2024年1月　　博士論文

Mathematical Models and Algorithms for Tensor Completion

（テンソル補完の数理モデルとアルゴリズム）

指導教員
横田 達也 准教授

名古屋工業大学　大学院　工学研究科
情報工学専攻

高山 拓夢

Contents

Chapter 1 Introduction **1**
1.1 Tensor completion 1
1.2 Preliminaries for handling tensors 5
1.3 Tensor decompositions and tensor ranks 7
1.4 General observation model for tensor completion 10
1.5 Low-rank tensor completion (LRTC) 10
1.6 Tensor completion by convolution 14
1.7 Proposed method 16

Chapter 2 Tensor completion by Automatic Rank Determination with Multiplicative Gamma Process (MGP-ARD) **19**
2.1 Related works 21
2.2 Review of ARD 21
2.2.1 Modeling 21
2.2.2 Inference 23
2.2.3 Overestimation of the CP rank by ARD 24
2.3 Proposed method 26
2.3.1 Modeling 27
2.3.2 Inference 27
2.3.3 Computional complexity 32
2.4 Experiment 32
2.4.1 Experiments on artificial data (Rank is known) 33
2.4.2 Experiments with image data (Rank is unknown) 39
2.4.3 Experiments with trafic data (Rank is unknown) 40
2.4.4 Hyper-parameter sensitivities of rank estimation 42

Chapter 3 Tensor completion by Smooth Convolution Tensor Factorization (SCTF) **46**
3.1 Related works 48
3.2 Review of MDT 49
3.2.1 Delay-embedding Transform (DT) 49

3.2.2 Multiway Delay-embedding Transform (MDT) 51
3.2.3 Relationship between MDT and similarity 53
3.2.4 Fast-MDT-Tucker 55
3.3 Proposed Method 55
3.3.1 Key theory of proposed method 55
3.3.2 Smoothness constraints 58
3.3.3 Optimization formulas and algorithm 59
3.3.4 Computational complexity 63
3.3.5 Extension to non-periodic signals 64
3.4 Experiment 64
3.4.1 Completion of clipped data 64
3.4.2 Completion of random missing data 68
3.4.3 Applications to audio inpainting 72
3.4.4 Signal extrapolation 74
3.4.5 Analysis of algorithm 74

Chapter 4 Conclusions **77**

Appendices **79**

Acknowledgement **92**

Achievements **94**

Chapter 1

Introduction

1.1 Tensor completion

Several real-world data are multidimensional. For example, a recommender system is based on customer purchase history data of *customer* × *merchandise* × *time* [1], image processing is based on three-dimensional data of *height* × *width* × *channel* [2], [3], video processing is based on four-dimensional data of *frame* × *height* × *width* × *channel* [4], [5], knowledge graph is facts in the triple form of *subject entities* × *relation* × *object entities* [6], and EEG analysis is based on three-dimensional data of *sensors* × *time* × *frequency* [7], [8], [9]. Tensors are mathematical models that represent such data. Tensor is defined as a multidimensional array and is a generalization of a vector and matrix [10]. The data modeled as tensor is often corrupted by measurement errors and missing observations [11], [12], [3], [13], [14], [15]. Also, in the case of the recommender system, unrated items are considered missing values (not every customer can evaluate every item). Tensor completion is the task of filling-in the missing values of the tensor data using the

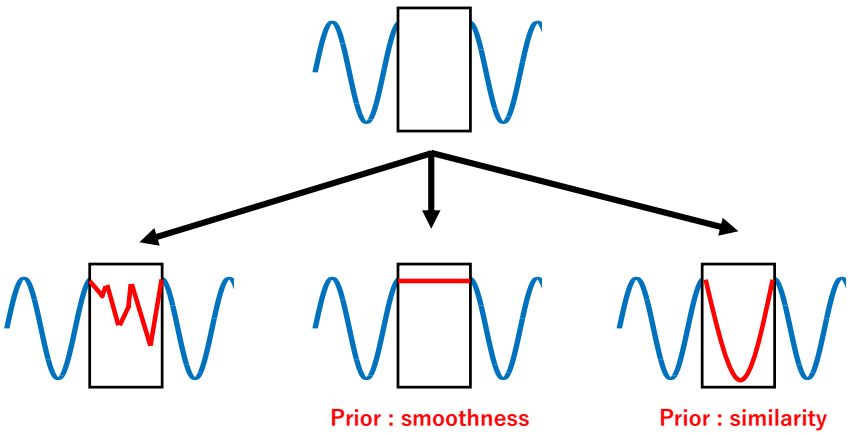


Figure 1.1 An example of the signal completion by prior. In this example, a completion based on *similarity prior* is the natural.

	Item1	Item2	Item3	Item4	Item5
Person A	4	1	1	4	1
Person B	5	2	1	5	1
Person C	1	5	5	1	5
Person D	5	1	2	5	1
Person E	1	5	4	1	5
Person F	1	?	?	1	?

	Item1	Item2	Item3	Item4	Item5
Person A	4	1	1	4	1
Person B	5	2	1	5	1
Person C	1	5	5	1	5
Person D	5	1	2	5	1
Person E	1	5	4	1	5
Person F	1	5	5	1	5

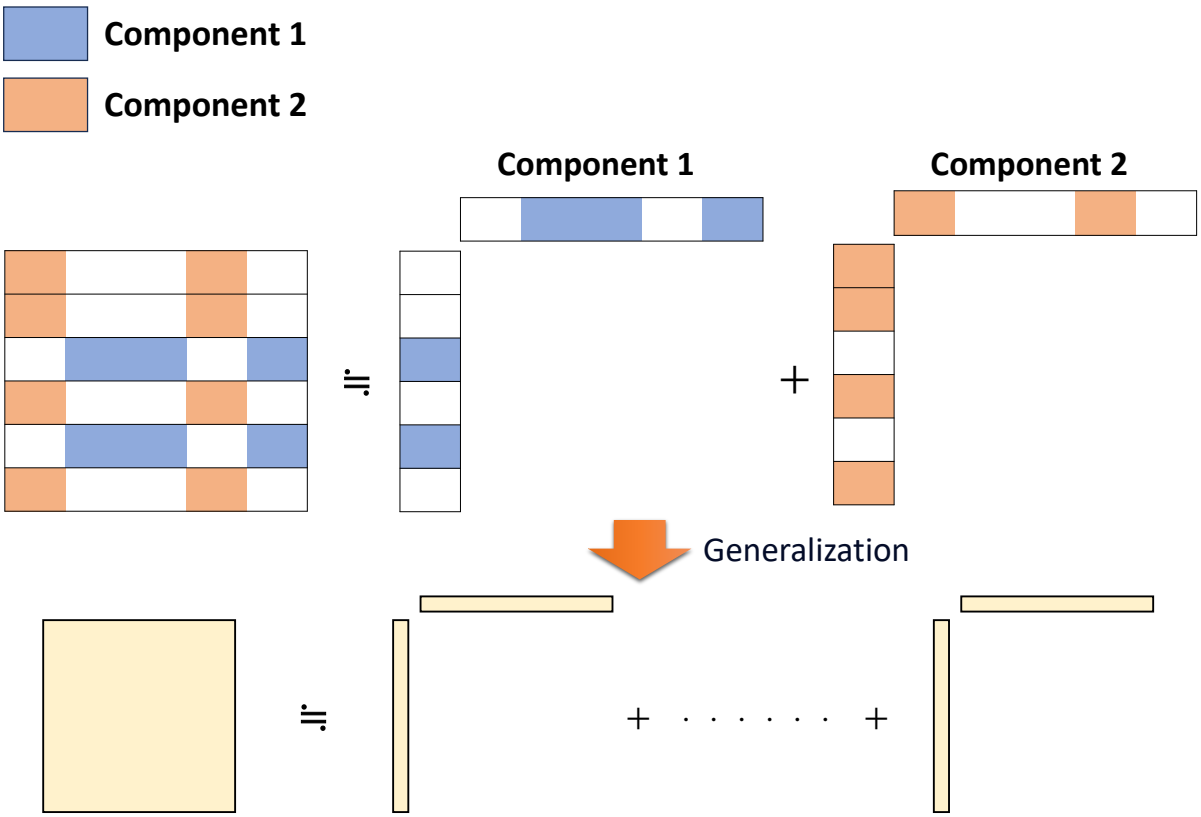


Figure 1.2 An example of collaborative filltering by *low-rank prior*. In this example, the assumption is made that the matrix is rank 2, and the two components are extracted. This information is used to predict (completion) the unrated items (missing) for new customers.

values of the reference elements [2], [16], [17], [18].

Tensor completion is an ill-posed problem that does not satisfy solution uniqueness [19] because the number of elements of the tensor to be estimated (number of parameters) is greater than the number of observations (number of equations). Thus, we consider the *prior* structure in the target tensor to narrow down the solution set. The completion value should be appropriate as per the properties of the analyzed data, and it is important to employ prior in accordance with these properties flexibly. Prior includes smoothness [20], [21], [16], nonnegativity [22], [23], [24], sparsity [25], low-rank [3], [2], etc.

We use several examples of tensor completion to explain the prior. First, we consider signal completion as an example. Figure 1.1 shows the completion of the signal (first-order tensor) with the center portion missing. As seen in Figure 1.1, there are countless candidates for the solution to complete the missing parts, and the solution varies greatly depending on the prior. In this example, since there is a periodic pattern in the observed area, it is natural to use *similarity prior*.

Recommender system, especially in collaborate filtering, often uses *low-rank prior* [26], [27]. The task of the collaborate filtering is to predict (completion) ratings for unrated items (missing values) [28], [29]. In collaborative filtering, customer reviews of the items are represented by a matrix of *customers* $\times$ *items*. Assuming that the matrix is low-rank, the idea is that items that have already been highly rated will also be highly rated by those who have yet to rate them. Also, the number of ranks corresponds to the number of latent shared features in the items. Figure 1.2 shows a collaborative filtering example. In the example in Figure 1.2, by assuming that the rank of a matrix is 2, the components of the two types of items are captured, and forecasts are executed for each type.

In the case of image completion, *smoothness prior* and *low-rank prior* are important factors. In image data, adjacent pixels tend to be of similar colors; smoothness often appears. Also, images tend to show the same pattern for straight lines (see Figure 1.3), which induces low-rankness of the image data. Figure 1.4 shows image completion performed using low-rank prior and smoothness prior. Both priors can achieve relatively high precision completion, but there are differences in the completion results. For example, smooth prior results are slightly blurred, and low-rank prior shows vertical and horizontal streaks.

Here are two things we note about the prior-based image completion. First, note that transforming the space changes the natural prior. For example, smoothness in the original space corresponds to *sparsity* in the frequency space. This is because smoothness means pixel values are concentrated in the low-frequency components in the frequency space. Figure 1.5 shows the image's intensity and Discrete Cosine Transform (DCT) coefficients. As can be seen in the figure, there is no sparsity in the image itself, but sparsity appears in the frequency space.

Also, note that smoothness and low-rankness are not valid prior at any time. Image

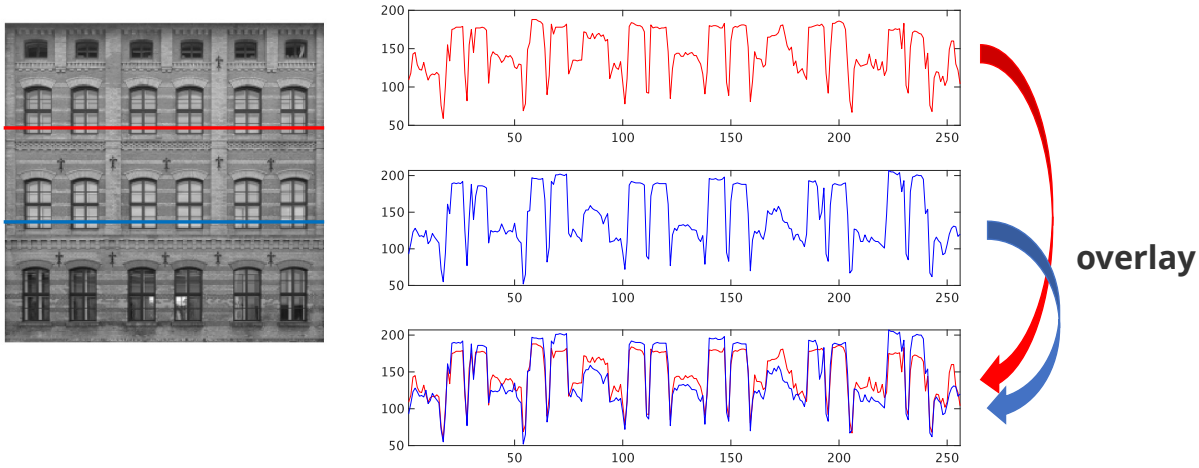
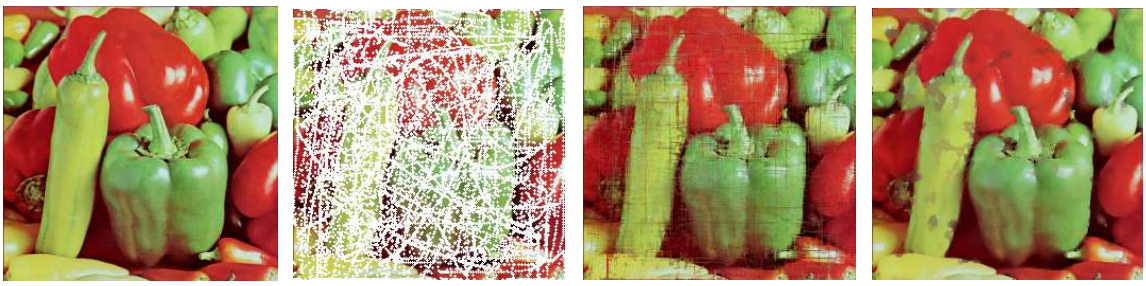


Figure 1.3 The figure shows the focus on the similarity of the lines in the image (low-rank prior). The graph on the right shows a cross-section of the image (red and blue lines), each of which shows many similar parts.



(a) True (b) Missing (c) Low-rankness (d) Total Variation

Figure 1.4 Example of RGB image completion by low-rank prior and smoothness prior. Total Variation is used for smooth prior. The experimental result is based on the existing method Linear Total Variation approximate regularized Nuclear Norm (LTVNN) [30].

smoothness tends to remove high-frequency components such as edges. On the other hand, image low-rank information has drawbacks such as difficulty in capturing diagonal features. In fact, it is known that rotational transformations can significantly alter the rank of an image [31]. Thus, prior should be employed flexibly according to the features of the image.

Our thesis focuses on *low-rank prior* in tensor completion. Unlike other priors, low-rank priors are tensor-specific features that appear only when data is represented as a tensor. In addition, the method using low-rankness in tensor completion is mainstream. Section 1.5 discusses Low-Rank Tensor Completion (LRTC).

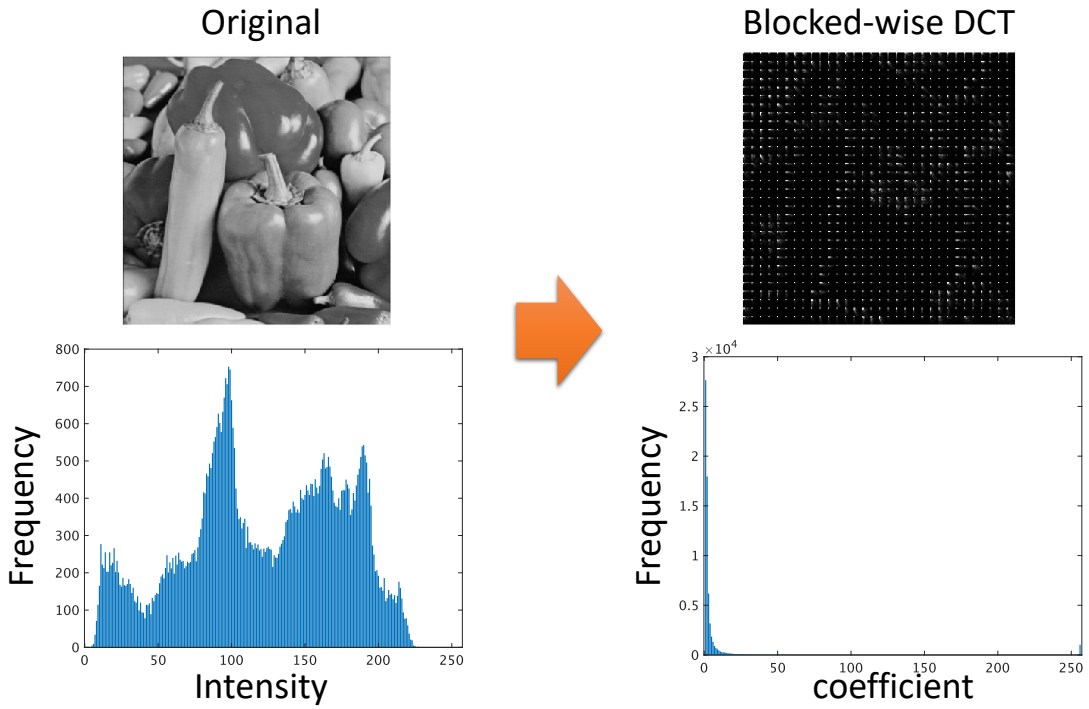


Figure 1.5 The figure shows the histogram of image intensities and the histogram of DCT coefficients when the blocked-wise DCT transform is applied. Although the image itself is not sparse, it is sparse in frequency space.

1.2 Preliminaries for handling tensors

Tensor is defined as *multidimensional array*, following the [10]. In mathematical notation, variables with subscripts such as x_{ij} or x_{ijk} are described, and the number of subscripts is referred to as the *order*. That is, a zero-order tensor is a scalar, a first-order tensor is a vector, a second-order tensor is a matrix. Figures 1.6, 1.7, 1.8 show examples of tensors. The axes of a tensor are defined as *mode*, and the operation that fixes a specific mode (mode n) and unfolds the tensor into a matrix is defined as *mode n-unfold* (See figure 1.9).

$$\begin{pmatrix} 10 & 4 & 9 \end{pmatrix}$$

Figure 1.6 vector (first order tensor)

$$\begin{pmatrix} 31 & 2 & 4 \\ 5 & 10 & 21 \end{pmatrix}$$

Figure 1.7 matrix (second order tensor)

$$\begin{pmatrix} \begin{pmatrix} 3 & 2 & \dots & 31 \\ 7 & 6 & \dots & 8 \\ 12 & 22 & \dots & 21 \\ \vdots & \vdots & \ddots & \vdots \\ 40 & 32 & \dots & 57 \end{pmatrix} & \begin{matrix} 9 \\ \vdots \\ 1 \end{matrix} \end{pmatrix} \quad \begin{pmatrix} \begin{pmatrix} 12 & 10 & \dots & 32 \end{pmatrix} & 21 \\ \vdots & \vdots \\ 43 & \end{pmatrix}$$

Figure 1.8 tensor (third or more order)

Vectors are represented as lowercase boldface $\boldsymbol{a} \in \mathbb{R}^I$, matrices as uppercase $\boldsymbol{A} \in \mathbb{R}^{I \times J}$, and higher-order tensors are written by calligraphic letters $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$. A single entry of a tensor is represented as $\mathcal{A}_{i_1,\ldots,i_N}$ (Only Chapter 3 expresses $\mathcal{A}_{i_1,\ldots,i_N}$ as $\mathcal{A}(i_1,\ldots,i_N)$.).

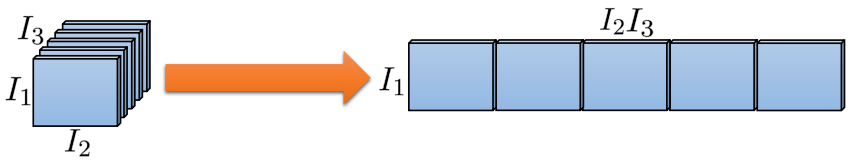


Figure 1.9 the example of mode 1-unfold of third order tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$

An $N-1$th order tensor that fixes only one mode of the tensor is denoted as $\boldsymbol{\mathcal{A}}_{i_n} \coloneqq \boldsymbol{\mathcal{A}}_{:,\ldots,i_n,\ldots,:}$

The inner product of a tensor is defined as $\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle$, where

$$\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \rangle \coloneqq \sum_{i_1=1}^{I_1} \cdots \sum_{i_N=1}^{I_N} \mathcal{A}_{i_1,\ldots,i_N} \mathcal{B}_{i_1,\ldots,i_N}. \tag{1.1}$$

The Frobenius norm is defined as $\|\boldsymbol{\mathcal{A}}\|_F \coloneqq \sqrt{\langle \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{A}} \rangle}$.

The Hadamard product of the matrices $\boldsymbol{A} \in \mathbb{R}^{I \times J}$ and $\boldsymbol{B} \in \mathbb{R}^{I \times J}$ is $\boldsymbol{A} \circledast \boldsymbol{B} \in \mathbb{R}^{I \times J}$, and the Kronecker product of the matrices $\boldsymbol{A} \in \mathbb{R}^{I \times J}$ and $\boldsymbol{B} \in \mathbb{R}^{K \times L}$ is $\boldsymbol{A} \otimes \boldsymbol{B} \in \mathbb{R}^{IK \times JL}$, and the Khatri-Rao product of the matrices $\boldsymbol{A} \in \mathbb{R}^{I \times K}$ and $\boldsymbol{B} \in \mathbb{R}^{J \times K}$ is denoted by $\boldsymbol{A} \odot \boldsymbol{B} \in \mathbb{R}^{IJ \times K}$, respectively. In particular, the Hadamard product of a set of matrices is denoted by

$$\bigcircledast_n \boldsymbol{A}^{(n)} \coloneqq \boldsymbol{A}^{(N)} \circledast \boldsymbol{A}^{(N-1)} \circledast \cdots \circledast \boldsymbol{A}^{(1)}, \tag{1.2}$$

and the Khatri-Rao product of a set of matrices in reverse order is denoted by

$$\bigodot_n \boldsymbol{A}^{(n)} \coloneqq \boldsymbol{A}^{(N)} \odot \boldsymbol{A}^{(N-1)} \odot \cdots \odot \boldsymbol{A}^{(1)}. \tag{1.3}$$

A mode-n unfold (matricization) of a tensor $\boldsymbol{\mathcal{X}}$ is denoted as $\boldsymbol{X}_{(n)} \in \mathbb{R}^{I_n \times \prod_{k \neq n} I_k}$. A mode-$n$ multiplication between a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and a matrix/vector $\boldsymbol{A} \in \mathbb{R}^{R \times I_n}$ is denoted by $\boldsymbol{\mathcal{X}} \times_n \boldsymbol{A} \in \mathbb{R}^{I_1 \times \cdots \times I_{n-1} \times R \times I_{n+1} \times \cdots \times I_N}$, where the entries are given by

$$y_{i_1,\ldots,i_{n-1},r,i_{n+1},\ldots,i_N} = \sum_{i_n=1}^{I_n} y_{i_1,\ldots,i_{n-1},i_n,i_{n+1},\ldots,i_N} a_{r,i_n}, \tag{1.4}$$

and we have $\boldsymbol{Y}_{(n)} = \boldsymbol{A}\boldsymbol{X}_{(n)}$. We consider N matrices $\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ $(n = 1, \ldots, N)$ and an N-th order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{R_1 \times \cdots \times R_N}$. The *all-mode product* is denoted as

$$\boldsymbol{\mathcal{X}} \times \{\boldsymbol{U}\} \coloneqq \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{U}^{(1)} \times_2 \cdots \times_N \boldsymbol{U}^{(N)}. \tag{1.5}$$

A outer product of N vectors $\boldsymbol{a}^{(1)} \in \mathbb{R}^{I_1}, \ldots, \boldsymbol{a}^{(N)} \in \mathbb{R}^{I_N}$ is denoted $\boldsymbol{a}^{(1)} \circ \boldsymbol{a}^{(2)} \circ \cdots \circ \boldsymbol{a}^{(N)} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, where the entries are given by

$$\{\boldsymbol{a}^{(1)} \circ \cdots \circ \boldsymbol{a}^{(N)}\}_{i_1,\ldots,i_N} \coloneqq a_{i_1}^{(1)} \cdots a_{i_N}^{(N)}.$$

1.3 Tensor decompositions and tensor ranks

The tensor increases exponentially in the number of its elements with the number of order. In addition, since many data contain noise, it is often required to extract essential features hidden in high-dimensional data. From the analogy of the matrix factorization [32], [33], which is a low-rank approximation method for matrices, we consider decomposing tensors into tensors with small degrees of freedom (latent factors). This is defined as a *tensor decomposition* [10]. There are two standard representative models of tensor decomposition: CANDECOMP/PARAFAC (CP) decomposition[1] and Tucker decomposition [10]. CP decomposition is a method of approximating an N-th order tensor of the size $I_1 \times \cdots \times I_N$ by a sum of R rank-1 tensors (the outer product of N vectors $\boldsymbol{a}_{:,r}^{(n)}$) [34], [35], [36]. Taking a N-th order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ as an example, CP decomposition of the tensor is defined as

$$\boldsymbol{\mathcal{X}} \coloneqq \sum_{r=1}^{R} \boldsymbol{a}_{:,r}^{(1)} \circ \cdots \circ \boldsymbol{a}_{:,r}^{(N)}. \tag{1.6}$$

The entries in $\boldsymbol{\mathcal{X}}$ can be computed individually as

$$\mathcal{X}_{i_1,i_2,\ldots,i_N} = \sum_{r=1}^{R} a_{i_1,r}^{(1)} a_{i_2,r}^{(2)} \cdots a_{i_N,r}^{(N)}. \tag{1.7}$$

Figure 1.10 shows the diagram for CP decomposition of the third-order tensor. A matrix of the size $I_n \times R$ with the column vector $\boldsymbol{a}_{:,r}^{(n)}$ is defined as a factor matrix and corresponds to a latent factor in CP decomposition. Latent factor matrix $\boldsymbol{A}^{(n)}$ $(n = 1, \cdots, N)$ is defined as

$$\boldsymbol{A}^{(n)} \coloneqq \left[\boldsymbol{a}_{:,1}^{(n)}, \ldots, \boldsymbol{a}_{:,r}^{(n)}, \ldots, \boldsymbol{a}_{:,R}^{(n)}\right] \in \mathbb{R}^{I_n \times R}. \tag{1.8}$$

Also, considering that tensors $\boldsymbol{A}^{(n)}$ are normalized to length 1 and their weights are introduced by $\boldsymbol{\lambda} \coloneqq [\lambda_1, \lambda_2, \ldots, \lambda_R] \in \mathbb{R}^R$, the CP decomposition is also defined as

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \tilde{\boldsymbol{a}}_{:,r}^{(1)} \circ \cdots \circ \tilde{\boldsymbol{a}}_{:,r}^{(N)}, \tag{1.9}$$

where $\|\tilde{\boldsymbol{a}}_{:,r}^{(n)}\| = 1$.

