aim to show the difference between TRLRF and TRLNN. The two TR-structured synthetic tensors are of size $12 \times 12 \times 12 \times 12$ with 30% random missing entries. For the first and the second synthetic tensors, the real TR-rank are set as $\{6, 6, 6, 6\}$ and $\{3, 6, 3, 6\}$, respectively. Three TR-based algorithms [i.e., TRLRF, TRLNN and TRALS (Wang et al. 2017)] are used to recover the two incomplete TR-structured tensors which own balanced TR-rank and unbalanced TR-rank, respectively. The TRALS algorithm is considered as the baseline because it cannot tune the TR-rank. With the increase of the prescribed TR-rank, the completion results are shown in Fig. 3. From Fig. 3a, we can see that TRALS obtains its best performance when the prescribed TR-rank equals the real rank of the synthetic tensor but it becomes overfitting when the prescribed TR-rank goes up. On the other hand, the performance of TRLRF and TRLNN are relatively stable when the prescribed TR-rank is increased over the real-rank. TRLRF performs better than TRLNN due to the strong low-rank regularization on each mode of the TR factors. However, in Fig. 3b, when the elements of the real TR-rank is unbalanced, TRLRF becomes less efficient than TRLNN when the prescribed TR-rank is $\{6, 6, 6, 6\}$. In this situation, only a subset of the modes of the TR-rank needs to be regularized. When the TR-rank continues to increase, the TRLRF and TRLNN show robustness to the rank-increasing, while the performance of TRALS shows a sharp decrease due to the overfitting problem.

¹ The Matlab code of our algorithms is available at www.github.com/yuanlonghao/TRLRF.



Fig. 4 The eight benchmark images of size $256 \times 256 \times 3$

4.2 Benchmark image inpainting

In this section, we adopt eight widely-used benchmark RGB images (Fig. 4) to validate the completion performance of our TRLRF and TRLNN. The original images can be considered as tensors of size $256 \times 256 \times 3$. The first experiment is conducted to demonstrate the rank-robustness performance of our algorithms. We treat TRALS and TRWOPT (Yuan et al. 2018) as the baseline because their TR-rank cannot be tuned automatically. We test these three algorithms on the “Lena” image with 80% random missing which is the case that the TRD-based algorithms are prone to be overfitting. The TR-rank for each independent experiment is set as $R = R_1 = R_2 = R_3$ and $R = \{2, 4, 6, 8, 10, 12\}$. Figure 5 shows the visual inpainting results of the compared algorithms when the TR-rank increases. We can see that when the TR-rank is $\{2, 2, 2\}$, all the algorithms show distinct underfitting, and when the TR-rank is $\{4, 4, 4\}$, all the algorithms show relatively good results due to the proper rank selection. However, when the TR-rank continues to increase, TRALS and TRWOPT show performance decrease due to the overfitting problem while our TRLRF and TRLNN are robust to rank increase and obtain even higher performance than the low TR-rank cases. The experiment results are in accordance with the synthetic data experiments in the previous section.

The next experiment is to testify the inpainting performance of our algorithms compared with other related low-rank-based algorithms. In addition to comparing the TR-based algorithms, the CP decomposition based TenALS and FBCP (Zhao et al. 2015), the Tucker-rank minimization based HaLRTC (Liu et al. 2013; Zhao et al. 2015) and the tensor SVD scheme based t-SVD (Zhang et al. 2014) are also included in our comparison. We test the eight algorithms on all the eight benchmark images with different missing rates: $\{0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95\}$. According to the parameter-tuning suggestions from each paper, the hyper-parameters are respectively tuned for the compared algorithm to try to exhibit their best performance.

Figure 6a, b show the average RSE and PSNR results of the eight images, respectively. The TRLRF and TRLNN show better results than other algorithms in most of the cases. The completion performance of all the algorithms decreases w.r.t. the increase of the missing rate. In particular, when the missing rate reaches 0.9 and 0.95, the per-

