

Higher-dimension Tensor Completion via Low-rank Tensor Ring Decomposition

Longhao Yuan^{*†}, Jianting Cao^{§*}, Xuyang Zhao^{*†}, Qiang Wu^{¶||} and Qibin Zhao^{†‡}

^{*} Graduate School of Engineering, Saitama Institute of Technology, Japan

[†] Tensor Learning Unit, RIKEN Center for Advanced Intelligence Project (AIP), Japan

[‡] School of Automation, Guangdong University of Technology, China

[§] School of Computer Science and Technology, Hangzhou Dianzi University, China

[¶] School of Information Science and Engineering, Shandong University, China

^{||} Institute of Brain and Brain-Inspired Science, Shandong University, China

Abstract—The problem of incomplete data is common in signal processing and machine learning. Tensor completion algorithms aim to recover the incomplete data from its partially observed entries. In this paper, taking advantages of high compressibility and flexibility of recently proposed tensor ring (TR) decomposition, we propose a new tensor completion approach named tensor ring weighted optimization (TR-WOPT). It finds the latent factors of the incomplete tensor by gradient descent algorithm, then the latent factors are employed to predict the missing entries of the tensor. We conduct various tensor completion experiments on synthetic data and real-world data. The simulation results show that TR-WOPT performs well in various high-dimension tensors. Furthermore, image completion results show that our proposed algorithm outperforms the state-of-the-art algorithms in many situations. Especially when the missing rate of the test images is high (e.g., over 0.9), the performance of our TR-WOPT is significantly better than the compared algorithms.

I. INTRODUCTION

Tensors are high-dimension representations of vectors and matrices. Many kinds of data in real-world, for example, color images ($length \times width \times RGB\ channels$), videos ($length \times width \times RGB\ channels \times time$) and electroencephalography (EEG) signals ($magnitude \times trails \times time$) are more than two dimensions. Usually, the high-dimension data is first transformed to vector or matrix, only then can the data be applied to traditional algorithms. In that way, the adjacent structure information of the original data will be lost, thus lead to redundant space cost and low-efficiency computation [20]. Tensor representation can retain the high dimension of data and solve the above problems. Tensor has been studied for more than a century, many methodologies have been proposed [14]. Moreover, tensor has been applied in various research field such as signal processing [6], machine learning [2], data completion [1], brain-computer interface (BCI) [16], etc..

CANDECOMP/PARAFAC (CP) decomposition [3] and Tucker decomposition [22] are the most popular and classic tensor decomposition models, of which many theories and applications have been proposed [21]. In recent years, a new

theory system named tensor network has drawn people's attention and becomes a promising aspect of tensor methodology [5], [7]. One of the most representative tensor decomposition models of tensor network is matrix product state (MPS), which is also known as tensor train (TT) decomposition [19]. TT decomposition shows high data compression ability and computational efficiency. One of the most significant features of TT decomposition is that it can overcome the "curse of dimensionality", i.e., for an N -dimension tensor, the number of parameters of Tucker decomposition is exponential in N , while the number of parameters of TT decomposition is linear in N . Although CP decomposition achieves high compression ability and the number of parameters is also linear in N , finding the optimal latent factors of CP decomposition is very difficult. Recently, a new tensor decomposition model named tensor ring (TR) decomposition which is more generalized than TT decomposition has been proposed [27]. Tensor ring owns all the dominant properties TT has and it has a more flexible model constraint than tensor train. TR relaxes the rank constraint of TT, thus lead to more interesting properties, e.g., enhanced compression ability, interpretability of latent factors and rotational invariance ability of latent factors [25]. Several papers have studied and applied TR decomposition in machine learning fields [23], [4], in which TR shows promising aspects in representation ability and computational efficiency.