On the other hand, the Tucker decomposition is represented by factor matrices and a core tensor that describes the relationships between the factors [37], [13]. Given a tensor

[1] CP decomposition is also called "canonical polyadic" in honor of Hitchcock [34], who is credited with first thinking of the concept.

$\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ and $\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n \times R_n}$ $(n = 1, 2, \ldots, N)$, Tucker decompostion of the tensor is

$$\boldsymbol{\mathcal{X}} \coloneqq \boldsymbol{\mathcal{G}} \times \{\boldsymbol{U}\}, \tag{1.10}$$

where $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R_1 \times R_2 \times \cdots \times R_N}$ is core tensor. The entries in $\boldsymbol{\mathcal{X}}$ can be computed individually as

$$\mathcal{X}_{i_1, i_2, \ldots, i_N} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_N=1}^{R_N} \mathcal{G}_{r_1, r_2, \ldots, r_N} a^{(1)}_{i_1, r_1} a^{(2)}_{i_2, r_2} \cdots a^{(N)}_{i_N, r_N}. \tag{1.11}$$

Figure 1.11 shows the diagram for Tucker decomposition of the third-order tensor.

Since the Tucker decomposition is equal to CP decomposition when its core tensor is super-diagonal (Figure 1.12), we can consider that CP decomposition is a more constrained model than Tucker decomposition [38]. Here, the super-diagonal (In case of $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{R \times R \times \cdots \times R}$, $\boldsymbol{U}^{(n)} \in \mathbb{R}^{I_n \times R}$ $(n = 1, 2, \ldots, N)$) is defined as

$$\mathcal{G}_{i_1, i_2, \ldots, i_N} \coloneqq \begin{cases} \lambda_r & i_1 = i_2 = \cdots = i_N = r \\ 0 & \text{otherwise} \end{cases}. \tag{1.12}$$

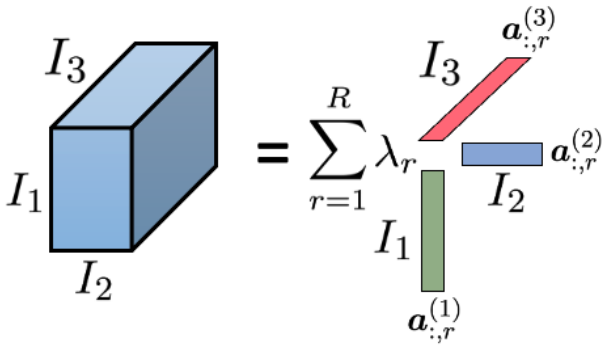


Figure 1.10 CP decomposition

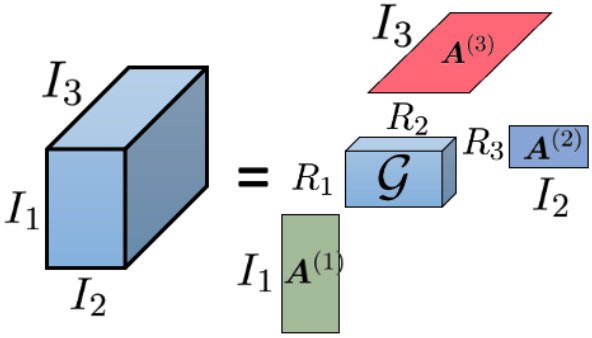


Figure 1.11 Tucker decomposition

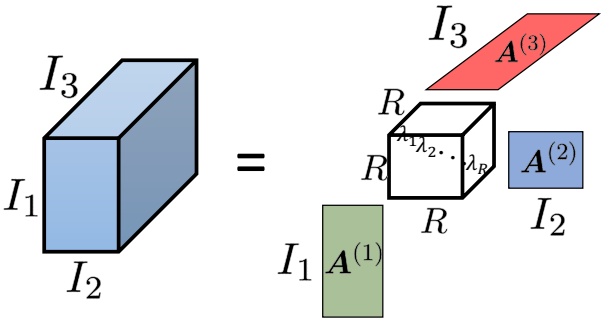


Figure 1.12 Tucker decomposition when the core tensor is super-diagonal

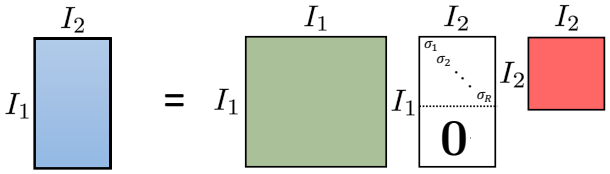


Figure 1.13 Singular value decomposition

CP decomposition is a natural extension of Singular Value Decomposition (SVD) [39], and we can interpret the core tensor as representing something like singular values in CP decomposition. Figure 1.13 shows the SVD.

The concept of rank exists in tensors as well as in matrices. In general, tensor rank refers to *CP rank*. CP rank is the minimum value of R in a decomposition that reconstructs the original tensor without error (referred to as exact decomposition) [34], [40]. As an example of CP rank, we present the third-order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$. Suppose that the CP decomposition of $\boldsymbol{\mathcal{X}}$ is expressed as

$$\boldsymbol{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \tilde{\boldsymbol{a}}_{:,r}^{(1)} \circ \tilde{\boldsymbol{a}}_{:,r}^{(2)} \circ \tilde{\boldsymbol{a}}_{:,r}^{(3)}, \tag{1.13}$$

then the minimum possible R is the CP rank. Here, we describe some peculiar properties of CP rank. First, CP rank is that there is no straightforward algorithm for determining the rank of a specific given tensor. In fact, it is NP-hard to find the CP rank of a given tensor [41]. Second, the uniqueness of the tensor decomposition is conditioned by CP rank R. When the CP decomposition of a tensor $\boldsymbol{\mathcal{X}}$ is expressed in Equation (1.6), the tensor $\boldsymbol{\mathcal{X}}$ satisfies uniqueness if the following conditions are satisfied.

$$\sum_{n=1}^{N} k_{\boldsymbol{A}^{(n)}} \geq 2R + (N-1), \tag{1.14}$$

where $k_{\boldsymbol{A}^{(n)}}$ is named *k-rank* and is defined as the maximum value k such that any k columns of $\boldsymbol{A}^{(n)}$ are linearly independent [40]. Finally, it is possible that the best rank-k approximation may not even exist. [42], [43] explain it with the following example. We consider the rank-3 tensor

$$\boldsymbol{\mathcal{X}} = \boldsymbol{a}_{:,1}^{(1)} \circ \boldsymbol{a}_{:,1}^{(2)} \circ \boldsymbol{a}_{:,2}^{(3)} + \boldsymbol{a}_{:,1}^{(1)} \circ \boldsymbol{a}_{:,2}^{(2)} \circ \boldsymbol{a}_{:,1}^{(3)} + \boldsymbol{a}_{:,2}^{(1)} \circ \boldsymbol{a}_{:,1}^{(2)} \circ \boldsymbol{a}_{:,1}^{(3)}, \tag{1.15}$$

where $\boldsymbol{A}^{(1)} \in \mathbb{R}^{I_1 \times 2}$, $\boldsymbol{A}^{(2)} \in \mathbb{R}^{I_2 \times 2}$, and $\boldsymbol{A}^{(3)} \in \mathbb{R}^{I_3 \times 2}$, and each has linearly independent columns. This tensor can be approximated arbitrarily closely by a rank-2 tensor

$$\boldsymbol{\mathcal{Y}} = \alpha \left(\boldsymbol{a}_{:,1}^{(1)} + \frac{1}{\alpha} \boldsymbol{a}_{:,2}^{(1)} \right) \circ \left(\boldsymbol{a}_{:,1}^{(2)} + \frac{1}{\alpha} \boldsymbol{a}_{:,2}^{(2)} \right) \circ \left(\boldsymbol{a}_{:,1}^{(3)} + \frac{1}{\alpha} \boldsymbol{a}_{:,2}^{(3)} \right) - \alpha \boldsymbol{a}_{:,1}^{(1)} \circ \boldsymbol{a}_{:,1}^{(2)} \circ \boldsymbol{a}_{:,1}^{(3)}. \tag{1.16}$$

In fact, from Equation

$$\|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Y}}\| = \frac{1}{\alpha} \|\boldsymbol{a}_{:,2}^{(1)} \circ \boldsymbol{a}_{:,2}^{(2)} \circ \boldsymbol{a}_{:,1}^{(3)} + \boldsymbol{a}_{:,2}^{(1)} \circ \boldsymbol{a}_{:,1}^{(2)} \circ \boldsymbol{a}_{:,2}^{(3)} + \boldsymbol{a}_{:,1}^{(1)} \circ \boldsymbol{a}_{:,2}^{(2)} \circ \boldsymbol{a}_{:,2}^{(3)} + \frac{1}{\alpha} \boldsymbol{a}_{:,2}^{(1)} \circ \boldsymbol{a}_{:,2}^{(2)} \circ \boldsymbol{a}_{:,2}^{(3)}\|, \tag{1.17}$$

the distance between two tensors can be described as $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{Y}}$ can be arbitrarily close at $\alpha \to \infty$. This example shows that the rank-2 tensor converges to a rank other than rank 2, indicating the difficulty of the best rank-k approximation.

On the other hand, there is also the concept of *n-rank* for tensor rank [44], [45]. *N-rank* is also known as *Tucker rank*. Tucker rank of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ is the ranks of matrices $\boldsymbol{X}_{(n)}$. That is, if the rank of the n-unfold is R_n, N matrices are generated for n-unfold, so Tucker rank are obtained in tuple $(R_1, \ldots, R_N)$. We can easily find an exact Tucker decomposition of rank $(R_1, \ldots, R_N)$ [10]. In general, the rank of a tensor refers to CP rank, but Tucker rank is also often considered when considering low-rank approximations of the tensor. Here, we define $f_{\text{Rank}}(\boldsymbol{\mathcal{X}})$ as the function that returns the rank of a tensor $\boldsymbol{\mathcal{X}}$. However, whether $f_{\text{Rank}}(\boldsymbol{\mathcal{X}})$ refers to the CP rank or the Tucker rank depends on the problem (we will specify the details in each case).

1.4 General observation model for tensor completion

The tensor completion discussed in Section 1.1 is again defined mathematically. Equation (1.18) shows the commonly used general observation model of the tensor data for tensor completion.

$$\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{O}} \circledast (\boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{E}}). \tag{1.18}$$

$\boldsymbol{\mathcal{X}}$ represents the true tensor, $\boldsymbol{\mathcal{O}}$ is the mask, $\boldsymbol{\mathcal{E}}$ is the noise, and $\boldsymbol{\mathcal{Y}}$ is the observed tensor. The mask $\boldsymbol{\mathcal{O}}$ is defined as

$$\mathcal{O}_{i_1,i_2,\ldots,i_N} = \begin{cases} 1 & \mathcal{Y}_{i_1,i_2,\ldots,i_N} \quad \text{is observed} \\ 0 & \text{otherwise} \end{cases}. \tag{1.19}$$

The unknown variable in Equation (1.18) are $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{E}}$. $\boldsymbol{\mathcal{X}}$ has a choice of prior structure, and $\boldsymbol{\mathcal{E}}$ has a choice such as Gaussian, Laplacian, Poisson, and so on. We consider the inverse problem of estimating an unknown true tensor $\boldsymbol{\mathcal{X}}$ from an observed tensor $\boldsymbol{\mathcal{Y}}$. Here, we assume Gaussian in the noise tensor $\boldsymbol{\mathcal{E}}$. Therefore, an important discussion in our study is about the mathematical model that represents the prior of the true tensor $\boldsymbol{\mathcal{X}}$. As described in Section 1.1, we focus on *low-rank prior*.

1.5 Low-rank tensor completion (LRTC)

Low-rank tensor completion (LRTC) is the most major tensor completion technique. LRTC has two approaches: *rank minimization* and *tensor decomposition*. Rank minimization can be defined as

$$\begin{aligned} \min_{\boldsymbol{\mathcal{X}}} \quad & f_{\text{rank}}(\boldsymbol{\mathcal{X}}) \\ \text{s.t.} \quad & \|\boldsymbol{\mathcal{O}} \circledast (\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}})\|_F < \delta, \end{aligned} \tag{1.20}$$

where δ is a value dependent on noise, and the tensor decomposition can be defined as

$$\min_{\boldsymbol{\mathcal{A}}^{(1)},\boldsymbol{\mathcal{A}}^{(2)},\ldots,\boldsymbol{\mathcal{A}}^{(N)}} \|\boldsymbol{\mathcal{O}} \circledast (\boldsymbol{\mathcal{Y}} - f(\boldsymbol{\mathcal{A}}^{(1)}, \boldsymbol{\mathcal{A}}^{(2)}, \ldots, \boldsymbol{\mathcal{A}}^{(N)}))\|_F^2, \tag{1.21}$$

where $f(\boldsymbol{\mathcal{A}}^{(1)}, \boldsymbol{\mathcal{A}}^{(2)}, \ldots, \boldsymbol{\mathcal{A}}^{(N)})$ is an arbitrary tensor decomposition.

First, we will discuss the rank minimization. Therefore, we consider rank minimization of the matrix $\boldsymbol{X} \in \mathbb{R}^{n \times n}$ before the tensor. Since rank minimization is NP-hard, we use the fact that nuclear norm minimization is a convex relaxation of the rank minimization problem [46]. The nuclear norm of the matrix $\boldsymbol{X}$ is defined

$$\|\boldsymbol{X}\|_* = \sum_{i=1}^{r} \sigma_i(\boldsymbol{X}), \tag{1.22}$$

where $\sigma_i(\boldsymbol{X})$ is the i-th singular value of the matrix $\boldsymbol{X}$. Subsequently, [47] introduced nuclear norm minimization was introduced for low-rank matrix completion. This work also theoretically guaranteed that solving the completion problem

$$\begin{aligned} \min_{\boldsymbol{X}} \quad & \|\boldsymbol{X}\|_* \\ \text{s.t.} \quad & \boldsymbol{O} \circledast \boldsymbol{Y} = \boldsymbol{O} \circledast \boldsymbol{X}, \end{aligned} \tag{1.23}$$

for a matrix $\boldsymbol{X}$ of true rank r is fully completable when the number of observations m satisfies

$$m \geq C n^{1.2} r \log n, \tag{1.24}$$

where C is a positive constant. Note that Equation (1.23), unlike Equation (1.20), describes the exact completion. A new method of LRTC based on nuclear norm minimization is proposed by [48]. In this study, the LRTC is defined as

$$\begin{aligned} \min_{\boldsymbol{\mathcal{X}}, \tilde{\boldsymbol{\mathcal{X}}}} \quad & \frac{1}{2}\|\boldsymbol{\mathcal{X}} - \tilde{\boldsymbol{\mathcal{X}}}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{O}} \circledast \tilde{\boldsymbol{\mathcal{X}}} \\ & \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{X}_{(i)}\|_* \leq c, \end{aligned} \tag{1.25}$$

which describes low-rankness by considering the nuclear norm of all modes of the tensor. This method is for applications to tensor completion and claims the superiority of methods that use the global information of low-rankness over methods that use neighborhood information, such as Markov Random Field [49] and anisotropic diffusion [50]. As a parallel study, [3] achieves LRTC in a form similar to Equation (1.25). However, [3] uses the extended lagrangian method and Alternating Direction Method of Multipliers

(ADMM) [51] to provide theoretical guarantees regarding convergence. Also, this work defines the observation model of the tensor as an inverse problem

$$\boldsymbol{y} = \boldsymbol{\mathcal{A}}(\boldsymbol{\mathcal{X}}) + \boldsymbol{\epsilon}, \tag{1.26}$$

where $\boldsymbol{\mathcal{A}}$ is a linear map $\boldsymbol{\mathcal{A}} : \mathbb{R}^{I_1 \times \cdots \times I_N} \to \mathbb{R}^p$ with $p \leq \prod_{n=1}^{N} I_n$ and given $\boldsymbol{y} \in \mathbb{R}^p$, find the tensor $\boldsymbol{\mathcal{X}}$ that minimizes a function of the n-rank of the tensor. Our observation model Equation (1.18) of the tensor is based on Equation (1.26), but Equation (1.18) changes $\boldsymbol{\mathcal{A}}$ in Equation (1.26) to the mask $\boldsymbol{\mathcal{O}}$. [52] imposes a smoothness constraint on the nuclear norm minimization for LRTC. The optimization Equation is defined as

$$\begin{aligned} \min_{\boldsymbol{\mathcal{X}}} \quad & \alpha f_{\mathrm{TV}}(\boldsymbol{\mathcal{X}}) + \beta \frac{1}{N} \sum_{i=1}^{N} \|\boldsymbol{X}_{(i)}\|_* \\ \text{s.t.} \quad & v_{\min} \leq \boldsymbol{\mathcal{X}} \leq v_{\max} \\ & \|\boldsymbol{\mathcal{O}} \circledast (\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}})\|_F < \delta, \end{aligned} \tag{1.27}$$

where f_{TV} is Total Variation (TV) regularization operator. The introduction of TV is based on the assumption that much real-world data, such as natural images/videos, spectral signals, and biomedical data, are smooth. There are several studies of TV+LR tensor completion [20], [53]. For example, [53] has set up weighted nuclear norm minimization problems. Furthermore, various other nuclear norm minimization methods exist, including the introduction of latent variables [54], setting convex relaxation stricter than the nuclear norm [55], and Robust-PCA based [56], [57].

Another LRTC is a tensor decomposition approach, such as CP decomposition or Tucker decomposition [13], [35], [58]. CP Weighted OPTimization (CP-WOPT) [11] proposed a weighted CP decomposition for tensor completion as

$$\min_{\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \ldots, \boldsymbol{A}^{(N)}} \quad \|\boldsymbol{\mathcal{O}} \circledast (\boldsymbol{\mathcal{Y}} - \sum_{r=1}^{R} \boldsymbol{a}_{:,r}^{(1)} \circ \boldsymbol{a}_{:,r}^{(2)} \circ \cdots \circ \boldsymbol{a}_{:,r}^{(N)})\|_F^2. \tag{1.28}$$

Here, the weight $\boldsymbol{\mathcal{O}}$ represents the mask. [59] defines

$$\min_{\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, \ldots, \boldsymbol{A}^{(N)}} \quad \|\boldsymbol{\mathcal{O}} \circledast (\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}} \times_1 \boldsymbol{A}^{(1)} \times_2 \boldsymbol{A}^{(2)} \times_3 \cdots \times_N \boldsymbol{A}^{(N)})\|_F^2, \tag{1.29}$$

as the Tucker decomposition model for CP-WOPT. Since it is difficult for both methods to solve optimization problems (1.28) and (1.29) directly, the majorization-minimization (MM) algorithm [60], [61] is used to solve them. Many of these approaches are nonconvex optimization and heuristic methods with no theoretical guarantees. On the other hand, they are superior to nuclear norm minimization approaches in that they have better completion performance and are low computational cost (avoiding the computation of SVD). Tensor completion by parallel Matrix factorization (TMac) proposed matricization

(n-unfold) of the tensor [62] and then consider a low-rank matrix factorization model defined as

$$\begin{aligned} \min_{\boldsymbol{\mathcal{X}},\boldsymbol{A}^{(1)},\ldots,\boldsymbol{A}^{(N)},\boldsymbol{B}^{(1)},\ldots,\boldsymbol{B}^{(N)}} \quad & \sum_{n=1}^{N} \|\boldsymbol{X}_{(n)} - \boldsymbol{A}^{(n)}\boldsymbol{B}^{(n)}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{X}} \\ & \boldsymbol{A}^{(n)} \in \mathbb{R}^{I_n \times R_n}, \boldsymbol{B}^{(n)} \in \mathbb{R}^{R_n \times \prod_{k \neq n} I_k}. \end{aligned} \tag{1.30}$$

TMac achieved a highly accurate completion and a highly efficient computation compared to the nuclear norm minimization approach. This research also found that an approach that greedily increases the rank R_n stepwise from 1 achieves particularly accurate completion. Furthermore, [16] introduced smoothness into each factor of the CP decomposition as Equation

$$\begin{aligned} \min_{\boldsymbol{\mathcal{X}},\boldsymbol{\mathcal{Z}},\boldsymbol{\mathcal{G}},\boldsymbol{U}^{(1)},\ldots,\boldsymbol{U}^{(N)}} \quad & \|\boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{Z}}\|_F^2 + \sum_{r=1}^{R} \frac{g_r^2}{2} \sum_{n=1}^{N} \rho^{(n)} \|\boldsymbol{L}^{(n)} \boldsymbol{u}_r^{(n)}\|_p^p \\ \text{s.t.} \quad & \boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{X}} \\ & (1 - \boldsymbol{\mathcal{O}}) \circledast \boldsymbol{\mathcal{Z}} = (1 - \boldsymbol{\mathcal{O}}) \circledast \boldsymbol{\mathcal{X}} \\ & \boldsymbol{\mathcal{Z}} = \sum_{r=1}^{R} g_r \boldsymbol{u}_r^{(1)} \circ \cdots \circ \boldsymbol{u}_r^{(n)}, \end{aligned} \tag{1.31}$$

where $\boldsymbol{L}^{(n)}$ is difference transformation matrix, which the second term in the equation corresponds to the smooth constraint. Similar to the method in [62], this method is a stepwise increase in rank, and although computationally expensive, it has very high completion performance. Tensor decomposition approaches are often incorporated simultaneously with other prior structures. For example, non-negative tensor factorization (NTF) [63], [64] has been applied to various fields, such as sparse coding of images [65], traffic analysis [66], and EEG analysis [7]. Other prior structures include studies incorporating graph structures and sparse regularization to achieve a super-resolution in multispectral images [67]. Also, some methods apply tensor decomposition to stochastic modeling, and one of our proposed methods is a kind of stochastic tensor decomposition [68], [69].

The two approaches of LRTC are summarized here in Table 1.1. The approach with rank minimization is a convex optimization, and there are several theoretical guarantees regarding completion accuracy. Furthermore, the ranks can be roughly estimated from the nuclear norm values. However, it lacks scalability, and many methods have poorer completion accuracy than the tensor decomposition approach. On the other hand, the tensor decomposition approach is scalable and flexible. In addition, the completion accuracy tends to be higher than that of the rank minimization approach. However, it requires hyperparameter adjustment for rank setting.

Table 1.1 This table summarizes the advantages and disadvantages of the two approaches to LRTC. Note that they are only trends.

	Rank minimization	Tensor decomposition
Theoretical guarantees on completion	✓	
Automatic rank determination	✓	
Scalability		✓
Performance of the completion		✓

In our study, the tensor decomposition model is employed as the LRTC. The reasons are described as follows:

- The tensor decomposition represents the structure of the data appropriate to the problem and suitable for data analysis [59], [62], [16]. Hence, high completion accuracy is often achieved. Note that this advantage is shared in matrix completion by using matrix factorization methods [70], [71].
- In the case of rank minimization (nuclear norm minimization), since the computation of the SVD is unavoidable, the model is often not scalable, and the computational cost of the SVD is high. In the case of tensor decomposition, there are other alternatives, such as computing with the gradient method [59].

On the other hand, the tensor decomposition approach has the problem that rank determination is difficult. Rank is basically determined heuristically by the algorithm, and in particular, overly large estimates of ranks lead to increased computation time and worse completion performance due to noise tolerance. One of the proposed methods is a framework that can automatically and more accurately estimate rank while simultaneously achieving efficient and accurate tensor completion.

1.6 Tensor completion by convolution

We have focused on tensor decomposition as a method for LRTC. Here, we explain tensor decomposition by convolution, a new tensor completion framework in recent years. In particular, this thesis focuses on the t-SVD model [72]. t-SVD model is a method for third-order tensors and was initially conceived as a tensor decomposition method for a video whose time direction corresponds to the third mode [73]. Therefore, a new operation, t-product, is defined in t-SVD, focusing on the third mode. T-product of tensors $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{n_1 \times r \times n_3}$ and $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{r \times n_2 \times n_3}$ is defined as

$$\boldsymbol{\mathcal{A}} *_3 \boldsymbol{\mathcal{B}} := \text{permute}(\text{fold}(\text{bcric}(\boldsymbol{\mathcal{A}})\boldsymbol{B}_{(2)}^{\mathrm{T}}), [1, 3, 2]) \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \tag{1.32}$$

where bcirc$(\cdot)$ of $\boldsymbol{\mathcal{A}}$ is

$$\text{bcric}(\boldsymbol{\mathcal{A}}) \coloneqq \begin{pmatrix} \boldsymbol{A}_{:,:,1} & \boldsymbol{A}_{:,:,n_3} & \boldsymbol{A}_{:,:,n_3-1} & \cdots & \boldsymbol{A}_{:,:,2} \\ \boldsymbol{A}_{:,:,2} & \boldsymbol{A}_{:,:,1} & \boldsymbol{A}_{:,:,n_3} & \cdots & \boldsymbol{A}_{:,:,3} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \boldsymbol{A}_{:,:,n_3} & \boldsymbol{A}_{:,:,n_3-1} & \ddots & \boldsymbol{A}_{:,:,2} & \boldsymbol{A}_{:,:,1} \end{pmatrix} \in \mathbb{R}^{n_1 n_3 \times r n_3}, \tag{1.33}$$

fold$(\cdot)$ is the inverse of unfold and is an operation that folds the tensor to a higher order, and permute$(\cdot)$ is an operation to reorder mode into the permutation of $[\cdot]$. This operation is equivalent to performing a convolution operation only in the third mode and a matrix product operation in the other modes. Based on this operation, t-SVD is formulated as

$$\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{U}} *_3 \boldsymbol{\mathcal{D}} *_3 \boldsymbol{\mathcal{V}}, \tag{1.34}$$

where $\boldsymbol{\mathcal{U}}$ and $\boldsymbol{\mathcal{V}}$ are orthogonal[2]. Figure 1.14 shows the t-SVD concept. The tensor tubal rank of $\boldsymbol{\mathcal{X}}$ is defined to be the number of non-zero singular tubes of $\boldsymbol{\mathcal{D}}$. The minimization of the tubal rank is a convex relaxation of the minimization of the tensor nuclear norm. Tensor nuclear norm is defined as

$$\|\boldsymbol{\mathcal{X}}\|_{\text{TNN}} \coloneqq \left\| \begin{pmatrix} \hat{\boldsymbol{X}}_{:,:,1} & & & \\ & \hat{\boldsymbol{X}}_{:,:,2} & & \\ & & \ddots & \\ & & & \hat{\boldsymbol{X}}_{:,:,n_3} \end{pmatrix} \right\|_* \in \mathbb{R}^{n_1 n_3 \times n_2 n_3}, \tag{1.35}$$

where $\hat{\boldsymbol{X}}_{:,:,1}, \hat{\boldsymbol{X}}_{:,:,2}, \ldots, \hat{\boldsymbol{X}}_{:,:,n_3}$ are the Fourier transform of $\boldsymbol{X}_{:,:,1}, \boldsymbol{X}_{:,:,2}, \ldots, \boldsymbol{X}_{:,:,n_3}$ along the third mode. Based on this idea, [74] reported on video completion with tensor completion using nuclear norm minimization, defined as

$$\begin{aligned} \min_{\boldsymbol{\mathcal{X}}} \quad & \|\boldsymbol{\mathcal{X}}\|_{\text{TNN}} \\ \text{s.t.} \quad & \boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{Y}}. \end{aligned} \tag{1.36}$$

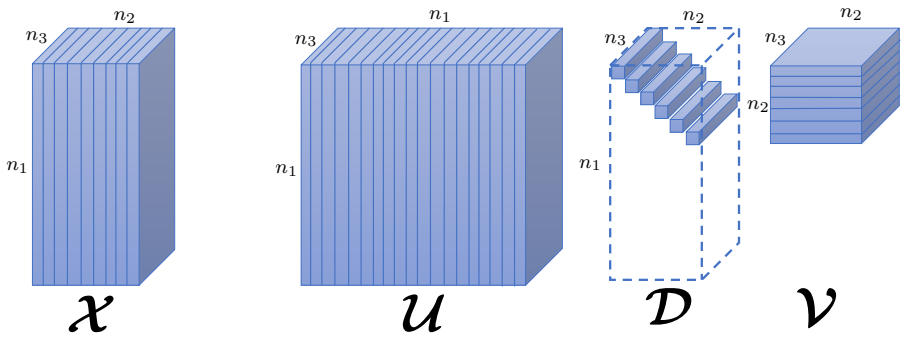


Figure 1.14 The t-SVD of an $n_1 \times n_2 \times n_3$ tensor

[2] See [72] for the definition of *orthogonal* in t-SVD.