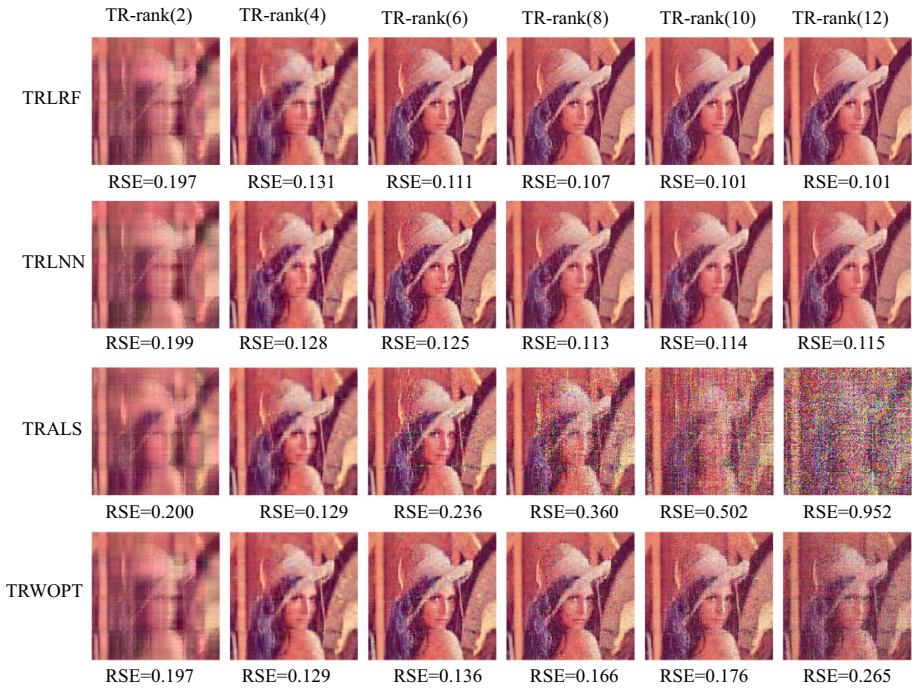


Fig. 5 Visual completion results of the reshaped RGB image “Lena” of size $256 \times 256 \times 3$

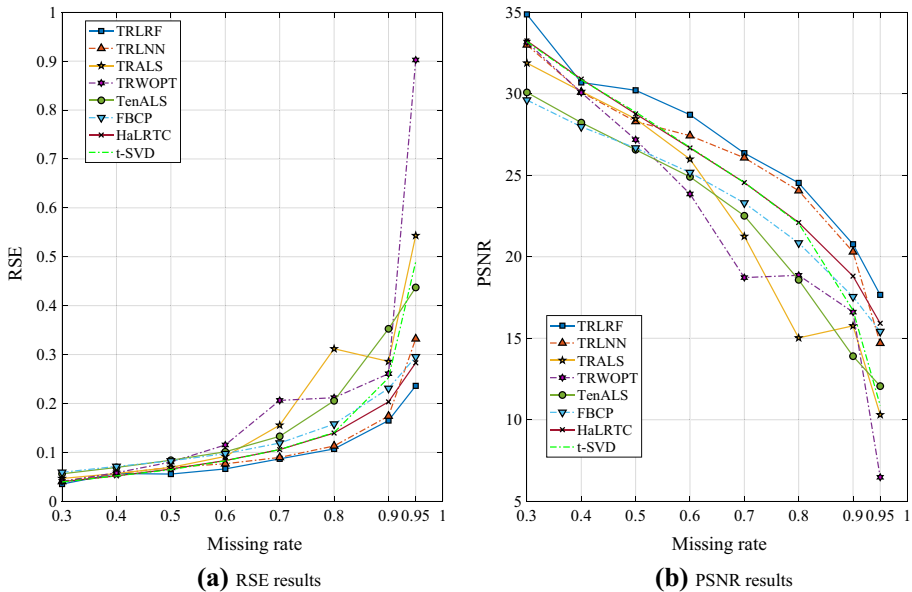


Fig. 6 Average completion results of the eight RGB images of size $256 \times 256 \times 3$ with different missing rate. The smaller RSE values and the larger PSNR values indicate higher performance

formance of most algorithms falls drastically. It should be noted that finding the best TR-rank to obtain the best completion results is very laborious, and tuning the rank for each image in different missing rate is not practical in real applications, especially for TR-based algorithms which need to tune multilinear rank. However, the rank selection is much easier for our proposed algorithms because the performance of TRLRF and TRLNN are fairly stable even though the TR-rank is selected from a wide range. As for running time, the average running time for a single image of each algorithm is 11.2, 10.7, 37.8, 22.9, 14.5, 20.3, 8.5, 13.4 (seconds) respectively, which shows the efficiency of our algorithms.