Tensor completion is to recover an incomplete tensor from the partially observed entries of the tensor, which has been applied in various completion problems such as image/video completion [15], [28], compressed sensing [10], link prediction [18], recommendation system [13], etc.. The theoretical key point of matrix completion and tensor completion is the low-rank assumption of data. There exists strong theoretical support and various solutions for solving the low-rank problem of matrices, and the most studied convex relaxation of low-rank matrix is nuclear norm [12]. However, determining the rank of a tensor is an NP-hard problem [11]. To solve this problem, there are mainly two types of tensor completion methods, "rank minimization based" approach and "tensor decomposition based" approach [17]. The first approach formulates the convex surrogate models of low-rank tensors. The most representative model of this kind of tensor completion

Corresponding authors: Jianting Cao (cao@sit.ac.jp) and Qibin Zhao (qibin.zhao@riken.jp).

approach employs low-rank constraints on the matricization of every mode of the incomplete tensor [15]. For an N -dimension incomplete tensor $\mathcal{T}$, the low-rank tensor completion model is formulated by:

$$\min_{\mathcal{X}} \sum_{n=1}^N \|\mathbf{X}_{(n)}\|_*, \text{ s.t. } P_{\Omega}(\mathcal{X}) = P_{\Omega}(\mathcal{T}), \quad (1)$$

where $\mathcal{X}$ is the low-rank approximation tensor, $\|\cdot\|_*$ is the nuclear norm, and $P_{\Omega}(\mathcal{T})$ denotes the entries w.r.t. the set of indices of observed entries represented by Ω . The missing entries of $\mathcal{T}$ is approximated by $\mathcal{X}$ and the rank of the completed tensor $\mathcal{X}$ is determined automatically. Different from “rank minimization based” approach, the “tensor decomposition based” approach do not find the low-rank tensor directly, instead, it firstly finds the tensor decomposition of the incomplete data by observed entries, then the latent factors are used to predict the missing entries. This kind of approach sets the rank of tensor decomposition manually, and the optimization model is given below:

$$\min_{\{\mathcal{G}^{(n)}\}_{n=1}^N} \|\mathcal{W} * (\mathcal{T} - \mathcal{X}(\{\mathcal{G}^{(n)}\}_{n=1}^N))\|_F^2, \quad (2)$$

where $\|\cdot\|_F$ is the Frobenius norm, $\{\mathcal{G}^{(n)}\}_{n=1}^N$ is the sequence of latent factors under consideration and $\mathcal{X}(\{\mathcal{G}^{(n)}\}_{n=1}^N)$ is the tensor approximated by the latent factors. $\mathcal{W}$ is a weight tensor which is the same size as the incomplete tensor, and every entry of $\mathcal{W}$ meets:

$$w_{i_1 i_2 \dots i_N} = \begin{cases} 0 & \text{if } y_{i_1 i_2 \dots i_N} \text{ is a missing entry,} \\ 1 & \text{if } y_{i_1 i_2 \dots i_N} \text{ is an observed entry.} \end{cases} \quad (3)$$

More explanations of notations can be found in Section II.A. Based on different tensor decomposition models, various “tensor decomposition based” approaches have been proposed, e.g., CP weighted optimization [1], weighted tucker [9] and TT weighted optimization [24]. All the methods aim to find the specific structure of the incomplete data by different kinds of tensor decompositions. However, CP, Tucker and TT based WOPT algorithms apply tensor decomposition models which lack of flexibility, this may lead to bad convergence when considering different kinds of data. TR decomposition model is much more flexible, so TR-based WOPT method can get a better approximation of incomplete data, thus provide better completion results.

In this paper, we propose a novel tensor completion method which shows good performance in various experimental situations. The main works of this paper are listed below:

- Based on the recently proposed TR decomposition, we propose a new tensor completion algorithm named tensor ring weighted optimization (TR-WOPT).
- The TR latent factors are optimized by gradient descent method and then they are used to predict the missing entries of the incomplete tensor.
- We conduct several simulation experiments and real-world data experiments. The experiment results show that our method outperforms the state-of-the-art tensor completion algorithms in various situations. We also find

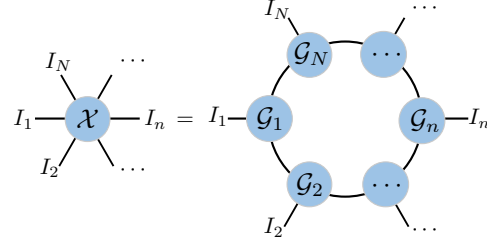


Figure 1. TR decomposition.

that our method is robust to tensor dimension. When the image data is tensorized to a proper high-dimension, the performance of our method can be enhanced.

