埼玉工業大学

博士後期学位論文

テンソルリング分解理論構築及び高次元データの補正と雑音除去に関する研究

原龍昊

埼玉工業大学大学院　工学研究科

博士後期課程　電子工学専攻

指導教員　曹 建庭 教授

令和元年 06 月 25 日

SAITAMA INSTITUTE OF TECHNOLOGY

DOCTORAL THESIS

Study on High-order Data Completion and Denoising via Tensor Ring Decomposition Theory

Author:
Longhao YUAN

Supervisor:
Jianting CAO

A thesis submitted in fulfillment of the requirements for the degree of

Doctor of Philosophy

in the

Graduate School of Engineering

June 26, 2019

論文概要

多視点センサとデータ記憶技術の開発に伴い、取得したデータはしばしば高次、大規模かつ高複雑性の特性を示します。これらのデータを如何に効率的に処理するかは重要な問題です。テンソル（Tensor）は行列とベクトルを一般化した高次的なものであり、自然にデータの高次の関係とオブジェクトを表すことができます。近年、テンソル法はデータ処理問題を解決するための強力なツールとなっています。ほんの数例を挙げると、信号処理、機械学習、データマイニング、画像処理、計算神経科学において、テンソル法で多数の応用が開発されています。

テンソル法の中で、テンソル分解は最も重要で基本的なツールの一つです。テンソル分解はテンソルを低次元の潜在的因子のセットに分解することです。潜在的な因子は、データの潜在的な特徴を含んでおり、データを強力に圧縮する方法で表現します。CANDECOMP/PARAFAC分解（CPD）とTucker分解（TKD）は、1世紀以上に渡って研究されてきた最も古典的なテンソル分解モデルです。しかし、これらのテンソル分解モデルは、大規模または非常に高次のテンソルを扱うときに計算の限界を示します。ごく最近では、テンソルリング分解（tensor ring decomposition, TRD）と呼ばれる最新のテンソル分解モデルが、その高い表現能力および多重線形特性のために人々の注目を集めています。 TRDの最も重要な利点は、モデルの複雑さがテンソルの次数によって指数関数的に増加しないことです。このようにして、TRDは「次元の呪い」を効果的に克服し、大規模で高次のテンソルを処理するための強力なツールとなりました。

本論文は、TRD の理論と応用を探ることに焦点を当てます。主な貢献は、高効率と高性能の様々な TRD ベースのアルゴリズムを提案することです。第一に、画像修復の問題において、勾配ベースのテンソル補完アルゴリズムを開発しました。既存の方法と比較して、我々が提案するアルゴリズムは、高次と高い欠測率の画像の修復において著しく良い性能を示しました。第二に、TR 因子の核規範に基づく２つのランクロバスト TRD アルゴリズムを提案しました。これらのアルゴリズムは、テンソルと TR 因子の間のランク関係の理論的証明に基づいており、テンソル補完の課題におけるランク選択の負担を著しく軽減することができました。第三に、独創的な TR マックス規範（TR-max-norm）に基づいた効率的な TRD アルゴリズムを提案しました。この規範はテンソルの TR ランクを正則化できることが、我々の研究で理論的に証明されました。既存のアルゴリズムと比較して、我々が提案したアルゴリズムは大規模なテンソル補完の課題において、安定した収束性と高い性能を示しました。第四に、テンソルランダムプロジェクション（tensor random projection）と呼ばれる最新の技術に基づいて、高速 TRD アルゴリズムを提案しました。このアルゴリズムは既存のものと比較して計算コストを大幅に削減し、大規模テンソル雑音除去に適用できます。この論文は TRD の理論研究と応用をより充実にし、それによってテンソル方法論に貢献し、研究と産業分野においても良い参考となります。

Abstract

Study on High-order Data Completion and Denoising via Tensor Ring Decomposition Theory

by Longhao YUAN

Doctor of Philosophy

SAITAMA INSTITUTE OF TECHNOLOGY
Graduate School of Engineering

With the development of multi-view sensors and data storage technology, the acquired data often show the properties of high-order, large-scale and high-complexity. How to efficiently process these data is a significant problem. Tensor is the generalization of matrix and vector, which can naturally represent high-order relations and objects of the data. In recent years, tensor methods have become powerful tools to solve the data processing problem. Numerous applications of tensor methods have been developed in signal processing, machine learning, data mining, image processing, computational neuroscience, to name a few.

Among the tensor methods, tensor decomposition is one of the most important and fundamental tools, which is to decompose a tensor into a set of latent factors of low dimensionality. The latent factors are powerful to reveal the latent feature of the data and represent the data in a highly compressive way. CANDECOMP/PARAFAC decomposition (CPD) and Tucker decomposition (TKD) are the most classical tensor decomposition models which have been studied for over a century. However, these models show computational limitations when dealing with tensors of large-scale or very high-order. In very recent years, a novel tensor decomposition model termed as tensor ring decomposition (TRD) has drawn people's attention due to its high representation ability and multi-linear property. The most significant advantage of TRD is that the model complexity does not grow exponentially in the tensor order. In this way, TRD can effectively overcome the 'curse of dimensionality' and becomes a powerful tool to process large-scale and high-order tensors.

This thesis focuses on exploring the theories and applications of TRD. The main contribution is to propose various TRD-based algorithms of high efficiency and high performance. Firstly, aiming at the problem of image recovery, a gradient-based tensor completion algorithm is developed. Compared with the traditional methods, our algorithm performs significantly better in the image of high-order and high missing rate. Secondly, two rank-robust TRD algorithms are proposed by imposing nuclear norm on TR factors. The algorithms are based on the theoretical proof of the rank relationship between the tensor and the TR factors, and they can successfully alleviate the burden of rank selection in tensor completion tasks. Thirdly, an efficient TRD algorithm is proposed based on the novel TR-max-norm regularizer which is theoretically proved to regularize the TR-rank of the tensor. Compared with the traditional algorithms, the proposed algorithm shows steady convergence and higher performance in large-scale tensor completion tasks. Fourthly, a fast TRD algorithm is provided based on the novel technique named tensor random projection. The proposed algorithm largely reduces the computational cost in comparison with the traditional ones and can be applied to large-scale tensor denoising. The work in the thesis has enriched the theoretical study and applications of TRD, which contribute to the tensor methodology and will be a good reference in the research and industry fields.

Acknowledgements

I would like to first give my sincere gratitude to my supervisor, Professor Jianting Cao, who gave full support to my research career during my PhD study. He always has inspirations and enthusiasm in research, and his constructive ideas and way of thinking have helped me a lot. On the first-year of my PhD study, he encouraged me to work on deep learning, which is a new research direction not only in our lab but also for me. I followed his guidance and tried to apply the deep learning technique to solve the major research problem of our lab. At the end of the first-year study, my work was accepted and I had an opportunity to attend my first memorable international conference in Portugal, which had totally opened my eyes and intrigued my passion of research. After that, I continued to go deeper into my research and he always gave essential advice to help me fight through the difficulties and confusions in my pursuit of PhD.

I also would like to thank Dr. Qibin Zhao who is an outstanding researcher and an impressive team leader. He enrolled me as a part-time researcher in RIKEN AIP from my second-year PhD till now. During the stay in his team and under his supervision, I have made fast progress in knowledge accumulation about tensor and machining learning. He was always a reliable mentor and a good model for me, and he used his abundant knowledge and advancing research ideas to help me improve my research ability and through numerous difficulties. The days in RIKEN AIP is with the memories of hard-working days, the joy of success, the disappointment and failures, which contribute to making me be the person I am today.

Last but not least, I would like to appreciate all my friends, colleagues and families, who have helped me a lot and provided great support during these years. The kindness, encouragement, positive thought, accompany and love of all of you have made me through an unforgettable PhD career.

Contents

Chapter 1

Introduction

Tensors are the high-order generalizations of vectors and matrices. Representing data by tensor can retain the high-order form of data and retain adjacent structure information of data. Most of the real-world data are more than two orders. For example, RGB images are order-3 tensors ($height \times width \times channel$), videos are order-4 tensors ($height \times width \times channel \times time$) and electroencephalography (EEG) signals are order-3 tensors ($magnitude \times trails \times time$). When facing data with more than two orders, traditional methods usually transform data into matrices or vectors by concatenation, which leads to spatial redundancy and less efficient factorization [1]. In recent years, many theories, algorithms and applications of tensor methodologies have been studied and proposed [2–4]. Due to the high compression ability and data representation ability of tensor methods, many applications have been proposed in a variety of fields such as image and video completion [5, 6], signal processing [7, 8], brain-computer interface [9], image classification [10], etc.

One of the most important tools for tensor is tensor decomposition, which aims to find the latent factors of tensor-valued data (i.e. the generalization of multi-dimensional arrays), thereby casting large-scale tensors into a multilinear tensor space of low-dimensionality (very few degrees of freedom designated by the rank). Tensor factors can then be considered as latent features of data, and in this way can represent the data economically and predict missing entries when the data is incomplete. The specific form and operations among latent factors define the type of tensor decomposition. A variety of tensor decomposition models have been applied in diverse fields such as machine learning [11–13] and signal processing [14, 15]. Tucker decomposition (TKD) and CANDECOMP/PARAFAC decomposition (CPD) are classical tensor decomposition models, which have been studied for nearly half a century [2, 16, 17]. In recent years, a novel tensor decomposition named tensor ring decomposition (TRD) [18] has drawn people's attention, due to its super compressive ability and multi-linear representation. Compared with CPD and TKD, the tensor ring decomposition owns the good numerical property and the model complexity grows linearly in tensor order, thus can achieve fast decomposition of high-order and large-scale tensor efficiently.

Tensor completion aims to recover an incomplete tensor from partially observed entries. The theoretical lynchpin in matrix or tensor completion problems is the low-rank assumption, and tensor completion has been applied in various applications such as image/video completion [6, 19], recommendation systems [20], link prediction [21], compressed sensing [22], to name but a few. Since the determination of tensor rank is an NP-hard problem [2, 23], many tensor low-rank surrogates were proposed for tensor completion. One such surrogate is the nuclear norm (a.k.a. Schatten norm, or trace norm), which is defined as the sum of singular values of a matrix, and is the most popular convex surrogate for rank regularization. Unlike matrix completion problems, the Schatten norm model of a tensor is hard to formulate. Recent studies mainly focus on two convex relaxation models of tensor Schatten norm, the "overlapped" model [19, 24–27] and the "latent" [24, 28] model.

The thesis studies on tensor decomposition and tensor completion based on tensor ring decomposition (TRD), aiming to develop efficient and high-performance data processing methods based on tensor methodologies, and employ them to various practical applications. *Chapter* 1 firstly introduces the contributions of this thesis. Then the background of tensor, tensor decomposition and tensor completion is provided. In *Chapter* 2, the tensor ring weighted optimization algorithm (TR-WOPT) is introduced, which is a gradient-based tensor completion algorithm and can be applied to high-order completion. Moreover, a visual data tensorization (VDT) method is provided to transform visual data into higher-order tensors to find more structure information of the data. IN order to solve the multi-linear rank selection problem of TRD, in *Chapter* 3, the relation between the rank of tensor and rank of TR factors are theoretically deduced, then, the tensor nuclear norm is imposed on the TR-factors by different schemes. Then, two tensor completion methods are proposed and efficiently solved by alternating direction method of multipliers (ADMM) algorithm. In *Chapter* 4, a novel low-TR-rank regularizer termed as TR-max-norm is proposed, which extends the matrix max-norm to tensor in TR format. The TRD model with TR-max-norm is formulated and efficiently solved by projected mini-batch stochastic gradient descent algorithm. The proposed algorithm shows faster convergence and higher performance than its traditional counterpart. In order to fill the gap that there lack large-scale TRD algorithms, in *Chapter* 5, a computational scheme based on tensor random projection (TRP) is provided. The scheme can be applied to various existing TRD algorithms and largely decrease the computation cost. The reconstruction and denoising experiments of large-scale tensor show the superior performance and computational speed of the proposed scheme. *Chapter* 6 provides the overall conclusion of the thesis and the future outlook.

1.1 Summary of contributions

1.1.1 High-order tensor completion by tensor ring decomposition

Taking advantages of high compressibility and good performance in high-order tensor decomposition of TRD, a new tensor completion approach named tensor ring weighted optimization (TR-WOPT) is proposed. 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. In addition, a method named Visual Data Tensorization (VDT) is proposed to transform visual data into higher-order tensors, resulting in the performance improvement of our algorithms. Furthermore, image completion results show that our proposed algorithm outperforms the related algorithms in many situations, especially in the high-order and high missing rate situations.

1.1.2 Rank-robust tensor ring decomposition and completion

In tensor completion tasks, the traditional low-rank tensor decomposition models suffer from the laborious model selection problem due to their high model sensitivity. In particular, for TRD, the number of model possibilities grows exponentially with the tensor order, which makes it rather challenging to find the optimal TR-rank for the tensor. By exploiting the low-rank structure of the TR latent space, this work proposes a novel tensor completion method which is robust to model selection. In contrast to imposing the low-rank constraint on the data space, we introduce nuclear norm regularization on the latent TR factors, resulting in the optimization step using singular value decomposition (SVD) being performed at a much smaller scale. By leveraging the alternating direction method of multipliers (ADMM) scheme, the latent TR factors with optimal rank and the recovered tensor can be obtained simultaneously. The proposed algorithm is shown to effectively alleviate the burden of TR-rank selection, thereby greatly reducing the computational cost. The extensive experimental results on both synthetic and real-world data demonstrate the superior performance and efficiency of the proposed approach against state-of-the-art algorithms.

1.1.3 Max-norm regularized tensor ring decomposition

The existing TRD-based algorithms are of high computational cost and the convergence instability is another challenging problem which remains unsolved. To this end, by leveraging the high efficiency and stability of the max-norm in matrix completion, the matrix max-norm is extended it to the tensor field by TRD and the TR-max-norm is developed which is proved to be a low-TR-rank regularizer for tensors. Compared to the nuclear norm regularization which has to conduct multiple singular value decomposition (SVD) on the whole tensor scale, the TR-max-norm is processed on the TR latent space, which drastically reduces the computational cost. An efficient TRD algorithm with TR-max-norm regularization is thus developed based on the projected mini-batch stochastic gradient descent (PMSGD) scheme which can be applied to large-scale data processing. The experimental results on simulation experiments show the fast and stable convergence of our algorithm.

1.1.4 Large-scale tensor denoising via tensor ring decomposition

Dimensionality reduction is an essential technique for multi-way large-scale data, i.e., tensor. The traditional TRD algorithms suffer from high computational cost when facing large-scale data. Taking advantages of the recently proposed tensor random projection (TRP) method, a randomized TRD scheme is proposed. By employing random projection on every mode of the large-scale tensor, the TRD can be processed at a much smaller scale. The large-scale tensor reconstruction and denoising experiments show the huge speed-up without loss of accuracy and the superior performance of the proposed scheme compared to the traditional counterparts and the other randomized algorithms.

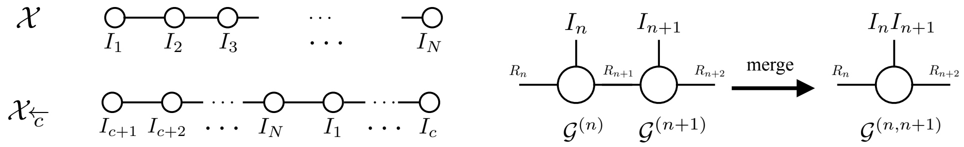
1.2 Tensor preliminaries

1.2.1 Notations

Notations in [2] are adopted in this thesis. A scalar is denoted by a normal lowercase/uppercase letter, e.g., $x, X \in \mathbb{R}$, 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 order $N \geq 3$ is denoted by an Euler script letter, e.g., $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$.

A sequence of tensor $\{\boldsymbol{\mathcal{X}}^{(1)}, \boldsymbol{\mathcal{X}}^{(2)}, \ldots, \boldsymbol{\mathcal{X}}^{(N)}\}$ is denoted by $\{\boldsymbol{\mathcal{X}}^{(n)}\}_{n=1}^{N}$, or simply $[\boldsymbol{\mathcal{X}}]$, in which $\boldsymbol{\mathcal{X}}^{(n)}$ is the n-th tensor of the sequence. The matrix sequences and vector sequences are defined in the same way. An element of tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ of index $\{i_1, i_2, \cdots, i_N\}$ is denoted by $x_{i_1 i_2 \cdots i_N}$ or $\boldsymbol{\mathcal{X}}(i_1, i_2, \cdots, i_N)$.

Furthermore, the inner product of two tensor $\boldsymbol{\mathcal{X}}$, $\boldsymbol{\mathcal{Y}}$ with the same size $\mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is defined as $\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \rangle = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_N} x_{i_1 i_2 \cdots i_N} y_{i_1 i_2 \cdots i_N}$. The Frobenius norm of $\boldsymbol{\mathcal{X}}$ is defined by $\|\boldsymbol{\mathcal{X}}\|_F = \sqrt{\langle \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{X}} \rangle}$. The Hadamard product is denoted by "$*$" and it is an element-wise product of vectors, matrices or tensors of the same size. For instance, given tensors $\boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{Y}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, $\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{X}} * \boldsymbol{\mathcal{Y}}$, then $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and $z_{i_1 i_2 \cdots i_N} = x_{i_1 i_2 \cdots i_N} y_{i_1 i_2 \cdots i_N}$ are satisfied. The Kronecker product of two matrices $\mathbf{X} \in \mathbb{R}^{I \times K}$ and $\mathbf{Y} \in \mathbb{R}^{J \times L}$ is $\mathbf{X} \otimes \mathbf{Y} \in \mathbb{R}^{IJ \times KL}$, see more details in [2].