Since this framework is based on matrix completion by nuclear norm minimization (see Equation 1.23), [72] shows theoretical guarantees regarding completion performance. There are also various extension techniques, such as studies that reduce the cost of nuclear norm minimization [75], [76], studies that introduce higher compression performance through framelet representation [77], and studies that improve completion performance by introducing arbitrary linear transformations [78]. However, t-SVD approaches are only for third-order tensor methods, and their completion capability is inferior to that of the common tensor decomposition model.

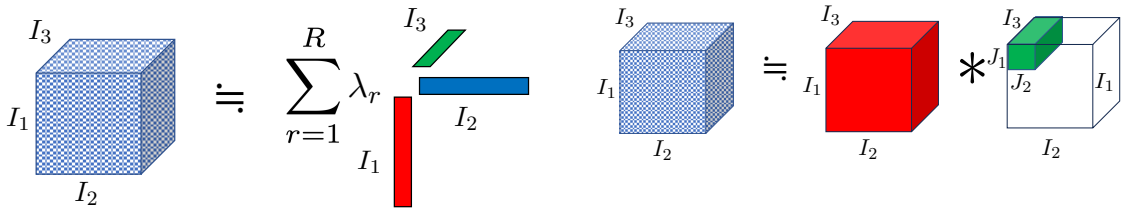
One of the proposed methods is also a convolutional tensor decomposition framework. However, the proposed method is much more accurate than the t-SVD model. This method has no restriction on the number of tensor orders and performs convolution in all modes. Also, our thesis shows that the proposed method is strongly related to LRTC on *delay-embedded space* [79], which has recently achieved highly accurate completion. In addition, by imposing smooth constraints on the tensor factors, the proposed method achieves more accurate completion.

1.7 Proposed method

Here, we propose two methods for accurate and efficient tensor completion: *Automatic Rank Determination with Multiplicative Gamma Process (MGP-ARD)* and *Smooth Convolution Tensor Factorization (SCTF).* Figure 1.15 shows an overview of the two proposed methods.

MGP-ARD is kind of LRTC. This method aims to achieve tensor completion and rank determination simultaneously. This can be achieved using Bayesian CP decomposition with Multiplicative Gamma Process (MGP) as the prior distribution. MGP is a distribution that decays the components. Using MGP, the proposed method avoids duplication of components and enables highly accurate rank estimation in Bayesian tensor modeling. In addition, MGP helps to reduce noise sensitivity and estimation time, which achieves highly accurate and highly efficient completion. Numerical experiments using artificial data and image data demonstrate the effectiveness of this method. Details of the method are described in Chapter 2.

On the other hand, SCTF is a kind of completion method by convolutional tensor decomposition. The concept of SCTF is based on a delay-embedded space. Recently, Multiway Delay-embedding Transform (MDT), which considers a low-dimensional space in a delay-embedded space with high expressive capability, has attracted attention as a tensor completion method. Although MDT has a high complementary performance, its computational cost is considerably high. SCTF is small in computational complexity because of its concise model of rank-1 decomposition in the delay-embedded space and



(a) MGP-ARD (b) SCTF

Figure 1.15 Concept of tensor decomposition model of the two proposed methods.

because it does not directly perform optimization in the delay-embedded space. In addition, a smooth constraint term is assigned to the factor tensors as a prior data structure in the optimization to further improve the completion accuracy. In our experiments, we completed clipped and randomly missing image data and confirmed that the proposed method achieved high completion accuracy without high computational cost. Details of the method are described in Chapter 3.

At first glance, both MGP-ARD and SCTF appear to be different frameworks, LRTC, and convolutional tensor decomposition. However, when considered in terms of the concept of *low-rankness*, we can think of MGP-ARD as a low-rank model in the original space (CP decomposition), and SCTF as a low-rank model in the delay-embedded space (rank 1 decomposition), and can see two methods in a unified way. Figure 1.16 summarizes the proposed method from the unifying perspective of low-rankness. Note that low-rankness on delay-embedded space is a *similarity prior* (see Section 3.2.3), so it is not the same as

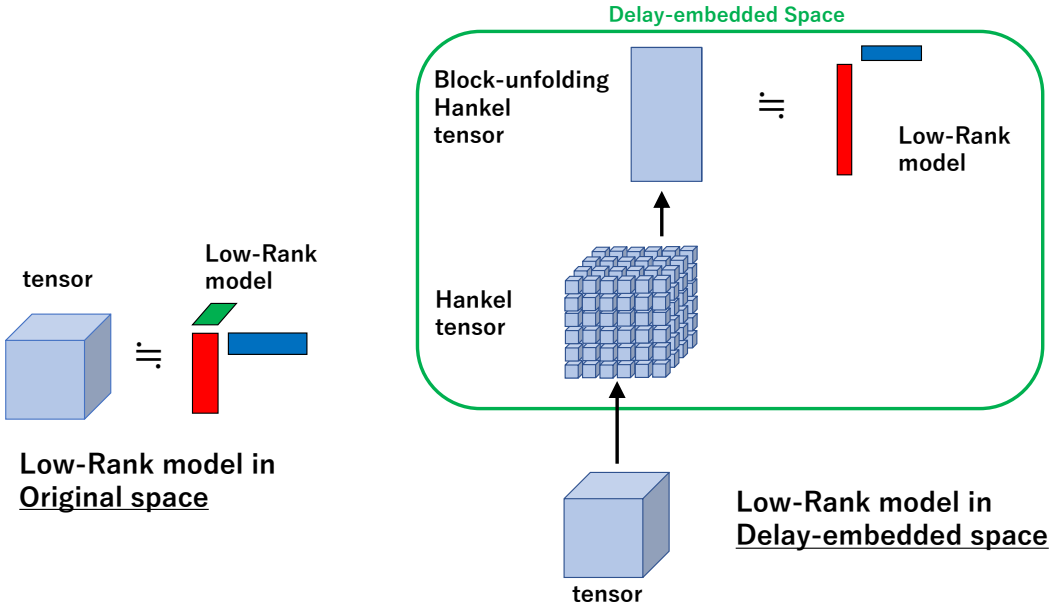


Figure 1.16 Concept of the two our proposed models through the unifying perspective of low-rankedness.

a low-rank prior after all. The main goal of our thesis is to achieve highly accurate and efficient tensor completion by focusing on the prior of an unknown true tensor.

Here, we describe the structure of this thesis. The remainder of this thesis is organized as follows: *Automatic Rank Determination with Multiplicative Gamma Process (MGP-ARD)* in Chapter 2, *Smooth Convolution Tensor Factorization (SCTF)* in Chapter 3, and their summary in Chapter 4.

Chapter 2

Tensor completion by Automatic Rank Determination with Multiplicative Gamma Process (MGP-ARD)

There are many approaches for tensor completion using CP decomposition. A typical way is minimizing the loss function such as Euclidean distance between a low-rank CP decomposition model and the observed tensor. CP Weighted Optimization (CPWOPT) is a method that formulates CP decomposition with missing data as an weighted least squares method [11] and has been applied to extract traffic patterns in Intelligent Transportation Systems (ITS) where missing data is a common problem [80]. However, tensor completion based on the least-squares method without regularization may not uniquely determine the solution and tends to be sensitive to noise when the rank is estimated to be larger than true rank. In this sense, this is considered as overfitting in tensor decompositions, which causes a severe deterioration of estimation accuracy. In addition, these methods require the rank to be determined in advance, which incurs a high computational cost in rank selection.

Another way to perform CP decomposition is Bayesian approaches. It is a type of method to infer the posterior probability distribution of parameters, such as a factor matrix. Unlike optimization methods, it has the advantage of being able to evaluate not only the estimated value but also its ambiguity. Some of the reported works include network structure analysis and collaborative filtering using tensor decomposition estimated by MCMC [81], [82]. However, the convergence speed of the MCMC estimation method is very slow.

ARD (Automatic Rank Determination) is a technique for tensor completion using CP decomposition with variational Bayes [69]. ARD is an algorithm that can perform rank

estimation as well as tensor completion and employs a hierarchical Bayesian model with a prior distribution that induces group sparsity in all factor matrices to improve robustness against noise. ARD allows us to avoid costly rank selection and achieve efficient tensor completion and rank estimation. However, ARD often causes duplication in the column vectors of factor matrices in replicated experiments. This leads to over-estimation of the CP rank, which deteriorates completion accuracy, estimation time and reduces compression performance.

In this chapter, we propose a new tensor completion method based on Bayesian CP decomposition with ARD using Multiplicative Gamma Process (MGP) shrinkage prior. MGP shrinkage prior is a distribution so that the core tensor of ARD is shrunk as much as possible [83], [84]. Since the core tensor and the factor matrix are linked, when the core tensor decays, the column vectors of factor matrices is ordered, and duplicates are removed. By applying MGP shrinkage prior to ARD, the proposed method can improve the accuracy of rank estimation, reduce the estimation time, and enhance robustness to noise.

Our contribution can be summarized as follows:

- We confirmed duplication in the column vectors of factor matrices in ARD by numerical experiments. We also gave mathematical proof of a property concerning the duplication of bases in particular situations. It is a significant contribution to point out this issue because duplication of the column vectors of the factor matrix leads to an overestimation of the rank, resulting in worse estimation accuracy, an increase in the estimation time, and reduced compression performance.

- To reduce the redundancy of the model based on the duplication of the column vectors of the factor matrix, which is a drawback of ARD, We proposed a new probabilistic model by using MGP. We also derived a variational Bayesian inference algorithm based on this probabilistic model that simultaneously performs tensor completion, denoising, and rank estimation.

- Experiments using synthetic data showed an increase in the accuracy of rank estimation and a decrease in estimation error. Experiments on real-world image and traffic data show that the estimation time is significantly reduced.

The rest of the paper is organized as follows: related works are described in Section 2.1, the review of conventional ARD method and its problems are described in Section 2.2, the proposed MGP-ARD method is described in Section 2.3, the experiments of the proposed method are described in Section 2.4.

2.1 Related works

There are four approaches for the CP rank estimation: Supervised learning, optimization, probabilistic estimation, and Bayesian inference.

First, supervised learning can be used for rank estimation. [85] proposed to use a CNN for CP rank estimation. Note that this approach requires training data.

On the other hand, there is a method of rank estimation based on optimization. In some studies, this has been applied to tensor completion [2], [59], [86].

There are also several methods based on probabilistic approaches. One method is to use probabilistic CP decomposition to estimate the CP rank by MAP estimation [87], which has been applied to channel estimation in MIMO systems [88]. Another approach is to use the EM algorithm to perform CP rank estimation and image denoising [89]. Although these studies are also stochastic approaches, they differ from our method in that they are based on point estimation and therefore cannot infer uncertainty in the solution.

On the other hand, there are methods to estimate the tensor rank using Bayesian estimation. For example, there is a method to obtain the CP rank for binary and real number tensors [84]. This method is different from our method in that it uses MCMC and convergence is slow. Also, this method is challenging to incorporate constraints such as smoothness into the Equation because the Equation is complicated due to the explicit mathematical model of the core tensor. There is also a method that uses variational Bayes to estimate the Tucker rank instead of the CP rank [90]. The ARD method is a variational Bayesian method that can simultaneously perform tensor completion, denoising, and rank estimation [69], and has been applied to the spatiotemporal traffic data imputation [91]. However, ARD has the disadvantage of duplication in the column vectors of factor matrices, making the model redundant.

2.2 Review of ARD

ARD is an algorithm for tensor completion based on Bayesian CP decomposition [69]. ARD can be applied to noisy data and can also perform rank estimation simultaneously. The modeling is constructed using hierarchical Bayes with a prior distribution that induces sparsity, and variational Bayes is used as the inference method.

2.2.1 Modeling

The $\boldsymbol{\mathcal{Y}}$ is an N-order tensor containing missing entries of size $I_1 \times I_2 \times \cdots \times I_N$. We define $(i_1, i_2, \ldots, i_N) \in \Omega$ as the index of the observation part, and the $\boldsymbol{\mathcal{O}}$ is the mask tensor such that the observed part is 1 and the missing part is 0. The $\boldsymbol{\mathcal{Y}}_\Omega \coloneqq \boldsymbol{\mathcal{Y}} \circledast \boldsymbol{\mathcal{O}}$ is defined as the element being observed. We also assume that $\boldsymbol{\mathcal{Y}}$ is the observed data with noise added to the latent tensor $\boldsymbol{\mathcal{X}}$, and formulate it as $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{E}}$. Here, $\boldsymbol{\epsilon}$ is assumed

to follow the independently and identically distributed (i.i.d.) Gaussian distribution. The latent tensor $\boldsymbol{\mathcal{X}}$ is defined as

$$\boldsymbol{\mathcal{X}} \coloneqq \sum_{r=1}^{R} \boldsymbol{a}_{:,r}^{(1)} \circ \cdots \circ \boldsymbol{a}_{:,r}^{(N)}, \tag{2.1}$$

which is represented by CP decomposition model. We define $\{\boldsymbol{A}^{(n)}\}_{n=1}^{N}$ as the factor matrix such that

$$\boldsymbol{A}^{(n)} \coloneqq \left[\boldsymbol{a}_{1,:}^{(n)}, \ldots, \boldsymbol{a}_{i_n,:}^{(n)}, \ldots, \boldsymbol{a}_{I_n,:}^{(n)}\right]^{\mathrm{T}} = \left[\boldsymbol{a}_{:,1}^{(n)}, \ldots, \boldsymbol{a}_{:,r}^{(n)}, \ldots, \boldsymbol{a}_{:,R}^{(n)}\right] \in \mathbb{R}^{I_n \times R}. \tag{2.2}$$

The probabilistic models for CP decomposition is represented by

$$p\left(\boldsymbol{\mathcal{Y}}_{\Omega} \middle| \{\boldsymbol{A}^{(n)}\}_{n=1}^{N}, \tau_c\right) \coloneqq \prod_{i_1=1}^{I_1} \cdots \prod_{i_N=1}^{I_N} \mathcal{N}\left(\mathcal{Y}_{i_1,i_2,\ldots,i_N} \middle| \langle \boldsymbol{a}_{i_1,:}^{(1)}, \ldots, \boldsymbol{a}_{i_N,:}^{(N)} \rangle, \tau_c^{-1}\right)^{\mathcal{O}_{i_1,\ldots,i_N}}, \tag{2.3}$$

where $\mathcal{N}(x|\mu,\sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\{-\frac{(x-\mu)^2}{2\sigma^2}\}$ is Gaussian distribution, and τ_c is precision, $\langle \boldsymbol{a}_{i_1,:}^{(1)}, \boldsymbol{a}_{i_2,:}^{(2)}, \ldots, \boldsymbol{a}_{i_N,:}^{(N)} \rangle \coloneqq \sum_r \prod_n a_{i_n,r}^{(n)}$. $\mathcal{O}_{i_1,\ldots,i_N}$ is the value of mask tensor $\boldsymbol{\mathcal{O}}$.

Next, we will discuss rank determination in ARD. In general, it is very difficult to estimate the dimension R of the latent space with the least redundancy, i.e., CP rank [92]. By the definition of CP rank [10], R should be a minimum value. ARD attempts to automatically determine CP rank in the process of Bayesian inference by setting up a prior distribution that induces sparsity for all factor matrices. This is based on the idea of sparse Bayesian learning [93], automatic relevance determination [94], [95], [96], and automatic association decision [97]. We will explain the discussion so far in more detail using formulas.

For all the factor matrices $\boldsymbol{A}^{(n)}$, prior distribution of ARD is defined as

$$p(\boldsymbol{A}^{(n)}|\boldsymbol{\lambda}) \coloneqq \prod_{i_n=1}^{I_n} \mathcal{N}(\boldsymbol{a}_{i_n,:}^{(n)}|\boldsymbol{0}, \boldsymbol{\Lambda}^{-1}), \tag{2.4}$$

where $\boldsymbol{\lambda} = [\lambda_1, \ldots, \lambda_R]$ ($\boldsymbol{\Lambda} = \mathrm{diag}(\boldsymbol{\lambda})$) is precision. In addition, prior distribution of precision is defined as

$$p(\boldsymbol{\lambda}) \coloneqq \prod_{r=1}^{R} \mathrm{Ga}(\lambda_r|c_0^r, d_0^r), \tag{2.5}$$

where $\mathrm{Ga}(x|a,b) = \frac{b^a x^{a-1} e^{-bx}}{\Gamma(a)}$ is Gamma distribution. The prior distribution of the factor matrix (Equation (2.4)) has a mean 0, so its value approaches 0 as the precision increases. Furthermore, since all factor matrices share the precision parameters, that is, the inverse

of the precision can be interpreted as super diagonal entries of the core tensor. This prior distribution is sparse because it sets the unimportant components of the factor matrix to zero. The number of components of the factor matrix obtained from the inference corresponds to CP rank. Since the model is robust to noise because of its sparsity, denoising can also be achieved from this prior distribution. With this prior distribution, ARD can efficiently derive CP rank.

The precision of CP decomposition model is also set as a probability distribution, and the distribution is defined as

$$p(\tau_c) \coloneqq \mathrm{Ga}(\tau_c | a_0, b_0). \tag{2.6}$$

In summary, we define the unobserved latent parameters as $\boldsymbol{\Theta} = \{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)}, \boldsymbol{\lambda}\ , \tau_c\}$, probabilistic modeling of ARD is defined as

$$p(\boldsymbol{\mathcal{Y}}_\Omega, \boldsymbol{\Theta}) \coloneqq p(\boldsymbol{\mathcal{Y}}_\Omega | \{\boldsymbol{A}^{(n)}\}_{n=1}^N, \tau_c) p(\tau_c) \prod_{n=1}^N p(\boldsymbol{A}^{(n)} | \boldsymbol{\lambda}) p(\boldsymbol{\lambda}). \tag{2.7}$$

2.2.2 Inference

Next, we will give an overview of inference methods. The goal of Bayesian inference is to derive the posterior distribution of parameters. The posterior distribution is derived as

$$p(\boldsymbol{\Theta} | \boldsymbol{\mathcal{Y}}_\Omega) = \frac{p(\boldsymbol{\mathcal{Y}}_\Omega, \boldsymbol{\Theta})}{\int p(\boldsymbol{\mathcal{Y}}_\Omega, \boldsymbol{\Theta}) d\boldsymbol{\Theta}}. \tag{2.8}$$

The missing entries are estimated by calculating the predictive distribution, and the predictive distribution is derived by

$$p(\boldsymbol{\mathcal{Y}}_{\backslash\Omega} | \boldsymbol{\mathcal{Y}}_\Omega) = \int p(\boldsymbol{\mathcal{Y}}_{\backslash\Omega} | \boldsymbol{\Theta}) p(\boldsymbol{\Theta} | \boldsymbol{\mathcal{Y}}_\Omega) d\boldsymbol{\Theta}. \tag{2.9}$$

In Equations (2.8), (2.9), we need to calculate the multiple integrals with parameter $\boldsymbol{\Theta}$, which is very difficult with complex latent variables.

In variational Bayes, in order to find a distribution $q(\boldsymbol{\Theta})$ that approximates the true posterior distribution $p(\boldsymbol{\Theta} | \boldsymbol{\mathcal{Y}}_\Omega)$, we derive q such that the KL divergence is minimized. This derivation is

$$\arg\min_q KL\left(q(\boldsymbol{\Theta}) || p(\boldsymbol{\Theta} | \boldsymbol{\mathcal{Y}}_\Omega)\right) = \arg\max_q \mathcal{L}(q), \tag{2.10}$$

where $\mathcal{L}(q) \coloneqq \int q(\boldsymbol{\Theta}) \ln \left\{ \frac{p(\boldsymbol{\mathcal{Y}}_\Omega, \boldsymbol{\Theta})}{q(\boldsymbol{\Theta})} \right\} d\boldsymbol{\Theta}$ is the variational lower bound. In this study, we employ the mean-field approximation for $q(\boldsymbol{\Theta})$, is defined as

$$q(\boldsymbol{\Theta}) \coloneqq q_\lambda(\boldsymbol{\lambda}) q_\tau(\tau) \prod_{n=1}^N q_n\left(\boldsymbol{A}^{(n)}\right). \tag{2.11}$$

Computing Equations (2.10), (2.11) after the mean field approximation, we get

$$\ln q_j(\boldsymbol{\Theta}_j) = \mathbb{E}_{q(\boldsymbol{\Theta}\setminus\boldsymbol{\Theta}_j)}[\ln p(\boldsymbol{\mathcal{Y}}_\Omega, \boldsymbol{\Theta})] + \text{const}, \tag{2.12}$$

and the approximate distribution $q_j(\boldsymbol{\Theta}_j)$ is obtained in a closed form. Note that $\mathbb{E}_{q(\boldsymbol{\Theta}\setminus\boldsymbol{\Theta}_j)}[\cdot]$ is the expected value for $\boldsymbol{\Theta}$ of all variables except $\boldsymbol{\Theta}_j$.

From Equation (2.12), the parameters $\boldsymbol{\Theta} = \{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)}, \boldsymbol{\lambda}, \tau\}$ can be inferred alternately. Since there are dependencies among the variables, it is necessary to obtain them using an iterative method.

After obtaining the approximate posterior distribution, the approximate predictive distribution can be obtained from

$$p(\boldsymbol{\mathcal{Y}}_{\setminus\Omega}|\boldsymbol{\mathcal{Y}}_\Omega) = \int p(\boldsymbol{\mathcal{Y}}_{\setminus\Omega}|\boldsymbol{\Theta})q(\boldsymbol{\Theta})d\boldsymbol{\Theta}, \tag{2.13}$$

and tensor completion, including ambiguity, can be achieved.

In ARD, the rank is obtained simultaneously with tensor completion. In the process of Bayesian inference, the result of the update of $\boldsymbol{\lambda}$ affects the new posterior distribution of the entire factor matrix $\boldsymbol{A}^{(n)}$, which in turn affects the next update of $\boldsymbol{\lambda}$. Therefore, when λ_r becomes very large, the prior distribution forces the r-th component of $\boldsymbol{A}^{(n)}$ toward zero. The tensor rank can then be obtained by counting the number of non-zero components of the factor matrix.

2.2.3 Overestimation of the CP rank by ARD

In this section, we discuss a phenomenon that ARD sometimes generates redundancy in CP decomposition model. In our experiments, we found that ARD can cause duplication in the column vectors of the factor matrix, and it was difficult to improve even with more iterations. Duplication in the experiment is shown in Figure 2.1. This is the factor matrix obtained by the ARD inference results when using a third-order tensor with true rank 3 and size $30 \times 30 \times 30$ as the data for completion. We can see that the ARD estimated the CP rank as 5 with redundant bases. This phenomenon occurs quite frequently. The results of the previous experiment after 100 trials show that in as many as 40 out of 100 trials, the estimated rank is greater than the true rank of 3.

It can also be shown mathematically that, under very specific conditions, when duplication in the column vectors of one factor matrix, the other factor matrices will also overlap.

Theorem 1. *Let the n-th factor matrix $\boldsymbol{A}^{(n)}$ be from Equation (2.2). If factor matrices $\boldsymbol{A}^{(k\neq n)}(k = 1, \ldots, n-1, n+1, \ldots, N)$ are rank 1 for all k, then the mean of $\boldsymbol{A}^{(n)}$ is less than or equal to 1.*

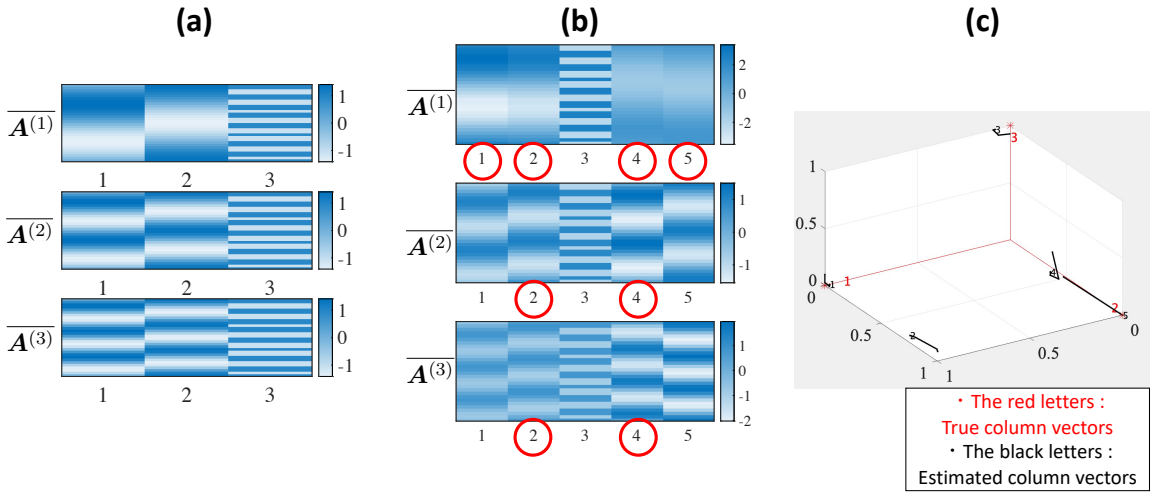


Figure 2.1 An example of duplication of the column vectors of the factor matrices. True rank is 3, it is shown in (a). The matrices are shown in (b). The 1st, 2nd, 4th, and 5th bases are essentially the same.(c) shows the estimation of the column vectors in 3-D space. The red letters represent the true column vectors, and the black trajectory represents the estimation process. We can see that the 2nd, 5th and 4th trajectories overlap.