II. PRELIMINARIES

A. Notations

In this paper, a scalar is denoted by a normal lowercase or capital letter, e.g., $x \in \mathbb{R}$ and $X \in \mathbb{R}$. A vector is denoted by a boldface lowercase letter, e.g., $\mathbf{x} \in \mathbb{R}^I$. A matrix is denoted by a boldface capital letter, e.g., $\mathbf{X} \in \mathbb{R}^{I \times J}$. A tensor of dimension $N \geq 3$ is denoted by Euler script letters, e.g., $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$. For $n = 1, \dots, N$, $\{\mathcal{X}^{(n)}\}_{n=1}^N := \{\mathcal{X}^{(1)}, \mathcal{X}^{(2)}, \dots, \mathcal{X}^{(N)}\}$ is defined as a tensor sequence, and $\mathcal{X}^{(n)}$ is the n th tensor of the sequence. The representations of matrix sequence and vector sequence are denoted in the same way. An element of tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ of index $\{i_1, i_2, \dots, i_N\}$ is denoted by $x_{i_1 i_2 \dots i_N}$ or $\mathcal{X}(i_1, i_2, \dots, i_N)$. One type of the mode- n matricization (unfolding) of tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is denoted by $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times I_1 \dots I_{n-1} I_{n+1} \dots I_N}$. Another type of mode- n matricization of tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is denoted by $\mathbf{X}_{<n>} \in \mathbb{R}^{I_n \times I_{n+1} \dots I_N I_1 \dots I_{n-1}}$. The inner product of two tensors $\mathcal{X}, \mathcal{Y}$ with the same size $\mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is defined as $\langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{i_1} \sum_{i_2} \dots \sum_{i_N} x_{i_1 i_2 \dots i_N} y_{i_1 i_2 \dots i_N}$. Furthermore, the Frobenius norm of $\mathcal{X}$ is defined by $\|\mathcal{X}\|_F = \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}$. The Hadamard product is denoted by ‘ $*$ ’ and it is an element-wise product of vectors, matrices or tensors of same sizes. For example, given tensor $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, $\mathcal{Z} = \mathcal{X} * \mathcal{Y}$, then $\mathcal{Z} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, and $z_{i_1 i_2 \dots i_N} = x_{i_1 i_2 \dots i_N} y_{i_1 i_2 \dots i_N}$.

B. Tensor ring decomposition

Tensor ring (TR) decomposition is the generalization of tensor-train (TT) decomposition. It represents a tensor by circular multilinear products over a sequence of latent factors of lower-dimension [27]. The latent factors are named core tensors. All of the core tensors of TR decomposition are three-dimension tensors, and are denoted by $\mathcal{G}^{(n)} \in \mathbb{R}^{R_1 \times I_n \times R_{n+1}}$, $n = 1, \dots, N$, where $R_1, \dots, R_N$ denotes the TR-rank which controls the model complexity of TR decomposition. The representation regulation of tensor network in monography [5] is adopted and the diagram of TR decomposition is shown in Figure 1. Similar as TT, the TR decomposition linearly scales the tensor dimension, and thus it can overcome the “curse of

dimensionality". The TR decomposition relaxes the rank constraints on the first and last core tensors of TT to $R_1 = R_{N+1}$, while the original constraints on TT decomposition is rather stringent, i.e., $R_1 = R_{N+1} = 1$. The rank constraints of TT is because of the relation of core tensors and original tensor has to be deduced by multiple multiplication operation. TR applies trace operation, so all the core tensors can be constrained to be third-dimension equivalently. Therefore, TR is considered as a linear combination of TT and it offers a more powerful and generalized representation ability than TT decomposition. The element-wise relation of the TR core tensors and the original tensor is given by:

$$\mathcal{X}(i_1, i_2, \dots, i_N) = \text{Trace} \left(\prod_{n=1}^N \mathbf{G}^{(n)}(i_n) \right), \quad (4)$$

where $\text{Trace}(\cdot)$ is the matrix trace operator, $\mathbf{G}^{(n)}(i_n) \in \mathbb{R}^{R_n \times R_{n+1}}$ is the i_n th mode-2 slice matrix of $\mathcal{G}^{(n)}$, which also can be denoted by $\mathcal{G}^{(n)}(:, i_n, :)$. Every element of the tensor can be calculated by the trace of the matrices multiple multiplication of the according mode-2 slice of every TR core tensors.