$\mathcal{X}$
I_1 I_2 I_3 $\cdots$ I_N
$\mathcal{X}_{\overleftarrow{c}}$
I_{c+1} I_{c+2} $\cdots$ I_N I_1 $\cdots$ I_c
I_n I_{n+1}
R_n R_{n+1} R_{n+2}
merge
$\mathcal{G}^{(n)}$ $\mathcal{G}^{(n+1)}$
$I_n I_{n+1}$
R_n R_{n+2}
$\mathcal{G}^{(n,n+1)}$

(a) Circular permutation of tensor

(b) Merging of TR factors

FIGURE 1.1: Diagrams of two tensor operations.

1.2.2 Tensor operations

Tensor unfolding. We employ three types of tensor unfolding (matricization) operations in this thesis. The standard mode-n unfolding [2] of tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ 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 $\boldsymbol{\mathcal{X}}$ which is often used in TR operations [18] is denoted by $\mathbf{X}_{<n>} \in \mathbb{R}^{I_n \times I_{n+1} \cdots I_N I_1 \cdots I_{n-1}}$. The third kind of mode-n unfolding of tensor $\boldsymbol{\mathcal{X}}$ 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 [29]. Furthermore, the inverse operation of 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 $\text{fold}_{(n)}(\cdot)$, $\text{fold}_{<n>}(\cdot)$ and $\text{fold}_{[n]}(\cdot)$ respectively, i.e., for a tensor $\boldsymbol{\mathcal{X}}$, we have $\text{fold}_{(n)}(\mathbf{X}_{(n)}) = \boldsymbol{\mathcal{X}}$.

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

Tensor product. The mode-n product of tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_n \times \cdots \times I_N}$ and matrix $\mathbf{B} \in \mathbb{R}^{J \times I_n}$ is denoted by $\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{X}} \times_n \mathbf{B} = \text{fold}_{(n)}(\mathbf{B}\mathbf{X}_{(n)})$.

Tensor factors merging [30]. Taking the TR factors merging as an example, the adjacent TR factors can be merged by reshaping and multiple matrix multiplication operations. For TR factors of size $\boldsymbol{\mathcal{G}}^{(n)} \in \mathbb{R}^{R_n \times I_n \times R_{n+1}}$, the contiguous subchain of the TR factors denoted by $\{\boldsymbol{\mathcal{G}}^{(i)}, \boldsymbol{\mathcal{G}}^{(i+1)}, \cdots, \boldsymbol{\mathcal{G}}^{(j)}\}$ can be merged as: $\boldsymbol{\mathcal{G}}^{(i,i+1,\dots,j)} \in \mathbb{R}^{R_i \times \prod_{k=i}^{j} I_k \times R_{j+1}}$

These basic tensor operations will be used in the following demonstrations of our work. The diagram of tensor circular permutation and TR factors merging are shown in Figure 1.1.

1.3 Tensor decomposition models

CANDECOMP/PARAFAC decomposition (CPD) [31] and Tucker decomposition (TKD) [16] are the most classical and well-studied tensor decomposition models, after which tensor train decomposition (TTD) [29] and tensor ring decomposition (TRD) [18] become popular because of their high compression performance in high-order and large-scale tensor. TT decomposition and TR decomposition provide natural solutions for the 'curse of dimensionality'. For instance, for an Nth-order tensor, the space complexity of Tucker grows exponentially in N, while the cases of TT, TR and CP are linear in N. Although CP is a highly compact decomposition model of which the space complexity is also linear in N, it has difficulties in finding the optimal latent tensor factors [32]. In the following sections, we mainly introduce the background of TTD and TRD.

1.3.1 Tensor train decomposition

Tensor train decomposition (TTD) is to decompose a tensor into a sequence of two matrices and $N-2$ order-three core tensors (factor tensors): $\mathbf{G}^{(1)}, \boldsymbol{\mathcal{G}}^{(2)}, \cdots, \mathbf{G}^{(N)}$. The relation between the approximated tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ and core tensors can be expressed as

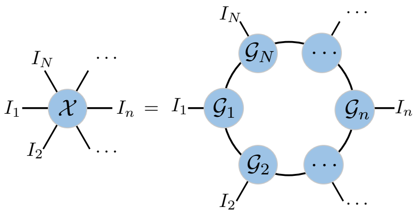


FIGURE 1.2: TR decomposition.

follow:

$$\boldsymbol{\mathcal{X}} = \ll \mathbf{G}^{(1)}, \boldsymbol{\mathcal{G}}^{(2)}, \cdots, \mathbf{G}^{(N)} \gg, \tag{1.1}$$

where for $n = 1, \cdots, N$, $\boldsymbol{\mathcal{G}}^{(n)} \in \mathbb{R}^{R_{n-1} \times I_n \times R_n}$, $R_0 = R_N = 1$, and the notation $\ll \cdot \gg$ is the operation to transform the core tensors to the approximated tensor. $\mathbf{G}^{(1)} \in \mathbb{R}^{I_1 \times R_1}$ and $\mathbf{G}^{(N)} \in \mathbb{R}^{R_{N-1} \times I_N}$ are two matrices in the first and the last positions. The sequence $R_0, R_1, \cdots, R_N$ is named TT-rank which limits the size of every core tensor. Furthermore, the $(i_1, i_2, \cdots, i_N)$th element of tensor $\boldsymbol{\mathcal{X}}$ can be represented by the multiple product of the corresponding mode-2 slices of the core tensors as:

$$x_{i_1 i_2 \cdots i_N} = \prod_{n=1}^{N} \mathbf{G}_{i_n}^{(n)}, \tag{1.2}$$

where $\mathbf{g}_{i_1}^{(1)}, \mathbf{G}_{i_1}^{(1)}, \cdots, \mathbf{g}_{i_N}^{(N)}$ is the sequence of slices from each core tensor. For $n = 1, 2, \cdots, N$, $\mathbf{G}_{i_n}^{(n)} \in \mathbb{R}^{R_{n-1} \times R_n}$ is the mode-2 slice extracted from $\boldsymbol{\mathcal{G}}^{(n)}$ according to each mode of the element index of $x_{i_1 i_2 \cdots i_N}$. $\mathbf{g}_{i_1}^{(1)} \in \mathbb{R}^{R_1}$ and $\mathbf{g}_{i_N}^{(N)} \in \mathbb{R}^{R_{N-1}}$ are extracted from first core tensor and last core tensor, they are considered as two order-one matrices for overall expression convenience.

1.3.2 Tensor ring decomposition

In recent years, the concept of tensor networks has been proposed and has become a powerful and promising aspect of tensor methodology [33, 34]. One of the most recent and popular tensor networks, named the matrix product state/tensor-train (MPS/TT), is studied across disciplines owing to its super compression and computational efficiency properties [11, 29]. For a tensor of N-dimensions, the most significant property of TT decomposition is that the space complexity will not grow exponentially with N, thus providing a natural remedy for the “curse of dimensionality”, while the number of parameters of Tucker decomposition is exponential in N. Although the CP decomposition is a highly compact representation which has the desirable property of being linear in N, it has difficulties in finding the optimal latent tensor factors. To address these issues, recent studies propose a generalization of TT decomposition, termed the tensor ring (TR) decomposition, in order to relax the rank constraint of TT, thus offering an enhanced representation ability, latent factors permutation flexibility (i.e. tensor permutation is directly related to the permutation of tensor factors) and structure information interpretability (i.e. each tensor factor can represent a specific feature of original tensor) [18, 35].

TRD is a more general decomposition than TT decomposition. It represents a high-order tensor by circular multilinear products over a sequence of core tensors (i.e., TR factors). The diagram of TRD is shown in Figure 1.2. All of the TR factors are 3rd-order tensors, which are denoted by $\{\mathcal{G}_n\}_{n=1}^{N}$, $\mathcal{G}_n \in \mathbb{R}^{R_n \times I_n \times R_{n+1}}$, $n = 1, \ldots, N$. In the same way as the TT decomposition, the TR decomposition linearly scales to the order of the tensor, thus it can overcome the “curse of dimensionality”. $R_1, R_2, \ldots, R_N$ denotes the TR-rank which controls the model complexity of TR decomposition. Compared to the TT decomposition, the TR decomposition relaxes the rank constraint on the first and the last core tensors to $R_1 = R_{N+1}$, while the original constraint on TT is rather stringent, i.e., $R_1 = R_{N+1} = 1$. TR applies trace operation and all the TR factors are constrained to be third-order equivalently. In this case, TR can be considered as a linear combination of TT and thus it offers a more powerful and generalized representation ability than TT. The element-wise relation and global relation of the TR decomposition and the tensor is given by equations (1.3) and (1.4):

$$\boldsymbol{\mathcal{X}}(i_1, i_2, \ldots, i_N) = \text{Trace}\left\{\prod_{n=1}^{N} \mathbf{G}_n(i_n)\right\}, \tag{1.3}$$

$$\mathbf{X}_{<n>} = \mathbf{G}_{n,(2)}(\mathbf{G}_{\neq n,<2>})^T, \tag{1.4}$$

where Trace$\{\cdot\}$ is the matrix trace operator, $\mathbf{G}_n(i_n) \in \mathbb{R}^{R_n \times R_{n+1}}$ is the i_nth mode-2 slice of $\boldsymbol{\mathcal{G}}_n$, which also can be denoted by $\boldsymbol{\mathcal{G}}_n(:, i_n, :)$ according to Matlab syntax. $\boldsymbol{\mathcal{G}}_{\neq n} \in \mathbb{R}^{R_{n+1} \times \prod_{i=1, i \neq n}^{N} I_i \times R_n}$ is a subchain tensor by merging all TR factors except the nth core tensor, see more details in [35].

1.4 Tensor completion

In practical situations, data missing is ubiquitous due to the error and the noise in data collecting process, resulting in the generation of data outliers and unwanted data entries. Generally, the lynchpin of tensor completion is to find the correlations between the missing entries and the observed entries. 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 [19, 36], compressed sensing [22], link prediction [37], recommendation system [20], to name a few. There exist 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 [38]. However, determining the rank of a tensor is an NP-hard problem [23]. To solve this problem, there are mainly two types of tensor completion methods: (i) rank-minimization-based approach and (ii) tensor-decomposition-based approach.

1.4.1 Rank-minimization-based completion

The first approach formulates the convex surrogate models of low-rank tensors. The low-rank tensor completion problem can be formulated as:

$$\min_{\boldsymbol{\mathcal{X}}} \text{ Rank}(\boldsymbol{\mathcal{X}}), \ s.t. \ P_\Omega(\boldsymbol{\mathcal{X}}) = P_\Omega(\boldsymbol{\mathcal{T}}), \tag{1.5}$$

and the model can be written in a unconstrained form by:

$$\min_{\boldsymbol{\mathcal{X}}} \text{ Rank}(\boldsymbol{\mathcal{X}}) + \frac{1}{\lambda}\|P_\Omega(\boldsymbol{\mathcal{X}}) - P_\Omega(\boldsymbol{\mathcal{T}})\|_F^2, \tag{1.6}$$

where $\boldsymbol{\mathcal{X}}$ is the low-rank approximation tensor, Rank$(\cdot)$ is a rank regularizer, $P_\Omega(\boldsymbol{\mathcal{T}})$ denotes all the observed entries w.r.t. the set of indices of observed entries represented by Ω, and $\|\cdot\|_F$ is the Frobenius norm. For the low-rank tensor completion problem, determining the rank of a tensor is an NP-hard problem. Work in [19] and [26] extends the concept of low-rank matrix completion and defines the tensor rank as the sum of the rank of mode-n matricization of the tensor. This surrogate is named "overlapped" model, and it simultaneously regularizes all the mode-n matricizations of a tensor into low-rankness by nuclear norm. For an N-dimension incomplete tensor $\mathcal{T}$, the low-rank tensor completion model is formulated by:

$$\min_{\mathcal{X}} \sum_{n=1}^{N} \|\mathbf{X}_{(n)}\|_*, \; s.t. \; P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{T}), \tag{1.7}$$

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. Moreover, based on different definitions of tensor rank, various nuclear norm regularized algorithms have been proposed [19,27,39,40]. Three algorithms based on nuclear norm minimization are proposed in [19], i.e., SiLRTC, FaLRTC, and HaLRTC. They extend the nuclear norm regularization for matrix completion to tensor completion by minimizing the Tucker rank of the incomplete tensor. In [41], Tucker low-n-rank tensor completion (TLnR) is proposed, and the experiments show better results than the traditional nuclear norm minimization methods.

Rank minimization based methods do not need to specify the rank of the employed tensor decompositions beforehand, and the rank of the recovered tensor will be automatically learned from the limited observations. However, these algorithms face multiple large-scale singular value decomposition (SVD) operations on the 2D unfoldings of the tensor when employing the nuclear norm and numerous hyper-parameter tuning, which in turn leads to high computational cost and low efficiency. To solve this problem, we provide methods in *Chapter* 3, 4.

1.4.2 Tensor-decomposition-based completion

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:

$$\min_{\{\mathcal{G}^{(n)}\}_{n=1}^{N}} \|\mathcal{W} * (\mathcal{T} - \mathcal{X}(\{\mathcal{G}^{(n)}\}_{n=1}^{N})\|_F^2, \tag{1.8}$$

where $\|\cdot\|_F$ is the Frobenius norm, $\{\mathcal{G}^{(n)}\}_{n=1}^{N}$ is the sequence of latent factors under consideration (here $[\mathcal{G}]$ stands for the factors of arbitrary decomposition) 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:

$$w_{i_1 i_2 \cdots i_N} = \begin{cases} 0 & \text{if } y_{i_1 i_2 \cdots i_N} \text{ is a missing entry,} \\ 1 & \text{if } y_{i_1 i_2 \cdots i_N} \text{ is an observed entry.} \end{cases} \tag{1.9}$$

Many completion algorithms have been proposed based on alternating least squares (ALS) method [42,43], gradient-based method [5,44], to mention but a few. Moreover, based on different tensor decomposition models, various tensor-decomposition-base approaches have been proposed, e.g., CP weighted optimization [5], weighted tucker [41] and TT weighted optimization [44]. 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.

Though ALS and gradient-based algorithms are free from burdensome hyper-parameter tuning, the performance of these algorithms is rather sensitive to model selection, i.e., rank selection of the tensor decomposition. Moreover, since the optimal rank is generally data-dependent, it is very challenging to specify the optimal rank beforehand. This is especially the case for Tucker, TT, and TR decompositions, for which the rank is defined as a vector; it is therefore impossible to find the optimal ranks by cross-validation due to the immense possibilities. To solve this problem, we provide a method in *Chapter* 3.

Chapter 2

High-order Tensor Completion by Tensor Ring Decomposition

Based on different tensor decomposition models, various tensor-decomposition-based tensor completion methods have been proposed, e.g., CP weighted optimization [5], weighted tucker [41] and TT weighted optimization [44]. 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 flexibility, this may lead to bad convergence when considering different kinds of data. TRD model is much more flexible and high performance in many applications, so TR-based WOPT method can get a better approximation of incomplete data, thus provide better completion results. In this work, we formulate the tensor ring weighted optimization (TR-WOPT) scheme which can be applied to tensor completion.

2.1 Tensor ring weighted optimization

2.1.1 Model formulation

Based on TRD, we propose the tensor ring weighted optimization (TR-WOPT) algorithm which is illustrated as follows. Define $\mathcal{T} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ 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 (1.8). 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:

$$f(\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(n)}) = \frac{1}{2} \left\| \mathcal{W} * (\mathcal{T} - \mathcal{X}(\{\mathcal{G}^{(n)}\}_{n=1}^{N})) \right\|_F^2 . \tag{2.1}$$

This is an objective function of an optimization problem and all the core tensors are the optimization objective. From [18], 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:

$$\mathbf{X}_{<n>} = \mathbf{G}_{(2)}^{(n)} (\mathbf{G}_{<2>}^{(\neq n)})^T, \tag{2.2}$$

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 nth core tensor, i.e., $\mathcal{G}^{(\neq n)} := \mathcal{X}(\{\mathcal{G}^{(n+1)}, \ldots, \mathcal{G}^{(n)}, \mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(n-1)}\})$. Because each of the core tensors is independent, we can optimize them independently. The optimization

Algorithm 1 Tensor ring weighted optimization (TR-WOPT)

```
input: incomplete tensor $\mathcal{T}$, weight tensor $\mathcal{W}$, TR-rank $R_1, \ldots, R_N$, and randomly initialized $\{\mathcal{G}^{(n)}\}_{n=1}^N$.
repeat
  for $n = 1$ to $N$ do
    Compute gradients of $\{\mathcal{G}^{(n)}\}_{n=1}^N$ according to (2.4).
  end for
  Update $\{\mathcal{G}^{(n)}\}_{n=1}^N$ by gradient descent algorithm.
until Stopping condition is satisfied
$\mathcal{Y} = P_\Omega(\mathcal{T}) + P_{\bar{\Omega}}(\mathcal{X}(\{\mathcal{G}^{(n)}\}_{n=1}^N))$
output: completed tensor $\mathcal{Y}$.
```

function w.r.t. $\mathcal{G}^{(n)}$ can be written as:

$$f(\mathcal{G}^{(n)}) = \frac{1}{2} \left\| \mathbf{W}_{<n>} * (\mathbf{T}_{<n>} - \mathbf{G}^{(n)}_{(2)} (\mathbf{G}^{(\neq n)}_{<2>})^T) \right\|_F^2, \tag{2.3}$$

where we consider other tensor cores remain fixed.