Proof. In the proof, we show that the approximate distribution $q(\boldsymbol{A}^{(n)})$ is Gaussian, and the mean matrix $\overline{\boldsymbol{A}^{(n)}}$ with mean vector $\tilde{\boldsymbol{a}}^{(n)}_{i_n,:}$ of a Gaussian is of rank 1 or less. $\overline{\boldsymbol{A}^{(n)}}$ is defined as

$$\overline{\boldsymbol{A}^{(n)}} := \left[\tilde{\boldsymbol{a}}^{(n)}_{1,:}, \ldots, \tilde{\boldsymbol{a}}^{(n)}_{i_n,:}, \ldots, \tilde{\boldsymbol{a}}^{(n)}_{I_n,:}\right]^{\mathrm{T}}. \tag{2.14}$$

By computing Equation (2.12), the approximated posterior distribution of the factor matrix $\boldsymbol{A}^{(n)}$ is defined as

$$q_n(\boldsymbol{A}^{(n)}) \;=\; \prod_{i_n=1}^{I_n} \mathcal{N}(\boldsymbol{a}_{i_n,:} | \tilde{\boldsymbol{a}}^{(n)}_{i_n,:}, \boldsymbol{V}^{(n)}_{i_n}), \tag{2.15}$$

and its parameters can be calculated by

$$\begin{aligned}
\tilde{\boldsymbol{a}}^{(n)}_{i_n,:} &= \mathbb{E}_q[\tau_c]\boldsymbol{V}^{(n)}_{i_n}\mathbb{E}_q\left[\boldsymbol{A}^{(\backslash n)\mathrm{T}}\right]\boldsymbol{O}_{i_n}\boldsymbol{y}_{i_n}, \\
\boldsymbol{V}^{(n)}_{i_n} &= \left(\mathbb{E}_q\left[\boldsymbol{A}^{(\backslash n)\mathrm{T}}\boldsymbol{O}_{i_n}\boldsymbol{A}^{(\backslash n)}\right]\mathbb{E}_q[\tau_c] + \mathbb{E}_q\left[\boldsymbol{\Lambda}\right]\right)^{-1},
\end{aligned}$$

where $\boldsymbol{A}^{(\backslash n)}, \boldsymbol{y}_{i_n}, \boldsymbol{O}_{i_n}$ denote parameters is defined as

$$\begin{aligned}
\boldsymbol{A}^{(\backslash n)} &= \bigodot_{k\neq n} \boldsymbol{A}^{(k)} \in \mathbb{R}^{\Pi_{k\neq n} I_k \times R}, \\
\boldsymbol{y}_{i_n} &= \mathrm{vec}(\boldsymbol{\mathcal{Y}}_{i_n}) \in \mathbb{R}^{\Pi_{k\neq n} I_k}, \\
\boldsymbol{O}_{i_n} &= \mathrm{diag}(\mathbb{I}(\boldsymbol{\mathcal{O}}_{i_n} = 1)) \in \mathbb{R}^{\Pi_{k\neq n} I_k \times \Pi_{k\neq n} I_k},
\end{aligned}$$

where the $\boldsymbol{\mathcal{Y}}_{i_n}$ denotes the n-mode i_n-th $N-1$th order tensor of the observation tensor, and $\boldsymbol{\mathcal{O}}_{i_n}$ denotes the n-mode i_n-th $N-1$th order tensor of the mask tensor. $\mathbb{I}(\cdot)$ is the indicator function and $\mathrm{vec}(\cdot)$ is the vectorization function of the tensor.

The mean matrix can be calculated from Equation (2.14), (2.15) to

$$\overline{\boldsymbol{A}^{(n)}}^{\mathrm{T}} \propto \left(\mathbb{E}_q\left[\boldsymbol{\Lambda}\right] + \mathbb{E}_q\left[\left(\bigodot_{k\neq n} \boldsymbol{A}^{(k)} \right)^{\mathrm{T}} \boldsymbol{O}_{i_n} \left(\bigodot_{k\neq n} \boldsymbol{A}^{(k)} \right) \right] \right)^{-1} \mathbb{E}_q\left[\bigodot_{k\neq n} \boldsymbol{A}^{(k)\mathrm{T}} \right] \boldsymbol{O}_{i_n} \boldsymbol{Y}^{(n)}, \tag{2.16}$$

where $\boldsymbol{Y}^{(n)}$ is defined as

$$\boldsymbol{Y}^{(n)} = \left[\boldsymbol{y}_1, \ldots, \boldsymbol{y}_{i_n}, \ldots, \boldsymbol{y}_{I_n} \right].$$

Since rank of $\boldsymbol{A}^{(k\neq n)}$ is 1, it is represented as

$$\boldsymbol{A}^{(k\neq n)} = \left[c_{k,1}\boldsymbol{a}^{(k)}_{:,1}, \ldots, c_{k,r}\boldsymbol{a}^{(k)}_{:,1}, \ldots, c_{k,R}\boldsymbol{a}^{(k)}_{:,1} \right] \in \mathbb{R}^{I_k \times R} \quad (k = 1, \ldots, n-1, n+1, \ldots, N).$$

Now, from

$$\mathbb{E}_q\left[\bigodot_{k\neq n} \boldsymbol{A}^{(k)} \right] = \left[\otimes_{k\neq n} c_{k,1} \overline{\boldsymbol{a}}^{(k)}_{:,1}, \ldots, \otimes_{k\neq n} c_{k,r} \overline{\boldsymbol{a}}^{(k)}_{:,1}, \ldots, \otimes_{k\neq n} c_{k,R} \overline{\boldsymbol{a}}^{(k)}_{:,1} \right],$$

$\mathrm{rank}\left(\mathbb{E}_q\left[\bigodot_{k\neq n} \boldsymbol{A}^{(k)} \right] \right) = 1$. From $\mathrm{rank}(\boldsymbol{XY}) \leqq \mathrm{rank}(\boldsymbol{X})$ and Equation (2.16), $\mathrm{rank}\left(\overline{\boldsymbol{A}^{(n)}} \right) \leqq 1$, so the rank of $\overline{\boldsymbol{A}^{(n)}}$ is less than or equal to 1. □

Since basis duplication is equivalent to rank reduction, Theorem 1 shows mathematically that the basis duplication of the Nth factor matrix is induced by the duplication of the basis of the non-Nth factor matrix. However, note that since this is a theorem under the very restrictive conditions of rank 1, basis duplication is essentially an assertion based on experimental results.

Overestimation of the CP rank causes at least three kinds of issues in applications. First, it directly reduces the data compression performance of the CP decomposition. Second, it reduces robustness to noise because the extra components out of the true rank may fit to the noise parameters. Third, it increases the computational cost of the algorithm since the computational complexity of ARD is proportional to R^3.

2.3 Proposed method

In this section, we propose a new tensor completion/decomposition method that uses a prior distribution in which the core tensor is decayed with Multiplicative Gamma Process (MGP). We call the proposed method as MGP-ARD. Because of the effects of MGP shrinkage prior, MGP-ARD reduces the problem of model redundancy in ARD.

2.3.1 Modeling

MGP-ARD is a method in which the MGP [84] distribution is set as the prior distribution of accuracy $\boldsymbol{\lambda}$ in ARD (see Equation (2.4)). The distribution of MGP used in the proposed method is defined as

$$\begin{aligned} p(\lambda_r|\tau_r) &\coloneqq \mathrm{Ga}(\lambda_r|c_0, \tau_r), \\ \tau_r &\coloneqq \prod_{l=1}^{r} \delta_l \quad (0 < \delta_l < 1), \\ p(\delta_r) &\coloneqq \mathrm{Ga}(\delta_r|e_0, f_0). \end{aligned} \tag{2.17}$$

In this study, we employ somewhat different formulation/modeling of MGP of original one. Equation (2.17) is a model that expresses that as the index r increases, the accuracy λ increases, and the core tensor decays. The accuracy increases as the index r increases because δ_r is a truncated gamma distribution from 0 to 1, and the scale parameter τ_r of the gamma distribution is a multiplication of δ_r. An overview of the decay of the core tensor is illustrated in Figure 2.2. MGP is based on the idea of nonparametric factor analysis [83]. In nonparametric factor analysis, to resolve the indistinguishability [98] caused by the rotational invariance of factor analysis, a gamma distribution has been introduced such that the values decay to zero as the index increases in the column direction of the loading matrix, achieving a significant reduction in the number of parameters. MGP-ARD avoids duplication of bases by having the factor matrices be ordered, thus improving the redundancy of the model, which is an issue in ARD. An overview of the improvement of duplication is illustrated in Figure 2.3. It is also expected that the order invariance of CP decomposition is resolved in the proposed method, which narrows down the solution space of parameters and achieves efficient estimation.

The graphical model of MGP-ARD is shown in Figure 2.4. The parameters of the MGP-ARD are $\boldsymbol{\Theta} = \{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)}, \boldsymbol{\lambda}, \boldsymbol{\delta}, \tau_c\}$, and the probability modeling of the MGP-ARD is defined as

$$p(\boldsymbol{\mathcal{Y}}_\Omega, \boldsymbol{\Theta}) \coloneqq p(\boldsymbol{\mathcal{Y}}_\Omega|\{\boldsymbol{A}^{(n)}\}_{n=1}^N, \tau_c)p(\tau_c) \prod_{n=1}^{N} p(\boldsymbol{A}^{(n)}|\boldsymbol{\lambda}) \prod_{r=1}^{R} p(\lambda_r|\tau_r)p(\delta_r), \tag{2.18}$$

from (2.17). The prior distribution of $\boldsymbol{A}^{(n)}, \tau_c$ other than MGP shrinkage prior distribution is identical to ARD, i.e., Equations (2.4) and (2.6).

2.3.2 Inference

The distribution of the parameters to be estimated is $\boldsymbol{\Theta} = \{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)}, \boldsymbol{\lambda}, \boldsymbol{\delta}, \tau_c\}$, derived from the Equation (2.12), respectively.

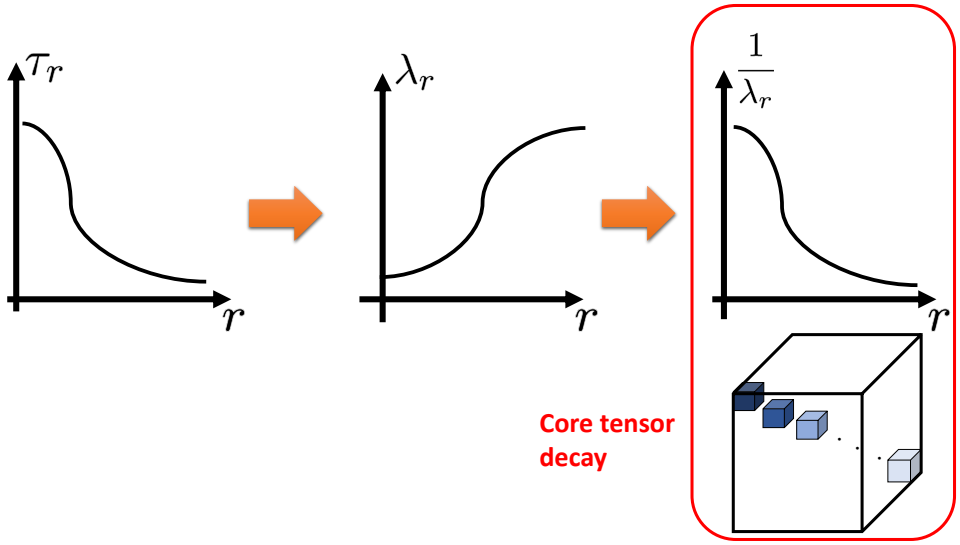


Figure 2.2 An overview of the decay of the core tensor by MGP. The scale parameter τ_r of the gamma distribution decreases as the index r increases. Thus, λ_r increases as the index r increases, resulting in a decrease in core tensor $\frac{1}{\lambda_r}$.

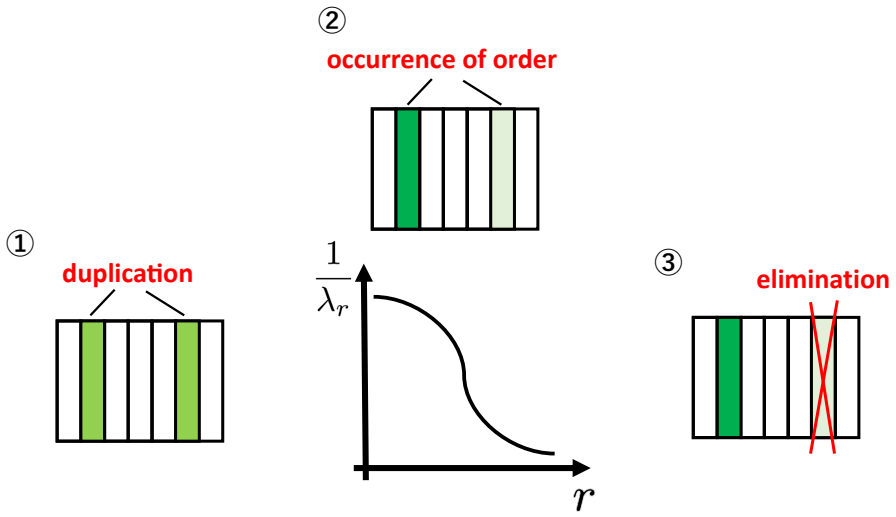


Figure 2.3 An overview of how duplication is improved by MGP. The core tensor $\frac{1}{\lambda_r}$ decays as the index r increases. This results in an ordering of the factor matrices linked to the core tensor, which improves the duplication in the factor matrices.

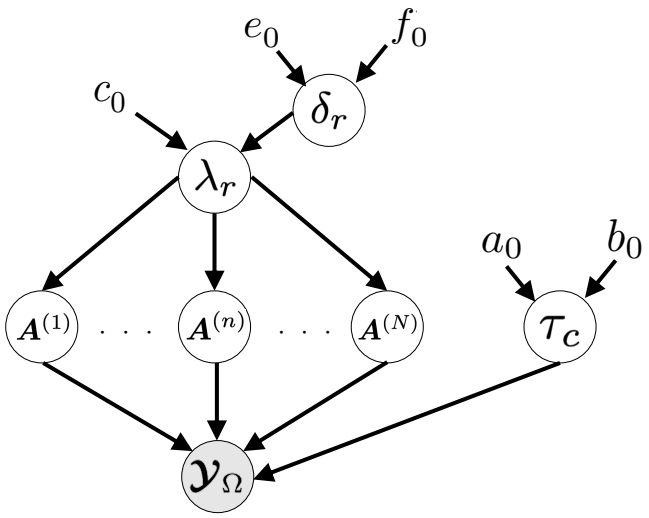


Figure 2.4 Graphical model of MGP-ARD. The parameters to be estimated is $\boldsymbol{\Theta} = \{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)}, \boldsymbol{\lambda}, \boldsymbol{\delta}, \tau_c\}$. Observed data are represented by colored circles and unobserved data by white circles.

The approximated posterior distribution of the factor matrix $\boldsymbol{A}^{(n)}$ is defined as

$$q_n(\boldsymbol{A}^{(n)}) = \prod_{i_n=1}^{I_n} \mathcal{N}(\boldsymbol{a}_{i_n,:} | \tilde{\boldsymbol{a}}_{i_n,:}^{(n)}, \boldsymbol{V}_{i_n}^{(n)}), \tag{2.19}$$

and its parameters can be calculated by

$$\begin{aligned} \tilde{\boldsymbol{a}}_{i_n,:}^{(n)} &= \mathbb{E}_q[\tau_c] \boldsymbol{V}_{i_n}^{(n)} \mathbb{E}_q\left[\boldsymbol{A}^{(\backslash n)\mathrm{T}}\right] \boldsymbol{O}_{i_n} \boldsymbol{y}_{i_n}, \\ \boldsymbol{V}_{i_n}^{(n)} &= \left(\mathbb{E}_q\left[\boldsymbol{A}^{(\backslash n)\mathrm{T}} \boldsymbol{O}_{i_n} \boldsymbol{A}^{(\backslash n)}\right] \mathbb{E}_q[\tau_c] + \mathbb{E}_q\left[\boldsymbol{\Lambda}\right]\right)^{-1}, \end{aligned}$$

where $\boldsymbol{A}^{(\backslash n)}, \boldsymbol{y}_{i_n}, \boldsymbol{O}_{i_n}$ denote parameters is defined as

$$\begin{aligned} \boldsymbol{A}^{(\backslash n)} &\coloneqq \bigodot_{k \neq n} \boldsymbol{A}^{(k)} \in \mathbb{R}^{\Pi_{k \neq n} I_k \times R}, \\ \boldsymbol{y}_{i_n} &\coloneqq \mathrm{vec}(\boldsymbol{\mathcal{Y}}_{i_n}) \in \mathbb{R}^{\Pi_{k \neq n} I_k}, \\ \boldsymbol{O}_{i_n} &\coloneqq \mathrm{diag}(\mathbb{I}(\boldsymbol{\mathcal{O}}_{i_n} = 1)) \in \mathbb{R}^{\Pi_{k \neq n} I_k \times \Pi_{k \neq n} I_k}, \end{aligned}$$

where the $\boldsymbol{\mathcal{Y}}_{i_n}$ denotes the n-mode i_n-th (N-1)th order tensor of the observation tensor, and $\boldsymbol{\mathcal{O}}_{i_n}$ denotes the n-mode i_n-th (N-1)th order tensor of the mask tensor. $\mathbb{I}(\cdot)$ is the indicator function and $\mathrm{vec}(\cdot)$ is the vectorization function of the tensor.

Focusing on the expression of the posterior covariance $\boldsymbol{V}_{i_n}^{(n)}$, we can see that it is controlled by the noise precision parameter τ_c of CP decomposition model. In other words, if τ_c is large, the contribution of the factor matrix $\boldsymbol{A}^{(\backslash n)}$, which is a model term, will be large, and if it is small, the contribution of $\boldsymbol{\Lambda}$, which is a term related to decay (MGP), will be enormous. Focusing on the expression for the posterior mean $\boldsymbol{V}_{i_n}^{(n)}$, we can see

that $\boldsymbol{\mathcal{Y}}$, which represents the observed data, and $\boldsymbol{A}^{(\backslash n)}$, which represents the factor matrix of the model, are correlated.

The approximated posterior distribution of the factor matrix accuracy λ_r is defined as

$$q(\lambda_r) = \mathrm{Ga}(\lambda_r | c_M^r, d_M^r), \tag{2.20}$$

and its parameters can be calculated by

$$\begin{aligned} c_M^r &= c_0 + \frac{1}{2}\sum_{n=1}^{N} I_n, \\ d_M^r &= \mathbb{E}_q[\tau_r] + \frac{1}{2}\sum_{n=1}^{N} \mathbb{E}_q[\boldsymbol{a}_{:,r}^{(n)\mathrm{T}} \boldsymbol{a}_{:,r}^{(n)}]. \end{aligned}$$

Focusing on the expression for d_M^r, the first term is τ_r, which is related to the decay (MGP), and the second term is $\mathbb{E}_q[\boldsymbol{a}_{:,r}^{(n)\mathrm{T}} \boldsymbol{a}_{:,r}^{(n)}]$, which is related to the model. In other words, in the case of ARD, when the r-th component of the factor matrix, $\|\boldsymbol{a}_{:,r}\|_2^2$, becomes small, the accuracy of the r-th component increases and induces sparsity, while in MGP-ARD, also, the decay mechanism τ_r also affects the accuracy.

The approximated distribution of the posterior distribution of the degeneracy mechanism δ_r is defined as

$$q(\delta_r) = \mathrm{Ga}(\delta_r | e_M^r, f_M^r), \tag{2.21}$$

and its parameters can be calculated by

$$\begin{aligned} e_M^r &= (R - r + 1)(c_0 - 1) + e_0, \\ f_M^r &= \sum_{h=r}^{R} \mathbb{E}_q[\lambda_r] \prod_{l=1, l\neq r}^{h} \mathbb{E}_q[\delta_l] + f_0. \end{aligned}$$

The approximated posterior distribution of the accuracy τ_c of the model for CP decomposition is defined as

$$q(\tau_c) = \mathrm{Ga}(\tau_c | a_M, b_M), \tag{2.22}$$

and its parameters can be calculated by

$$\begin{aligned} a_M &= a_0 + \frac{1}{2}\sum_{i_1,\ldots,i_N} \mathcal{O}_{i_1,\ldots,i_N}, \\ b_M &= b_0 + \frac{1}{2}\mathbb{E}_q\left[\|\boldsymbol{\mathcal{O}} \circledast (\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}})\|_F^2\right]. \end{aligned}$$

Focusing on the expression for b_M, the second term represents the error between the observed data $\boldsymbol{\mathcal{Y}}$ and the latent tensor $\boldsymbol{\mathcal{X}}$ (factor matrix $\boldsymbol{A}^{(n)}$), which is the model.

Algorithm 1 MGP-ARD

Require: Observation tensor $\boldsymbol{\mathcal{Y}}_\Omega$, mask tensor $\boldsymbol{\mathcal{O}}_\Omega$

Initialize: $\boldsymbol{A}^{(n)}, \boldsymbol{V}_{i_n}^{(n)}, \forall i_n \in [1, I_n], \forall n \in [1, N], a_0, b_0, c_0, d_0, e_0, f_0, \text{and}\, \tau_c = \frac{a_0}{b_0}, \lambda_r = \frac{c_0^r}{\tau_r}, \delta_r = \frac{e_0^r}{f_0^r} \forall, r \in [1, R]$.

repeat

for $n = 1, \ldots, N$ **do**

Update the approximated posterior distribution $q(\boldsymbol{A}^{(n)})$ using Equation (2.19).

end for

Update the approximated posterior distribution $q(\boldsymbol{\delta})$ using Equation (2.21).

Update the approximated posterior distribution $q(\boldsymbol{\lambda})$ using Equation (2.20).

Update the approximated posterior distribution $q(\tau)$ using Equation (2.22).

Calculate the variational lower bound $\mathcal{L}(q)$ using Equation (2.24).

Reduce the rank R by removing the 0 component of $\{\boldsymbol{A}^{(n)}\}$.

until $\mathcal{L}(q)$ converges by checking Equation (2.23).

Compute the predictive distribution from Equation (2.25).

Next, we will discuss the specific formulation for the variational lower bound. The convergence of the algorithm is determined by

$$\left| \frac{\mathcal{L}(q)^{(t)} - \mathcal{L}(q)^{(t-1)}}{\mathcal{L}(q)^{(2)}} \right| < \epsilon, \tag{2.23}$$

where ϵ is the convergence threshold. The variational lower bound $\mathcal{L}(q)$ can be calculated by

$$\begin{aligned}
\mathcal{L}(q) = & -\frac{a_M}{2b_M} \mathbb{E}_q \left[\| \boldsymbol{\mathcal{O}} \circledast (\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}}) \|_F^2 \right] \\
& - \frac{1}{2} \mathrm{Tr} \left\{ \tilde{\boldsymbol{\Lambda}} \sum_n \left(\tilde{\boldsymbol{A}}^{(n)\mathrm{T}} \tilde{\boldsymbol{A}}^{(n)} + \sum_{i_n} \boldsymbol{V}_{i_n}^{(n)} \right) \right\} + \frac{1}{2} \sum_n \sum_{i_n} \ln \left| \boldsymbol{V}_{i_n}^{(n)} \right| \\
& + \sum_{r=1}^{R} \{ \ln \Gamma(c_M^r) + \ln \Gamma(e_M^r) + c_M^r (1 - \ln b_M) \\
& + e_M^r \left(1 - e_M^r \ln f_M^r - \frac{1}{f_M^r} \left(\frac{c_M^r}{d_M^r} \prod_{l=1, l \neq r}^{r} \frac{e_M^l}{f_M^l} + f_0 \right) \right) \Big\} \\
& + \ln \Gamma(a_M) + a_M \left(1 - \ln b_M - \frac{b_0}{b_M} \right).
\end{aligned} \tag{2.24}$$

Finally, we discuss the specific formula for predictive distribution. The purpose of

predictive distributions is tensor completion. The predictive distribution can be approximately calculated by

$$\begin{aligned} p(\mathcal{Y}_{i_1,\ldots,i_N}|\boldsymbol{\mathcal{Y}}_\Omega) &= \int p(\mathcal{Y}_{i_1,\ldots,i_N}|\boldsymbol{\Theta})p(\boldsymbol{\Theta}|\boldsymbol{\mathcal{Y}}_\Omega)d\boldsymbol{\Theta} \\ &\simeq \int p\left(\mathcal{Y}_{i_1,\ldots,i_N}\middle|\left\{\boldsymbol{a}^{(n)}_{i_n,:}\right\},\tau^{-1}\right) q\left(\left\{\boldsymbol{a}^{(n)}_{i_n,:}\right\}\right) q(\tau) d\boldsymbol{a}^{(n)}_{i_n,:} d\tau \\ &\simeq \mathcal{T}\left(\tilde{\mathcal{Y}}_{i_1,\ldots,i_N}, \mathcal{S}_{i_1,\ldots,i_N}, \nu_y\right), \end{aligned} \tag{2.25}$$

where $\mathcal{T}$ is Student's t-distribution and

$$\begin{aligned} \tilde{\mathcal{Y}}_{i_1,\ldots,i_N} &= \langle \tilde{\boldsymbol{a}}^{(1)}_{i_1,:}, \ldots, \tilde{\boldsymbol{a}}^{(N)}_{i_N,:} \rangle, \\ \nu_y &= 2a_M, \\ \mathcal{S}_{i_1,\ldots,i_N} &= \left\{ \frac{b_M}{a_M} + \sum_n \left\{ \left(\circledast_{k\neq n} \tilde{\boldsymbol{a}}^{(k)}_{i_k,:} \right)^{\mathrm{T}} \boldsymbol{V}^{(n)}_{i_n} \left(\circledast_{k\neq n} \tilde{\boldsymbol{a}}^{(k)}_{i_k,:} \right) \right\} \right\}^{-1}. \end{aligned}$$

The process of deriving Equation (2.25) is described in [99].

An overview of the algorithm is shown in Algorithm 1. Similar to ARD, when the value of the r-th element of the factor matrix $\boldsymbol{A}^{(n)}$ becomes zero (below the threshold) during the inference process, the r-th factor is removed. The number of components in the factor matrix is the rank, enabling rank determination.

2.3.3 Computional complexity

The computation cost of the factor matrices $\boldsymbol{A}^{(n)}$ in Equation (2.19) is $O(NR^2M + R^3\sum_n I_n)$, where N is the order of the tensor, M denotes the number of observations, i.e., the input data size. R is the number of latent components in each $\boldsymbol{A}^{(n)}$, i.e., model complexity or tensor rank. The computation cost of the hyperparameter $\boldsymbol{\lambda}$ in Equation (2.20) is $O(R^2\sum_n I_n)$. The computation cost of the hyperparameter $\boldsymbol{\delta}$ in Equation (2.21) is $O(R^2)$. The computation cost of the hyperparameter τ in Equation (2.22) is $O(R^2M)$. Therefore, the overall complexity of our algorithm is $O(NR^2M+R^3)$, which scales linearly with the data size but polynomially with the model complexity. It can be seen that the algorithm strongly depends on the tensor rank, i.e., the complexity R of the model. The computational complexity of ARD is $O(NR^2M + R^3)$, which is the same as the proposed method. However, unlike ARD, MGP-ARD does not evaluate R excessively high, so the computation time is shorter, and the algorithm is faster.