III. TENSOR RING WEIGHTED OPTIMIZATION

Based on TR decomposition, we propose the TR-WOPT algorithm which is illustrated as follows. Define $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ is the incomplete tensor with missing entries filled with zero, $\mathcal{X}(\{\mathcal{G}^{(n)}\}_{n=1}^N)$ is the tensor approximated by the core tensors of TR decomposition. The proposed algorithm is based on "tensor decomposition based" approach and the model is described in (2). The model is to find the core tensors of TR decomposition of an incomplete tensor, then use the TR core tensors to approximate the missing entries. To minimize the model by gradient-based algorithm, the problem is reformulated by the below optimization model:

$$f(\mathcal{G}^{(1)}, \dots, \mathcal{G}^{(n)}) = \frac{1}{2} \left\| \mathcal{W} * (\mathcal{T} - \mathcal{X}(\{\mathcal{G}^{(n)}\}_{n=1}^N)) \right\|_F^2. \quad (5)$$

This is an objective function of an optimization problem and all the core tensors are the optimization objective. From [27], the relation between the approximated tensor $\mathcal{X}$ and the core tensors $\{\mathcal{G}^{(n)}\}_{n=1}^N$ can be deduced as the following equation:

$$\mathbf{X}_{<n>} = \mathbf{G}_{(2)}^{(n)} (\mathbf{G}_{<2>}^{(\neq n)})^T, \quad (6)$$

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 all core tensors except the n th core tensor, i.e., $\mathcal{G}^{(\neq n)} := \mathcal{X}(\{\mathcal{G}^{(n+1)}, \dots, \mathcal{G}^{(n)}, \mathcal{G}^{(1)}, \dots, \mathcal{G}^{(n-1)}\})$. Because each of the core tensors is independent, we can optimize them independently. The optimization function w.r.t. $\mathcal{G}^{(n)}$ can be written as:

$$f(\mathcal{G}^{(n)}) = \frac{1}{2} \left\| \mathbf{W}_{<n>} * (\mathbf{T}_{<n>} - \mathbf{G}_{(2)}^{(n)} (\mathbf{G}_{<2>}^{(\neq n)})^T) \right\|_F^2, \quad (7)$$

where we consider other tensor cores remain fixed. Next, we can deduce the partial derivatives of the objective function (7)

w.r.t. $\mathbf{G}_{(2)}^{(n)}$ as follow:

$$\frac{\partial f}{\partial \mathbf{G}_{(2)}^{(n)}} = (\mathbf{W}_{<n>} * (\mathbf{G}_{(2)}^{(n)} (\mathbf{G}_{<2>}^{(\neq n)})^T - \mathbf{T}_{<n>})) \mathbf{G}_{<2>}^{(\neq n)}. \quad (8)$$

For $n = 1, \dots, N$, the gradients of all the core tensors can be obtained, and the core tensors can be optimized by any gradient-based optimization algorithms. Furthermore, if there is no missing entries in tensor data, our algorithm can also be used as a TR decomposition algorithm. The whole process of applying TR-WOPT to tensor completion is listed in Algorithm 1. The details of gradient descent algorithm and parameter settings in this paper will be explained in Section IV.

Algorithm 1 Tensor ring weighted optimization (TR-WOPT)

```

1: Input: incomplete tensor  $\mathcal{T}$ , weight tensor  $\mathcal{W}$ , TR-rank
    $R_1, \dots, R_N$ , and randomly initialized  $\{\mathcal{G}^{(n)}\}_{n=1}^N$ .
2: While the stopping condition is not satisfied
3:   For  $n=1:N$ 
4:     Compute gradients of  $\{\mathcal{G}^{(n)}\}_{n=1}^N$  according to (8).
5:   End
6:   Update  $\{\mathcal{G}^{(n)}\}_{n=1}^N$  by gradient descend algorithm.
7: End while
8:  $\mathcal{Y} = P_{\Omega}(\mathcal{T}) + P_{\Omega}(\mathcal{X}(\{\mathcal{G}^{(n)}\}_{n=1}^N))$ 
9: Output: completed tensor  $\mathcal{Y}$ .
```

IV. EXPERIMENT RESULTS

We conduct several synthetic data experiments and RGB image data experiments to test the performance of our TR-WOPT algorithm. We compare our algorithm with some tensor completion algorithms which are similar to our algorithm, e.g., TT-WOPT [24] and CP-WOPT [1]. Moreover, some other state-of-the-art algorithms, e.g., FBCP [26] and FaLRCT [16] are also tested in the next experiments.