4.3 Hyperspectral image completion

A hyperspectral image (HSI) of size $200 \times 200 \times 80$ which records an area of the urban landscape is tested in this section.² In order to test the performance of the algorithms on higher-order tensors, the HSI data is also reshaped to higher-order tensors, which makes it easy to find more low-rank features of the data. We compare our TRLRF and TRLNN to the other five tensor completion algorithms in 3rd-order tensor ($200 \times 200 \times 80$) case and 8th-order tensor ($8 \times 5 \times 5 \times 8 \times 5 \times 8 \times 5 \times 10$) case. The higher-order tensors are generated from original HSI data by directly reshaping it to the specified order and size. Moreover, we test two missing patterns: the dead line missing which often happens in HSI recording process (He et al. 2015; Zhang et al. 2014), and the 90% random missing case to test the reconstruction performance of the algorithms.

The compared algorithms are TRALS (Wang et al. 2017), TTSGD (Yuan et al. 2019), FBCP (Zhao et al. 2015), HaLRTC (Liu et al. 2013) and t-SVD Zhang et al. (2014) which apply the low-rank approximation of TR-rank, TT-rank, CP-rank, Tucker-rank and tensor tubal-rank, respectively. It should be noted that in order to make a clear comparison of the algorithms which apply different kinds of rank, we set the prescribed rank equally for all the algorithms in each missing case. All the tuning parameters of every algorithm were set according to the previous experiments. The completion performance of RSE, PSNR and running time are listed in Table 1 and the visual completion results are shown in Fig. 7. From the completion results we can see, our TRLRF provides the best recovery performance for the HSI image in almost all the cases, and TRLNN also performs well in all the completion tasks. In the case of 3rd-order dead line missing, most of the algorithms perform well. However, when the tensor is reshaped to 8th-order, the performance of FBCP and HaLRTC decreases. Moreover, in the 90% random missing case, the TR-based and TT-based algorithms show steady performance from 3rd-order tensor to 8th-order tensor, while the other algorithms show a sharp performance decrease and HaLRTC even fails to complete the 8th-order 90% missing case. As for running time, our algorithms own an ordinary speed but a lot faster than TRALS. Moreover, the running time for 3rd-order tensors is much less than the running time of 8th-order tensors for most algorithms, even though the rank is set lower for the high-order tensor cases. This is because the processing of high-order tensors always leads to more computational loops and it will cost more time than low-order tensors.

² http://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes.

Table 1 HSI completion results (RSE, PSNR and running time) under two different tensor orders and two missing situations. The PSNR is calculated by the three color bands: 80, 34 and 9

Algorithms		TRLRF	TRLNN	TRALS	TTSGD	FBCP	HaLRTC	t-SVD
Rank tuning		✓	✓			✓	✓	✓
deadline missing (3rd-order)	RSE	0.0840	0.0874	0.109	0.403	0.101	0.128	0.122
	PSNR	34.55	33.93	32.02	20.89	32.87	30.83	31.27
	time	167.4	128.1	619.2	151.1	99.3	106.2	71.1
0.9 random missing (3rd-order)	RSE	0.0806	0.165	0.157	0.328	0.388	0.345	0.268
	PSNR	31.50	27.48	28.17	22.67	21.15	22.17	21.79
	time	170.8	198.6	289.4	418.0	71.5	105.7	128.6
deadline missing (8th-order)	RSE	0.0924	0.0996	0.101	0.309	0.133	0.136	0.0956
	PSNR	33.58	31.91	32.71	23.19	30.45	30.40	33.26
	time	583.0	560.2	616.2	126.4	221.45	111.1	374.9
0.9 random missing (8th-order)	RSE	0.109	0.118	0.127	0.251	0.401	0.910	0.262
	PSNR	31.03	30.43	30.10	24.78	20.87	13.78	24.45
	time	337.5	328.9	480.9	188.5	145.36	68.9	380.9