2.1.2 Gradient-based solving scheme

Next, we can deduce the partial derivatives of the objective function (2.3) w.r.t. $\mathbf{G}^{(n)}_{(2)}$ as follow:

$$\frac{\partial f}{\partial \mathbf{G}^{(n)}_{(2)}} = (\mathbf{W}_{<n>} * (\mathbf{G}^{(n)}_{(2)} (\mathbf{G}^{(\neq n)}_{<2>})^T - \mathbf{T}_{<n>}) \mathbf{G}^{(\neq n)}_{<2>}. \tag{2.4}$$

For $n = 1, ..., N$, 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.

For optimization method of TR-WOPT, in order to have a clear comparison with CP-WOPT which is also based on gradient descent methods, we adopt the same optimization method as paper [5]. The paper applies nonlinear conjugate gradient (NCG) with Hestenes-Stiefel updates [45] and the Moré-Thuente line search method [46]. The optimization method is implemented by an optimization toolbox named Pablano Toolbox [47].

2.2 Higher-order tensorization scheme for visual data

2.2.1 Visual data tensorization

TR-based algorithms can achieve high and stable performance in high-order tensors. In this section, we provide a Visual Data Tensorization (VDT) method to transform low-order tensor into higher-order tensor and improve the performance of our algorithms. The VDT method is derived from an image compression and entanglement methodology [48] which is to transform a gray-scale image of size $2^l \times 2^l$ into a real ket of a Hilbert space. The method cast the image to a higher-order tensor structure with an appropriate block structured addressing. Similar method named KA augmentation is proposed in [49] which extends the method in [48] to order-three visual data of size $2^l \times 2^l \times 3$.

Our VDT method is a generalization of the KA augmentation, and the visual data of various data sizes can be applied to our tensorization method. For visual data like RGB image, video, hyperspectral image, the first two orders of the tensor (e.g., $\mathbf{Y} \in \mathbb{R}^{U \times V}$) are named as the image modes. The 2D representation of the image modes cannot fully exploit the correlation and local structure of the data, so we propose the VDT method to strengthen the local structure correlation of visual data. The VDT method operates as follows: if the first two orders of a visual data tensor is $U \times V$ and can be reshaped to $u_1 \times u_2 \times \cdots \times u_l \times v_1 \times v_2 \times \cdots \times v_l$, then VDT method permutes and reshapes the data to size $u_1v_1 \times u_2v_2 \times \cdots \times u_lv_l$ and obtain the higher-order representation of the visual data. This higher-order tensor is a new structure of the original data: the first order of this higher-order tensor corresponds to a $u_1 \times v_1$ pixel block of the image, and the following orders of $u_2v_2, \cdots, u_lv_l$ describe the expanding larger-scale partition of the image. Based on VDT method, TR-based algorithms can efficiently exploit the structure information of visual data and achieve better low-rank representation. After the tensorized data is calculated by the completion algorithms, reverse operation of VDT is conducted to get the original image structure. The diagrams to explain the procedure of VDT are shown in Figure 2.1.

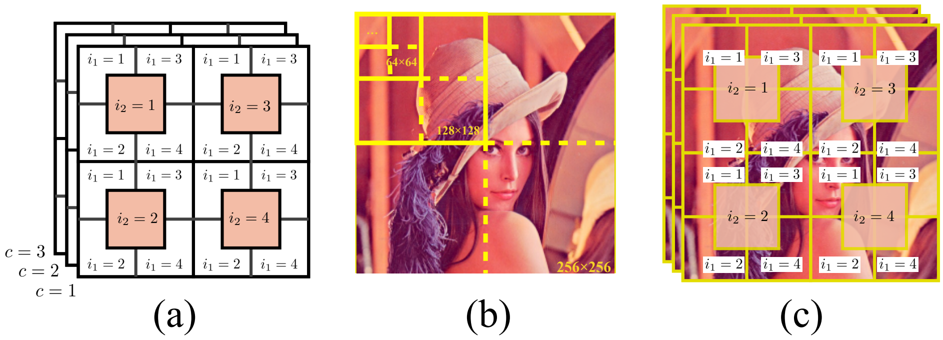


FIGURE 2.1: Illustration of the proposed VDT method. Figure (a) is the example of applying the VDT method on a $I \times I \times C$ tensor. Figure (b) and Figure (c) shows the example of the VDT operation on a $256 \times 256 \times 3$ image.

2.2.2 Validation of VDT on benchmark images

To verify the effectiveness of our VDT method, we choose a benchmark image “Lena” with 0.9 missing rate. We compare the performance of the five algorithms (TR-WOPT, CP-WOPT [5], FBCP [36], HaLRTC [19] and TLnR [41]) under three different data structures: order-3 tensor, order-9 tensor without VDT, order-9 tensor generated by VDT method. The order-3 tensor applies original image data structure of size $256 \times 256 \times 3$. The order-9 tensor without VDT is generated by directly reshaping data to the size $4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 3$. For order-9 tensor with VDT method, firstly the original data is reshaped to a order-seventeen tensor of size $2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 3$ and then it is permuted according to the order of $\{1\ 9\ 2\ 10\ 3\ 11\ 4\ 12\ 5\ 13\ 6\ 14\ 7\ 15\ 8\ 16\ 17\}$. Finally we reshape the tensor to a nine-order tensor of size $4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 4 \times 3$. This nine-order tensor with VDT is considered to be a better structure of the image data. The first order of the nine way tensor contains the data of a 2×2 pixel block of the image and the following orders of the tensor describe the expanding pixel blocks of the image. Most of the parameter settings follow the previous synthetic data experiments, and we tune the TR-rank, CP-rank and Tucker-rank of the corresponding algorithms to try to obtain better performance. Figure 2.2 shows the visual results of the five algorithms under the three different data structure. We can see that in the three-order tensor case, the results among the algorithms are similar. However, for nine-order cases, other algorithms fail the completion task while

TR-WOPT performs well. Furthermore, when the image is transformed to nine-order tensor by VDT method, we see the distinct improvement of our algorithm.

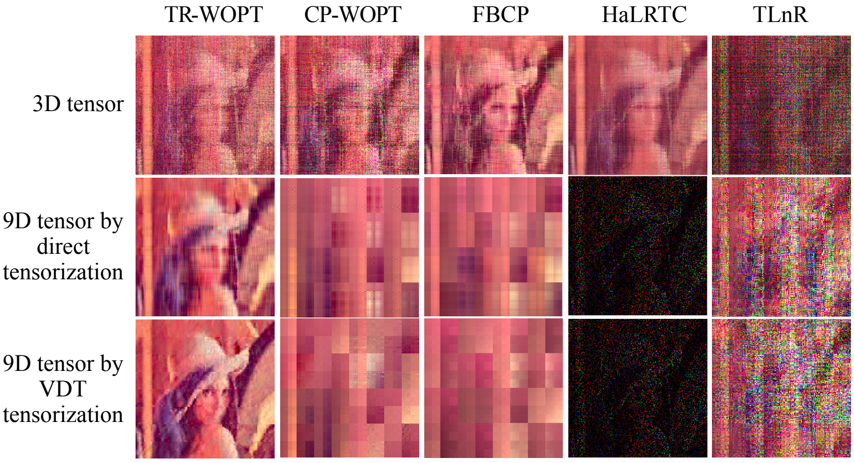

TR-WOPT
CP-WOPT
FBCP
HaLRTC
TLnR
3D tensor
9D tensor by direct tensorization
9D tensor by VDT tensorization

FIGURE 2.2: Visual results for completion of the 0.9 random missing "Lena" image under five algorithms. The first row applies original order-three tensor data, the second row applies order-nine tensor data without VDT method, and the third row applies order-nine tensor data generated by VDT method.

2.3 Irregular missing experiments of benchmark images

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 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 $= 10\log_{10}(255^2/\text{MSE})$, 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. It should be noted that all the experiments in this thesis are implemented on Matlab software and all the computations are conducted by using a Mac computer with Intel Core i7 and 64GB DDR3 memory.

The 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 and 0.9. We tune ranks and hyper-parameters of each algorithm and record the best completion results of each algorithm. Figure 2.3 and Table 2.1 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 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 of the algorithm drops quickly.

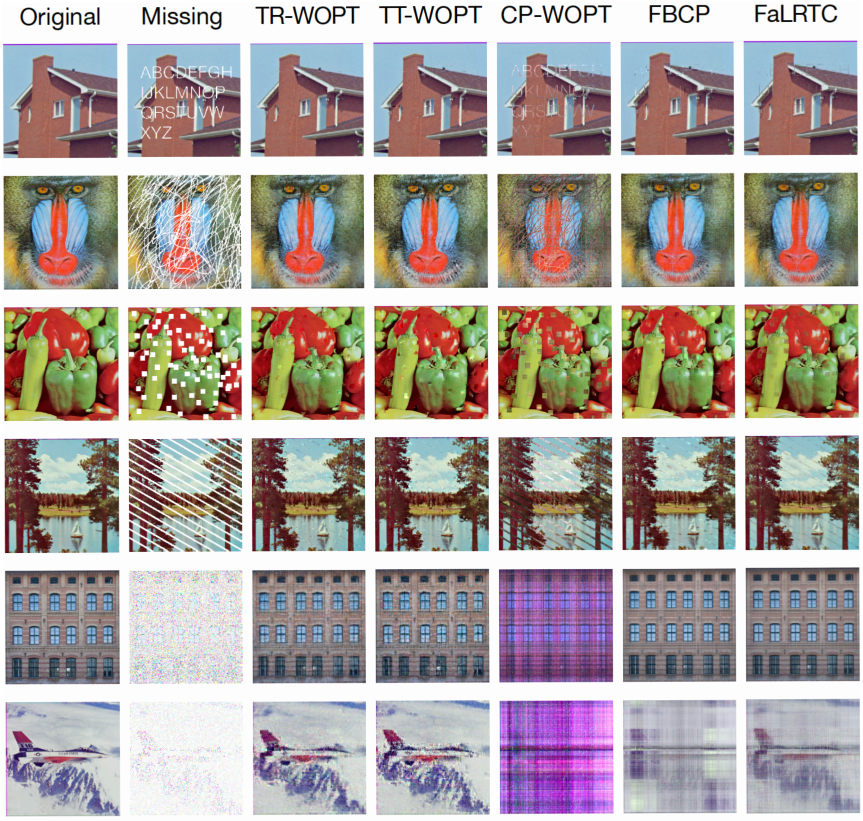


FIGURE 2.3: Visual completion results of five algorithms under six image missing situations. The first column and the second column are original images and images with specified missing patterns respectively. The following columns are the completion results of the five algorithms respectively. The first 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 and 0.9 respectively.

TABLE 2.1: Numerical completion results of five algorithms under six 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	**0.105**	0.119	0.231	0.114	0.106
	PSNR	**25.93**	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

2.4 Conclusion

Based on low-rank TRD, in this work, we proposed a new tensor completion algorithm named tensor-ring weighted optimization (TR-WOPT). The TR factors are optimized by the gradient-based method and used to predict the missing entries of the incomplete tensors. The experiments on benchmark images and the results show that TR-WOPT outperforms the related tensor completion 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 improving 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.

Chapter 3

Rank-robust Tensor Ring Decomposition and Completion

Based on the assumption that the underlying tensor is in TR structure, several tensor completion methods have been proposed and show high performance and efficiency [43,50,51]. However, the performance of TR-based completion methods is very sensitive to model selection. Due to the interdependence between the optimal rank of the decomposition model, the different data structure and the missing settings, it is rather challenging to determine the optimal rank for the data approximation. Moreover, the above algorithms have to manually tune the TR-rank to get the optimal solution which is time-consuming and inefficient. 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).

Furthermore, other tensor completion methods with automatic rank selection are difficult to be extended to TR-based method or less efficient. [6] proposes an 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 [52] and Tucker decomposition [53] exhibit poor efficiency when facing large-scale tensor and multi-linear rank model. In consideration of the problem, this chapter proposes two novel methods which can effectively alleviate the rank selection problem of TRD.

3.1 The rank of tensor ring decomposition

In this subsection, we first prove that the rank of the unfolded tensor is upper bounded by the TR-rank. Then, we prove that the low-rank constraint on TR factors will impose more low-rank constraint on the underlying tensor. For all the tensor decomposition models, model rank is the most significant parameter for tensor decomposition, as it controls the model complexity which is related to the approximation performance.

In recent studies, the rank of the tensor decomposition is always corresponding to a kind of low-rank structure of the underlying tensor. For instance, CP decomposition is to decompose the tensor into R rank-1 tensors. In this way, CP decomposition is to find the rank-1 latent components of the tensor [2]. However, directly minimizing CP-rank is an NP-hard problem, and all the existing CP-based algorithms are decomposition-based. For Tucker decomposition, the Tucker-rank is related to the rank of the mode-n unfoldings of the underlying tensor [41] (i.e.., $\mathrm{Rank}(\boldsymbol{\mathcal{X}}) := \sum_{n=1}^{N} \|\mathbf{X}_{(n)}\|_*$). Moreover, TT-rank is related to the rank of the n-unfoldings of the tensor [49] (i.e., $\mathrm{Rank}(\boldsymbol{\mathcal{X}}) := \sum_{n=1}^{N} \|\mathbf{X}_{[n]}\|_*$). So, minimizing Tucker-rank and TT-rank can be cast into solving a series of low-rank matrix completion problems. For TR decomposition, to the best of our knowledge, there are no studies about the relationship between the TR-rank and the underlying tensor low-rank structure. Next, we prove that all the n-unfoldings of the tensor after the tensor circular permutation are bounded by the TR rank. We prove that all the tensor circular permutation is bounded by TR-rank by the following theorem.

Theorem 1. *If the tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is in TR-format of rank $R_1, R_2, \dots, R_N$, then the rank of $\mathbf{X}_{\overleftarrow{c},[n]}$ is bounded by TR-rank as:*

$$Rank(\mathbf{X}_{\overleftarrow{c},[n]}) \leq R_c R_{t+1}, \tag{3.1}$$

where

$$t = \begin{cases} c+n, \ n \leq N-c; \\ N-n+1, \ otherwise. \end{cases} \tag{3.2}$$

Proof. Assume the TR factors of a tensor $\boldsymbol{\mathcal{X}}$ is $\{\boldsymbol{\mathcal{G}}^{(n)}\}_{n=1}^N \in \mathbb{R}^{R_n \times I_n \times R_{n+1}}$. By applying the property of circular permutation invariance of TR decomposition [18] (Theorem 2.1), $\boldsymbol{\mathcal{X}}_{\overleftarrow{c}} \in \mathbb{R}^{I_{c+1} \times \cdots \times I_N \times I_1 \times \cdots \times I_c}$ can be decomposed by the TR-factors of $\{\boldsymbol{\mathcal{G}}^{(c+1)}, \cdots, \boldsymbol{\mathcal{G}}^{(N)}, \boldsymbol{\mathcal{G}}^{(1)}, \cdots, \boldsymbol{\mathcal{G}}^{(c)}\}$, and for all the element in $\boldsymbol{\mathcal{X}}$, we have

$$\boldsymbol{\mathcal{X}}_{\overleftarrow{c}}(i_1, \dots, i_N) = \text{Trace}(\prod_{n=c+1}^{N} \mathbf{G}_{i_n}^{(n)} \prod_{n=1}^{c} \mathbf{G}_{i_n}^{(n)}). \tag{3.3}$$

By merging $\{\boldsymbol{\mathcal{G}}^{(c+1)}, \dots, \boldsymbol{\mathcal{G}}^{(t)}\}$ and $\{\boldsymbol{\mathcal{G}}^{(t+1)} \dots, \boldsymbol{\mathcal{G}}^{(c)}\}$ respectively, we have $\boldsymbol{\mathcal{G}}^{\leq n, \overleftarrow{c}}$ and $\boldsymbol{\mathcal{G}}^{>n, \overleftarrow{c}}$. and obtain the following equation:

$$\mathbf{X}_{\overleftarrow{c},[n]} = \mathbf{G}_{(2)}^{\leq n, \overleftarrow{c}} (\mathbf{G}_{<2>}^{>n, \overleftarrow{c}})^{\top}, \tag{3.4}$$

where $\boldsymbol{\mathcal{G}}^{\leq n, \overleftarrow{c}} \in \mathbb{R}^{R_c \times I_{c+1} \cdots I_{t+1} \times R_{t+1}}$ and $\boldsymbol{\mathcal{G}}^{(t+1,\dots,c)} \in \mathbb{R}^{R_{t+1} \times I_{t+1} \cdots I_c \times R_c}$. This indicates that there exist a matrix decomposition for $\mathbf{X}_{\overleftarrow{c},[n]}$ of rank $R_c R_{t+1}$, so we have $\text{Rank}(\mathbf{X}_{\overleftarrow{c},[n]}) \leq R_c R_{t+1}$. □

Theorem 1 proves that the unfoldings of the arbitrary circular permuted tensor have the rank upper bounds which are constrained by TR-rank. Compared to the Tucker-rank and TT-rank, which are the bound 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. Therefore, it can be inferred that unbalanced values of the unfolding rank may lead to unbalanced TR-rank. Meanwhile, we can also infer that the regularization on the TR-rank is equivalent to minimizing the rank of the tensor under unfoldings.

Next, we deduce the rank of the TR factors and the rank of the tensor unfoldings by the below theorem.