2.4 Experiment

In this section, we describe the experiments to verify the effectiveness of the proposed MGP-ARD. Since MGP-ARD is proposed to correctly estimate the CP rank, which was

overestimated in ARD, the following points are verified from both artificial data and image data.

1. By attempting to estimate the CP rank for artificial tensor data, we examine whether the estimation accuracy of MGP-ARD is improved compared to ARD [69], an original method. We also examine whether the estimation error is reduced by not overestimating the rank in the problem of noisy tensor completion.

2. We attempt to recover incomplete images with noise. We verify that the estimation time is reduced by suppressing the overestimation of the rank while maintaining a high estimation system by comparing it to ARD [69] and MGP-a [84].

The experiment was conducted in the following environments: CPU: Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz, 12 cores, Memory: 512GByte, Software: Matlab R2019a.

2.4.1 Experiments on artificial data (Rank is known)

In tensor completion, we verify whether the duplication of column vectors is improved and the accuracy of rank estimation is improved compared to the existing original ARD method. In order to check the accuracy of the rank estimation, we use synthetic data. In this experiment, the number of tensor orders is 3, and the sizes are $30 \times 30 \times 30$ and $20 \times 40 \times 10$. The true rank is 3 or 5, respectively. To investigate the robustness to the missing, we experiment with different observation rates (0.2, 0.5, and 0.9). The noise is 20 [dB], and 50 trials are performed for each experimental pattern. The convergence threshold ϵ is 1.0×10^{-5}, and the estimation accuracy is RSE $= \frac{\|\hat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}\|_F}{\|\boldsymbol{\mathcal{X}}\|_F}$. The initial value of R is twice the true rank.

The experimental results are shown in Figure 2.5. In all cases, the accuracy of rank estimation of MGP-ARD (the proposed method) was higher than that of ARD (the original method). In particular, when the loss rate is low, ARD tends to overestimate the rank because the amount of noise in the data is large. At the same time, MGP-ARD estimates the true rank, resulting in a significant difference in estimation accuracy.

Table 2.1 shows the mean and median RSE of the proposed method (MGP-ARD) and the existing method (ARD). Both mean and median RSE were lower in MGP-ARD under most conditions, suggesting that RSE was improved. This suggests that the MGP-ARD does not overestimate the rank, thereby reducing the redundancy of the model and avoiding sensitivity to noise.

Next, we also compared ARD and MGP-ARD in various noise levels: SNR is 0, 10, and 20 [dB]. The size of the tensor is $30 \times 30 \times 30$, the missing rate is 0.5, the true rank is 3 and 5, and each combination is tried 50 times. The experimental results are shown in Figure 2.6, Table 2.2. In all cases, the accuracy of rank estimation of MGP-ARD (the proposed method) was higher than that of ARD (the original method). The proposed

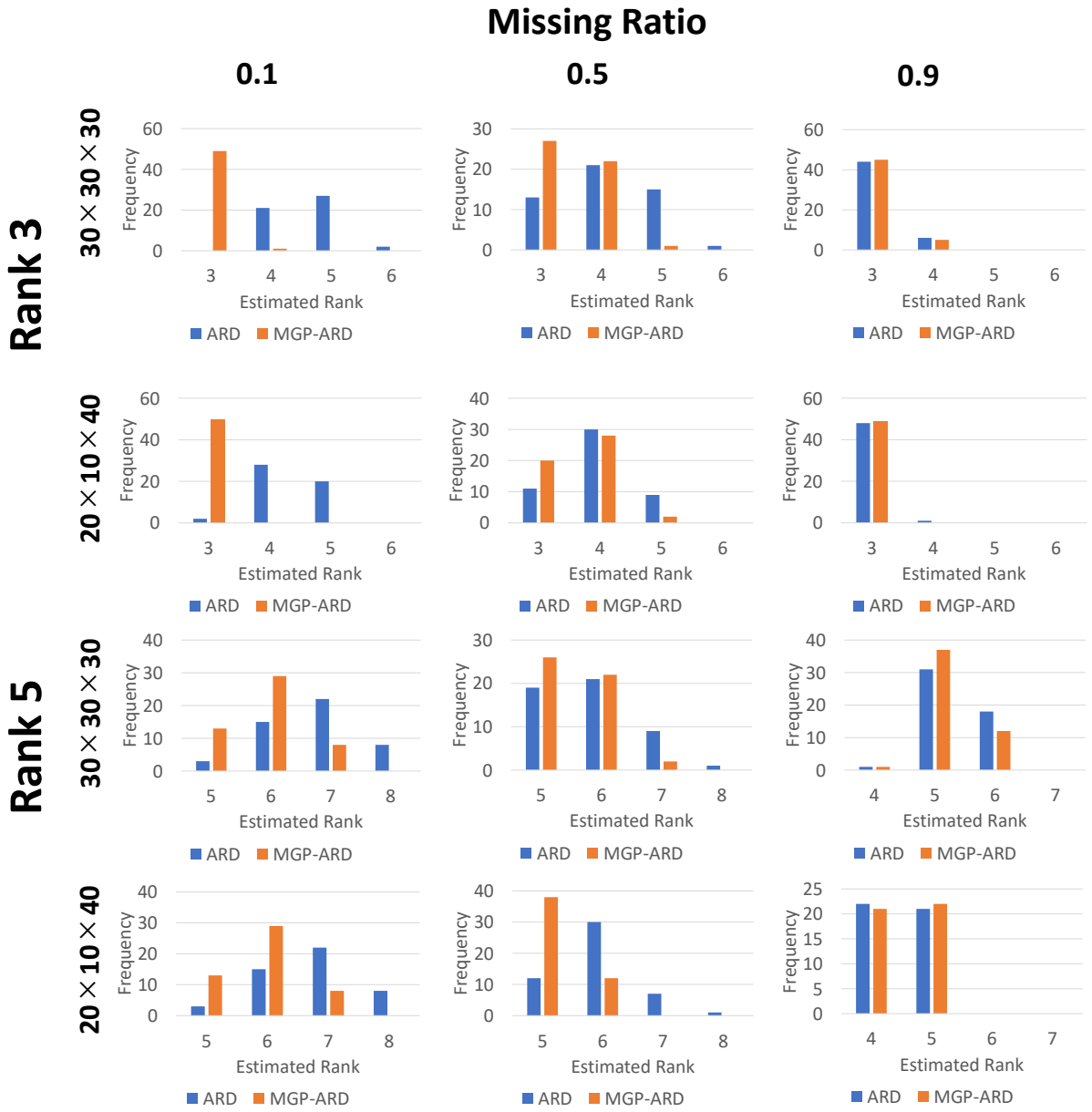


Figure 2.5 Rank estimation results for tensor completion using MGP (original method) and MGP-ARD (proposed method). The top two rows show the results when the true rank is 3, and the bottom two rows show the results when the true rank is 5. The loss rate is 0.1, 0.5, and 0.9, and the SNR is 20 [dB]. 50 trials were performed for each pattern.

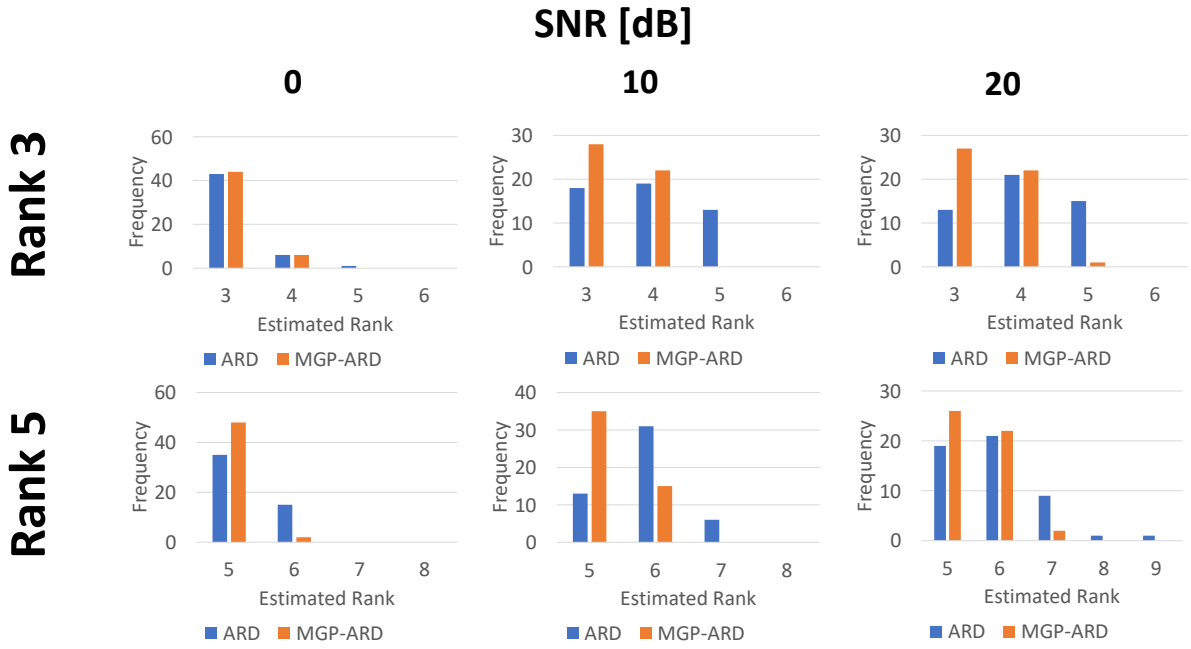


Figure 2.6 Rank estimation results for tensor completion using MGP (original method) and MGP-ARD (proposed method). The top row shows the results when the true rank is 3, and the bottom row shows the results when the true rank is 5. The size of the tensor is $30 \times 30 \times 30$, and the missing rate is 0.5, and the SNR is 0, 10, 20 [dB]. 50 trials were performed for each setting.

method has a minor change in the distribution in response to noise than the conventional method.

The actual improvement in duplication is shown in Figure 2.7. This figure shows the factor matrices finally obtained by ARD and MGP-ARD when the true rank is 5, the observation rate is 0.5, and the SNR is 20 [dB]. In ARD, the number of components in the obtained factor matrix is 8, so the rank estimation result is 8. On the other hand, the number of components in the factor matrix obtained by MGP-ARD is 5, which results in a rank estimation result of 5. Since the true rank is 5, we can see that MGP-

Table 2.1 Mean and median of the estimation error (RSE) in tensor completion. True ranks are 3 and 5, sizes are $30 \times 30 \times 30$ and $20 \times 10 \times 40$, and missing rates are 0.1, 0.5, and 0.9. Each pattern consists of 50 trials.

True rank	Size	Missing rate	Mean		Median	
			ARD	MGP-ARD	ARD	MGP-ARD
3	$30 \times 30 \times 30$	0.1	0.0040	**0.0033**	0.0040	**0.0033**
		0.5	0.0115	**0.0108**	0.0115	**0.0107**
		0.9	0.0322	0.0322	0.0320	0.0320
	$20 \times 10 \times 40$	0.1	0.0065	**0.0054**	0.0064	**0.0054**
		0.5	0.0181	**0.0175**	0.0183	**0.0172**
		0.9	0.0674	**0.0672**	0.0564	**0.0563**
5	$30 \times 30 \times 30$	0.1	0.0049	**0.0046**	0.0049	**0.0047**
		0.5	0.0139	**0.0136**	0.0140	**0.0134**
		0.9	0.0522	**0.0517**	0.0437	**0.0435**
	$20 \times 10 \times 40$	0.1	0.0049	**0.0046**	0.0049	**0.0047**
		0.5	0.0077	**0.0073**	0.0077	**0.0073**
		0.9	0.4748	**0.4674**	0.5461	**0.5434**

Table 2.2 Mean and median of the estimation error (RSE) in tensor completion. True ranks are 3 and 5, sizes are $30 \times 30 \times 30$, and SNR are 0, 10, and 20 [dB]. Each pattern consists of 50 trials.

True rank	SNR [dB]	Mean		Median	
		ARD	MGP-ARD	ARD	MGP-ARD
3	0	0.0991	**0.0987**	0.1012	**0.1010**
	10	0.0341	**0.0327**	0.0337	**0.0324**
	20	0.0115	**0.0108**	0.0115	**0.0107**
5	0	0.1317	**0.1305**	0.1301	**0.1295**
	10	0.0423	**0.0415**	0.0419	**0.0412**
	20	0.0139	**0.0136**	0.0140	**0.0134**

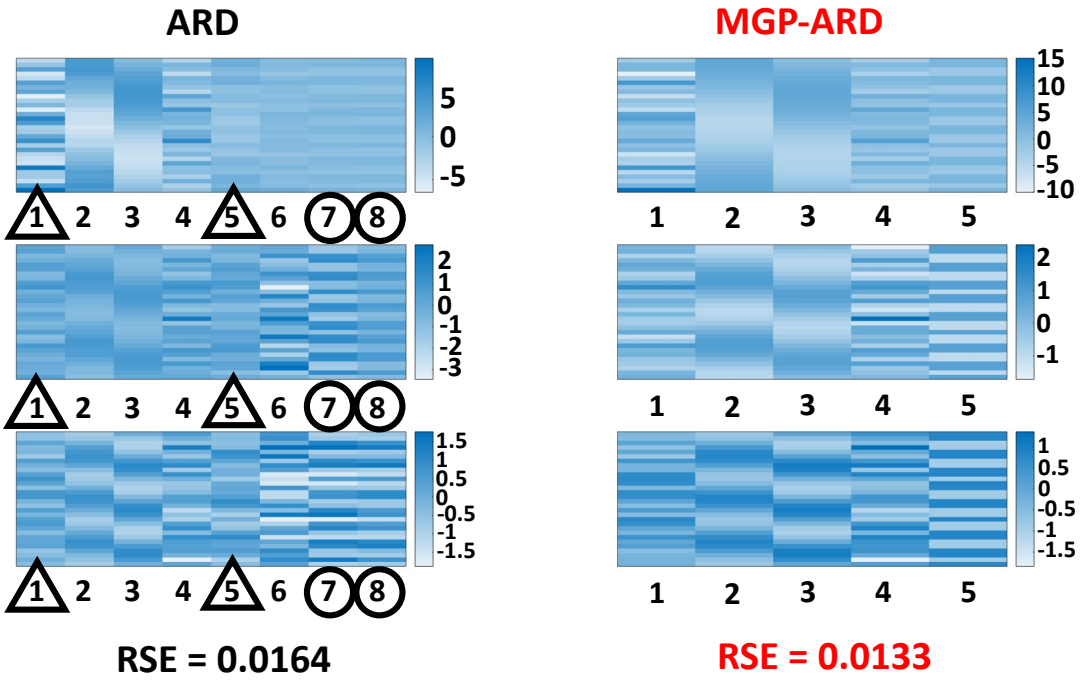


Figure 2.7 The improvement in the duplication of the column vectors. true rank is 5, observation rate is 0.5, SNR is 20 [dB]. In ARD, the 1st and 5th, 7th, and 8th components overlap, but MGP-ARD improves this and correctly estimates the true rank of 5.

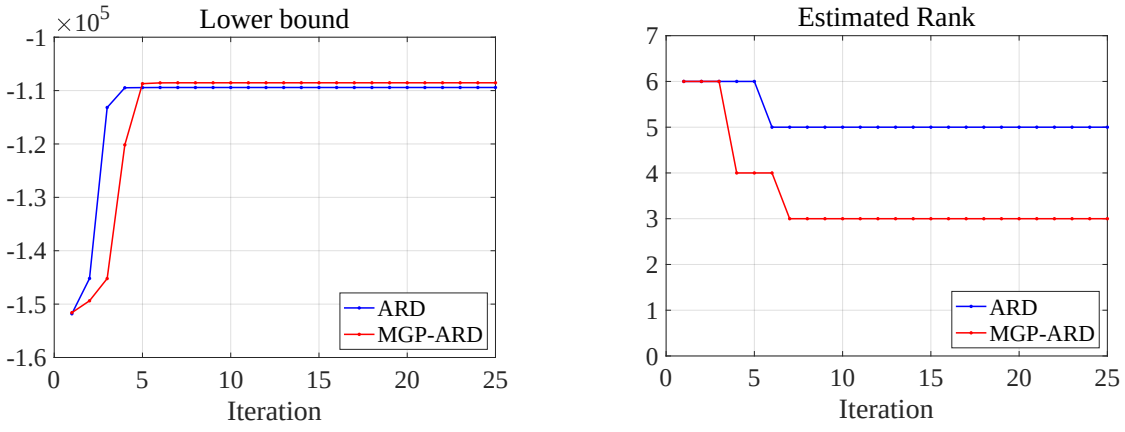


Figure 2.8 The figure shows the convergence of ARD and MGP-ARD. The left figure shows the variational lower bound and the right figure shows the process of rank estimation. The size of the tensor is $30 \times 30 \times 30$, true rank is 3, observation rate is 0.5, SNR $= 20$ [dB].

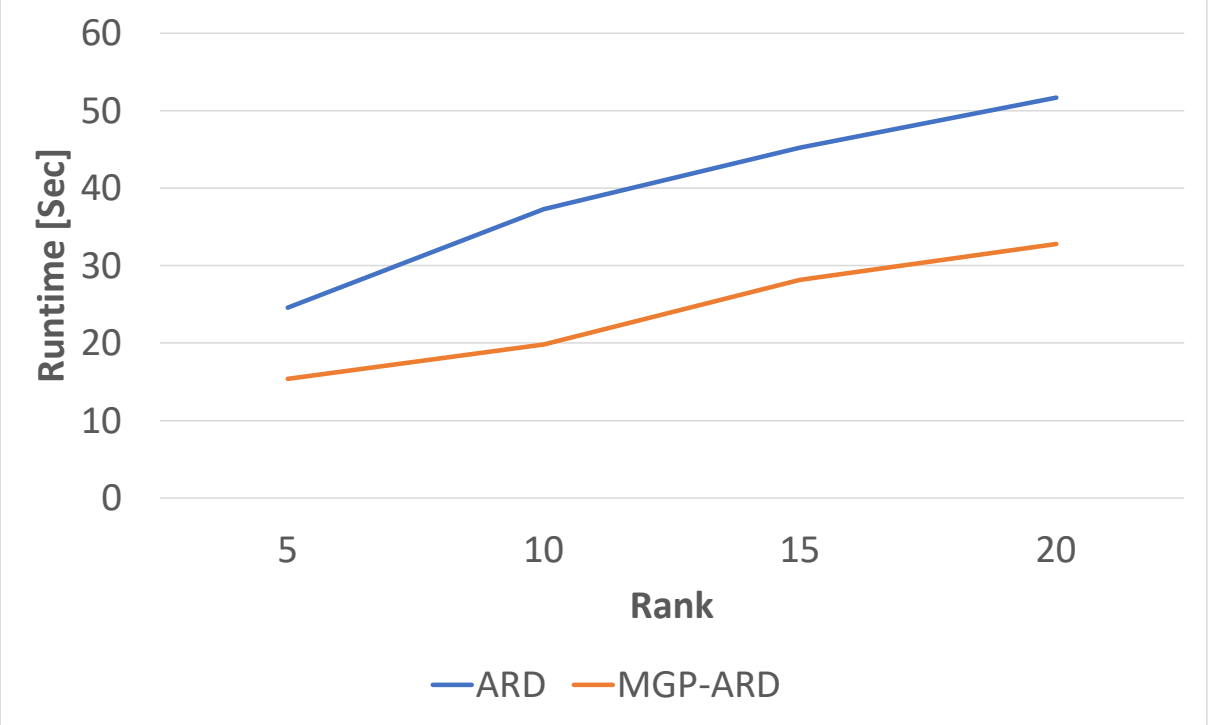


Figure 2.9 The average runtime (seconds) in tensor completion. True ranks are 5, 10, 15, and 20, size is $30 \times 30 \times 30$, SNR is 20 [dB], and missing rate is 0.5. Each setting consists of 50 trials.

ARD more accurately estimates the rank, while ARD overestimates 3. This is because of the duplication in the column vectors of the factor matrix of ARD, and MGP-ARD has improved this. Furthermore, the RSE of MGP-ARD is lower than that of ARD, suggesting that sensitivity to noise is avoided by not overestimating the ranks.

Figure 2.8 shows the convergence of the ARD and MGP-ARD algorithms, i.e. the variational lower bound and rank. The size of the tensor is $30 \times 30 \times 30$, true rank is 3, observation rate is 0.5, SNR is 20 [dB]. From Figure 2.8, the convergence of MGP-ARD was slightly slower than ARD, but the variational lower bound of MGP-ARD was better than that of ARD finally. In addition, the estimated rank of MGP-ARD was significantly decreased in early stage of the iterations.

Figure 2.9 shows estimation time against the rank values in both methods for ARD and MGP-ARD. True ranks are 5, 10, 15, and 20, size is $30 \times 30 \times 30$, SNR is 20 [dB], and missing rate is 0.5. It can be seen that MGP-ARD is faster than ARD in all ranks. The gap of computation times between ARD and MGP-ARD becomes larger as the true rank increases. This is because the computational complexity of both methods is dominated by $O(R^3)$.

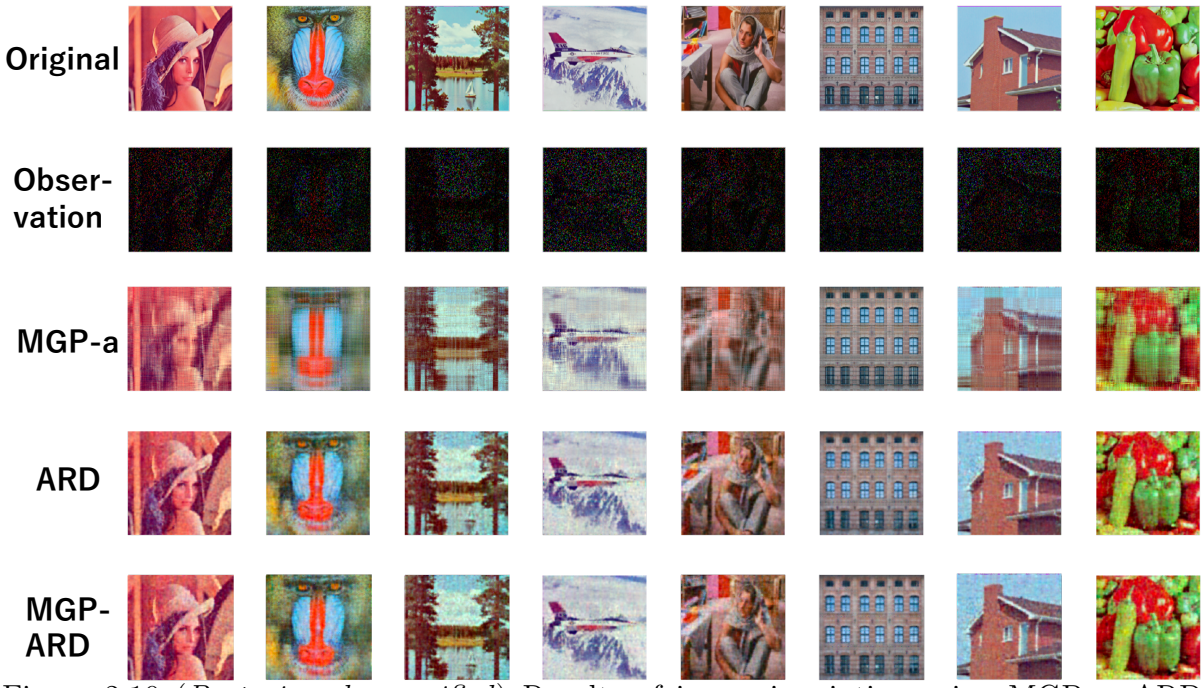


Figure 2.10 (*Best viewed magnified*) Results of image inpainting using MGP-a, ARD, and MGP-ARD. There are 8 types of experimental images with an uniform area missing. The 1st row is the original complete image, the 2nd row is the image with noise added (SNR = 20 [dB]) plus 90% missing. The 3rd and subsequent rows are the complementary results of each method.

2.4.2 Experiments with image data (Rank is unknown)

We experiment with MGP-ARD using real data as well as artificial data. In this experiment, we try to recover the missing image with noise for image inpainting. Since the true rank is not yet known, we mainly evaluate the estimation accuracy and time.

First, we use 8 types of uniform missing images. The observation rate is 0.1 (90% missing) and the SNR is 10 [dB]. The original image and the image with further missing data that contains noise are shown in Figure 2.10. We experiment with image inpainting using MGP-a [84], ARD and MGP-ARD. The convergence threshold ϵ is 1.0×10^{-4}, and the estimation accuracy is RSE $= \frac{\|\hat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}\|_F}{\|\boldsymbol{\mathcal{X}}\|_F}$. We also use PSNR and SSIM as other assessment measures. The initial value of R is set to 100.

The completion results are shown in Figure 2.10. The recovery performance (RSE, PSNR, SSIM) and runtime are also summarized in Table 2.3. Table 2.3, MGP-ARD is the fastest while maintaining a high performance among the three methods. MGP-ARD is only slightly less accurate than ARD, but not enough to be distinguished by the naked eye, referring to the image inpainting results in Figure 2.10. In terms of estimation time, we achieved completion in about half the estimation time of ARD. This can be attributed to the fact that the computation time of MGP-ARD is proportional to R^3, and the improvement of redundancy reduces R, which in turn reduces the computation time.

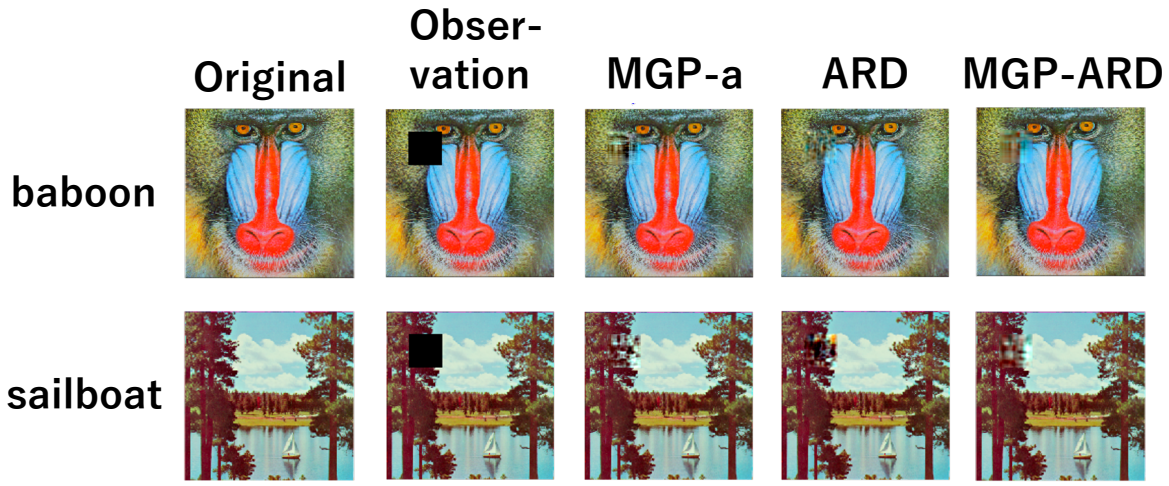


Figure 2.11 Results of image inpainting using MGP-a, ARD, and MGP-ARD. There are 2 types of experimental images with a specific area missing. The 1st column is the original complete image, the 2nd column is the image with large missing upper left part. The 3rd and subsequent rows are the complementary results of each method.