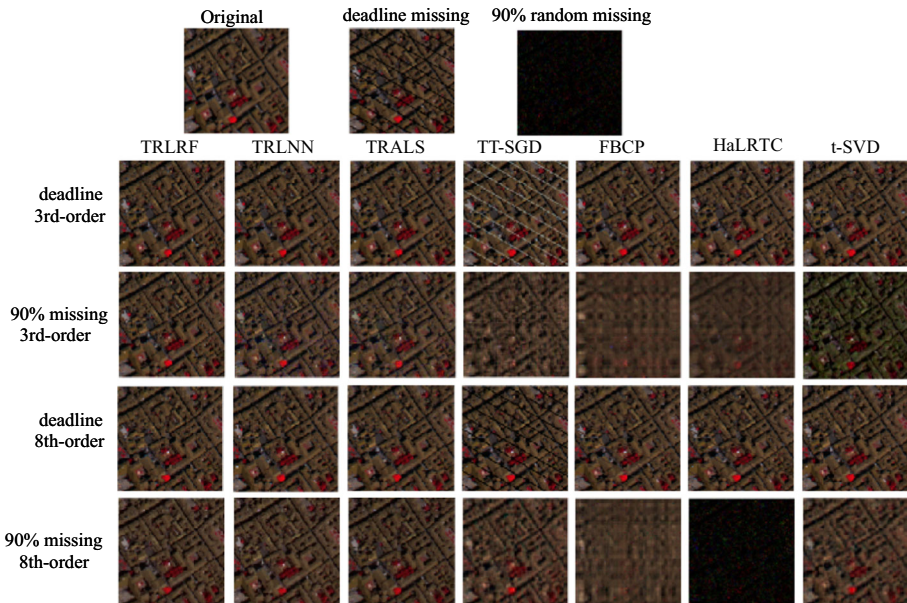


Fig. 7 Visual results of the HSI completion tasks. The the color bands 80, 34, 9 are chosen for the image display

5 Conclusion

In this paper, we make the virtue of applying both the nuclear norm regularization and tensor ring decomposition, to formulate a new tensor completion approach that achieves tensor completion and decomposition simultaneously. By exploiting the rank relationship between the tensor and the TR latent space, we employ low-rank constraints on the TR

factors by two different low-rank regularizations, thus providing more robust completion performance. We develop the ADMM solving scheme to optimize the proposed models, resulting in satisfying running time. The experimental results on various datasets not only verify the robustness and efficiency of the proposed algorithms but also demonstrate the superior performance of our method against the compared algorithms. Furthermore, it is expected that the idea of imposing rank regularization constraint on tensor latent space can be extended to various tensor decomposition models in order to develop more efficient and robust algorithms.

Acknowledgements This work was supported by JSPS KAKENHI (Grant Nos. 17K00326, 18K04178), JST CREST (Grant No. JPMJCR1784).

References

- Bengua, J. A., Phien, H. N., Tuan, H. D., & Do, M. N. (2017). Efficient tensor completion for color image and video recovery: Low-rank tensor train. *IEEE Transactions on Image Processing*, 26(5), 2466–2479.
- Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1), 1–122.
- Cai, J. F., Candès, E. J., & Shen, Z. (2010). A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4), 1956–1982.
- Chen, Y., Jin, X., Kang, B., Feng, J., & Yan, S. (2018). Sharing residual units through collective tensor factorization to improve deep neural networks. In *IJCAI* (pp. 635–641).
- Cichocki, A., Mandic, D., De Lathauwer, L., Zhou, G., Zhao, Q., Caiafa, C., et al. (2015). Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2), 145–163.
- Du, B., Zhang, M., Zhang, L., Hu, R., & Tao, D. (2017). PLTD: Patch-based low-rank tensor decomposition for hyperspectral images. *IEEE Transactions on Multimedia*, 19(1), 67–79.
- He, W., Zhang, H., Zhang, L., & Shen, H. (2015). Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration. *IEEE Transactions on Geoscience and Remote Sensing*, 54(1), 178–188.
- Hu, Y., Yi, X., & Davis, L.S. (2015). Collaborative fashion recommendation: A functional tensor factorization approach. In *Proceedings of the 23rd ACM international conference on Multimedia* (pp. 129–138). ACM.
- Khoo, Y., Lu, J., & Ying, L. (2017). Efficient construction of tensor ring representations from sampling. [arXiv:1711.00954](https://arxiv.org/abs/1711.00954).
- Kolda, T. G., & Bader, B. W. (2009). Tensor decompositions and applications. *SIAM Review*, 51(3), 455–500.
- Liu, J., Musialski, P., Wonka, P., & Ye, J. (2013). Tensor completion for estimating missing values in visual data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1), 208–220.
- Liu, Y., Long, Z., & Zhu, C. (2019). Image completion using low tensor tree rank and total variation minimization. *IEEE Transactions on Multimedia*, 21(2), 338–350.
- Liu, Y., Shang, F., Fan, W., Cheng, J., & Cheng, H. (2016). Generalized higher order orthogonal iteration for tensor learning and decomposition. *IEEE Transactions on Neural Networks and Learning Systems*, 27(12), 2551–2563.
- Liu, Y., Shang, F., Jiao, L., Cheng, J., & Cheng, H. (2015). Trace norm regularized CANDECOMP/PARAFAC decomposition with missing data. *IEEE Transactions on Cybernetics*, 45(11), 2437–2448.
- Long, Z., Liu, Y., Chen, L., & Zhu, C. (2018). Low rank tensor completion for multiway visual data. *Signal Processing*.
- Novikov, A., Podoprikin, D., Osokin, A., & Vetrov, D. P. (2015). Tensorizing neural networks. In *Advances in neural information processing systems* (pp. 442–450).
- Oseledets, I. V. (2011). Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5), 2295–2317.
- Sidiropoulos, N. D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E. E., & Faloutsos, C. (2017). Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13), 3551–3582.
- Song, Q., Ge, H., Caverlee, J., & Hu, X. (2019). Tensor completion algorithms in big data analytics. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 13(1), 6.
- Tomioka, R., & Suzuki, T. (2013). Convex tensor decomposition via structured Schatten norm regularization. In *Advances in neural information processing systems* (pp. 1331–1339).