Theorem 2. *Given an N-th order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ which is in TR-format with the TR-rank $[R_1, R_2, \dots, R_N]^{\top}$ of which the TR factors are denoted by $\{\boldsymbol{\mathcal{G}}^{(n)}\}_{n=1}^N \in \mathbb{R}^{R_n \times I_n \times R_{n+1}}$, then the following inequality holds for all $n = 1, \dots, N$:*

$$Rank(\mathbf{G}_{(2)}^{(n)}) \geq Rank(\mathbf{X}_{(n)}). \tag{3.5}$$

Proof. For the n-th TR factor $\boldsymbol{\mathcal{G}}^{(n)}$, according to the work in [18], we have:

$$\mathbf{X}_{<n>} = \mathbf{G}_{(2)}^{(n)} (\mathbf{G}_{<2>}^{(\neq n)})^{\top}, \tag{3.6}$$

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 generated by merging all but the n-th TR factor. Hence, the relationship between the rank of the TR factors mode-2 unfoldings and the rank of the tensor unfoldings satisfies:

$$\begin{aligned}\mathrm{Rank}(\mathbf{X}_{<n>}) &\leq \min\{\mathrm{Rank}(\mathbf{G}^{(n)}_{(2)}), \mathrm{Rank}(\mathbf{G}^{(\neq n)}_{<n>})\}\\ &\leq \mathrm{Rank}(\mathbf{G}^{(n)}_{(2)}).\end{aligned} \tag{3.7}$$

The proof is completed by

$$\mathrm{Rank}(\mathbf{X}_{<n>}) = \mathrm{Rank}(\mathbf{X}_{(n)}) \leq \mathrm{Rank}(\mathbf{G}^{(n)}_{(2)}). \tag{3.8}$$

□

Theorem 2 proves the relation 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.

3.2 Tensor ring low-rank factors

3.2.1 Model formulation

We impose low-rank regularizations on each of the TR factors and so that our basic tensor completion model is formulated as follow:

$$\begin{aligned}\min_{[\mathcal{G}],\mathcal{X}} \ &\sum_{n=1}^{N} \mathrm{Rank}(\mathcal{G}^{(n)}) + \frac{\lambda}{2}\|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2,\\ &s.t.\ P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{T}).\end{aligned} \tag{3.9}$$

According to theorem 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 $\sum_{n=1}^{N} \|\mathbf{G}^{(n)}_{(1)}\|_*$+$\sum_{n=1}^{N} \|\mathbf{G}^{(n)}_{(3)}\|_*$. 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. Moreover, by theorem 2, we continue to impose nuclear norm regularization on the dimension-mode of the TR factors (i.e., $\|\mathbf{G}^{(n)}_{(2)}\|_*$), to explore more low-rank structure of the underlying tensor. Finally, by imposing overlapped nuclear norm on TR factors, the tensor ring low-rank factors (TRLRF) model can be expressed as:

$$\begin{aligned}\min_{[\mathcal{G}],\mathcal{X}} \ &\sum_{n=1}^{N}\sum_{i=1}^{3} \|\mathbf{G}^{(n)}_{(i)}\|_* + \frac{\lambda}{2}\|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2,\\ &s.t.\ P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{T}),\end{aligned} \tag{3.10}$$

where the optimization objectives are the recovered underlying tensor $\mathcal{X}$ and the TR factors $[\mathcal{G}]$, $\lambda > 0$ is the tuning parameter.

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. By the nuclear norm regularizations on the TR factors, we can largely decrease the computational complexity of our model compared to the algorithms which are based on model (1.7). Our TRLRF model has two distinctive advantages. Firstly, the low-rank assumption 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 optimal TR-rank and reduce the computational cost in the implementation.

3.2.2 ADMM solving scheme

The alternating direction method of multipliers (ADMM) [54] 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 [19,40], we can utilize ADMM to achieve more efficient solving schemes of our models. To solve the model in (3.10) 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

$$\begin{aligned}
&\min_{[\mathcal{M}],[\mathcal{G}],\mathcal{X}} \sum_{n=1}^{N}\sum_{i=1}^{3} \|\mathbf{M}_{(i)}^{(n,i)}\|_* + \frac{\lambda}{2}\|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2, \\
&s.t.\ \mathbf{M}_{(i)}^{(n,i)} = \mathbf{G}_{(i)}^{(n)}, n = 1,\dots,N,\ i = 1,2,3, \\
&\quad P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{T}),
\end{aligned} \tag{3.11}$$

where $[\mathcal{M}] := \{\mathcal{M}^{(n,i)}\}_{n=1,i=1}^{N,3}$ 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

$$\begin{aligned}
&L\left([\mathcal{G}],\mathcal{X},[\mathcal{M}],[\mathcal{Y}]\right) \\
&= \sum_{n=1}^{N}\sum_{i=1}^{3} \Big(\|\mathbf{M}_{(i)}^{(n,i)}\|_* + < \mathcal{Y}^{(n,i)}, \mathcal{M}^{(n,i)} - \mathcal{G}^{(n)} > \\
&+ \frac{\mu}{2}\|\mathcal{M}^{(n,i)} - \mathcal{G}^{(n)}\|_F^2\Big) + \frac{\lambda}{2}\|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2, \\
&s.t.\ P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{T}),
\end{aligned} \tag{3.12}$$

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.

Update $\mathcal{G}^{(n)}$. By using (3.23), the augmented Lagrangian function w.r.t. $\mathcal{G}^{(n)}$ can be simplified as

$$\begin{aligned}
L(\mathcal{G}^{(n)}) &= \sum_{i=1}^{3} \frac{\mu}{2}\left\|\mathcal{M}^{(n,i)} - \mathcal{G}^{(n)} + \frac{1}{\mu}\mathcal{Y}^{(n,i)}\right\|_F^2 \\
&+ \frac{\lambda}{2}\|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2 + C_{\mathcal{G}},
\end{aligned} \tag{3.13}$$

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 $n = 1, \ldots, N$, $\mathcal{G}^{(n)}$ can be updated by

$$\begin{aligned}\mathcal{G}_{+}^{(n)} = \text{fold}_{(2)}\Big(\big(\sum_{i=1}^{3} (\mu \mathbf{M}_{(2)}^{(n,i)} + \mathbf{Y}_{(2)}^{(n,i)}) \\ + \lambda \mathbf{X}_{<n>} \mathbf{G}_{<2>}^{(\neq n)} \big) \big(\lambda \mathbf{G}_{<2>}^{(\neq n),\top} \mathbf{G}_{<2>}^{(\neq n)} + 3\mu \mathbf{I} \big)^{-1} \Big),\end{aligned} \tag{3.14}$$

where $\mathbf{I} \in \mathbb{R}^{R_n^2 \times R_n^2}$ denotes the identity matrix.

Update $\mathcal{M}^{(n,i)}$. For $i = 1, 2, 3$, the augmented Lagrangian functions w.r.t. $[\mathcal{M}]$ is expressed as

$$\begin{aligned}L(\mathcal{M}^{(n,i)}) = \frac{\mu}{2} \| \mathcal{M}^{(n,i)} - \mathcal{G}^{(n)} + \frac{1}{\mu} \mathcal{Y}^{(n,i)} \|_F^2 \\ + \| \mathbf{M}_{(i)}^{(n,i)} \|_* + C_{\mathcal{M}}.\end{aligned} \tag{3.15}$$

The above formulation has a closed-form [55], which is given by

$$\mathcal{M}_{+}^{(n,i)} = \text{fold}_{(i)} \Big(D_{\frac{1}{\mu}} \big(\mathbf{G}_{(i)}^{(n)} - \frac{1}{\mu} \mathbf{Y}_{(i)}^{(n,i)} \big) \Big), \tag{3.16}$$

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 $D_{\beta}(\mathbf{A}) = \mathbf{U} max\{\mathbf{S} - \beta \mathbf{I}, 0\} \mathbf{V}^{\top}$.

Update $\mathcal{X}$. The augmented Lagrangian functions w.r.t. $\mathcal{X}$ is given by

$$\begin{aligned}& L(\mathcal{X}) = \frac{\lambda}{2} \| \mathcal{X} - \Psi([\mathcal{G}]) \|_F^2 + C_{\mathcal{X}}, \\ & s.t.\ P_{\Omega}(\mathcal{X}) = P_{\Omega}(\mathcal{T}),\end{aligned} \tag{3.17}$$

which is equivalent to the tensor decomposition based model in (1.6). The expression for $\mathcal{X}$ is updated by inputing the observed values in the corresponding entries, and by approximating the missing entries by updated TR factors $[\mathcal{G}]$ for every iteration, i.e.,

$$\mathcal{X}_{+} = P_{\Omega}(\mathcal{T}) + P_{\bar{\Omega}}(\Psi([\mathcal{G}])), \tag{3.18}$$

where $\bar{\Omega}$ is the set of indices of missing entries which is a complement to Ω.

Update $\mathcal{Y}^{(n,i)}$. For $n = 1, \ldots, N$ and $i = 1, 2, 3$, the Lagrangian multiplier $\mathcal{Y}^{(n,i)}$ is updated as

$$\mathcal{Y}_{+}^{(n,i)} = \mathcal{Y}^{(n,i)} + \mu (\mathcal{M}^{(n,i)} - \mathcal{G}^{(n)}). \tag{3.19}$$

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 Tensor ring latent nuclear norm

3.3.1 Latent nuclear norm

Latent tensor nuclear norm is first proposed in [24], whose form equals the infimum of sum of a sequence of matrix nuclear norm. Given an N-th order tensor $\mathcal{W}$, its latent nuclear norm is defined as

$$\|\mathcal{W}\|_L = \inf_{\mathcal{W}^{(1)} + \mathcal{W}^{(2)} + \ldots + \mathcal{W}^{(N)} = \mathcal{W}} \sum_{n=1}^{N} \| \mathbf{W}_{(n)}^{(n)} \|_*, \tag{3.20}$$

where $\|\cdot\|_*$ denotes the matrix nuclear norm. Compared to the conventional overlapped tensor nuclear norm [19], the latent model is proved to provide more precise completion results especially in the unbalanced case of tensor rank [24].

3.3.2 Model formulation

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, which is a more flexible low-rank constraint than overlapped nuclear norm, we employ the norm (3.20) to the factors. The work in [24] first proposes the "latent" norm model and shows that the mean square error of a "latent" norm method scales no greater than the "overlapped" norm method. Under the low-rank regularization of the latent model, the underlying tensor does not need to be low-rank at every mode. In this respect, we further decompose each TR factor into a sum of latent components. , i.e. for a given factor $\mathcal{G}^{(n)}$ we have

$$\mathcal{G}^{(n)} = \sum_{i=1}^{3} \mathcal{W}^{(n,i)}. \tag{3.21}$$

By regularizing the nuclear norm of the mode-i unfoldings of each component, our tensor ring latent nuclear norm (TRLNN) model is formulated as:

$$\begin{aligned} \min_{[\mathcal{G}],\mathcal{X}} \; & \sum_{n=1}^{N}\sum_{i=1}^{3} \|\mathbf{W}_{(i)}^{(n,i)}\|_* + \frac{\lambda}{2}\|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2 \\ & s.t. \; P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{T}), \mathcal{G}^{(n)} = \sum_{i=1}^{3} \mathcal{W}^{(n,i)}, \\ & n = 1,\dots,N, \end{aligned} \tag{3.22}$$

Similar to the TRLRF model, the TRLNN is also to optimize the TR factors and the underlying tensor simultaneously. From the model (3.22), we can see that the rank along different modes for each TR factor $\mathcal{G}^{(n)}$ is independently regularized by different components $\mathcal{W}^{(n,i)}$, 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 [24].

3.3.3 ADMM solving scheme

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:

$$\begin{aligned} & L\left([\mathcal{G}], \mathcal{X}, [\mathcal{W}], [\mathcal{Y}]\right) \\ & = \sum_{n=1}^{N} \Big(\sum_{i=1}^{3} \|\mathbf{W}_{(i)}^{(n,i)}\|_* + < \mathcal{Y}^{(n)}, \sum_{i=1}^{3} \mathcal{W}^{(n,i)} - \mathcal{G}^{(n)} > \\ & + \frac{\mu}{2} \| \sum_{i=1}^{3} \mathcal{W}^{(n,i)} - \mathcal{G}^{(n)}\|_F^2 \Big) + \frac{\lambda}{2}\|\mathcal{X} - \Psi([\mathcal{G}])\|_F^2, \\ & s.t. \; P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{T}). \end{aligned} \tag{3.23}$$

Due to the interdependence of $[\mathcal{G}]$, $[\mathcal{W}]$ and $[\mathcal{Y}]$, we provide the updating scheme of these variables as below.

Update $\mathcal{G}^{(n)}$. To update $[\mathcal{G}]$, for $n = 1, \ldots, N$, model (3.23) can be rewritten by:

$$\begin{aligned} L(\mathcal{G}^{(n)}) = \frac{\mu}{2} \Big\| \sum_{i=1}^{3} \mathcal{W}^{(n,i)} - \mathcal{G}^{(n)} + \frac{1}{\mu} \mathcal{Y}^{(n)} \Big\|_F^2 \\ + \frac{\lambda}{2} \| \mathcal{X} - \Psi([\mathcal{G}]) \|_F^2 + C_{\mathcal{G}}, \end{aligned} \tag{3.24}$$

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:

$$\begin{aligned} \mathcal{G}_+^{(n)} = \mathrm{fold}_{(2)} \Big(\big(\lambda \mathbf{X}_{<n>} \mathbf{G}_{<2>}^{(\neq n)} + \mu \sum_{i=1}^{3} \mathbf{W}_{(2)}^{(n,i)} \\ + \mathbf{Y}_{(2)}^{(n)} \big) \big(\lambda \mathbf{G}_{<2>}^{(\neq n),\top} \mathbf{G}_{<2>}^{(\neq n)} + \mu \mathbf{I} \big)^{-1} \Big), \end{aligned} \tag{3.25}$$

where $\mathbf{I} \in \mathbb{R}^{R_n^2 \times R_n^2}$ is an identity matrix.

Update $\mathcal{W}^{(n,i)}$. Similarly, for $n = 1, \ldots, N$, $i = 1, 2, 3$, in order to update $[\mathcal{W}]$, function (3.23) can be rewritten by:

$$\begin{aligned} L(\mathcal{W}^{(n,i)}) = \frac{\mu}{2} \| \mathcal{W}^{(n,i)} + \sum_{j=1, j\neq i}^{3} \mathcal{W}^{(n,i)} - \mathcal{G}^{(n)} \\ + \frac{1}{\mu} \mathcal{Y}^{(n,i)} \|_F^2 + \| \mathbf{W}_{(i)}^{(n,i)} \|_* + C_{\mathcal{W}}, \end{aligned} \tag{3.26}$$

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:

$$\mathcal{W}_+^{(n,i)} = \mathrm{fold}_{(i)} \Big(D_{\frac{1}{\mu}} \big(\mathbf{G}_{(i)}^{(n)} - \frac{1}{\mu} \mathbf{Y}_{(i)}^{(n,i)} - \sum_{j=1, j\neq i}^{3} \mathcal{W}^{(n,i)} \big) \Big). \tag{3.27}$$

Update $\mathcal{X}$. Next, to update $\mathcal{X}$, the augmented Lagrangian function (3.23) can be rewritten by:

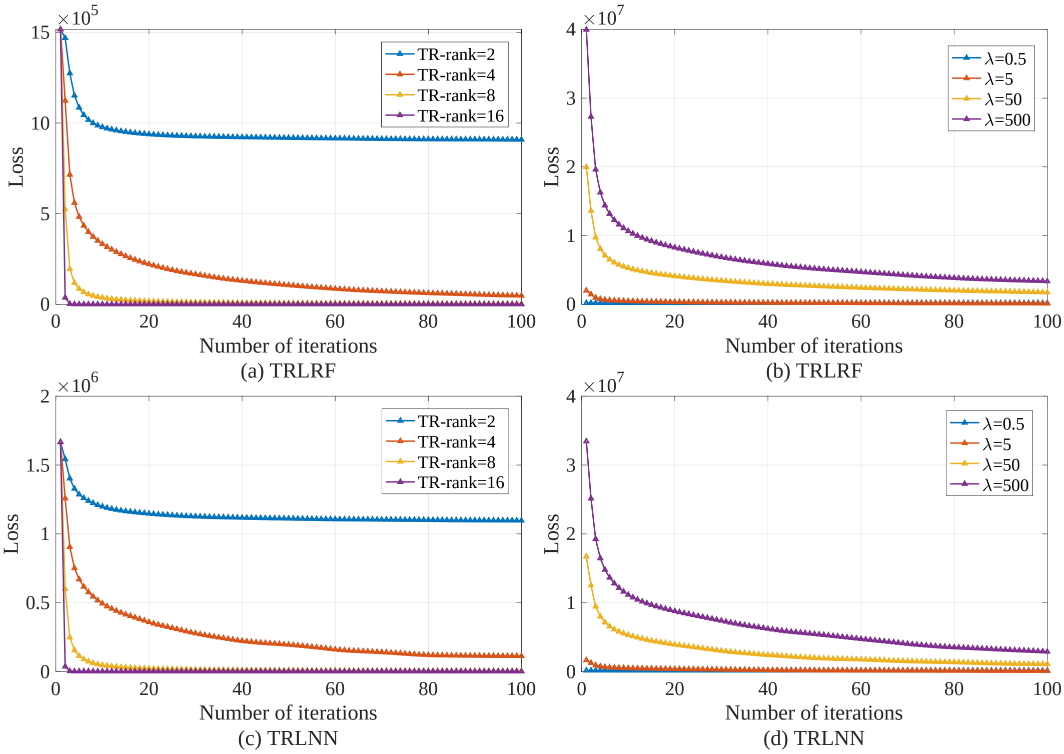
$$\begin{aligned} L(\mathcal{X}) &= \frac{\lambda}{2} \| \mathcal{X} - \Psi([\mathcal{G}]) \|_F^2 + C_{\mathcal{X}}, \\ s.t.\ & P_\Omega(\mathcal{X}) = P_\Omega(\mathcal{T}), \end{aligned} \tag{3.28}$$

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

$$\mathcal{X}_+ = P_\Omega(\mathcal{T}) + P_{\bar{\Omega}}(\Psi([\mathcal{G}])). \tag{3.29}$$

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

$$\mathcal{Y}_+^{(n)} = \mathcal{Y}^{(n)} + \mu \big(\sum_{i=1}^{3} \mathcal{W}^{(n,i)} - \mathcal{G}^{(n)} \big). \tag{3.30}$$



(a) TRLRF

(b) TRLRF

(c) TRLNN

(d) TRLNN

FIGURE 3.1: 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. Panel (a) and panel (b) show the convergence curves of TRLRF under different TR-rank and λ respectively. The convergence curves of TRLNN are presented in panel (c) and panel (d).