Next, we have also experimented with images in which certain areas are completely missing. We use two types of images (baboon, sailboat) in which the upper left corner is largely missing in a rectangular shape. The original image and the image with further missing data are shown in Figure 2.11. The algorithm, the convergence threshold, initial R, and the metrics are the same as for the uniform missing images experiment.

The completion results are shown in Figure 2.11. The recovery performance (RSE, PSNR, SSIM) and runtime are also summarized in Table 2.4. Table 2.4 shows that MGP-ARD has the highest accuracy and the fastest estimation for both images. Figure 2.11 shows that MGP-ARD performed the smoothest completion due to its low rank property, which is considered to increase the completion accuracy.

In summary, we confirmed that MGP-ARD significantly reduces the estimation time while maintaining the same level of estimation accuracy compared to ARD in image inpainting.

2.4.3 Experiments with trafic data (Rank is unknown)

With Intelligent Transportation Systems (ITS) operation, the analysis of large-scale traffic data in urban centers is becoming more and more critical. In general, traffic data contains information about time, space, and individual attributes, and the number of data is enormous. Such high-dimensional data with multiple characteristics can be regarded as tensor data. The problem in analyzing such high-dimensional data is missing values due to hardware/software or communication network failures. In this experiment, we

Table 2.3 The recovery performance (RSE, PSNR, SSIM) and runtime (seconds) on uniform missing eight images. For methods that need to tune parameters, the runtime with the best tuning parameter.

	MGP-a	ARD	MGP-ARD
RSE	0.2031	**0.1481**	0.1485
PSNR	19.44	**22.06**	22.03
SSIM	0.4204	**0.6107**	0.6099
Runtime	111	177	**97**

Table 2.4 The averaged recovery performance (RSE, PSNR, SSIM) and runtime (seconds) on two types of images with specific areas missing. For methods that need to tune parameters, the runtime with the best tuning parameter.

	baboon			sailboat		
	MGP-a	ARD	MGP-ARD	MGP-a	ARD	MGP-ARD
RSE	0.3995	0.3532	**0.3447**	0.5937	0.7349	**0.5565**
PSNR	28.06	29.1266	**29.3396**	24.2892	22.4362	**24.8514**
SSIM	0.9602	0.9613	**0.9631**	0.9545	0.9579	**0.9599**
Runtime	393	335	**60**	395	142	**57**

use MGP-ARD to perform tensor completion on incomplete traffic data and estimate the missing values.

In this section, we conduct numerical experiments based on a traffic speed dataset collected in Guangzhou, China. This experiment is based on the work of Chen et al [91]. This data set is available at `https://doi.org/10.5281/zenodo.1205229`. This dataset consists of travel speeds observed at 10-minute intervals (144-time intervals per day) from 214 road segments over two months (61 days from August 1, 2016, to September 30, 2016). The speed data can be organized as a third-order tensor (road segment $\times$ day $\times$ time interval, $214 \times 61 \times 144$). Of the approximately 1.88 million data, about 1.29% are not observed or provided in raw data. To evaluate how well the method works in cases with more missing observations, we create an artificial version of the data where 70% of the entry is assumed missing. The algorithms used in the experiments are ARD and MGP-ARD. The convergence threshold ϵ is 1.0×10^{-4}, and the estimation accuracy is RSE $= \frac{\|\hat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}\|_F}{\|\boldsymbol{\mathcal{X}}\|_F}$. The initial value of R is set to 100.

The completion results are shown in Figure 2.12. The recovery performance (RSE) and runtime are also summarized in Table 2.5. Table 2.5 shows that better estimation can be achieved with a faster estimation time. Figure 2.12 also shows that MGP-ARD estimates better than ARD at the 1:00 p.m. time point on the first and third days, where the true value drops significantly.

In summary, we confirmed that MGP-ARD significantly reduces the estimation time

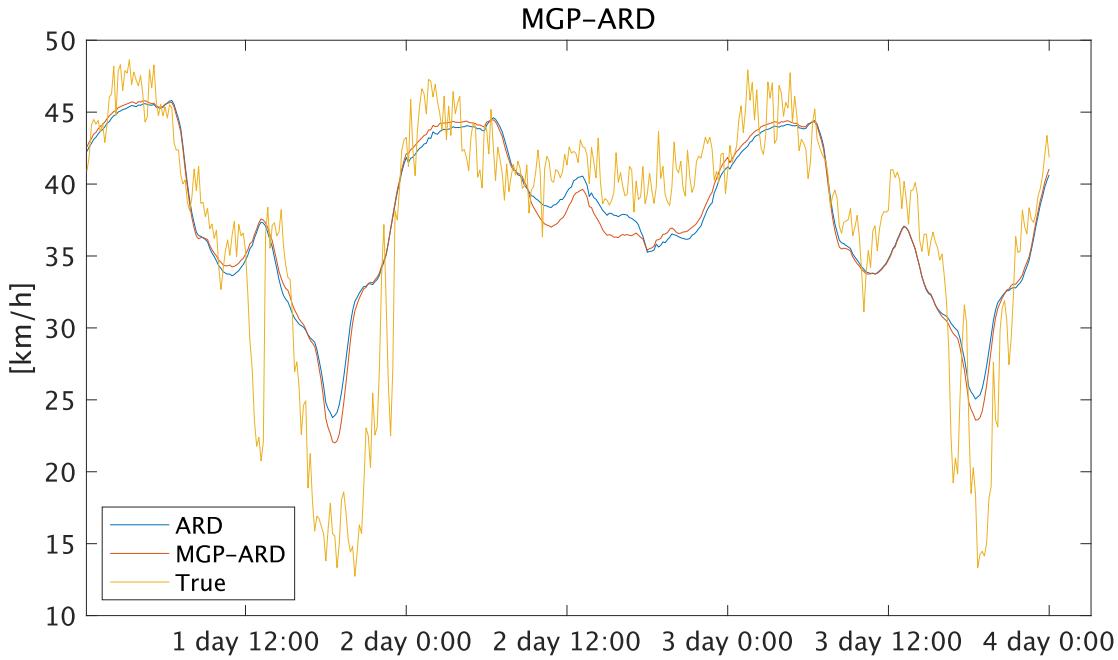


Figure 2.12 Completion results of traffic speed data [km/h]. The graph shows the ARD of the data with 70% of the elements missing, the completion result by MGP-ARD, and the data before the missing elements (True). The range of the data is three days.

and improves estimation accuracy compared to ARD in speed traffic data completion.

Table 2.5 The recovery performance (RSE) and runtime (seconds) on missing speed traffic data. For methods that need to tune parameters, the runtime with the best tuning parameter.

	ARD	MGP-ARD
RSE [km/h]	4.9047	**4.8377**
Runtime	477	**219**

2.4.4 Hyper-parameter sensitivities of rank estimation

In this section, we conduct numerical experiments on the dependence hyper-parameter sensitivities of rank estimation results in MGP-ARD and discuss the experimental results. The representative hyperparameter of MGP-ARD is e_0, which is related to the degree of degeneracy of the core tensor (see Equation (2.17)). When e_0 is large, the degree of degeneracy is small, and when e_0 is small, the degree of degeneracy is large. In other words, the results of rank estimation vary greatly depending on the value of the hyperparameters, and MGP-ARD may not be able to automatically estimate ranks from data, unlike ARD. Therefore, we experiment to confirm that MGP-ARD estimates rank not only based on the hyperparameters that determine the degree of shrinkage but also on information from

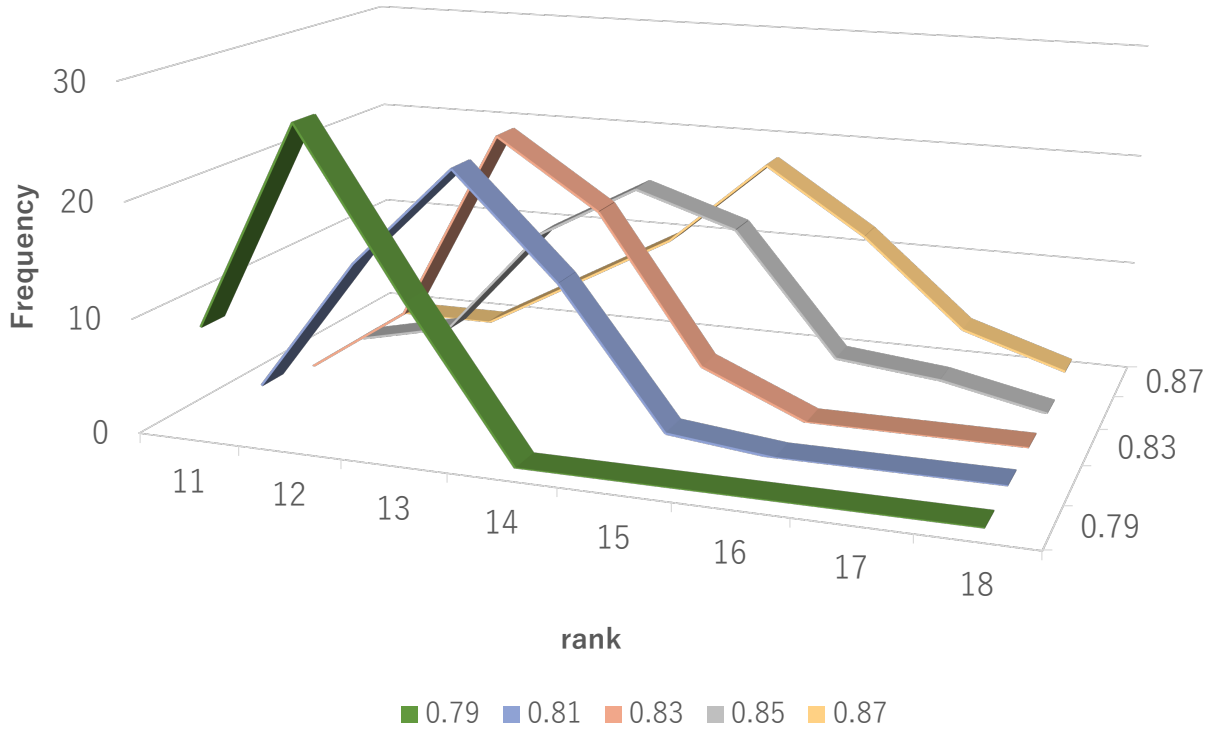


Figure 2.13 The result of rank estimation when the hyperparameters are varied (true rank is fixed at 12)

the data. In the experiment, we show the rank estimation results of MGP-ARD when only the hyperparameters are changed while the true rank is fixed. We use a third-order tensor of size $30 \times 30 \times 30$ and two true ranks of 12 and 15. The hyperparameters e_0 are 0.79, 0.81, 0.83, 0.85, and 0.87 for both, and 50 trials are made for each. SNR is 20 [dB], and the observation rate is 0.5. The initial value of R is twice the true rank.

Figure 2.13 and 2.14 show the estimation results when the true rank is 12 and 15. The rank estimated is prone to be larger as hyperparameters increase for both true ranks of 12 and 15. However, the rank estimated is larger when the true rank is 15 than when the true rank is 12, indicating that the rank is estimated at around 12 when the true rank is 12 and around 15 when the true rank is 15. In other words, MGP-ARD is not only influence of the hyperparameters, but is also influenced by the data.

Next, we conducted the experiment of rank estimation of MGP-ARD when the true rank is varied under fixed hyperparameter conditions. The data is a third-order tensor of size $30 \times 30 \times 30$, the SNR is 20 [dB], and the observation rate is 0.5. The hyperparameter e_0 is fixed at 0.8, and we confirm how the estimation results change when the true rank is changed from 10, 11, 12, 13, 14, to 15.

Figure 2.15 shows the experimental results. We confirm that as the true rank increases, the rank estimation results also increase (11, 12 $\rightarrow$ 13, 14). This indicates that information from the data also influenced the rank determination.

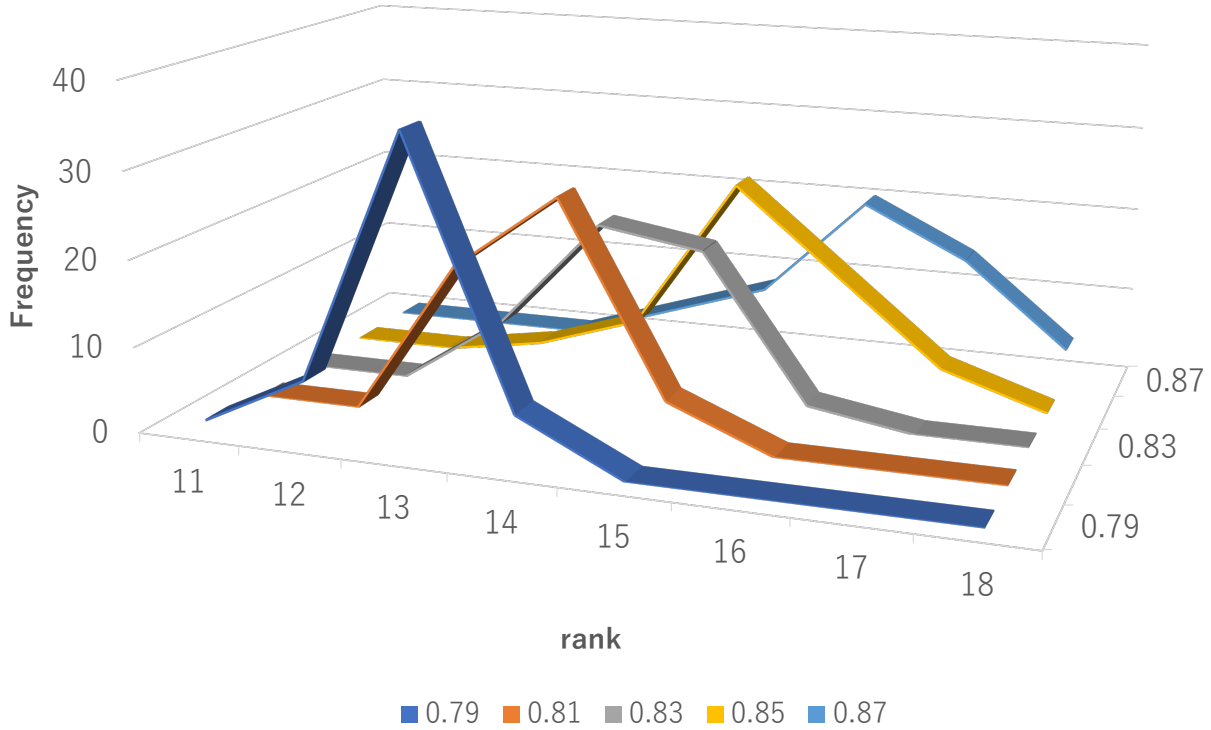


Figure 2.14 The result of rank estimation when the hyperparameters are changed. The color type indicates the true rank. (true rank is fixed at 15)

In summary, MGP-ARD is affected by hyperparameter but the information from the data absorbs the influence. For example, if the hyperparameter e_0 is set too large and the reduction is too strongly, the information from the data can suppress the excessive removal of rank.

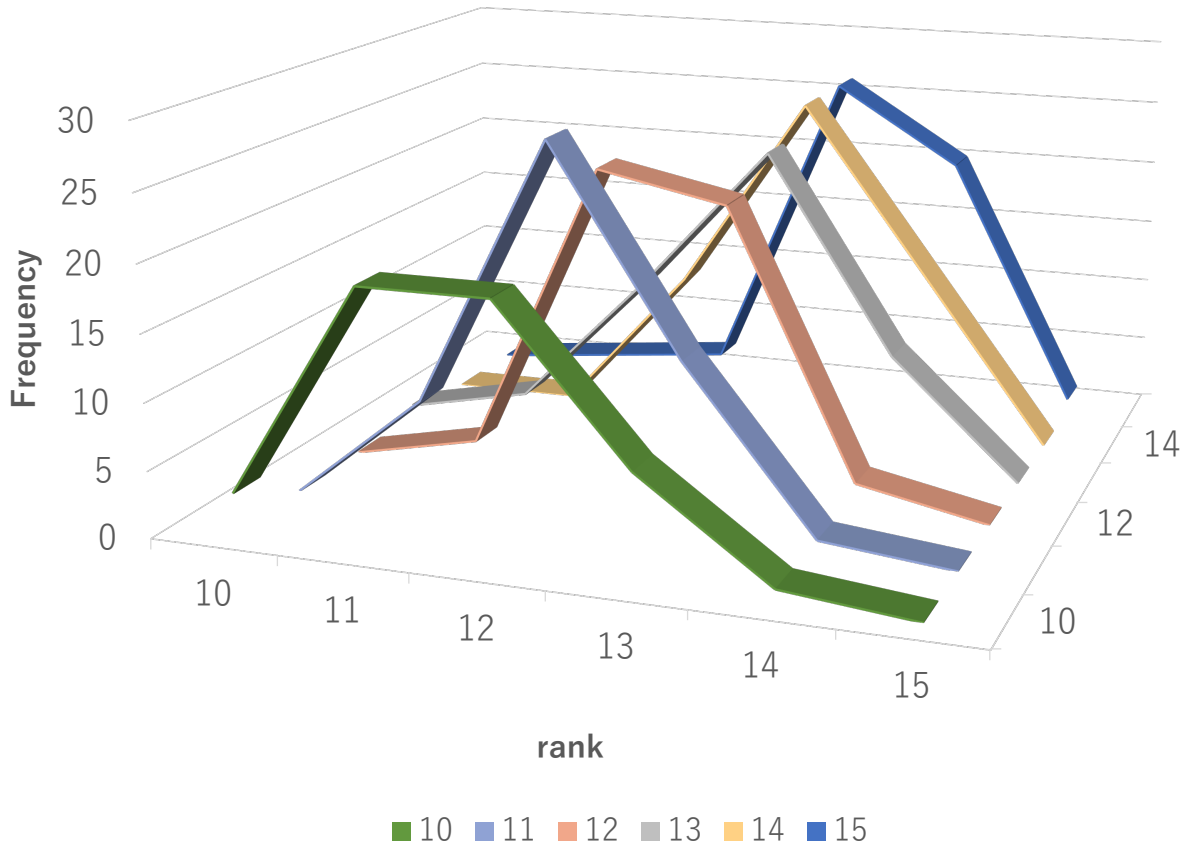


Figure 2.15 The result of rank estimation when varying the true rank (hyperparameters fixed at 0.8)

Chapter 3

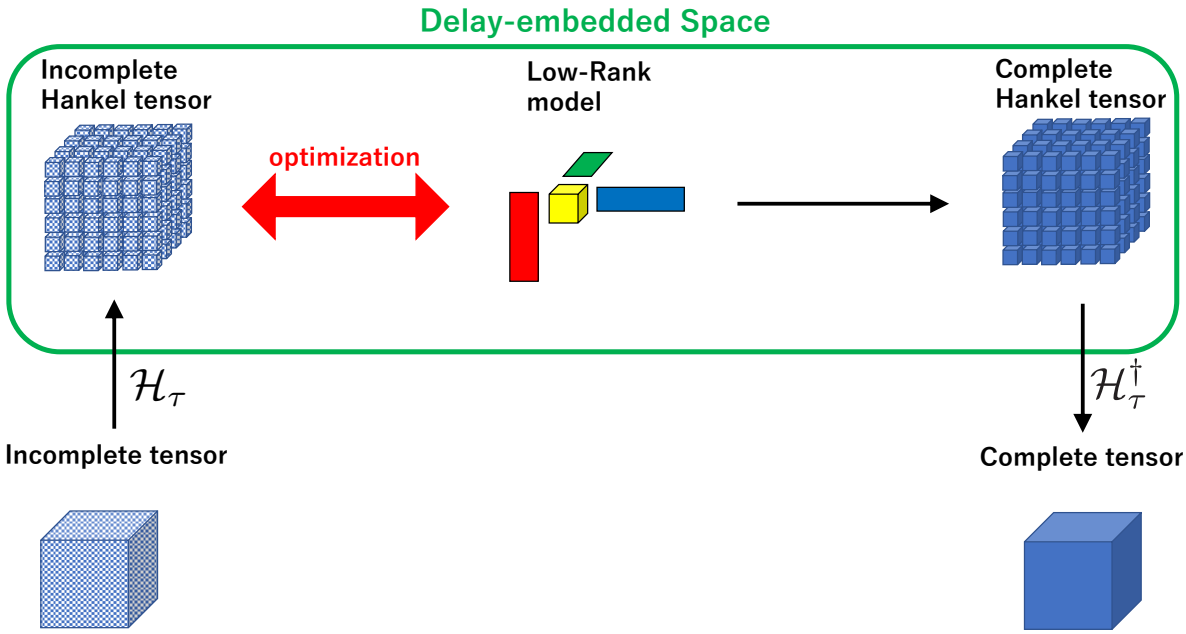
Tensor completion by Smooth Convolution Tensor Factorization (SCTF)

Recently, tensor completion for considering low-rank structures in delay-embedded spaces has attracted attention [79], [100]. A delay-embedded space is a high-dimensional space that represents time delay. In particular, the embedding of tensor data is called Multiway Delay-embedding Transform (MDT), which is mathematically equivalent to multi-level Hankelization. MDT has been widely applied to the tensor completion of images and videos [101], [102], [100], [17], [5], [79]. MDT-Tucker [79], the original model of tensor completion using MDT, consists of the following steps:

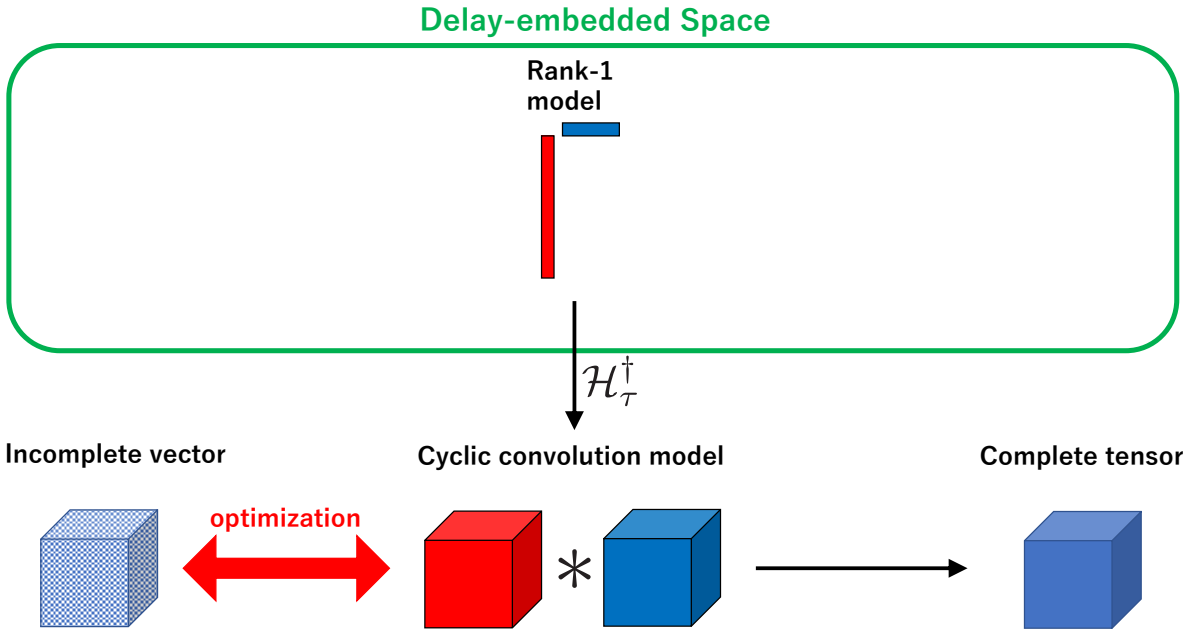
1. Hankelization of the observed tensor by MDT.
2. Completion of the Hankelized tensor using Tucker decomposition.
3. Inverse MDT of the completed tensor.

This method considers a delay-embedded space with a high expressive capability and exhibits higher completion accuracy than existing methods [2], [21], [52], [68], [16]. However, MDT-Tucker has the disadvantages of considerable time requirement and space computational complexity. For example, for an Nth-order tensor of average size T, if the delay window size is τ, the space complexity is $\mathcal{O}(\tau^N T^N)$ and the time complexity is $\mathcal{O}(\tau^{N+1} T^N)$; thus, the complexity increases exponentially with order.

In this chapter, we propose a novel smooth convolutional tensor factorization (SCTF) model, which decomposes a tensor into two smooth factor tensors by convolution instead of a product. Figure 3.1 shows a schematic of the algorithm. This model implicitly implements tensor decomposition in the delay-embedded space, whereas optimization is performed in the original space. The model is based on the relationship between the



(a) Existing method



(b) Proposed method

Figure 3.1 Comparison between the existing MDT-based and proposed convolution-based methods. (a) The existing method computes the optimization on the delay-embedded space. (b) Whereas, the proposed method computes the optimization in the original space, but implicitly considers the delay-emabedded space.

inverse MDT of the rank-1 model and the cyclic convolution of the factor tensors. In addition, because it is a rank 1 model, the SCTF is simpler than the MDT-Tucker model, which considers the Tucker decomposition model. These properties are expected to reduce computational complexity. In addition, a smoothness constraint was imposed on the factor tensors to further narrow the solution set. Our contributions can be summarized as follows:

- We have mathematically proven that tensor decomposition based on inverse MDT has sufficient representation ability in rank-1 decomposition.
- Based on the relationship of inverse MDT of rank 1 decomposition and cyclic convolution of factor tensors and the introduction of smooth prior structure into factor tensors, we proposed a new tensor completion model named smooth convolutional tensor factorization (SCTF).
- We derived a solution method of the proposed SCTF with the Majorization-Mini mization (MM) algorithm [60], [61], which is expected to provide a stable optimization in which the cost function decreases monotonically. Moreover, we exploit the equivalence of cyclic convolution in the time domain and Hadamard product in the frequency domain to reduce computation time.

The remainder of this chapter is organized as follows: related works in Section 3.1, a review of MDT in Section 3.2, the proposed method in Section 3.3, experiments using the proposed method in Section 3.4.