For the gradient descent method applied in our TR-WOPT, we use the nonlinear conjugate gradient (NCG) with line search method, which is implemented by a Matlab toolbox named Poblano toolbox [8]. We use two stopping conditions to all the algorithms: the number of iteration reaches 500 and the error between two iterations satisfies $\|\mathcal{Y}_2 - \mathcal{Y}_1\| / \|\mathcal{Y}_2\| < tol = 10^{-6}$. When one of the stopping conditions is satisfied, the optimization will be stopped.

For completion performance evaluation, we adopt relative square error (RSE) and peak signal-to-noise ratio (PSNR). RSE is calculated by:

$$\text{RSE} = \frac{\|\mathcal{T}_{real} - \mathcal{Y}\|_F}{\|\mathcal{T}_{real}\|_F}, \quad (9)$$

where $\mathcal{T}_{real}$ is the real tensor with full observations, $\mathcal{Y}$ is the completed tensor. Moreover, for RGB image data, PSNR is obtained by:

$$\text{PSNR} = 10 \log_{10}(255^2 / \text{MSE}). \quad (10)$$

MSE is deduced by:

$$\text{MSE} = \|\mathcal{T}_{real} - \mathcal{Y}\|_F^2 / \text{num}(\mathcal{T}_{real}), \quad (11)$$

where $\text{num}(\cdot)$ denotes the number of element of the tensor. In addition, for tensor experiments of random missing cases, we define missing rate as $1 - M/\text{num}(\mathcal{T}_{\text{real}})$, where M is the number of sampled entries (i.e. observed entries).

A. Synthetic data

In this section, we conduct synthetic data experiments to see the performance of our algorithm and the compared algorithms under different tensor dimensions and different missing rates. TR-WOPT, TT-WOPT, CP-WOPT, FBCP, and FaLRTC are tested in the experiments. The synthetic data is generated by a highly oscillating function: $f(x) = \sin(\pi/4)\cos(x^2)$ in vector form, then the synthetic data is reshaped to generate the required tensors of different dimensions. We test four different tensor dimensions: $48 \times 48 \times 48$ (3D), $16 \times 16 \times 16 \times 16$ (4D), $10 \times 10 \times 10 \times 10 \times 10$ (5D) and $7 \times 7 \times 7 \times 7 \times 7 \times 7$ (6D). The missing rate varies from 0.1 to 0.95. For the rank selection of TR-WOPT, TT-WOPT, CP-WOPT, we adjust the ranks of each algorithm to make the model parameters to be as close as possible, thus provide a fair situation to compare model representation ability. FBCP can tune CP rank automatically, and FaLRTC does not need to set tensor ranks beforehand, thus we only need to tune hyper-parameters of the two algorithms and record the best performance. The five algorithms are tested and the RSE results are shown in Figure 2.

From the figure we can see, TR-WOPT performs well in all the four tensor dimensions. Moreover, when the tensor dimension increases, many compared algorithms show performance degradation. TT-WOPT is the closest algorithm to TR-WOPT, though the number of model parameters is set similarly, TR-WOPT performs better than TT-WOPT almost in all situations due to higher representation ability of the TR-based model.

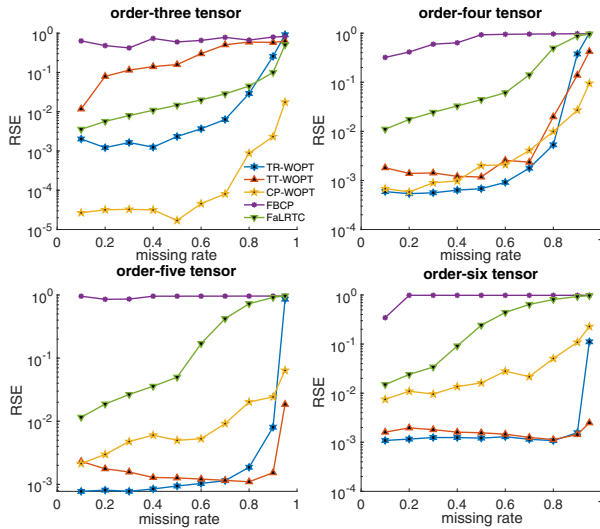


Figure 2. Synthetic data completion results of five algorithms under different tensor dimensions and different missing rates.