3.4 Algorithm analysis

3.4.1 Computational complexity

To recover a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I \times I \times \cdots \times I}$, for simplicity, the TR-rank is set equally as $R_1 = R_2 = \cdots = R_N = R$, then the computational complexity of updating $[\boldsymbol{\mathcal{M}}]$ for TRLRF and $[\boldsymbol{\mathcal{W}}]$ for TRLNN are mainly spent on the SVD calculation, which are $\mathcal{O}(NIR^3 + I^2R^2)$ equally. The complexity of HaLRTC [19] 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 by updating $[\boldsymbol{\mathcal{G}}]$, which are both $\mathcal{O}(NR^2I^N + NR^6)$. It is comparable to the computational complexity of TRALS [43] 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.

Algorithm 2 Solving scheme and parameter settings of TRLRF and TRLNN.

```
input: incomplete observation $P_\Omega(\mathcal{T})$, initial TR-rank $\{R_n\}_{n=1}^N$.
initialization: For $n = 1, \dots, N$, $i = 1, 2, 3$, random sample $\mathcal{G}^{(n)}$ by Gaussian distribution $N \sim (0, 1)$, $[\mathcal{Y}] = 0$, $[\mathcal{M}] = 0$, $[\mathcal{W}] = 0$, $\lambda = 5$, $\mu^0 = 1$, $\mu_{max} = 10^2$, $\rho = 1.01$, $tol = 10^{-6}$, $k = 0$, $k_{max} = 300$.
for $k = 1$ to $k_{max}$ do
  $\mathcal{X}_{last} = \mathcal{X}$.
  For TRLRF, update the variables by (3.14), (3.16), (3.18), (3.19)
  For TRLRF, update the variables by (3.25), (3.27), (3.29), (3.30).
  $\mu = max(\rho\mu, \mu_{max})$
  if $\|\mathcal{X} - \mathcal{X}_{last}\|_F / \|\mathcal{X}\|_F < tol$ then
    break
  end if
end for
output: completed tensor $\mathcal{X}$ and TR factors $[\mathcal{G}]$.
```

3.4.2 Convergence analysis

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 Figure 3.1). 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 Figure 3.1, 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.

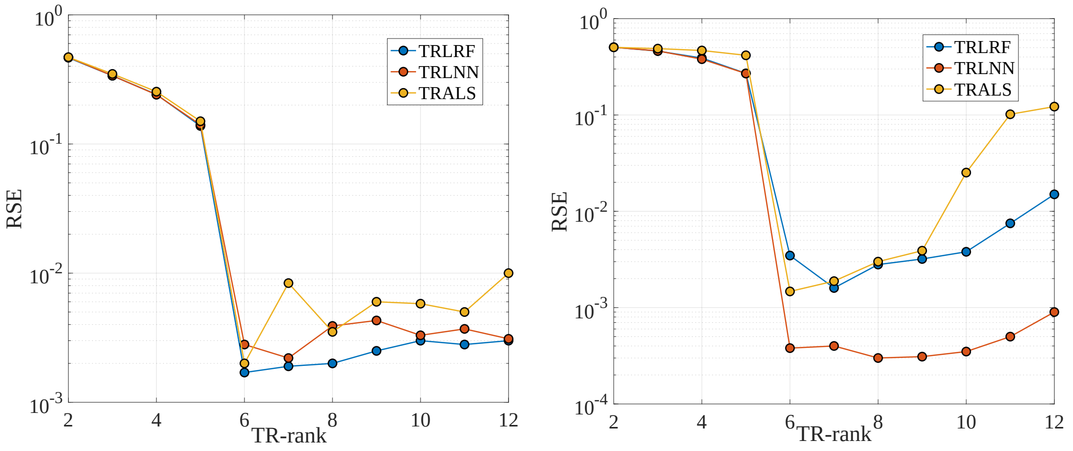
3.5 Experimental results

[1]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 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 2.

3.5.1 Synthetic data experiment

In the simulation experiment, three TR-based algorithms (i.e., TRLRF, TRLNN and TRALS [43]) are used to recover the two incomplete TR-structured tensors which have unbalanced TR-rank and balanced TR-rank, respectively. The two TR-structured synthetic tensors are of size $12 \times 12 \times 12 \times 12$ with 30% random missing. 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. From the results in Figure 3.2 (a), we can see 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

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



(a) Balanced case, real TR-rank: {6,6,6,6}

(b) Unbalanced case, real TR-rank: {3,6,3,6}

FIGURE 3.2: 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.

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. However, in Figure 3.2 (b), when the elements of the real TR-rank is unbalanced, TRLRF becomes less efficient than TRLNN when the TR-rank is selected as $\{6,6,6,6\}$, because only a subset of 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.

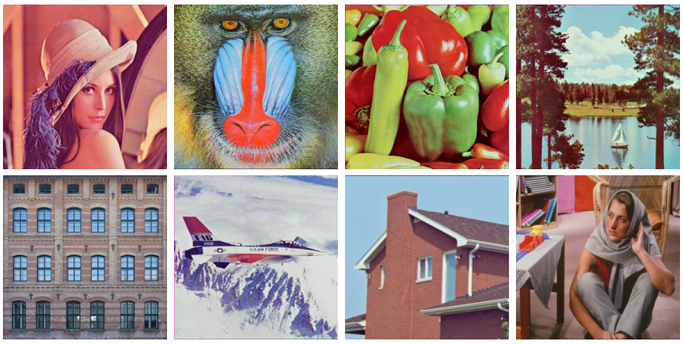


FIGURE 3.3: The eight benchmark images of size $256 \times 256 \times 3$.

3.5.2 Benchmark image inpainting

In this section, we adopt eight widely-used benchmark RGB images (Figure 3.3) 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 [50] 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 3.4 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.

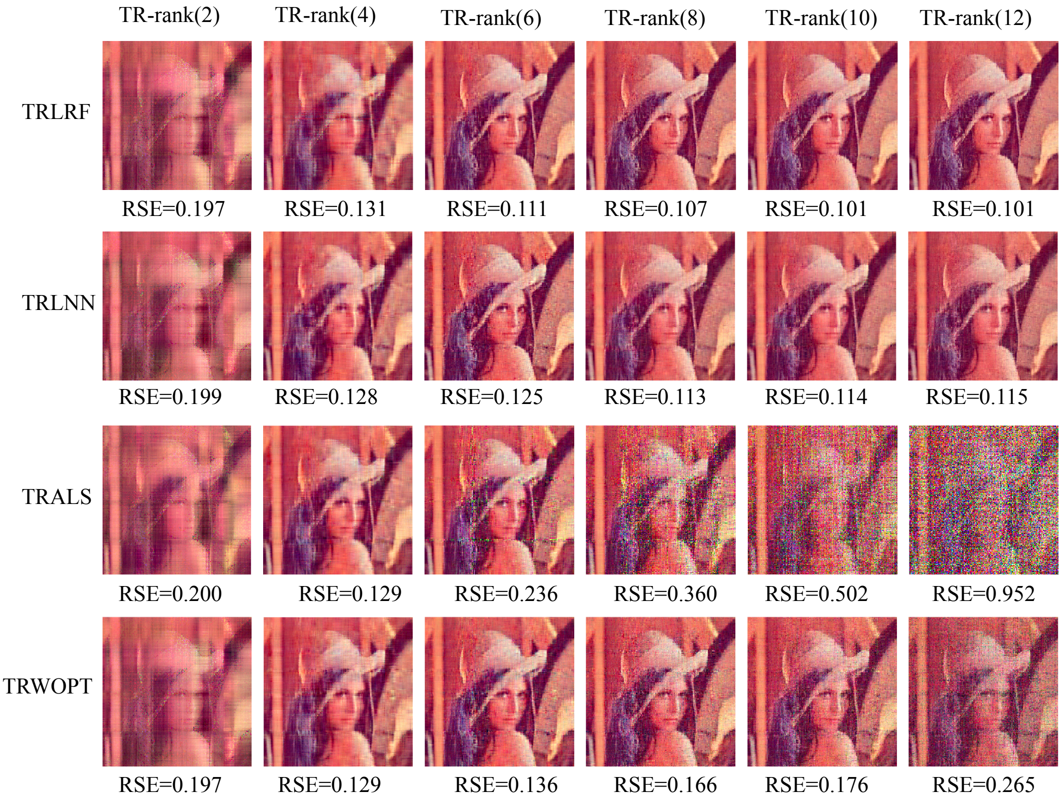
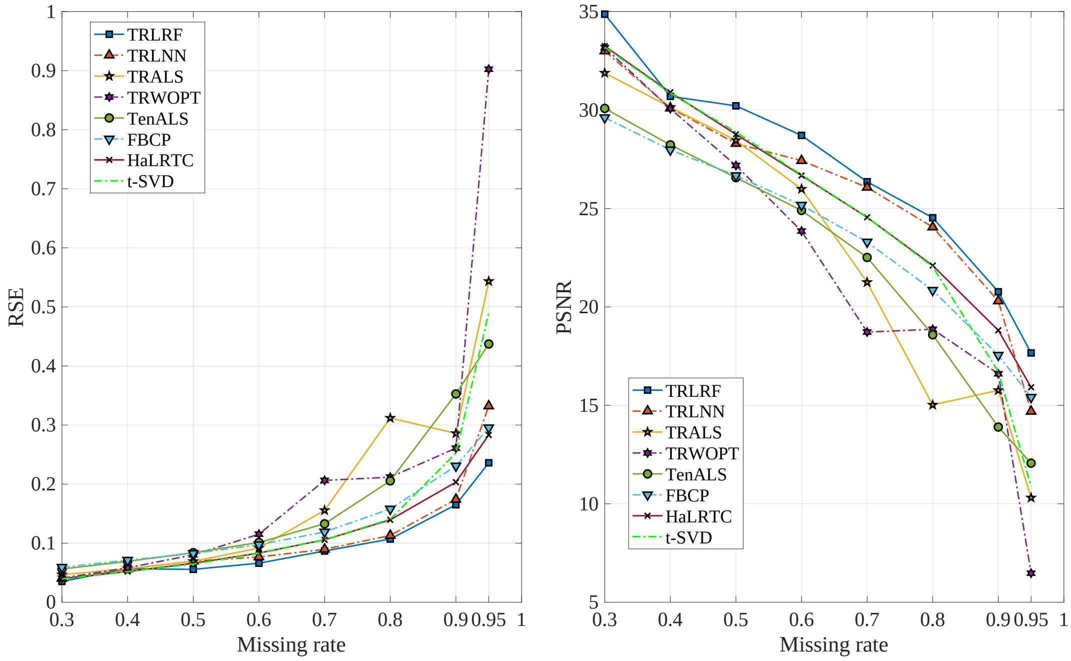


FIGURE 3.4: Visual completion results of the reshaped RGB image "Lena" of size 256 × 256 × 3.

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 CPD-based TenALS and FBCP [6], the Tucker-rank minimization based HaLRTC [19] and the tensor SVD scheme based t-SVD [56] 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 3.5 (a) and Figure 3.5 (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 performance 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 the tuning the optimal 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



(a) RSE results

(b) PSNR results

FIGURE 3.5: 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.

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.

3.6 Conclusion

In order to solve the rank selection problems of TRD and the computational efficiency problem most algorithms experience, in this work, 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. We propose a novel tensor completion approach which exploits the low-rankness of TR latent space by nuclear norm regularizations. Firstly, 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. Then, 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 task. Finally, the alternating direction method of multipliers (ADMM) solving scheme of the two models is 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. Furthermore, it is expected that the idea of imposing rank minimization constraint on tensor latent space can be extended to various tensor decomposition models in order to develop more efficient and robust algorithms.

Chapter 4

Max-norm Regularized Tensor Ring Decomposition

TRD decomposition is one of the most recently proposed and successful tensor decompositions which is based on the matrix product state (MPS) model [18]. It has been applied to various fields such as image recovery [43, 57], neural network acceleration [58], data compression and denoising [59]. However, the existing TR-based methods face two main drawbacks: lack of scalable algorithm and instability of the decomposition due to the non-convex model and difficulty of TR-rank selection, which lead to low computational efficiency and performance deficiency.

In order to solve these problems, in this section, we propose a novel completion method based on TRD. We define a novel regularizer termed TR-max-norm which is proved to be able to minimize the TR-rank. Next, we provide an algorithm based on stochastic gradient descent (SGD) to solve the proposed model. Finally, the algorithm is tested on a large-scale and highly sparse tensor, and the results show that our algorithm owns faster convergence and higher performance in comparison with the traditional TR-SGD algorithm.

4.1 Tensor max-norm regularization in TR-format

4.1.1 Matrix max-norm

The traditional low-rank tensor completion model illustrated by equation (1.6) is based on nuclear norm $||\cdot||_*$ regularization which is the summation of matrix singular values. It needs to process multiple singular value decompositions (SVD) to update the underlying tensor or matrix, which is of high computational cost. In consideration of the shortage of nuclear norm, we introduce an alternative low-rank regularizer of nuclear norm termed as max-norm. The related studies show that the matrix max-norm is a tighter bound of rank and a better regularizer for non-uniform missing, compared to nuclear norm [60]. The max-norm is applied as a convex regularizer for matrix [61]. Similar to the nuclear norm regularizer, max-norm also promotes the low-rankness of the matrix. The max-norm of a matrix $\mathbf{X} \in \mathbb{R}^{I_1 \times I_2}$ is defined by:

$$||\mathbf{X}||_{max} = \inf\{||\mathbf{U}||_{2,\infty}^2 ||\mathbf{V}||_{2,\infty}^2\},\ s.t.\ \mathbf{X} = \mathbf{U}\mathbf{V}^\top, \tag{4.1}$$

where the infimum of the model is obtained by enumerating all the possible factorizations. Moreover, if $\mathbf{U} \in \mathbb{R}^{I_1 \times r}$, the 2-infinity norm of $\mathbf{U}$ is calculated by $||\mathbf{U}||_{2,\infty} = \max_i^{I_1} ||\vec{u}_i||_2$ which denotes the maximum l_2-norm of all the rows in $\mathbf{U}$, and the 2-infinity norm of $\mathbf{V}$ is calculated in the same way. The matrix completion model with max-norm regularizer is formulated by:

$$\min_{\mathbf{X},\mathbf{U},\mathbf{V}} ||P_\Omega(\mathbf{T} - \mathbf{X})||_F^2, s.t.\ \mathbf{X} = \mathbf{U}\mathbf{V}^\top, ||\mathbf{X}||_{max} \leq \lambda, \tag{4.2}$$

where $\mathbf{T}$ is the partially observed matrix, and λ is the upper bound of the matrix max-norm which controls the strength of the regularizer. Moreover, by the semidefinite program solver, the max-norm can be cast into a new form

$$\|\mathbf{X}\|_{max} = \min_{(\mathbf{U},\mathbf{V}):\mathbf{X}=\mathbf{U}\mathbf{V}^\top} \max\{\|\mathbf{U}\|_{2,\infty}^2, \|\mathbf{V}\|_{2,\infty}^2\}, \tag{4.3}$$

so the matrix completion model can be written as a decomposition model with regularization on the decomposition factors:

$$\min_{\mathbf{U},\mathbf{V}} \|P_\Omega(\mathbf{T}-\mathbf{U}\mathbf{V}^\top)\|_F^2, s.t.\ \|\mathbf{U}\|_{2,\infty} \leq \lambda, \|\mathbf{V}\|_{2,\infty} \leq \lambda. \tag{4.4}$$

The model is easy to be solved by projected gradient descent methods and it has been applied to matrix completion problems [62], and it has obtained empirically better results than nuclear norm regularization in non-uniform data, e.g., collaborative filtering. Moreover, in tensor field, high performance methods, especially for non-uniform missing data completion is in demand. So we attempt to apply the idea of max-norm on tensor field, to develop high performance algorithms.

4.1.2 Tensor ring max-norm

Inspired by the efficient formulation of max-norm-based completion, in this section, we propose a new optimization model for low-rank tensor approximation based on TRD. We first prove that the TR-rank bounds the rank of the circular unfoldings of the TR-structured tensor. Then, we propose a new tensor regularizer in TR-format. Next, we show how the new regularizer imposes low-rank constraint to minimize the TR-rank. Finally, we provide an efficient algorithm to solve the TR decomposition problem with incomplete tensor.