3.1 Related works

t-SVD [72] is a convolutional tensor decomposition method as well as the proposed method. It can achieve accurate tensor recovery based on group theory. t-SVD considers a new SVD for tensors by using some convolution, and the rank in t-SVD (tubal rank) is defined as the number of non-zero singular values. Since it is difficult to minimize the tubal rank directly, its convex relaxation is usually employed. The convex relaxation of tubal rank is given by the sum of singular values based on t-SVD, and it is called as the tensor nuclear norm (TNN). Low-rank approximation in the t-SVD is substituted for a problem of minimizing TNN, which has the advantage of incorporating a global structure. TNN [72] is a typical model for tensor completion problems based on t-SVD, and PSTNN [103] is a further developed model. PSTNN suppresses the excessively low rank of the estimated tensor by considering partial sums of only small singular values in the tensor nuclear norm. RTF [75] and UTF [76] have also been proposed as models that avoid the high computational cost of these t-SVD models. RTF considers a factorization model

of a low-rank tensor of small size and a dictionary (orthogonal) tensor. Since t-SVD is applied only to low-rank tensors of small size, the computational cost is lower than that of other t-SVD models. On the other hand, UTF uses the fact that the TNN is transformed to the minimum sum of the Frobenius norms of the two low-rank tensors so that the algorithm does not directly compute t-SVD in its calculation. UTF achieves very fast inference despite t-SVD model. However, these methods differ from the proposed method because it uses only the third-order tensor, and the convolution operation is performed only in the channel direction. Also, unlike the proposed model, these models do not have a smoothness term.

CNNM [104], [105] is a mathematical model of nuclear norm minimization of convolutional tensors applied to image completion and time series prediction. This research shows the equivalence of nuclear norm minimization of the convolution tensor and sparse approximation in Fourier space. However, the relationship between inverse MDT and cyclic convolution, and smoothness constraints is not discussed.

3.2 Review of MDT

This section summarizes the Multiway Delay-embedding Transform (MDT). Note that there are two types of MDT: noncyclic MDT [79] and cyclic MDT [5]. In this study, we consider a cyclic MDT. First, we discuss the Delay-embedding Transform (DT) for one-dimensional data (vectors), which is then extended to multidimensional data. Next, we introduce the overview of Multiway Delay-embedding Transform (MDT) and describe the outstanding points and drawbacks of MDT. Finally, Fast-MDT-Tucker [5], for avoiding drawbacks of MDT.

3.2.1 Delay-embedding Transform (DT)

Overview of DT

A delay-embedding Transform (DT) is the transformation of data into a high-dimensional space representing a time delay. In physics, DT has been studied by reconstructing dynamic attractors from time-series data in a delay-embedded space [106]. Mathematically, the DT converts a vector into a Hankel matrix (Hankelization) [107]. When embedding an observed signal from the original space into a high-dimensional space, it is assumed that the signal is represented by a low-rank and smooth manifold in the delay-embedded space [108], [109], [110]. Figure 3.2 shows the results of DT of the signal generated by the Lorenz system, indicating that the transformed signal is smooth and low-dimensional in the delay-embedded space. Based on this assumption, a low-rank approximation of the Hankel matrix is used in the data analysis [111], [107].

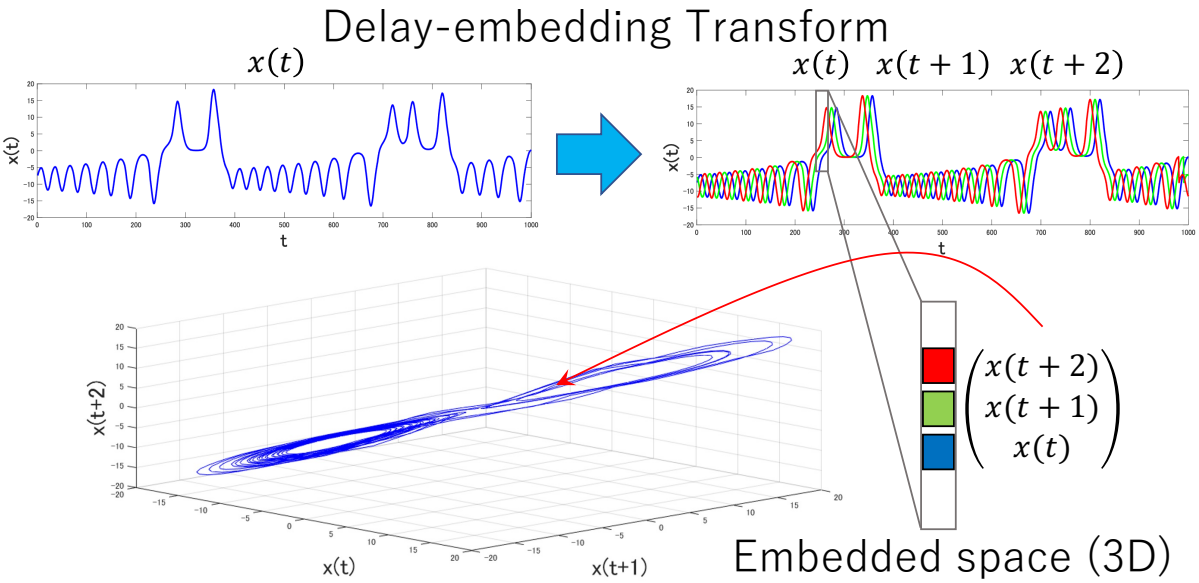


Figure 3.2 The figure shows the delay-embedded of signals generated from the Lorenz system. The embedded signal is smooth and low-dimensional in the delay-embedded space.

Mathematical operation of DT

Following [5], the DT for an observation vector $\boldsymbol{x} = (x_1, \ldots, x_T)^{\mathrm{T}} \in \mathbb{R}^T$ with a delay window size τ is defined as

$$\boldsymbol{X} \coloneqq \mathcal{H}_\tau(\boldsymbol{x}) = \begin{pmatrix} x_1 & x_2 & \cdots & x_{\tau-1} & x_\tau \\ x_2 & x_3 & \cdots & x_\tau & x_{\tau+1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{T-1} & x_T & \ldots & x_{\tau-3} & x_{\tau-2} \\ x_T & x_1 & \ldots & x_{\tau-2} & x_{\tau-1} \end{pmatrix} \in \mathbb{R}^{T\times\tau}, \tag{3.1}$$

where the DT operation is denoted as $\mathcal{H}_\tau$. Because $\boldsymbol{X}$ is a Hankel matrix, DT is also called Hankelization. Each row of $\boldsymbol{X}$ is identical to the local window of vector $\boldsymbol{x}$. Notably, this study assumes that the signal is cyclic. Figure 3.3a shows a concrete example of a DT operation.

The DT can be considered a linear operation. By using the duplication matrix $\boldsymbol{S} \in \mathbb{R}^{T\tau\times T}$, we obtain

$$\boldsymbol{S}(i,j) = \begin{cases} 1 & j = (((i-1) \bmod T) + \lfloor (i-1)/T \rfloor) \bmod T + 1 \\ 0 & \text{otherwise} \end{cases}. \tag{3.2}$$

DT can be given by

$$\mathcal{H}_\tau(\boldsymbol{x}) = \mathrm{fold}_{(T,\tau)}(\boldsymbol{S}\boldsymbol{x}), \tag{3.3}$$

where $\text{fold}_{(V,v)}$: $\mathbb{R}^{Vv} \to \mathbb{R}^{V\times v}$ is a folding operator, that is reshaping from a vector to a matrix.

The pseudo-inverse of the DT can be expressed as the average of the anti-diagonal entries. Considering a matrix $\boldsymbol{X} \in \mathbb{R}^{T\times\tau}$, the inverse DT operation $\mathcal{H}_\tau^\dagger$ can be given by

$$\mathcal{H}_\tau^\dagger(\boldsymbol{X}) = \boldsymbol{S}^\dagger \text{vec}(\boldsymbol{X}), \tag{3.4}$$

where $\boldsymbol{S}^\dagger \coloneqq (\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S})^{-1}\boldsymbol{S}^{\mathrm{T}}$ is a Moore-Penrose pseudoinverse of $\boldsymbol{S}$, and $\text{vec}(\cdot)$ represents an operation of vectorization. We note that $(\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S})^{-1} = \frac{1}{\tau}\boldsymbol{I}$. The tth element of the inverse DT is given by

$$\begin{aligned}[\mathcal{H}_\tau^\dagger(\boldsymbol{X})](t) &= [\boldsymbol{S}^\dagger \text{vec}(\boldsymbol{X})](t) \\ &= \frac{1}{\tau}\sum_{k=1}^{\tau} X((t-k \bmod T)+1, k).\end{aligned} \tag{3.5}$$

Figure 3.3 illustrates the DT and inverse DT matrix computations.

3.2.2 Multiway Delay-embedding Transform (MDT)

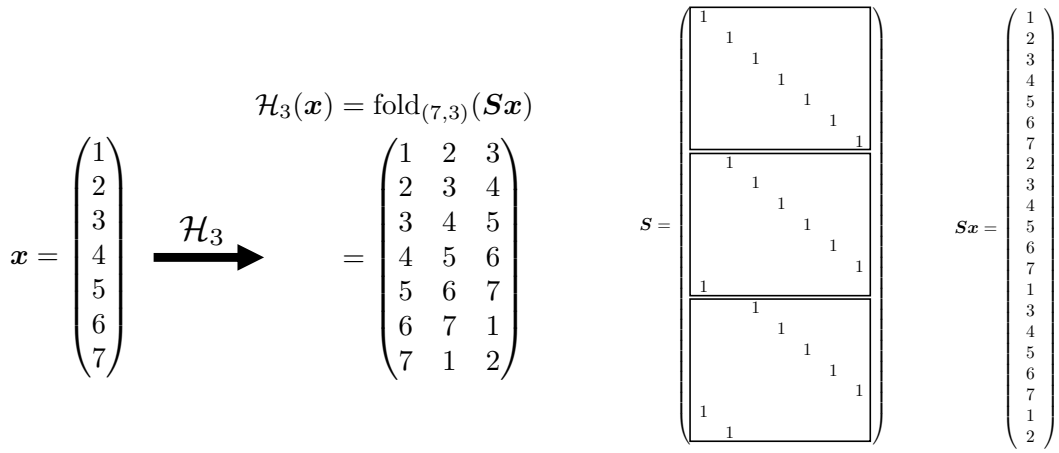
Overview of MDT

A Multiway Delay-embedding transform (MDT) is an Delay-embedding transform (DT) for tensor data of two or more orders. Tensor decomposition based on the MDT has various applications, particularly for tensor completion. A representative model is MDT-Tucker, which considers the Tucker decomposition model of the Hankel tensor and has been applied to image completion [79] and time-series data [112]. Another model, the HT-RPCA, was proposed in [113]. Unlike general RPCA, HT-RPCA solves the rank minimization of the tensor Hankelized by MDT, instead of the rank minimization of the matrix. This method enables anomaly detection by considering the time series. Furthermore, the TT and TR decomposition models of the Hankel tensor have been proposed and applied to image completion and time-series data [102], [100].

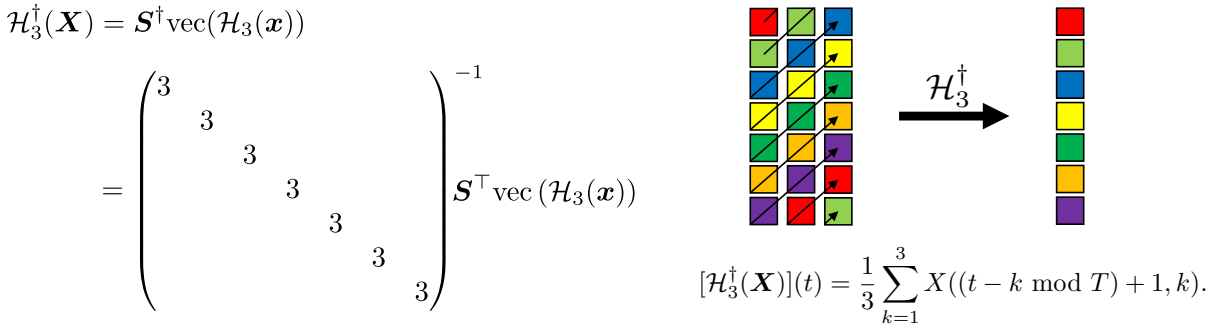
Mathematical operation of MDT

The DT can be naturally extended to an N-th order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{T_1\times\cdots\times T_N}$ of size $\boldsymbol{T} = (T_1, \ldots, T_N) \in \mathbb{R}^N$. Let us consider N duplication matrices $\boldsymbol{S}_n \in \{0,1\}^{T_n\tau_n\times T_n}$ $(n = 1, \ldots, N)$ with a window size $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_N) \in \mathbb{R}^N$ (see Equation (3.2)). The MDT is defined using an all-mode product and folding as follows:

$$\mathcal{H}_\tau(\boldsymbol{\mathcal{X}}) \coloneqq \text{fold}_{(\boldsymbol{T},\boldsymbol{\tau})}(\boldsymbol{\mathcal{X}} \times \{\boldsymbol{S}\}), \tag{3.6}$$



(a) DT operation



(b) inverse DT operation

Figure 3.3 Matrix computation of DT operation and inverse DT operation. In particular, the computation of the pseudo-inverse matrix in the inverse DT corresponds to the average of anti-diagonal elements of the matrix.

where $\mathrm{fold}_{(\boldsymbol{V},\boldsymbol{v})} : \mathbb{R}^{V_1 v_1 \times \cdots \times V_N v_N} \to \mathbb{R}^{V_1 \times v_1 \times \cdots \times V_N \times v_N}$ is the folding operator from an N-th order tensor to a $2N$-th order tensor. Conversely, the inverse MDT is defined as

$$\mathcal{H}_{\tau}^{\dagger}(\boldsymbol{\mathcal{X}}) \coloneqq \mathrm{unfold}_{(\boldsymbol{T},\boldsymbol{\tau})}(\boldsymbol{\mathcal{X}}) \times \{\boldsymbol{S}^{\dagger}\}, \tag{3.7}$$

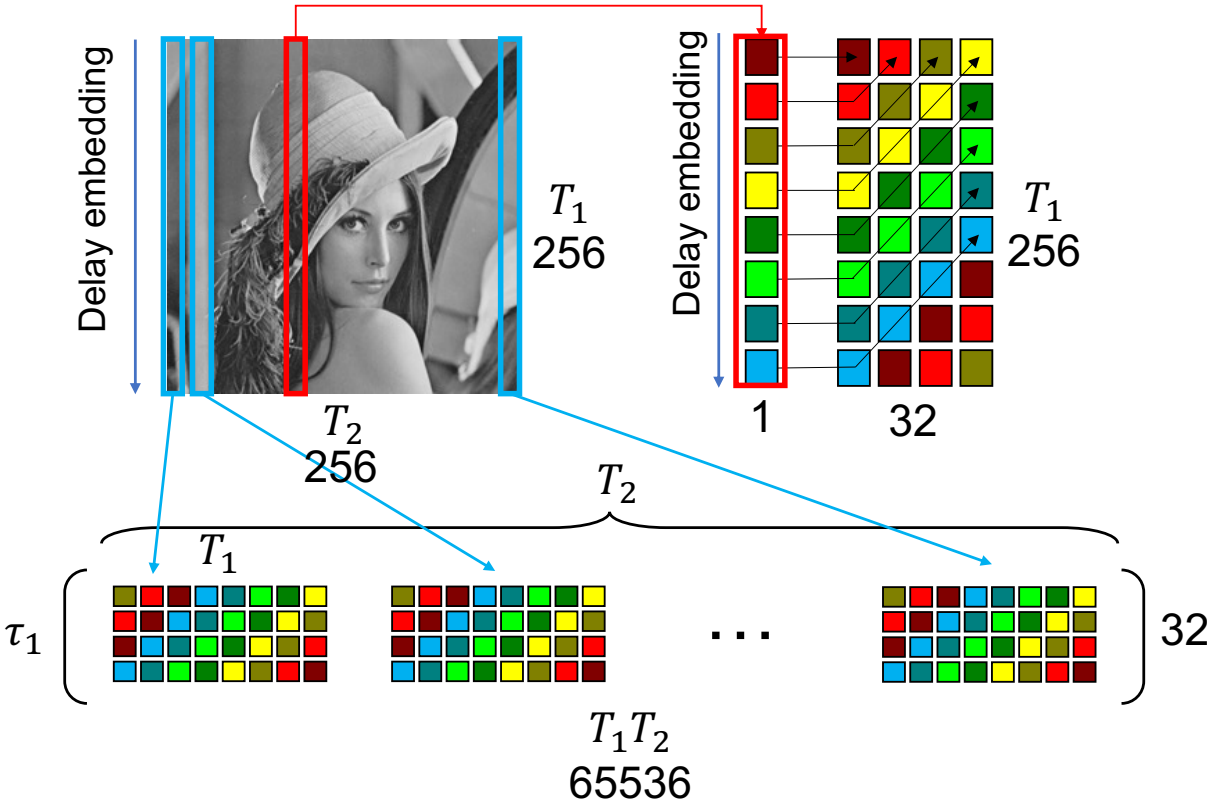
where $\mathrm{unfold}_{(\boldsymbol{V},\boldsymbol{v})} : \mathbb{R}^{V_1 \times v_1 \times \cdots \times V_N \times v_N} \to \mathbb{R}^{V_1 v_1 \times \cdots \times V_N v_N}$ is an unfolding operator from the $2N$-th order tensor of an N-th order tensor.

3.2.3 Relationship between MDT and similarity

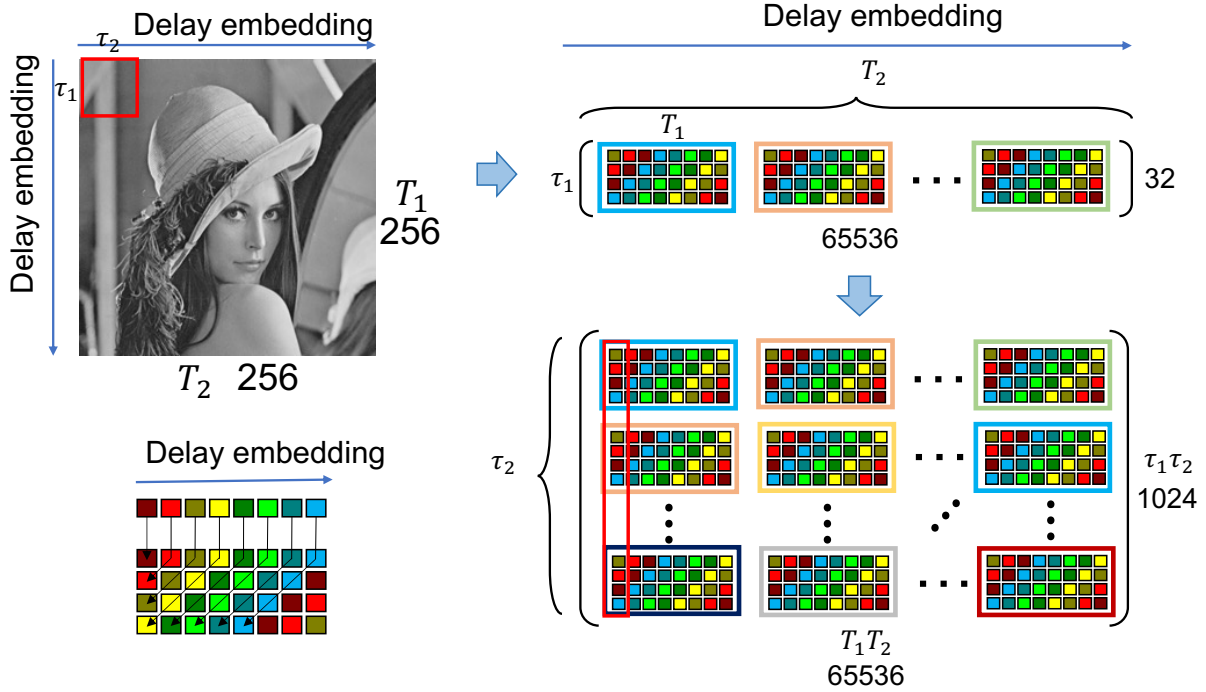
We discuss the relationship between MDT and non-local-similarity/self-similarity. Figure 3.4 explains the procedure of MDT using a graysclae image (2nd-order tensor) as an example. First, DT is performed on all columns of the image. Next, considering the matrix of DT created for each column as a single block, we can see that a vector was created with the block as a single element. DT again for that vector produces a block matrix, which is the result of the MDT of the 2nd-order tensor. As can be seen in Figure 3.4, the area of one column of the block matrix created by MDT corresponds to the area of the patch in the original image. That is, the *low-rankedness prior* on the delay-embedded space becomes another prior *patch similarity* in the original space. In image analysis, similarity based on patches, not neighborhoods, has been discussed in research as *nonlocal similarity*. In particular, *nonlocal similarity* has been studied as a denoising method, of which NL-means [114] and BM3D [115] are typical examples. These methods perform denoising by selecting a reference patch from an image, extracting similar patches by template matching, and averaging similar patches by weighted average. Low-rankness in the delay-emabedded space automatically performs reference and feature extraction simultaneously.

The proposed method is a completion problem not denoising, and this is where *self-similarity* is also important. Many images are constructed entirely from similar structures of the same patterns (lines, textures, etc.). Self-similarity is the repetition of a local pattern that constitutes a whole, called fractal in geometric properties [116]. Fractals occur frequently in many physical processes in nature, i.e., for example, in an image, similar lines and textures often appear repeatedly [117]. The proposed method is based on the belief that even if most of the structure is missing if the structure that appears locally remains, the whole can be completed.

There is also a study of low-rank patterns on delay-embedded space, which is for basic sine/cosine waves [118], [119]. For example, [118] is based on the fact that functions appearing in the world can be described on a Fourier basis and image data are represented by DCT, and [119] is based on the fact that time delays can be represented by multiplication in Z transform.



(a) MDT (1st-mode)



(b) MDT (2nd-mode)

Figure 3.4 This figure represents the process of MDT by grayscale image. It can be seen that the low-rankness in the delay-embedded space represents a patch of the image.

3.2.4 Fast-MDT-Tucker

The methods introduced in Section 3.2.2 have the disadvantage of considerable time requirement and space computational complexity because of the Hankelization in each mode of the tensor. Fast-MDT-Tucker [5] was proposed to improve the high computational complexity of MDT-Tucker. This method focuses on the redundant structure of the Hankel matrix and improves the time complexity to $\mathcal{O}(NT^N \log NT)$ and the space complexity to $\mathcal{O}(T^N)$ using two techniques:

1. Omission of duplicate computations.
2. Equivalence of cyclic convolution in the time domain and Hadamard product in the frequency domain.

In 2), the Fast-MDT-Tucker exploits the relationship between the inverse MDT and cyclic convolution. Fast-MDT-Tucker provides a fast and accurate completion; however, only low-rank priors are available.

The proposed method is also an algorithm based on the relationship between the inverse MDT and cyclic convolution and similarly avoids the issues of MDT. Note that the low-rank model of the proposed method is not a Tucker decomposition but a rank-1 decomposition. In addition, the proposed method imposes a smoothness constraint on the factor tensors.

3.3 Proposed Method

The proposed method solves the optimization problem by assuming that the observation tensor can be represented by a cyclic convolution of two smooth factors of the same size (See Figure 3.1). We describe the key theory behind the proposed method in Section 3.3.1, the smoothness constraints in Section 3.3.2, and the formulation and algorithm in Section 3.3.3.

3.3.1 Key theory of proposed method

Relationship between the inverse DT and cyclic convolution

Any rank-R matrix $\boldsymbol{X} \in \mathbb{R}^{T \times \tau}$ has a singular value decomposition that can be expressed as

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} = \sum_{r=1}^{R} \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^{\mathrm{T}}, \tag{3.8}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_R] \in \mathbb{R}^{T \times R}$ and $\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_R] \in \mathbb{R}^{\tau \times R}$ are orthonormal matrices and $\boldsymbol{\Sigma} = \mathrm{diag}(\sigma_1, \ldots, \sigma_R) \in \mathbb{R}^{R \times R}$ is a diagonal matrix. Because the inverse DT is a linear operation, $\mathcal{H}_\tau^\dagger(\boldsymbol{X})$ can be separated into rank-1 bases:

$$\mathcal{H}_\tau^\dagger(\boldsymbol{X}) = \sum_{r=1}^{R} \sigma_r \mathcal{H}_\tau^\dagger(\boldsymbol{u}_r \boldsymbol{v}_r^{\mathrm{T}}). \tag{3.9}$$

From Equations (3.5) and (3.9), the tth element of the inverse DT for a single basis $\boldsymbol{u}_r \boldsymbol{v}_r^{\mathrm{T}}$ is given by

$$[\mathcal{H}_\tau^\dagger(\boldsymbol{u}_r \boldsymbol{v}_r^{\mathrm{T}})](t) = \frac{1}{\tau} \sum_{k=1}^{\tau} u_r((t - k \bmod T) + 1) v_r(k). \tag{3.10}$$

Now, let us consider the matrix $\boldsymbol{P} = (\boldsymbol{I}_\tau \boldsymbol{O})^{\mathrm{T}} \in \mathbb{R}^{T \times \tau}$ and set $\boldsymbol{v} \in \mathbb{R}^\tau$ to be the same vector as the dimension of $\boldsymbol{u} \in \mathbb{R}^T$, i.e., zero padding operation is given by

$$\tilde{\boldsymbol{v}} \coloneqq \boldsymbol{P}\boldsymbol{v} = [v(1), v(2), \ldots v(\tau), \underbrace{0, \ldots, 0}_{T-\tau}]^{\mathrm{T}} \in \mathbb{R}^T. \tag{3.11}$$

Note that the sizes of $\boldsymbol{u}$ and $\tilde{\boldsymbol{v}}$ are equal and the elements of $\tilde{\boldsymbol{v}}$ are zero after the size of the delay window τ. From Equations (3.10) and (3.11), the inverse DT of the rank-1 basis $\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}$ is given by

$$\begin{aligned}
[\mathcal{H}_\tau^\dagger(\boldsymbol{u}_r \boldsymbol{v}_r^{\mathrm{T}})](t) &= \frac{1}{\tau} \sum_{k=1}^{T} u_r((t - k \bmod T) + 1) \tilde{v}_r(k) \\
&= \frac{1}{\tau} [\boldsymbol{u}_r * \tilde{\boldsymbol{v}}_r](t),
\end{aligned} \tag{3.12}$$

where $*$ denotes a cyclic convolution operation. From Equation (3.12), the inverse DT of the rank-1 basis can be formulated in terms of a cyclic convolution. Eventually, from Equations (3.9) and (3.12), the inverse DT of $\boldsymbol{X}$ is

$$\mathcal{H}_\tau^\dagger(\boldsymbol{X}) = \frac{1}{\tau} \sum_{r=1}^{R} \sigma_r \boldsymbol{u}_r * \tilde{\boldsymbol{v}}_r. \tag{3.13}$$

Sufficient representation ability even with a rank-1 matrix

We now discuss the rank-1 representation of $\boldsymbol{X}$. From Equation (3.13), the rank $\boldsymbol{X}$ denotes the number of convolutional bases. The degrees of freedom of each convolutional basis determine the representational ability of the model. In this study, we consider $\boldsymbol{X}$ to be rank-1 and show that it has sufficient representation ability for vector reconstruction. Rank-1 matrix model $\boldsymbol{X} = \boldsymbol{u}\boldsymbol{v}^{\mathrm{T}} \in \mathbb{R}^{T \times \tau}$ can generate any $\boldsymbol{x} \in \mathbb{R}^T$. Let us put

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{x} \end{bmatrix}, \quad \boldsymbol{v} = \begin{bmatrix} \tau \\ \boldsymbol{0}_{\tau-1} \end{bmatrix}, \tag{3.14}$$

where $\mathbf{0}_{\tau-1}$ is a $(\tau-1)$-dimensional vector of zeros and we have

$$\mathcal{H}_\tau^\dagger(\boldsymbol{u}\boldsymbol{v}^{\mathrm{T}}) = \mathcal{H}_\tau^\dagger\left(\left[\begin{array}{cc} \tau\boldsymbol{x} & \mathbf{0}_{T,\tau-1} \end{array}\right]\right) = \boldsymbol{x}. \tag{3.15}$$

This suggests that the inverse DT of a matrix, even rank-1, is over-parameterized and does not work as a model. Therefore, $\boldsymbol{u}$ and $\boldsymbol{v}$ must impose constraints to narrow the solution.