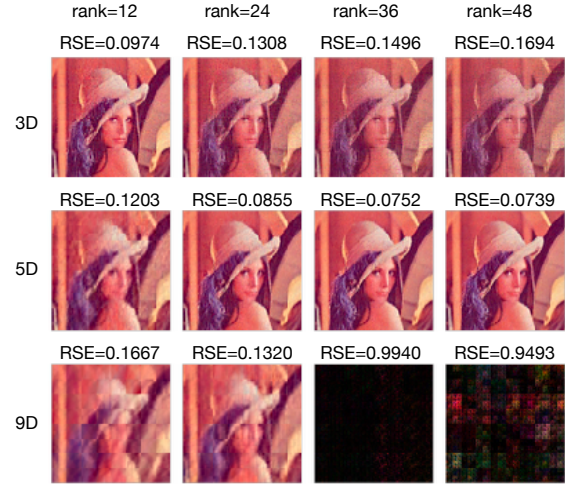


Figure 3. Completion results of TR-WOPT under different tensor dimensions and different TR-ranks using benchmark image "Lena" (missing rate is 0.7).

B. Image experiments

The synthetic data experiments results show that our TR-WOPT performs well in higher-dimension tensors, so in this section, we first test the performance of TR-WOPT under different tensor dimension and TR-ranks. Instead of directly reshaping image to higher-dimension, we apply a novel tensorization scheme mentioned in [24]. The idea is simple, if the size of a RGB image is $U \times V \times 3$, and $U = u_1 \times u_2 \times \dots \times u_l$ and $V = v_1 \times v_2 \times \dots \times v_l$ are satisfied, then the image can be tensorized to a $(l+1)$ -dimension tensor of size $u_1 v_1 \times u_2 v_2 \times \dots \times u_l v_l \times 3$. The first dimension of the higher-dimension tensor stands for a pixel block of the image, and the following dimensions are the extension blocks of the image. The higher-dimension tensor generated by the above tensorization scheme is considered to be a better structure of the image.

We choose the benchmark RGB image "Lena" (original size $256 \times 256 \times 3$) in this experiment and tensorize the image by the above tensorization scheme to 3-D ($256 \times 256 \times 3$), 5-D ($16 \times 16 \times 16 \times 16 \times 3$) and 9-D ($4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 3$). For simplicity, the TR-ranks of each experiment are set as the same value, i.e., $R_1 = R_2 = \dots = R_N$. We conduct the experiments when TR-ranks are 12, 24, 36 and 48 respectively. The missing rates of all the data are set as 0.7. Figure 3 shows the visual results and corresponding RSE values of each situation. From the results we can see, the best completion performance is obtained at the 5-D case when TR-rank is set as 48. So tensorizing data to a higher-dimension properly can enhance the performance of our algorithm. In the following RGB image experiments, we adopt this tensor structure for our algorithm.

The following experiments consider four different irregular missing situations, i.e., removing images by the shapes of the alphabet, missing by scratching, the block missing and the line

missing. Moreover, random missing cases with high missing rates are also considered. Because all the tensor completion algorithms perform well in low missing rate situations, we only test high random missing rate situations, i.e., missing rates are 0.8, 0.9, 0.95 and 0.99. We tune ranks and hyper-parameters of each algorithm and record the best completion results of each algorithm. Figure 4 and Table I show the visual and numerical completion results of the five algorithms respectively. From the results we can see, TR-WOPT performs better than TT-WOPT, CP-WOPT, and FBCP in all the tested situations. However, FaLRTC shows slightly better performance than TR-WOPT in the images of scratch missing and random missing (missing rate is 0.8). This is because the images own distinct low-rank property and the missing rate is relatively low, which is easy for “rank minimization based” algorithms to catch the low-rank structures of the tensor. However, when the missing rate of data is higher and most of the information is missing, FaLRTC cannot find the low-rank structure of the data, so the performance of random missing rate 0.9 to 0.99 of the algorithm drops quickly.