We extent the matrix max-norm to tensor by TRD and define a new max-norm regularizer named TR-max-norm as:

Definition 1. *(TR-max-norm) By rearranging the rank-modes of TR factors in to the same order, the TR-max-norm of tensor is defined as:*

$$\|\boldsymbol{\mathcal{X}}\|_{TR-max} := \inf_{\substack{[\boldsymbol{\mathcal{G}}]\in\mathbb{R}^{R_n\times I_n\times R_{n+1}} \\ \Psi([\boldsymbol{\mathcal{G}}])=\boldsymbol{\mathcal{X}}}} \{\prod_{n=1}^{N} \|\mathbf{G}_{(2)}^{(n)}\|_{2,\infty}\} \tag{4.5}$$

It should be noted that when $N=2$, $\mathbf{X} = \mathbf{G}_{(2)}^{(1)}(\mathbf{G}_{<n>}^{(2)})^\top$, we have the same definition with matrix max-norm if the matrix is in TR format. Next, we provide the relationship between the matrix max-norm of tensor circular unfoldings and the tensor TR-max-norm by the below theorem:

Theorem 3. *If $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1\times\cdots\times I_N}$ can be decomposed into TR factors $\boldsymbol{\mathcal{G}}^{(1)},\ldots,\boldsymbol{\mathcal{G}}^{(N)}$, then for $\forall c, \forall n$ we have:*

$$\|\mathbf{X}_{\overleftarrow{c},[n]}\|_{max} \leq \|\boldsymbol{\mathcal{X}}\|_{TR-max}. \tag{4.6}$$

Proof. By the definition of matrix max-norm, we have:

$$\|\mathbf{X}_{\overleftarrow{c},[n]}\|_{max} = \min_{\mathbf{X}_{\overleftarrow{c},[n]}=\mathbf{G}_{(2)}^{\leq n,\overleftarrow{c}}(\mathbf{G}_{<2>}^{>n,\overleftarrow{c}})^\top} \max\{\|\mathbf{G}_{(2)}^{\leq n,\overleftarrow{c}}\|_{2,\infty}^2, \|\mathbf{G}_{<2>}^{>n,\overleftarrow{c}}\|_{2,\infty}^2\}, \tag{4.7}$$

Moreover, for $n = 1,\ldots,N$, $i_n \in \{1,2,\ldots,N\}$, when we enumerate all the combinations of the index to obtain the $2,\infty$-norm of the two merged TR factors, $\exists\ \|\mathbf{G}_{(2)}^{\leq n,\overleftarrow{c}}\|_{2,\infty}^2 =$

$\| \prod_{n=c+1}^{t} \mathbf{G}_{i_n}^{(n)} \|_F^2$ and $\|\mathbf{G}_{<2>}^{>n,\overleftarrow{c}}\|_{2,\infty}^2 = \| \prod_{n=t+1}^{c} \mathbf{G}_{i_n}^{(n)} \|_F^2$, where By Cauchy-Schwarz inequality, we have $\|\mathbf{G}_{(2)}^{\leq n,\overleftarrow{c}}\|_{2,\infty}^2 \leq \prod_{n=c+1}^{t} \|\mathbf{G}_{i_n}^{(n)}\|_F^2$ and $\|\mathbf{G}_{<2>}^{>n,\overleftarrow{c}}\|_{2,\infty}^2 \leq \prod_{n=t+1}^{c} \|\mathbf{G}_{i_n}^{(n)}\|_F^2$. Finally, we have

$$\|\mathbf{X}_{\overleftarrow{c},[n]}\|_{max} \leq \min\{ \prod_{n=c+1}^{t} \|\mathbf{G}_{i_n}^{(n)}\|_{2,\infty}^2, \prod_{n=t+1}^{c} \|\mathbf{G}_{i_n}^{(n)}\|_F^2\} \leq \prod_{n=1}^{N} \|\mathbf{G}_{(2)}^{(n)}\|_{2,\infty}. \tag{4.8}$$

From the definition of TR-max-norm, we can deduce that $\prod_{n=1}^{N} \|\mathbf{G}_{(2)}^{(n)}\|_{2,\infty} \leq \|\boldsymbol{\mathcal{X}}\|_{TR-max}$, so the max-norm of all the tensor circular unfoldings are upper-bounded by the TR-max-norm. □

After the relationship between the tensor unfolding max-norm and the TR-max-norm is deduced, we also attempt to find the relationship of the max-norm and nuclear norm of the unfoldings.

Theorem 4. *The nuclear norm of* $\mathbf{X}_{\overleftarrow{c},[n]}$ *is upper-bounded by its weighted max-norm as:*

$$\|\mathbf{X}_{\overleftarrow{c},[n]}\|_* \leq R_c R_{t+1} \|\mathbf{X}_{\overleftarrow{c},[n]}\|_{max} \tag{4.9}$$

Proof. From the theory in [63], the nuclear norm of a matrix can be transformed into the form of the squared Frobenius norm of its factorizations. We can deduce that

$$\begin{aligned}
\|\mathbf{X}_{\overleftarrow{c},[n]}\|_* &= \min_{(\mathbf{U},\mathbf{V}):\mathbf{X}=\mathbf{U}\mathbf{V}^\top} \frac{1}{2}(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2) \\
&\leq \min_{\mathbf{L},\mathbf{R}} \sum_{i=1}^{R_c R_{t+1}} \frac{1}{2}(\|\mathbf{L}(i,:)\|_2^2 + \|\mathbf{R}(i,:)\|_2^2) \\
&\leq \min_{\mathbf{L},\mathbf{R}} R_c R_{t+1} \frac{1}{2}(\max_i \|\mathbf{L}(i,:)\|_2^2 + \max_i \|\mathbf{R}(i,:)\|_2^2) \\
&\leq \min_{\mathbf{L},\mathbf{R}} R_c R_{t+1} \max\{\|\mathbf{L}\|_{2,\infty}^2, \|\mathbf{R}\|_{2,\infty}^2) \\
&= R_c R_{t+1} \|\mathbf{X}_{\overleftarrow{c},[n]}\|_{max}
\end{aligned} \tag{4.10}$$

□

This indicates that minimizing the max-norm of the tensor circular unfoldings is to minimize the upper-bound of the corresponding nuclear norm. Therefore, the max-norm of the tensor circular unfoldings is a tighter regularization compared to its nuclear norm.

4.2 Max-norm regularized tensor ring completion

Though TR-WOPT mentioned in *Chapter* 2 achieves high performance in data completion task, it considers all the missing entries of data as zero, and it computes the whole scale of tensor in every iteration. If the data scale is huge and the missing rate is high, TR-WOPT will cost much computer memory space and be ineffective as it computes the whole scale tensor of which only a small percentage of entries is useful. In order to solve the problems of TR-WOPT as mentioned, we propose a TRD algorithm based on stochastic gradient descent, which only uses observed entries to compute the gradient of every core tensor is proposed.

4.2.1 Model formulation

From the above theorems, we establish the relationship between the TR-max-norm and the TR-rank, thus proving the TR-max-norm regularization can minimize the TR-rank. This imposes the abilities of convergence stability and enhanced decomposition performance. The proposed model is as follow:

$$\min_{\mathcal{X}=\Psi([\mathcal{G}])} f(\mathcal{X},\mathcal{T};\Omega),\ s.t.\|\mathcal{X}\|_{TR-max} \leq \lambda, \tag{4.11}$$

where $\mathcal{T}$ is the observed tensor, in which the observed entry is in the set Ω, and $\mathcal{X}$ is the underlying tensor with TR structure. Similar to matrix max-norm, we can cast the TR-max-norm into the semidefinite program which is equal to the following scheme:

$$\|\mathcal{X}\|_{TR-max} := \min_{\mathcal{X}=\Psi([\mathcal{G}])} \max\{\|\mathbf{G}_{(2)}^{(1)}\|_{2,\infty}, \cdots, \|\mathbf{G}_{(2)}^{(N)}\|_{2,\infty}\} \tag{4.12}$$

Then, if $\|\mathcal{X}\|_{TR-max}$ is bounded by λ, solving the regularizer is equal to project the $2,\infty$-norm of all the mode-2 slices of the TR factors onto the set of $\{\|\mathbf{G}_{(2)}^{(n)}\|_{2,\infty}\}_{n=1}^{N} \leq \lambda$, for $n = 1,\ldots,N$. So the original model can be transformed into solving the TR decomposition problem with conditional projection:

$$\min_{[\mathcal{G}]} f([\mathcal{G}],\mathcal{T};\Omega),\ s.t.\|\mathbf{G}_{(2)}^{(n)}\|_{2,\infty} \leq \lambda_n, n = 1,\ldots,N \tag{4.13}$$

The TRD model is non-convex, however, according to the experimental results of the following sections, with the regularization on the TR factors, the model is to have faster and more stable convergence. In the next section, we propose a scalable and efficient algorithm to solve the model based on gradient descent method. We adopt the mini-batch stochastic gradient descent (SGD) which takes the balance of low computational complexity and stable convergence.

4.2.2 Projected mini-batch SGD

We adopt the mini-batch stochastic gradient descent (SGD) algorithm in consideration of the low memory cost and low per-iteration complexity. From equation (1.3) we see that an entry of TR-approximated tensor is calculated by the multiple multiplication of the mode-2 slice of the TR factor w.r.t. the indices. The element-wise loss function can be formulated as:

$$\min_{\mathbf{G}_{i_1}^{(1)},\ldots,\mathbf{G}_{i_N}^{(N)}} \sum_{\{i_1\ldots,i_N\}\in\Omega} f(\mathrm{Trace}(\prod_{n=1}^{N}\mathbf{G}_{i_n}^{(n)}), t_{i_1,\ldots,i_N}),\ s.t.\{\|\mathbf{G}_{(2)}^{(n)}\|_{2,\infty}\}_{n=1}^{N} \leq \lambda, \tag{4.14}$$

where $t_{i_1,\ldots,i_N}$ is an observed entry of incomplete tensor $\mathcal{T}$. When we sample one entry of index $\{i_1, i_2, \ldots, i_N\}$, the gradient of the corresponding slice can be calculated by:

$$\frac{\partial f_{i_1,\ldots,i_N}}{\partial \mathbf{G}_{i_n}^{(n)}} = \nabla(f_{i_1,\ldots,i_N})(\prod_{k=n+1}^{N}\mathbf{G}_{i_k}^{(k)}\prod_{k=1}^{n-1}\mathbf{G}_{i_k}^{(k)})^{\top}. \tag{4.15}$$

When we calculate the batch-sized gradient of the TR model, the statistic gradients of the mode-2 slices are the average values of the gradient accumulations. Let M be the batch-size, and we define the index of the mth entry in the mini-batch as $\{i_1^m, i_2^m, \cdots, i_N^m\}$, $m = 1,\cdots,M$. For $i_n \in \{1,2,\ldots,I_n\}$, $n = 1,2,\ldots,N$, the statistic gradient of every mode-2 slice is calculated as:

$$\frac{\partial f}{\partial \mathbf{G}^{(n)}_{i_n}} = \frac{1}{N_{i_n^m=i_n}} \sum_{\substack{m=1\\ m:i_n^m=i_n}}^{M} \nabla(f_m)(\prod_{k=n+1}^{N} \mathbf{G}^{(k)}_{i_k} \prod_{k=1}^{n-1} \mathbf{G}^{(k)}_{i_k})^{\top}, \tag{4.16}$$

where $N_{i_n^m=i_n}$ is the number of repetition of the gradient calculation of a slice $\mathbf{G}^{(n)}_{i_n}$. Then, for every iteration of our algorithm, the updated TR-factors are evaluated and projected according to the regularizer. For $i_n \in \{1, 2, \ldots, I_n\}$ and $n = 1, \ldots, N$, the projector is as follows:

$$\mathbb{P}(\mathbf{G}^{(n)}_{i_n}) = \begin{cases} \frac{\lambda_n}{\|\mathbf{G}^{(n)}_{i_n}\|_{2,\infty}} \mathbf{G}^{(n)}_{i_n}, \ \|\mathbf{G}^{(n)}_{i_n}\|_{2,\infty} > \lambda_n; \\ \mathbf{G}^{(n)}_{i_n}, \text{ otherwise.} \end{cases} \tag{4.17}$$

4.3 Experimental results

4.3.1 Experimental settings

Adam gradient descent scheme. For gradient descent scheme, we employ an algorithm named Adaptive Moment Estimation (Adam) as our gradient descent method, it has been widely used in stochastic-gradient-based optimization [64]. We take the simple update rule of Adam algorithm as: $\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{v_t+\epsilon}} m_t$, where θ is the objective variables, t is the number of iteration, η is the learning rate, m_t and v_t are termed as the first moment estimate and second moment estimate, which are defined as $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$ and $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$ respectively, g_t is the gradient (i.e., calculated by (4.16) in our method). β_1, β_2 and ϵ are hyper parameters which are set as 0.9, 0.999 and 10^{-8} respectively according to the author.

Stopping condition. Furthermore, we consider to use the relative tolerance as one of the the stopping conditions in the following experiments to increase efficiency. For the TR factors of iteration t and $t + 1$, the relative tolerance of the TR factors is defined as:

$$tol = \max\{\frac{\|\boldsymbol{\mathcal{G}}^{(1)}_{t+1} - \boldsymbol{\mathcal{G}}^{(1)}_{t}\|_F}{\|\boldsymbol{\mathcal{G}}^{(1)}_{t+1}\|_F}, \ldots, \frac{\|\boldsymbol{\mathcal{G}}^{(N)}_{t+1} - \boldsymbol{\mathcal{G}}^{(N)}_{t}\|_F}{\|\boldsymbol{\mathcal{G}}^{(N)}_{t+1}\|_F}\} \tag{4.18}$$

When the tol reaches the expected value, or the iteration reaches the prescribed time, the iteration will be stopped.

The bound of TR-max-norm. The selection of the upper bound of the TR-max-norm is the most important parameter for the model, as it largely influences the convergence speed and decomposition performance. We provide a simple scheme for changing the upper bound λ adaptively. For every iteration, for $n = 1, \ldots, N$, we calculate λ_n as follows:

$$\lambda_n = \kappa \|\boldsymbol{\mathcal{G}}^{(n)}\|_F / I_n \tag{4.19}$$

The selection scheme applies the average l_2-norm of all the rows of the mode-2 slice of the TR factors as the basic reference, and κ is the tuning parameter which controls the influence of the max-norm regularizer. We empirically choose κ from 5 to 10 w.r.t the different data set to obtain the best performance. Finally, the algorithm is summarized in Algorithm 3.

4.3.2 Convergence analysis

It should be noted that the global convergence of our algorithm cannot be theoretically guaranteed, because the proposed algorithm is non-convex. However, the convergence property

Algorithm 3 Projected mini-batch SGD for (4.11)

```
input: $P_{\Omega}(\mathcal{T})$, initial TR rank $\{R_n\}_{n=1}^{N}$,
initialization: For $n = 1, \dots, N$, random sample $\mathcal{G}^{(n)}$ by distribution $N \sim (0,1)$, batch size $1e4$, maximum iteration $1e3$, $tol = 1e-4$, $\kappa = 10$.
repeat
  Random sample entries from $\Omega$ w.r.t. the batch size.
  repeat
    for $n = 1$ to $N$ do
      Compute gradients according to (4.15)
    end for
  until All the gradients of the batch are computed
  Compute the statistical gradients according to (4.16)
  Do gradient descent according to Adam scheme.
  Project the updated TR-factor according to (4.17) and (4.19).
until Maximum iteration is reached, or (4.18) is satisfied.
output: optimized TR factors $[\mathcal{G}]$.
```

can be verified empirically (see the simulation experiment in Figure 4.1). Moreover, the extensive results in the experimental section also illustrate the stability and effectiveness of the proposed algorithm.

4.3.3 Synthetic data analysis

In this section, we test our method in a large and sparse synthetic tensor. All the loss function in the experiments are set as the squared Frobenius norm, i.e., $f(\cdot) = \|\cdot\|_F^2$.

In the simulation experiment, we aim to test the algorithm stability in comparison with the baseline algorithm TR-SGD [30] by different rank selection. Moreover, the influence of the hyper-parameter κ to the completion performance is also investigated. We use the synthetic values which are sampled from a highly oscillating function: $f(x) = sin\frac{x}{4}cos(x^2)$ to generate the tensor data. We sample 10^8 values from the function, then the sampled values are reshaped to the tensor of size $100 \times 100 \times 100 \times 100$. The tensor data is expected to own good low-rank properties and easy to be approximated by tensor decompositions. We randomly remove 99.9% entries of the tensor and conduct the completion experiments by standard TR-SGD and the TR-max-norm regularized SGD (proposed). For TR-rank selection, we choose the TR-rank as $\{2,2,2,2\}$ and $\{5,5,5,5\}$ respectively. All the other optimization parameters of standard TR-SGD and the proposed method follow the statement in Algorithm 3.