- Wang, W., Aggarwal, V., & Aeron, S. (2017). Efficient low rank tensor ring completion. In *2017 IEEE international conference on computer vision (ICCV)* (pp. 5698–5706). IEEE.
- Yokota, T., Erem, B., Guler, S., Warfield, S. K., & Hontani, H. (2018). Missing slice recovery for tensors using a low-rank model in embedded space. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8251–8259).
- Yokota, T., Zhao, Q., & Cichocki, A. (2016). Smooth PARAFAC decomposition for tensor completion. *IEEE Transactions on Signal Processing*, *64*(20), 5423–5436.
- Yu, J., Li, C., Zhao, Q., & Zhou, G. (2019). Tensor-ring nuclear norm minimization and application for visual data completion. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 3142–3146). IEEE.
- Yuan, L., Cao, J., Zhao, X., Wu, Q., & Zhao, Q. (2018). Higher-dimension tensor completion via low-rank tensor ring decomposition. In *Proceedings, APSIPA annual summit and conference* (Vol. 2018, pp. 12–15).
- Yuan, L., Li, C., Mandic, D., Cao, J., & Zhao, Q. (2018). Rank minimization on tensor ring: A new paradigm in scalable tensor decomposition and completion. [arXiv:1805.08468](https://arxiv.org/abs/1805.08468).
- Yuan, L., Zhao, Q., Gui, L., & Cao, J. (2019). High-order tensor completion via gradient-based optimization under tensor train format. *Signal Processing: Image Communication*, *73*, 53–61.
- Yuan, M., & Zhang, C. H. (2016). On tensor completion via nuclear norm minimization. *Foundations of Computational Mathematics*, *16*(4), 1031–1068.
- Zhang, H., He, W., Zhang, L., Shen, H., & Yuan, Q. (2014). Hyperspectral image restoration using low-rank matrix recovery. *IEEE Transactions on Geoscience and Remote Sensing*, *52*(8), 4729–4743.
- Zhang, Z., Ely, G., Aeron, S., Hao, N., & Kilmer, M. (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* (pp. 3842–3849).
- Zhao, Q., Caiafa, C. F., Mandic, D. P., Chao, Z. C., Nagasaka, Y., Fujii, N., et al. (2012). Higher order partial least squares (HOPLS): A generalized multilinear regression method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*(7), 1660–1673.
- Zhao, Q., Sugiyama, M., Yuan, L., & Cichocki, A. (2019). Learning efficient tensor representations with ring-structured networks. In *ICASSP 2019-2019 IEEE international conference on acoustics, speech and signal processing (ICASSP)* (pp. 8608–8612). IEEE.
- Zhao, Q., Zhang, L., & Cichocki, A. (2015). Bayesian CP factorization of incomplete tensors with automatic rank determination. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(9), 1751–1763.
- Zhao, Q., Zhou, G., Xie, S., Zhang, L., & Cichocki, A. (2016). Tensor ring decomposition. [arXiv:1606.05535](https://arxiv.org/abs/1606.05535).
- Zhao, Q., Zhou, G., Zhang, L., Cichocki, A., & Amari, S. I. (2015). Bayesian robust tensor factorization for incomplete multiway data. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(4), 736–748.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.