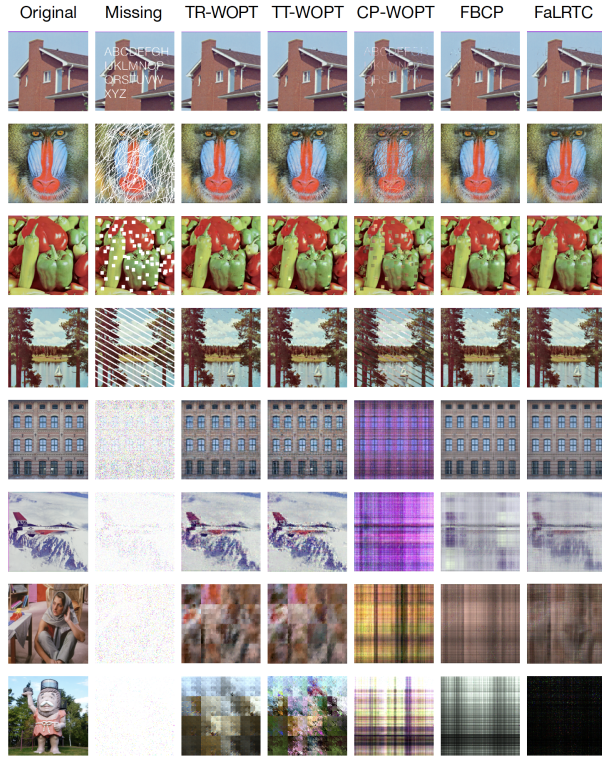


Figure 4. Visual completion results of five algorithms under eight image missing situations. The first column and the second column are original images and images with specified missing patterns respectively. The following columns are the completion results of the five algorithms respectively. The first row to the fourth row are the completion results of alphabet missing, scratch missing, block missing and line missing respectively. The fifth row to the last row are random missing completion results of missing rates 0.8, 0.9, 0.95 and 0.99 respectively.

Table I
NUMERICAL COMPLETION RESULTS OF FIVE ALGORITHMS UNDER EIGHT IMAGE MISSING SITUATIONS.

		TR-WOPT	TT-WOPT	CP-WOPT	FBCP	FaLRTC
Alphabet	RSE	0.0227	0.0282	0.0901	0.0397	0.0313
	PSNR	37.17	35.30	25.62	32.32	34.40
Scratch	RSE	0.110	0.119	0.231	0.114	0.106
	PSNR	24.55	23.83	18.08	24.20	24.84
Block	RSE	0.0891	0.124	0.176	0.115	0.104
	PSNR	26.21	23.31	20.32	24.01	24.84
Line	RSE	0.101	0.115	0.187	0.116	0.112
	PSNR	24.81	23.70	19.46	23.61	24.72
0.8	RSE	0.128	0.142	0.332	0.101	0.0839
	PSNR	23.59	22.71	15.32	25.70	27.27
0.9	RSE	0.125	0.134	0.414	0.175	0.146
	PSNR	19.97	19.35	9.562	17.01	18.62
0.95	RSE	0.252	0.276	0.530	0.351	0.343
	PSNR	18.20	17.42	11.74	15.32	15.52
0.99	RSE	0.398	0.657	0.679	0.457	0.942
	PSNR	12.59	8.226	7.948	11.38	5.099

V. CONCLUSIONS

Based on low-rank tensor ring decomposition, in this paper, we proposed a new tensor completion algorithm named tensorizing weighted optimization (TR-WOPT). The TR core tensors are optimized by the gradient-based method and used to predict the missing entries of the incomplete tensors. We conduct various synthetic data experiments and real-world data experiments, and the results show that TR-WOPT outperforms the state-of-the-art algorithms in many situations. In addition, we also find that tensorizing lower-dimension tensor to a proper higher-dimension tensor can give a better data structure and thus can improve the performance of our algorithm. Good performance of TR-WOPT in various completion tasks shows the high representation ability and flexibility of TR decomposition. It is also shown that the gradient-based algorithm is promising to optimize tensor decompositions. Furthermore, our method needs the TR-rank to be specified before the experiment, which is time-consuming to find the best TR-rank for the data. In our future work, we will study how to determine TR-ranks automatically.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI (Grant No. 17K00326, 15H04002, 18K04178), JST CREST (Grant No. JPMJCR1784) and the National Natural Science Foundation of China (Grant No. 61773129).