Figure 4.2 shows the changes in the loss function value with the iteration increases. From Figure 4.2 (a) we can see, when the TR-rank is set as $\{2,2,2,2\}$, our algorithm converges much faster than standard TR-SGD. Moreover, because of the high sparsity of the tensor data, when TR-rank is set as $\{5,5,5,5\}$, the standard TR-SGD becomes overfitting and fails to converge, while the algorithm with TR-max-norm regularization obtains fast convergence and good performance. The results also highlight the property of rank selection robustness of the proposed algorithm. This is because the TR-max-norm regularization provides the ability to lower the initial TR-rank if the prescribed TR-rank is higher than we need. For Figure 4.2 (b), the TR-rank is set as $\{5,5,5,5\}$ for all the cases and the hyper-parameter κ is set as 2, 5, 10 for the proposed algorithm, respectively. It should be noted that the lower value of κ provides higher regularization weight. From the results we can see, our algorithm fails when $\kappa = 2$ because the regularization is too strong, while the $\kappa = 5$ obtains the fastest convergence. When $\kappa = 10$, the convergence become slow because of the low regularization

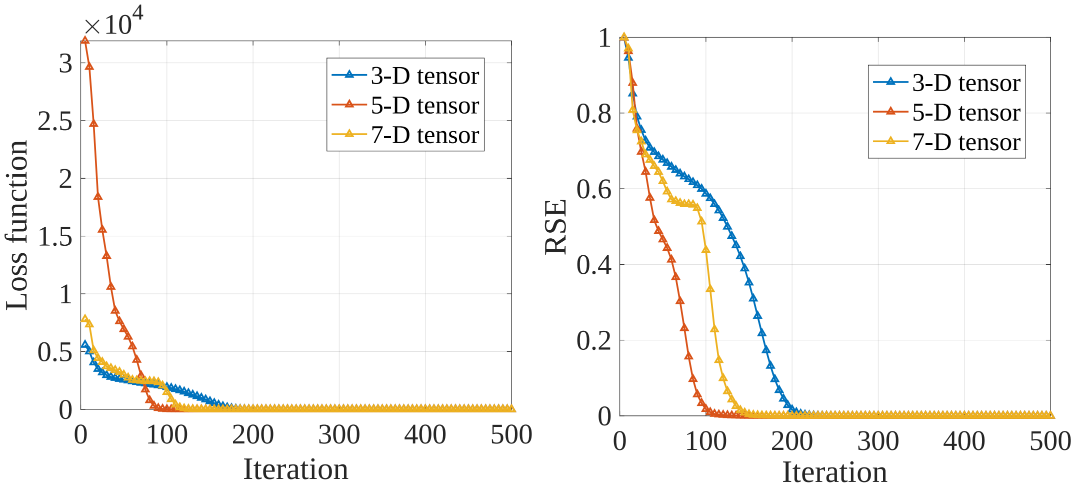
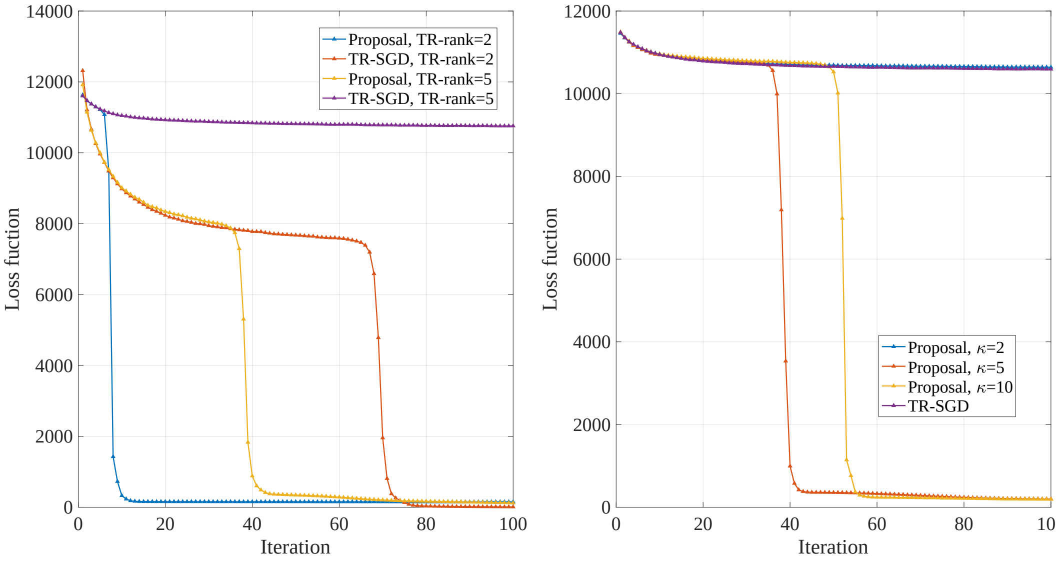


FIGURE 4.1: The convergence property for the proposed algorithm under different tensors. The tensors are of size $25 \times 25 \times 25$ (3D), $7 \times 7 \times 7 \times 7 \times 7$ (5D), $5 \times 5 \times 5 \times 5 \times 5 \times 5 \times 5$ (7D), respectively, and generated by randomly initialized TR factors with TR-rank 2. The missing rate of all the tensors are set as 0.9 equally. Each experiment is conducted 100 times independently and the graphs records the average results.



(a) different selection of TR-rank

(b) different bound of TR-max-norm

FIGURE 4.2: Convergence property of the proposed method and the standard TR-SGD w.r.t. the increase of the iteration. For (a), we set $\kappa = 5$ for our algorithm. For (b), we set the TR-rank of the two algorithms as $\{5, 5, 5, 5\}$.

weight. From the trend, we can see that the selection of the hyper-parameter is very essential to the performance of our algorithm.

4.4 Conclusion

In this work, by leveraging the latent space of the TR factors, we propose a novel tensor low-rank regularizer which minimizes the TR-rank of the underlying tensor. The novel regularizer is theoretically proved to regularize the low-TR-rank approximation of the underlying tensor. Then, to solve the TRD model with the TR-max-norm regularization, we propose an efficient and high-performance completion algorithm by projected mini-batch stochastic gradient descent. The proposed algorithm show fast and stable convergence in synthetic data in comparison with the standard TR-SGD algorithm. Furthermore, the tensor max-norm regularization is a promising framework and it should be extended to other tensor decomposition models to improve the performance of the existing algorithms.

Chapter 5

Large-scale Tensor Denoising via Tensor Ring Decomposition

With the development of data acquisition and storage technology, large-scale data (i.e., big data) becomes ubiquitous in many fields such as computational neuroscience, signal processing, machine learning, and pattern recognition [65]. Among these fields, large amounts of multi-dimensional data (i.e., tensors) of high dimensionality are generated. Big data is of large volume and complex, which is hard to process by traditional methods like singular value decomposition (SVD) and principal component analysis (PCA) due to their high computational complexity. Moreover, in order to fit in these algorithms, traditional methods need operations to transform tensor data to matrices and vectors, which leads to the loss of adjacent structure information and the redundant space cost of data.

Though tensor decomposition has the merits of data structure conservation and high data representation ability, when dealing with large-scale data, traditional deterministic algorithms like alternative least squares (ALS) and gradient descent (GD) are of low-efficiency due to their high computational cost and low convergence rate. Therefore, fast and efficient algorithms are of high demand for large-scale tensor decomposition.

5.1 Randomized algorithms

The randomization technology is a powerful computation acceleration technique which has been proposed and studied for decades [66, 67]. Recently, randomness-based tensor decomposition has drawn people's attention. Literature [68] proposes a randomized algorithm for large-scale tensors based on Tucker decomposition, it can process arbitrarily large-scale tensors with low multi-linear rank and the method shows robustness to various data set. A randomized least squares algorithm for CPD is proposed in [69], it is much faster than the traditional CP least squares algorithm and can keep the high performance at the same time. The work in [70] provides a different randomized CPD algorithm, they first find the CPD of the small tensor which is generated by tensor random projection of the large-scale tensor, then the CPD of the large-scale tensor is obtained by back projection of the CPD of the small tensor.

Many of these randomized tensor decomposition algorithms are efficient and perform well in simulation experiments. However, to the best of our knowledge, randomized techniques have not been applied to TRD, and few studies are conducted to explore the performance of randomized tensor decomposition algorithms in real-world data. Facing the fact that TRD lacks fast and efficient algorithms for large-scale tensor, in this work, we explore the effectiveness of tensor random projection method on TRD.

5.2 Randomized tensor ring decomposition

5.2.1 Tensor random projection

Tensor random projection (TRP) has drawn people's attention in the very recent years, and several studies have been conducted based on CPD and TKD [68, 70]. Similar to matrix projection, TRP method aims to process random projection at every mode of the tensor, then a much smaller subspace tensor is obtained which reserves most of the actions of the original tensor. The TRP is simply formulated as follows:

$$\begin{aligned} \mathcal{X} &\approx \mathcal{X} \times_1 \mathbf{Q}_1\mathbf{Q}_1^T \times_2 \cdots \times_N \mathbf{Q}_N\mathbf{Q}_N^T \\ &\approx \mathcal{P} \times_1 \mathbf{Q}_1 \times_2 \cdots \times_N \mathbf{Q}_N, \end{aligned} \tag{5.1}$$

where $\times_n$ is the mode-n tensor production, see details in [2], $[\mathbf{Q}_n]$ are the orthogonal matrices, and $\mathcal{P}$ is the projected tensor. After projection, the projected tensor $\mathcal{P}$ is employed to calculate the desired low-rank approximation of the original large-scale tensor. The implementation details of the TRP method are illustrated in the next subsection.

It should be noted that for randomized algorithms, several techniques can be applied to the projection step to improve the numerical stability of the projection, thus providing higher decomposition performance. For example, adopting structured projection matrices instead of Gaussian distribution [71] and applying power iterations method to update the projected tensor in order to achieve fast decay of the spectrum of the mode-n unfolding of the projected tensor [66].

5.2.2 Model formulation

The problem of finding TRD is formulated by the following model:

$$\min_{[\mathcal{G}_n]} \|\mathcal{X} - \Psi([\mathcal{G}_n])\|_F^2, \tag{5.2}$$

where $\mathcal{X}$ is the target tensor to be decomposed, $[\mathcal{G}_n]$ are the TR factors to be considered, and $\Psi(\cdot)$ is the function which transforms the TR factors into the approximated tensor. In [35], the model is solved by various methods like TRSVD, TRALS, TRSGD, etc. However, the SVD-based and ALS-based algorithms are of high computational cost. When facing large-scale data, tremendous computing resource is needed. In addition, though TRSGD owns low complexity on every iteration and is suitable for large-scale computation, the convergence speed is rather slow and the performance cannot be guaranteed. Under this situation, we combine the TRP technique with the traditional TRD algorithms, (e.g. TRALS and TRSVD), to make it possible for fast and reliable TRD of large-scale tensor. The randomized tensor ring decomposition (rTRD) algorithms which is based on ALS (i.e., rTRALS) and SVD (i.e., rTRSVD) are illustrated by Algorithm 4.

5.3 Experimental results

5.3.1 Large-scale RGB image denoising

The projection size $[K_n]$ is the most important hyper-parameter of the TRP step of our algorithm because it determines the amounts of residual features of the original tensor to be retained and it balances the computational speed and the accuracy. This experiment is to explore the influence of the different size of the projected tensor to the performance and running time of our algorithm, and compare the performance with the TRSVD algorithm. We compare the traditional TRSVD and our rTRD to see the performance and speed differences w.r.t. different projection size and noise strength. We choose the RGB image of size 5690 × 4234 × 3 as the target tensor and add noise of 0 dB and 10 dB respectively. The RGB image is

Algorithm 4 Randomized tensor ring decomposition (rTRD)

```
input: A large-scale tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, the projection size of every mode $[K_n]$, and the TR-rank $R_1, \ldots, R_N$.
output: TR factors $[\mathcal{G}_n]$ of the large-scale tensor $\mathcal{X}$.
for $n = 1$ to $N$ do
    Create matrix $\mathbf{M} \in \mathbb{R}^{\prod_{i=1, i \neq n}^{N} I_i \times K_n}$ following the Gaussian distribution.
    $\mathbf{Y}$=$\mathbf{X}_{(n)}\mathbf{M}$ % random projection
    $[\mathbf{Q}_n, \sim] = \mathrm{QR}(\mathbf{Y})$ % economy QR decomposition
    $\mathcal{P} \leftarrow \mathcal{X} \times_n \mathbf{Q}_n^T$
end for
Obtain TR factors $[\mathcal{Z}_n]$ of $\mathcal{P}$ by TRALS or TRSVD [18] .
for $n = 1$ to $N$ do
    $\mathcal{G}_n = \mathcal{Z}_n \times_2 \mathbf{Q}_n$.
end for
```

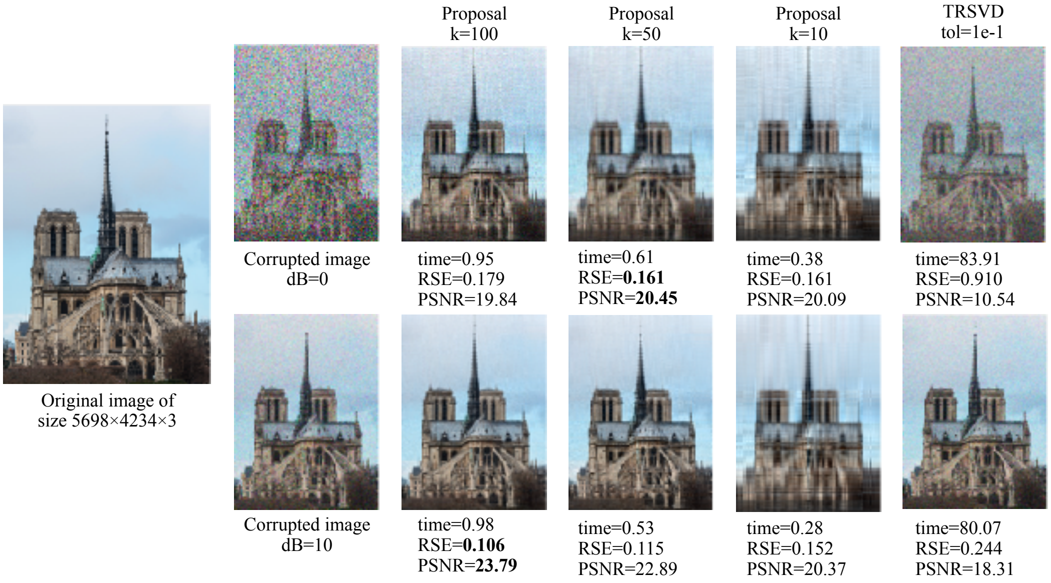

Proposal k=100
Proposal k=50
Proposal k=10
TRSVD tol=1e-1
Original image of size 5698×4234×3
Corrupted image dB=0
time=0.95 RSE=0.179 PSNR=19.84
time=0.61 RSE=**0.161** PSNR=**20.45**
time=0.38 RSE=0.161 PSNR=20.09
time=83.91 RSE=0.910 PSNR=10.54
Corrupted image dB=10
time=0.98 RSE=**0.106** PSNR=**23.79**
time=0.53 RSE=0.115 PSNR=22.89
time=0.28 RSE=0.152 PSNR=20.37
time=80.07 RSE=0.244 PSNR=18.31

FIGURE 5.1: Denoising results of a large-scale image with different projection size and noise strength.

a typical order-3 tensor of large-scale and the image modes are considered to have strong low-rankness, so the projection of mode-1 and mode-2 can largely reduce the computational cost. The projection size of mode-1 and mode-2 of the tensor data are chosen from $\{10, 50, 100\}$. The mode-3 of the tensor is small so it remains as 3. As for parameter settings, because only one iteration is needed and the TR-rank is automatically chosen in our algorithm, we set the tolerance of TRSVD as 0.01. Figure 5.1 shows the approximation error (i.e., RSE and PSNR) and computational time of the compared algorithms. From the experiment we can see, our method runs much faster than TRSVD and the performance is always higher than TRSVD. When the noise is 0 dB, the best denoising performance is obtained when the projection size is $\{50, 50, 3\}$, and the performance of the randomized algorithms remain steady and when the projection size is $\{10, 10, 3\}$. Moreover, when the noise is 10 dB, the best performance is obtained when the projection size is $\{100, 100, 3\}$, and the denoising performance falls when the projection size decreases. The results indicate that the running time increases when the projection size increases, and the image with less noise requires a higher projection size to reach a better performance.

5.3.2 Hyperspectral image denoising

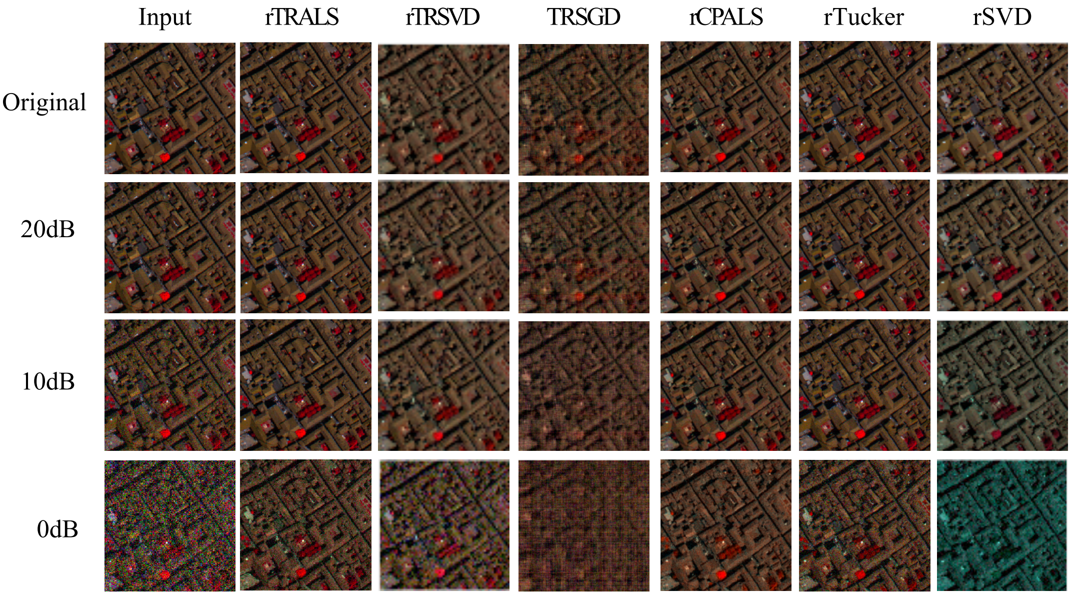


FIGURE 5.2: Visual results of HSI data reconstruction with different noise strength.