REFERENCES

- [1] Evrim Acar, Daniel M Dunlavy, Tamara G Kolda, and Morten Mørup. Scalable tensor factorizations for incomplete data. *Chemometrics and Intelligent Laboratory Systems*, 106(1):41–56, 2011.
- [2] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832, 2014.
- [3] Rasmus Bro. Parafac. tutorial and applications. *Chemometrics and intelligent laboratory systems*, 38(2):149–171, 1997.
- [4] Xingwei Cao and Qibin Zhao. Tensorizing generative adversarial nets. *arXiv preprint arXiv:1710.10772*, 2017.

- [5] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, Danilo P Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016.
- [6] Andrzej Cichocki, Danilo Mandic, Lieven De Lathauwer, Guoxu Zhou, Qibin Zhao, Cesar Caiafa, and Huy Anh Phan. Tensor decompositions for signal processing applications: From two-way to multiway component analysis. *IEEE Signal Processing Magazine*, 32(2):145–163, 2015.
- [7] Andrzej Cichocki, Anh-Huy Phan, Qibin Zhao, Namgil Lee, Ivan Oseledets, Masashi Sugiyama, Danilo P Mandic, et al. Tensor networks for dimensionality reduction and large-scale optimization: Part 2 applications and future perspectives. *Foundations and Trends® in Machine Learning*, 9(6):431–673, 2017.
- [8] Daniel M Dunlavy, Tamara G Kolda, and Evrim Acar. Poblano v1. 0: A matlab toolbox for gradient-based optimization. *Sandia National Laboratories, Albuquerque, NM and Livermore, CA, Tech. Rep. SAND2010-1422*, 2010.
- [9] Marko Filipović and Ante Jukić. Tucker factorization with missing data with application to low- n n-rank tensor completion. *Multidimensional systems and signal processing*, 26(3):677–692, 2015.
- [10] Silvia Gandy, Benjamin Recht, and Isao Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.
- [11] Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.
- [12] Yao Hu, Debing Zhang, Jieping Ye, Xuelong Li, and Xiaofei He. Fast and accurate matrix completion via truncated nuclear norm regularization. *IEEE transactions on pattern analysis and machine intelligence*, 35(9):2117–2130, 2013.
- [13] Alexandros Karatzoglou, Xavier Amatriain, Linas Baltrunas, and Nuria Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 79–86. ACM, 2010.
- [14] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.
- [15] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):208–220, 2013.
- [16] Ye Liu, Mingfen Li, Hao Zhang, Hang Wang, Junhua Li, Jie Jia, Yi Wu, and Liqing Zhang. A tensor-based scheme for stroke patients? motor imagery eeg analysis in bci-fes rehabilitation training. *Journal of neuroscience methods*, 222:238–249, 2014.
- [17] Zhen Long, Yipeng Liu, Longxi Chen, and Ce Zhu. Low rank tensor completion for multiway visual data. *arXiv preprint arXiv:1805.03967*, 2018.
- [18] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.
- [19] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.
- [20] Amnon Shashua and Tamir Hazan. Non-negative tensor factorization with applications to statistics and computer vision. In *Proceedings of the 22nd international conference on Machine learning*, pages 792–799. ACM, 2005.
- [21] Nicholas D Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos E Papalexakis, and Christos Faloutsos. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017.
- [22] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.
- [23] Wenqi Wang, Yifan Sun, Brian Eriksson, Wenlin Wang, and Vaneet Aggarwal. Wide compression: Tensor ring nets. *arXiv preprint arXiv:1802.09052*, 2018.
- [24] Longhao Yuan, Qibin Zhao, and Jianting Cao. Completion of high order tensor data with missing entries via tensor-train decomposition. In *International Conference on Neural Information Processing*, pages 222–229. Springer, 2017.
- [25] Qibin Zhao, Masashi Sugiyama, and Andrzej Cichocki. Learning efficient tensor representations with ring structure networks. *arXiv preprint arXiv:1705.08286*, 2017.
- [26] Qibin Zhao, Liqing Zhang, and Andrzej Cichocki. Bayesian cp factorization of incomplete tensors with automatic rank determination. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1751–1763, 2015.
- [27] Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016.
- [28] Qibin Zhao, Guoxu Zhou, Liqing Zhang, Andrzej Cichocki, and Shun-Ichi Amari. Bayesian robust tensor factorization for incomplete multiway data. *IEEE transactions on neural networks and learning systems*, 27(4):736–748, 2016.