Hyperspectral image (HSI) is a typical order-3 tensor (i.e., *height* × *weights* × *bands*) with large-scale. For HSI image, the spectrum-mode (mode-3) is usually considered to have strong low-rankness, so the projection of mode-3 can largely reduce the computational cost. We compare our algorithms to the related algorithms which are suitable for large-scale data, i.e., TRSGD [35], rTucker [68] and rCPALS [70]. We also employ matrix-based algorithm rSVD [66] which is often used in HSI image processing and rSVD is implemented by mode-3 unfolding operation. The projection size of all the algorithms are set as $100 \times 100 \times 6$ for the tested $200 \times 200 \times 80$ HSI image, and the other parameters are set to get the best performance. Figure 5.2 and Table 5.1 show the visual and numerical results respectively. rTRALS outperforms the compared algorithms in the experiment.

TABLE 5.1: Numerical results of HSI data reconstruction with different noise strength.

Noise		rTRALS	rTRSVD	TRSGD	rCPALS	rTucker	rSVD
-	RSE	**0.0150**	0.149	0.249	0.100	0.0110	0.0303
	Time	60.01	**0.45**	9.45	5.38	0.50	1.84
20dB	RSE	**0.0294**	0.143	0.253	0.101	0.0388	0.0594
	Time	60.21	1.20	206.82	3.97	**0.54**	2.33
10dB	RSE	**0.0811**	0.113	0.293	0.107	0.114	0.156
	Time	59.61	1.27	210.89	3.91	**0.46**	2.08
0dB	RSE	**0.285**	0.328	0.437	0.166	0.367	0.431
	Time	59.05	0.78	206.62	3.95	**0.44**	1.87

5.4 Conclusion

In this work, based on tensor random projection method, we proposed rTRALS and rTRSVD algorithms for fast and reliable TRD. Without losing accuracy, the two algorithms perform much faster than their traditional counterparts and outperform the compared randomized algorithms in large-scale RGB image and HSI image denoising experiments. The randomized method is a promising aspect for large-scale data processing. In our future work, we will focus on further improving the performance of decomposition and applying randomized algorithms to sparse and incomplete tensors.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, in order to solve various problems that are extensively concerned in the tensor field, several methods based on tensor ring decomposition (TRD) have been proposed which can be applied to tensor decomposition, tensor completion and tensor denoising. The contributions in the thesis prove that TRD is a promising tool which can be applied to various data processing tasks. In addition, the results can be an important reference to academic and industrial fields to achieve high performance when applying tensor methods. The main conclusion of the thesis are summarized as follows:

- TR-WOPT (Chapter 2): In consideration of the performance descent of the tensor decomposition and completion based on CP and Tucker in higher-order tensor, we apply the TRD model to deal with high-order tensor problems. First, we formulate an optimization model to find the optimal TRD by partially observed high-order tensor. Then, by applying gradient-based algorithms, the optimization model is solved to obtain the low-rank TRD approximation of the incomplete tensor. Finally, the TRD is converted to an underlying tensor to predict the missing entries of the data. Furthermore, we also develop a data reformulate method named visual data tensorization (VDT), to convert the low-order tensor to higher-order tensor which is a better structure of visual data. The experimental results show that our TRWOPT obtains high performance in high-order tensor completion tasks. Moreover, the VDT method can improve the performance of our method in visual data recovery tasks.

- TRLRF and TRLNN (Chapter 3): The model selection is a big problem for TRD due to the multilinear rank. We first explore the relationship between the rank of tensor unfoldings and TR-rank, and apply nuclear norm regularization on the rank-modes of the TR factors. Then, we give theoretical proof of the relationship between the rank of mode-2 unfoldings of TR factors and the Tucker-rank of the tensor, and impose nuclear norm on the mode-2 unfoldings of the TR factors. Based on the overlapped nuclear norm and latent nuclear norm respectively, we propose two solving models named tensor ring low-rank factor (TRLRF) and tensor ring latent nuclear norm (TRLNN), which are suitable for the different data structures. The models are solved by the alternating direction method of multipliers (ADMM) solving scheme. The experiments show that our methods are robust to different rank selection of the models and the complexity of singular value decomposition (SVD) operation is much lower than the traditional tensor completion method based on nuclear norm minimization.

- TR-max-norm (Chapter 4): Traditional low-rank tensor completion methods apply nuclear norm as the low-rank regularizer. However, the nuclear norm regularizer needs to process multiple SVD operations in every iteration which is of high computational cost. In consideration of this situation, we investigate a substitute of nuclear norm named max-norm which is an efficient matrix low-rank regularizer. We extend it to the tensor field by TRD, and formulate a TR-max-norm. The proposed norm is theoretically proved to be a low-TR-rank regularizer for low-rank tensor approximation. A tensor completion model is formulated with TR-max-norm regularization and is solved by mini-batch stochastic gradient descent (SGD) algorithm with projection in every iteration. In synthetic data experiments, the model shows faster convergence and rank robustness in comparison with the traditional TR-SGD algorithm.

- RTRD (Chapter 5): Though TRD is a very promising tensor decomposition model, it lacks large-scale algorithm. By applying the randomized algorithm which is a powerful tool for fast large-scale data processing, we proposed a randomized tensor ring decomposition (RTRD) scheme for fast large-scale tensor decomposition. By random tensor projection (RTP), the large-scale tensor is firstly projected into a small-sized tensor which contains the most of the actions of the original tensor. Then, the traditional TRD algorithms such as tensor ring alternating least squares (TRALS) and tensor ring singular value decomposition (TRSVD) are processed on the small-sized tensor. Finally, the obtained TRD of the small-sized tensor is back-projected to the TRD of the large-scale tensor. From the experiments, we can see the computational time of TRD is largely reduced without the loss of accuracy.

6.2 Future work

Though we have developed several algorithms based on TRD to solve the problems in the tensor field, there are still remained problems to be explored in the future:

- Our TRWOPT shows good performance in high-order and high missing rate tensor completion tasks. However, we have to set the reshape parameters for the proposed VDT method. In our future work, we will try to develop a method which can automatically learn the best data structure for the algorithm.
- For our TRLRF and TRLNN, they provide a good solution to solve the multilinear rank selection problem. However, the algorithms still have some limitations. For instance, if the selected rank is too low or too high, the performance will also be unsatisfied. The algorithms can only ensure that if the TR-rank is selected as a relatively high value, our algorithm can find the near optimal TR-rank to fit the data. Moreover, higher TR-rank selection means higher computational complexity. In consideration of the limitation of our algorithms, in our future work, we will develop more powerful algorithms which can determine the TR rank automatically.
- For our TR-max-norm and RTRD methods, they are very efficient algorithms in processing large-scale tensor ring decomposition. As a newly developed norm, we need to find more theoretical properties of our TR-max-norm. For example, sample complexity. Moreover, for RTRD scheme, in our future work, we will give the a theoretical error bound of the tensor projection. The algorithm development of the fast TRD of incomplete and highly sparse large-scale tensor should also be taken into consideration.
- Most of the proposed TRD-based methods are only tested on the synthetic data and benchmark data. In our future work, we will apply our methods to more real-world applications.

Bibliography

[1] 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.

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

[3] M Alex O Vasilescu and Demetri Terzopoulos. Multilinear subspace analysis of image ensembles. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–93. IEEE, 2003.

[4] Thomas Franz, Antje Schultz, Sergej Sizov, and Steffen Staab. Triplerank: Ranking semantic web data by tensor decomposition. *The Semantic Web-ISWC 2009*, pages 213–228, 2009.

[5] 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.

[6] 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.

[7] Lieven De Lathauwer and Joséphine Castaing. Blind identification of underdetermined mixtures by simultaneous matrix diagonalization. *IEEE Transactions on Signal Processing*, 56(3):1096–1105, 2008.

[8] Damien Muti and Salah Bourennane. Multidimensional filtering based on a tensor approach. *Signal Processing*, 85(12):2338–2353, 2005.

[9] J Mocks. Topographic components model for event-related potentials and some biophysical considerations. *IEEE transactions on biomedical engineering*, 35(6):482–484, 1988.

[10] Amnon Shashua and Anat Levin. Linear image coding for regression and classification using the tensor-rank principle. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2001.

[11] Alexander Novikov, Dmitrii Podoprikhin, Anton Osokin, and Dmitry P Vetrov. Tensorizing neural networks. In *Advances in Neural Information Processing Systems*, pages 442–450, 2015.

[12] 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.

[13] Heishiro Kanagawa, Taiji Suzuki, Hayato Kobayashi, Nobuyuki Shimizu, and Yukihiro Tagami. Gaussian process nonparametric tensor estimator and its minimax optimality. In *International Conference on Machine Learning*, pages 1632–1641, 2016.

[14] Guoxu Zhou, Qibin Zhao, Yu Zhang, Tülay Adalı, Shengli Xie, and Andrzej Cichocki. Linked component analysis from matrices to high-order tensors: Applications to biomedical data. *Proceedings of the IEEE*, 104(2):310–331, 2016.

[15] Fengyu Cong, Qiu-Hua Lin, Li-Dan Kuang, Xiao-Feng Gong, Piia Astikainen, and Tapani Ristaniemi. Tensor decomposition of eeg signals: a brief review. *Journal of neuroscience methods*, 248:59–69, 2015.

[16] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[17] RA Harshman. Foundations of the parafac procedure: Models and conditions for an" explanatory" multi-mode factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84, 1970.

[18] Qibin Zhao, Guoxu Zhou, Shengli Xie, Liqing Zhang, and Andrzej Cichocki. Tensor ring decomposition. *arXiv preprint arXiv:1606.05535*, 2016.

[19] 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.

[20] 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.

[21] Beyza Ermiş, Evrim Acar, and A Taylan Cemgil. Link prediction in heterogeneous data via generalized coupled tensor factorization. *Data Mining and Knowledge Discovery*, 29(1):203–236, 2015.

[22] Silvia Gandy, Benjamin Recht, and Isao Yamada. Tensor completion and low-n-rank tensor recovery via convex optimization. *Inverse Problems*, 27(2):025010, 2011.

[23] Christopher J Hillar and Lek-Heng Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):45, 2013.

[24] Ryota Tomioka and Taiji Suzuki. Convex tensor decomposition via structured schatten norm regularization. In *Advances in neural information processing systems*, pages 1331–1339, 2013.

[25] Hao Cheng, Yaoliang Yu, Xinhua Zhang, Eric Xing, and Dale Schuurmans. Scalable and sound low-rank tensor learning. In *Artificial Intelligence and Statistics*, pages 1114–1123, 2016.

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

[27] Masaaki Imaizumi, Takanori Maehara, and Kohei Hayashi. On tensor train rank minimization: Statistical efficiency and scalable algorithm. In *Advances in Neural Information Processing Systems*, pages 3933–3942, 2017.

[28] Xiawei Guo, Quanming Yao, and James Tin-Yau Kwok. Efficient sparse low-rank tensor completion using the Frank-Wolfe algorithm. In *AAAI*, pages 1948–1954, 2017.

[29] Ivan V Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[30] Qibin Zhao, Masashi Sugiyama, Longhao Yuan, and Andrzej Cichocki. Learning efficient tensor representations with ring-structured networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8608–8612. IEEE, 2019.

[31] Nicolaas Klaas M Faber, Rasmus Bro, and Philip K Hopke. Recent developments in CANDECOMP/PARAFAC algorithms: a critical review. *Chemometrics and Intelligent Laboratory Systems*, 65(1):119–137, 2003.

[32] Guoxu Zhou and Andrzej Cichocki. Canonical polyadic decomposition based on a single mode blind source separation. *IEEE Signal Processing Letters*, 19(8):523–526, 2012.

[33] 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.

[34] Andrzej Cichocki, Anh-Huy Phan, Qibin Zhao, Namgil Lee, Ivan Oseledets, Masashi Sugiyama, and Danilo P Mandic. 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.

[35] Qibin Zhao, Masashi Sugiyama, Longhao Yuan, and Andrzej Cichocki. Learning efficient tensor representations with ring structure networks. *International Conference on Learning Representations (ICLR Workshop)*, 2018.

[36] 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.

[37] Linyuan Lü and Tao Zhou. Link prediction in complex networks: A survey. *Physica A: statistical mechanics and its applications*, 390(6):1150–1170, 2011.

[38] 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.

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

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

[41] Marko Filipović and Ante Jukić. Tucker factorization with missing data with application to low-n-rank tensor completion. *Multidimensional systems and signal processing*, 26(3):677–692, 2015.

[42] Lars Grasedyck, Melanie Kluge, and Sebastian Kramer. Variants of alternating least squares tensor completion in the tensor train format. *SIAM Journal on Scientific Computing*, 37(5):A2424–A2450, 2015.

[43] Wenqi Wang, Vaneet Aggarwal, and Shuchin Aeron. Efficient low rank tensor ring completion. *Rn*, 1(r1):1, 2017.

[44] 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.

[45] Stephen Wright and Jorge Nocedal. Numerical optimization. *Springer Science*, 35(67-68):7, 1999.

[46] Jorge J Moré and David J Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Transactions on Mathematical Software (TOMS)*, 20(3):286–307, 1994.

[47] 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.

[48] Jose I Latorre. Image compression and entanglement. *arXiv preprint quant-ph/0510031*, 2005.

[49] Johann A Bengua, Ho N Phien, Hoang Duong Tuan, and Minh N Do. Efficient tensor completion for color image and video recovery: Low-rank tensor train. *IEEE Transactions on Image Processing*, 26(5):2466–2479, 2017.

[50] Longhao Yuan, Jianting Cao, Xuyang Zhao, Qiang Wu, and Qibin Zhao. Higher-dimension tensor completion via low-rank tensor ring decomposition. In *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1071–1076. IEEE, 2018.

[51] Yuehaw Khoo, Jianfeng Lu, and Lexing Ying. Efficient construction of tensor ring representations from sampling. *arXiv preprint arXiv:1711.00954*, 2017.

[52] Tatsuya Yokota, Qibin Zhao, and Andrzej Cichocki. Smooth PARAFAC decomposition for tensor completion. *IEEE Transactions on Signal Processing*, 64(20):5423–5436, 2016.

[53] Tatsuya Yokota, Burak Erem, Seyhmus Guler, Simon K Warfield, and Hidekata Hontani. 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*, pages 8251–8259, 2018.

[54] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[55] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[56] Zemin Zhang, Gregory Ely, Shuchin Aeron, Ning Hao, and Misha Kilmer. 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*, pages 3842–3849, 2014.

[57] Longhao Yuan, Chao Li, Danilo Mandic, Jianting Cao, and Qibin Zhao. Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion. *arXiv preprint arXiv:1809.02288*, 2018.

[58] Wenqi Wang, Yifan Sun, Brian Eriksson, Wenlin Wang, and Vaneet Aggarwal. Wide compression: Tensor ring nets. *learning*, 14(15):13–31, 2018.

[59] Longhao Yuan, Chao Li, Jianting Cao, and Qibin Zhao. Randomized tensor ring decomposition and its application to large-scale data reconstruction. *arXiv preprint arXiv:1901.01652*, 2019.

[60] Jason D Lee, Ben Recht, Nathan Srebro, Joel Tropp, and Ruslan R Salakhutdinov. Practical large-scale optimization for max-norm regularization. In *Advances in Neural Information Processing Systems*, pages 1297–1305, 2010.

[61] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. Maximum-margin matrix factorization. In *Advances in neural information processing systems*, pages 1329–1336, 2005.

[62] Rina Foygel and Nathan Srebro. Concentration-based guarantees for low-rank matrix reconstruction. In *Proceedings of the 24th Annual Conference on Learning Theory*, pages 315–340, 2011.

[63] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[64] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[65] Andrzej Cichocki. Era of big data processing: A new approach via tensor networks and tensor decompositions. *arXiv preprint arXiv:1403.2048*, 2014.

[66] Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

[67] Per-Gunnar Martinsson, Vladimir Rokhlin, and Mark Tygert. A randomized algorithm for the decomposition of matrices. *Applied and Computational Harmonic Analysis*, 30(1):47–68, 2011.

[68] Guoxu Zhou, Andrzej Cichocki, and Shengli Xie. Decomposition of big tensors with low multilinear rank. *arXiv preprint arXiv:1412.1885*, 2014.

[69] Casey Battaglino, Grey Ballard, and Tamara G Kolda. A practical randomized CP tensor decomposition. *SIAM Journal on Matrix Analysis and Applications*, 39(2):876–901, 2018.

[70] N Benjamin Erichson, Krithika Manohar, Steven L Brunton, and J Nathan Kutz. Randomized CP tensor decomposition. *arXiv preprint arXiv:1703.09074*, 2017.

[71] Franco Woolfe, Edo Liberty, Vladimir Rokhlin, and Mark Tygert. A fast randomized algorithm for the approximation of matrices. *Applied and Computational Harmonic Analysis*, 25(3):335–366, 2008.