Extention to MDT

The properties of DT can be applied to the MDT. First, we show the relationship between the inverse MDT and N-dimensional cyclic convolution. Let us consider factor tensors $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{T_1\times\cdots\times T_n}$, $\boldsymbol{\mathcal{B}} \in \mathbb{R}^{\tau_1\times\cdots\times\tau_n}$, and we define $\boldsymbol{a} \coloneqq \mathrm{vec}(\boldsymbol{\mathcal{A}}) \in \mathbb{R}^{\prod_n T_n}$, $\boldsymbol{b} \coloneqq \mathrm{vec}(\boldsymbol{\mathcal{B}}) \in \mathbb{R}^{\prod_n \tau_n}$. We assume $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{T_1\times\tau_1\times\cdots\times T_N\times\tau_N}$ is given by

$$\begin{aligned} \mathrm{bunfold}_{(\boldsymbol{T},\boldsymbol{\tau})}(\boldsymbol{\mathcal{X}}) &= \mathrm{vec}(\boldsymbol{\mathcal{A}})\mathrm{vec}(\boldsymbol{\mathcal{B}})^{\mathrm{T}} \\ &= \boldsymbol{a}\boldsymbol{b}^{\mathrm{T}} \in \mathbb{R}^{\prod_n T_n\times\prod_n \tau_n}, \end{aligned} \tag{3.16}$$

where $\mathrm{bunfold}_{(\boldsymbol{V},\boldsymbol{v})} : \mathbb{R}^{V_1\times v_1\times\cdots\times V_N\times v_N} \to \mathbb{R}^{\prod_n V_n\times\prod_n v_n}$ is the unfolding operator from an $2N$-th order tensor to the block matrix. We also define $\mathrm{bfold}_{(\boldsymbol{V},\boldsymbol{v})} : \mathbb{R}^{\prod_n V_n\times\prod_n v_n} \to \mathbb{R}^{V_1\times v_1\times\cdots\times V_N\times v_N}$ as the inverse transform of $\mathrm{bunfold}_{(\boldsymbol{V},\boldsymbol{v})}$. Using the zero padding matrix $\boldsymbol{P}_n = (\boldsymbol{I}_{\tau_n}\ \boldsymbol{O})^{\mathrm{T}} \in \{0,1\}^{T_n\times\tau_n}$ $(n = 1,\ldots,N)$, we define a tensor $\tilde{\boldsymbol{\mathcal{B}}} = \boldsymbol{\mathcal{B}} \times \{\boldsymbol{P}\} \in \mathbb{R}^{T_1\times\cdots\times T_n}$ of the same size as $\boldsymbol{\mathcal{A}}$. The inverse MDT of $\boldsymbol{\mathcal{X}} = \mathrm{bfold}_{(\boldsymbol{\tau},\boldsymbol{T})}(\boldsymbol{a}\boldsymbol{b}^{\mathrm{T}})$ is derived by

$$\begin{aligned} &\left[\mathcal{H}_\tau^\dagger(\mathrm{bfold}_{(\boldsymbol{\tau},\boldsymbol{T})}(\boldsymbol{a}\boldsymbol{b}^{\mathrm{T}}))\right](t_1,\ldots,t_N) \\ &= \frac{1}{\prod_n \tau_n}\sum_{k_1=0}^{\tau_1-1}\cdots\sum_{k_N=0}^{\tau_N-1} \\ &\quad \boldsymbol{\mathcal{A}}(t_1 - k_1 \bmod T_1,\ldots,t_N - k_N \bmod T_N)\boldsymbol{\mathcal{B}}(k_1,\ldots,k_N) \\ &= \frac{1}{\prod_n \tau_n}\sum_{k_1=0}^{T_1-1}\cdots\sum_{k_N=0}^{T_N-1} \\ &\quad \boldsymbol{\mathcal{A}}(t_1 - k_1 \bmod T_1,\ldots,t_N - k_N \bmod T_N)\tilde{\boldsymbol{\mathcal{B}}}(k_1,\ldots,k_N) \\ &= \frac{1}{\prod_n \tau_n}\left[\boldsymbol{\mathcal{A}} * \tilde{\boldsymbol{\mathcal{B}}}\right](t_1,\ldots,t_N). \end{aligned} \tag{3.17}$$

Thus, the inverse MDT is represented by an N-dimensional cyclic convolution.

Furthermore, we show that the tensor which is folded from rank-1 matrix has sufficient representation ability. Rank-1 tensor model of $\boldsymbol{X} \coloneqq \mathrm{bunfold}_{(\boldsymbol{T},\boldsymbol{\tau})}(\boldsymbol{\mathcal{X}}) = \boldsymbol{a}\boldsymbol{b}^{\mathrm{T}} \in \mathbb{R}^{\prod_n T_n\times\prod_n \tau_n}$ can generate any $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{T_1\times\cdots\times T_n}$. Let us

$$\boldsymbol{a} = \left[\begin{array}{c} \mathrm{vec}(\boldsymbol{\mathcal{X}}) \end{array}\right], \quad \boldsymbol{b} = \left[\begin{array}{c} \prod_n \tau_n \\ \mathbf{0}_{\prod_n \tau_n - 1} \end{array}\right], \tag{3.18}$$

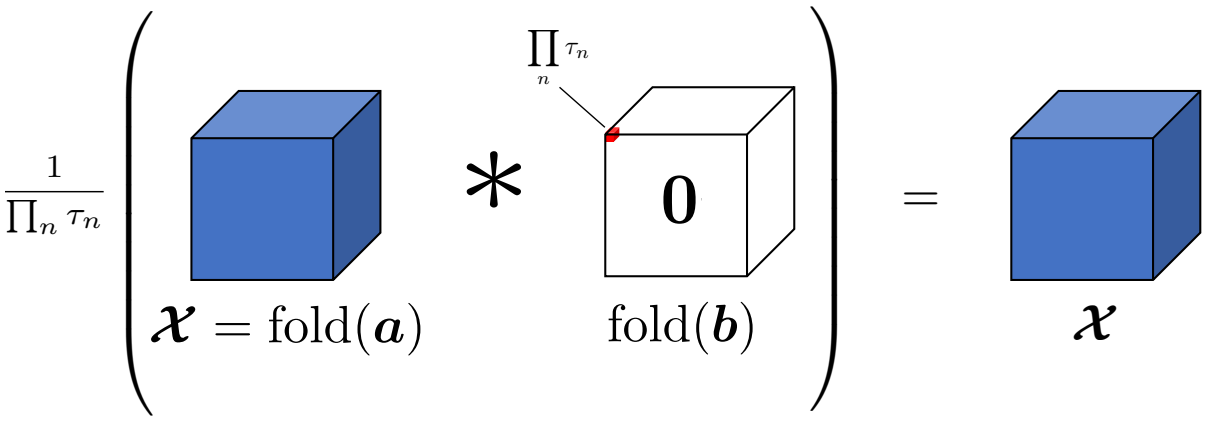


Figure 3.5 Cyclic convolution of a tensor $\mathcal{X}$ with only one element $\text{fold}(\boldsymbol{v})$ corresponds to the Equation (3.19). This operation is identically derived from the tensor $\mathcal{X}$.

where $\mathbf{0}_{\prod_n \tau_n - 1}$ is an $(\prod_n \tau_n - 1)$-dimensional vector of zeros, and then we have

$$\left[\mathcal{H}_{\tau}^{\dagger}(\text{bfold}_{(\boldsymbol{\tau},\boldsymbol{T})}(\boldsymbol{a}\boldsymbol{b}^{\mathrm{T}}))\right](t_1, \ldots, t_N) = \mathcal{X}. \tag{3.19}$$

Because this operation is equivalent to the cyclic convolution of a tensor $\mathcal{X}$ with only one element, the tensor $\mathcal{X}$ is derived identically (See Figure 3.5).

The inverse MDT of an unconstrained tensor, even rank-1, is over-parameterized and does not work as a model. In this study, additional constraints were imposed on $\mathcal{A}$ and $\mathcal{B}$ (see Section 3.3.2).

3.3.2 Smoothness constraints

Because the convolution of factor tensors can represent any tensor (even rank 1 models), it is necessary to impose constraints to narrow down the candidate solutions. In this study, smoothness is used as a constraint. The reasons for introducing smoothness as a constraint are as follows.

- As shown in Figure 3.2, the embedded data is represented by a smooth manifold on the delay embedded space.
- The data mainly targeted in our study are images, and there are many reports that smooth constraints are effective in image completion [20], [21], [16], [52].

Note that we do not introduce smoothness for the reconstructed tensor but the factor tensors. Unlike the model which smoothens the reconstructed tensor, the proposed model enables completion without excessive smoothing. We also set the scale adjustment terms for both $\mathcal{A}$ and $\mathcal{B}$ in the optimization equation to avoid smoothing by increasing only one factor of the tensors.

3.3.3 Optimization formulas and algorithm

Optimization formulas

In this thesis, we propose a new tensor completion model. We assume that the observed tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{T_1 \times \cdots \times T_N}$ is incomplete and that some entries have no values. The projection tensor $\boldsymbol{\mathcal{O}} \in \{0,1\}^{T_1 \times \cdots \times T_N}$ passes the observed entries and makes the missing entries equal to zero. The entries are given by

$$\mathcal{O}(t_1, \ldots, t_N) = \begin{cases} 1 & \mathcal{Y}(t_1, \ldots, t_N) \quad \text{is observed} \\ 0 & \text{otherwise} \end{cases}. \tag{3.20}$$

The problem involves obtaining the complete tensor $\boldsymbol{\mathcal{A}} * \boldsymbol{\mathcal{B}}$. In this study, we impose a smoothness constraint on $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$. The optimization problem is then given by

$$\begin{aligned} \min_{\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}} & \left\| \boldsymbol{\mathcal{O}} \circledast (\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{A}} * \tilde{\boldsymbol{\mathcal{B}}}) \right\|_F^2 \\ & + \sum_n \lambda_{A,n} \|\boldsymbol{\mathcal{L}}_n * \boldsymbol{\mathcal{A}}\|_F^2 + \sum_n \lambda_{B,n} \|\boldsymbol{\mathcal{L}}_n * \tilde{\boldsymbol{\mathcal{B}}}\|_F^2 \\ & + \eta_A \|\boldsymbol{\mathcal{A}}\|_F^2 + \eta_B \|\tilde{\boldsymbol{\mathcal{B}}}\|_F^2 \\ \text{s.t. } & \tilde{\boldsymbol{\mathcal{B}}} = \boldsymbol{\mathcal{B}} \times \{\boldsymbol{P}\}, \end{aligned} \tag{3.21}$$

where

$$\begin{aligned} \boldsymbol{\mathcal{L}}_n &\coloneqq \text{fold}_{\boldsymbol{T}}(\boldsymbol{l}_N \otimes \cdots \otimes \boldsymbol{l}_1) \in \mathbb{R}^{T_1 \times \cdots \times T_N} \\ & i = 1, \ldots, N \\ \boldsymbol{l}_i &\coloneqq \begin{cases} [1, -1, 0, \ldots, 0] & (i = n) \\ [1, 0, \ldots, 0] & (i \neq n) \end{cases} \in \mathbb{R}^{T_i} \end{aligned}$$

is a differential filter, and $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{T_1 \times \cdots \times T_N}, \boldsymbol{\mathcal{B}} \in \mathbb{R}^{\tau_1 \times \cdots \times \tau_N}$ are factor tensors and $\text{fold}_{\boldsymbol{V}} : \mathbb{R}^{V_1 V_2 \cdots V_N} \to \mathbb{R}^{V_1 \times V_2 \times \cdots \times V_N}$ is a folding operator from a vector to the N-th order tensor. Equation (3.21) evaluates the reconstruction loss in the first term. The second and third terms are smooth penalties for $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$, and the fourth and fifth terms adjust the scales of $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{B}}$. The equality constraint is for zero padding, based on Equation (3.11). Note that when $\tau = 1$, Equation (3.21) is equivalent to QV regularization. The relaxation of optimization problem (3.21) for an unconstrained optimization problem including a penalty term yields the following equation:

$$\begin{aligned}\min_{\boldsymbol{\mathcal{A}},\boldsymbol{\mathcal{B}}} L(\boldsymbol{\mathcal{A}},\boldsymbol{\mathcal{B}}) \coloneqq & \|\boldsymbol{\mathcal{O}} \circledast (\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{A}} * \boldsymbol{\mathcal{B}})\|_F^2 \\ & + \gamma \|\boldsymbol{\mathcal{I}}_\tau \circledast \boldsymbol{\mathcal{B}}\|_F^2 \\ & + \sum_n \lambda_{A,n} \|\boldsymbol{\mathcal{L}}_n * \boldsymbol{\mathcal{A}}\|_F^2 + \sum_n \lambda_{B,n} \|\boldsymbol{\mathcal{L}}_n * \boldsymbol{\mathcal{B}}\|_F^2 \\ & + \eta_A \|\boldsymbol{\mathcal{A}}\|_F^2 + \eta_B \|\boldsymbol{\mathcal{B}}\|_F^2, \end{aligned} \tag{3.22}$$

where $\boldsymbol{\mathcal{I}}_\tau \coloneqq \text{fold}_{\boldsymbol{\mathcal{T}}}(\boldsymbol{i}_{\tau_1} \otimes \ldots \otimes \boldsymbol{i}_{\tau_N}) \in \mathbb{R}^{T_1 \times \cdots \times T_N}$,
$\boldsymbol{i}_{\tau_n} \coloneqq [\underbrace{0, \ldots, 0}_{\tau_n}, \underbrace{1, \ldots, 1}_{T_n - \tau_n}] \in \mathbb{R}^{T_n}$. $\boldsymbol{i}_{\tau_n}$ serves as a penalty for $\boldsymbol{\mathcal{B}}$ and simulates zero padding $\{\boldsymbol{P}\}$. Note that we also redefine the size of $\boldsymbol{\mathcal{B}}$ as $T_1 \times \cdots \times T_N$.

Algorithm for solving optimization

In this study, we solved the optimization problem (3.22) using MM [60], [61]. The MM algorithm is an iterative method involving two steps.

1. Constructs a auxiliary function $h(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} | \boldsymbol{\mathcal{A}}^{(k)}, \boldsymbol{\mathcal{B}}^{(k)})$ for $L(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}})$ at $\boldsymbol{\mathcal{A}}^{(k)}, \boldsymbol{\mathcal{B}}^{(k)}$. Note,

$$\begin{aligned} \forall \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} \quad & L(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}) \leq h(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} | \boldsymbol{\mathcal{A}}^{(k)}, \boldsymbol{\mathcal{B}}^{(k)}) \\ & L(\boldsymbol{\mathcal{A}}^{(k)}, \boldsymbol{\mathcal{B}}^{(k)}) = h(\boldsymbol{\mathcal{A}}^{(k)}, \boldsymbol{\mathcal{B}}^{(k)} | \boldsymbol{\mathcal{A}}^{(k)}, \boldsymbol{\mathcal{B}}^{(k)}). \end{aligned}$$

2. Update as in

$$\boldsymbol{\mathcal{A}}^{(k+1)} \leftarrow \arg\min_{\boldsymbol{\mathcal{A}}} h(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}^{(k)} | \boldsymbol{\mathcal{A}}^{(k)}, \boldsymbol{\mathcal{B}}^{(k)}). \tag{3.23}$$

3. Update as in

$$\boldsymbol{\mathcal{B}}^{(k+1)} \leftarrow \arg\min_{\boldsymbol{\mathcal{B}}} h(\boldsymbol{\mathcal{A}}^{(k+1)}, \boldsymbol{\mathcal{B}} | \boldsymbol{\mathcal{A}}^{(k+1)}, \boldsymbol{\mathcal{B}}^{(k)}). \tag{3.24}$$

A conceptual diagram of the algorithm is shown in Figure 3.6. The MM algorithm was used because of convergence due to its monotonic convergence and ease of analytical computation.

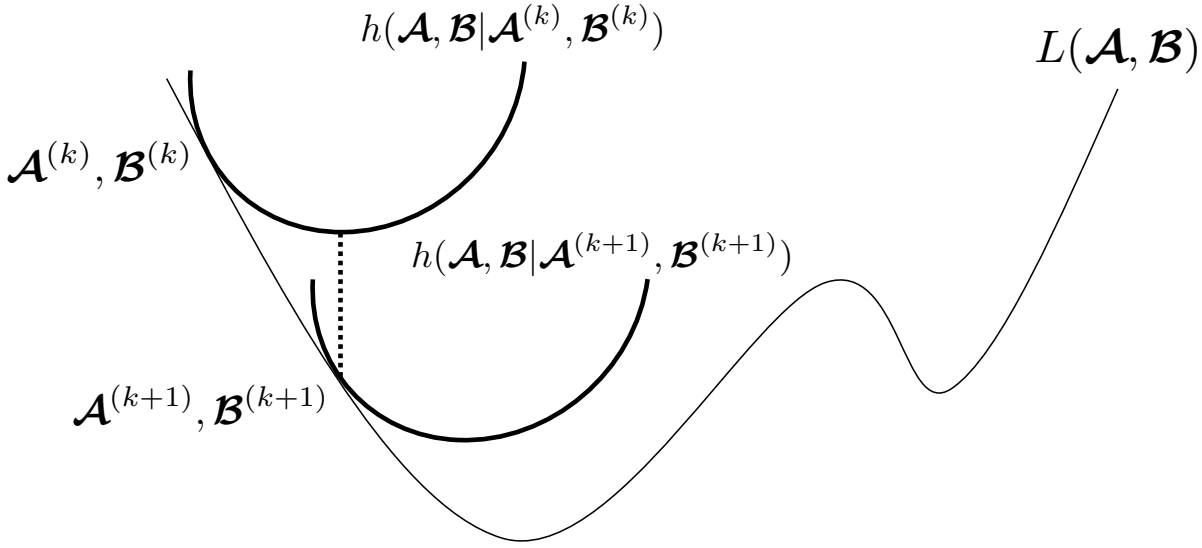


Figure 3.6 Concept of the two proposed models in our research.

The auxiliary function h is defined as follows:

$$\begin{aligned}
L(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}) &\leq h(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} | \boldsymbol{\mathcal{A}}^{(k)}, \boldsymbol{\mathcal{B}}^{(k)}) \\
&:= \|\boldsymbol{\mathcal{O}} \circledast (\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{A}} * \boldsymbol{\mathcal{B}})\|_F^2 + \gamma \|\boldsymbol{\mathcal{I}}_\tau \circledast \boldsymbol{\mathcal{B}}\|_F^2 \\
&\quad + \left\| \overline{\boldsymbol{\mathcal{O}}} \circledast (\boldsymbol{\mathcal{A}}^{(k)} * \boldsymbol{\mathcal{B}}^{(k)} - \boldsymbol{\mathcal{A}} * \boldsymbol{\mathcal{B}}) \right\|_F^2 \\
&\quad + \gamma \|\boldsymbol{\mathcal{I}}_{\overline{\tau}} \circledast (\boldsymbol{\mathcal{B}} - \boldsymbol{\mathcal{B}}^{(k)})\|_F^2 \\
&\quad + \sum_n \lambda_{A,n} \|\boldsymbol{\mathcal{L}}_n * \boldsymbol{\mathcal{A}}\|_F^2 + \sum_n \lambda_{B,n} \|\boldsymbol{\mathcal{L}}_n * \boldsymbol{\mathcal{B}}\|_F^2 \\
&\quad + \eta_A \|\boldsymbol{\mathcal{A}}\|_F^2 + \eta_B \|\boldsymbol{\mathcal{B}}\|_F^2 \\
&= \|\boldsymbol{\mathcal{Z}} - \boldsymbol{\mathcal{A}} * \boldsymbol{\mathcal{B}}\|_F^2 + \gamma \|\boldsymbol{\mathcal{W}} - \boldsymbol{\mathcal{B}}\|_F^2 \\
&\quad + \sum_n \lambda_{A,n} \|\boldsymbol{\mathcal{L}}_n * \boldsymbol{\mathcal{A}}\|_F^2 + \sum_n \lambda_{B,n} \|\boldsymbol{\mathcal{L}}_n * \boldsymbol{\mathcal{B}}\|_F^2 \\
&\quad + \eta_A \|\boldsymbol{\mathcal{A}}\|_F^2 + \eta_B \|\boldsymbol{\mathcal{B}}\|_F^2,
\end{aligned} \tag{3.25}$$

where

$$\boldsymbol{\mathcal{Z}} = \boldsymbol{\mathcal{O}} \circledast \boldsymbol{\mathcal{Y}} + \overline{\boldsymbol{\mathcal{O}}} \circledast \left(\boldsymbol{\mathcal{A}}^{(k)} * \boldsymbol{\mathcal{B}}^{(k)} \right), \tag{3.26}$$

and

$$\boldsymbol{\mathcal{W}} = \boldsymbol{\mathcal{I}}_{\overline{\tau}} \circledast \boldsymbol{\mathcal{B}}^{(k)}. \tag{3.27}$$

Furthermore, we exploit the property that the cyclic convolution of the time domain is a Hadamard product in the frequency domain to reduce the time complexity of the optimization problem. Because the Frobenius norm is invariant to the Fourier transform, the auxiliary function h is redefined as

$$\begin{aligned}
h(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}} | \boldsymbol{\mathcal{A}}^{(k)}, \boldsymbol{\mathcal{B}}^{(k)}) &= \hat{h}(\hat{\boldsymbol{\mathcal{A}}}, \hat{\boldsymbol{\mathcal{B}}} | \hat{\boldsymbol{\mathcal{A}}}^{(k)}, \hat{\boldsymbol{\mathcal{B}}}^{(k)}) \\
&:= \left\| \hat{\boldsymbol{\mathcal{Z}}} - \hat{\boldsymbol{\mathcal{A}}} \circledast \hat{\boldsymbol{\mathcal{B}}} \right\|_F^2 + \gamma \|\hat{\boldsymbol{\mathcal{W}}} - \hat{\boldsymbol{\mathcal{B}}}\|_F^2 \\
&\quad + \sum_n \lambda_{A,n} \|\hat{\boldsymbol{\mathcal{L}}}_n \circledast \hat{\boldsymbol{\mathcal{A}}}\|_F^2 \\
&\quad + \sum_n \lambda_{B,n} \|\hat{\boldsymbol{\mathcal{L}}}_n \circledast \hat{\boldsymbol{\mathcal{B}}}\|_F^2 \\
&\quad + \eta_A \|\hat{\boldsymbol{\mathcal{A}}}\|_F^2 + \eta_B \|\hat{\boldsymbol{\mathcal{B}}}\|_F^2,
\end{aligned} \tag{3.28}$$

where $\hat{\boldsymbol{\mathcal{L}}}_n, \hat{\boldsymbol{\mathcal{A}}}, \hat{\boldsymbol{\mathcal{B}}}, \hat{\boldsymbol{\mathcal{Z}}}, \hat{\boldsymbol{\mathcal{W}}}$ are the Fourier transform of $\boldsymbol{\mathcal{L}}_n, \boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}}, \boldsymbol{\mathcal{Z}}, \boldsymbol{\mathcal{W}}$.

The final auxiliary function is $\hat{h}$, which is minimized using Equations (3.23) and (3.24). To minimize $\hat{h}$, we derive $\hat{\boldsymbol{\mathcal{A}}}$ and $\hat{\boldsymbol{\mathcal{B}}}$ such that

$$\frac{\partial \hat{h}}{\partial \hat{\boldsymbol{\mathcal{A}}}} = \mathbf{0}, \qquad \frac{\partial \hat{h}}{\partial \hat{\boldsymbol{\mathcal{B}}}} = \mathbf{0}. \tag{3.29}$$

Algorithm 2 MM algorithm in the proposed method

Require: $\boldsymbol{\mathcal{Y}}$, $\boldsymbol{\mathcal{O}}$, $\boldsymbol{\mathcal{I}}_\tau$, γ, λ_A, λ_B, η_A, η_B, maxiter

$\hat{\boldsymbol{\mathcal{A}}} \leftarrow \mathrm{FFT}(\boldsymbol{\mathcal{A}})$
$\hat{\boldsymbol{\mathcal{B}}} \leftarrow \mathrm{FFT}(\boldsymbol{\mathcal{B}})$
for $i = 1$ to maxiter **do**
    Update $\boldsymbol{\mathcal{Z}}$ by (3.26)
    $\hat{\boldsymbol{\mathcal{Z}}} \leftarrow \mathrm{FFT}(\boldsymbol{\mathcal{Z}})$
    Update $\boldsymbol{\mathcal{W}}$ by (3.27)
    $\hat{\boldsymbol{\mathcal{W}}} \leftarrow \mathrm{FFT}(\boldsymbol{\mathcal{W}})$
    Update $\hat{\boldsymbol{\mathcal{A}}}$ by (3.30)
    Update $\hat{\boldsymbol{\mathcal{B}}}$ by (3.31)
    $\boldsymbol{\mathcal{A}} \leftarrow \mathrm{IFFT}(\hat{\boldsymbol{\mathcal{A}}})$
    $\boldsymbol{\mathcal{B}} \leftarrow \mathrm{IFFT}(\hat{\boldsymbol{\mathcal{B}}})$
    Calculate $L(\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{B}})$.
    if convergence of L **then** break
    end if
end for

Note that (3.29) is substituted by alternating the optimization with (3.23) and (3.24) because they are not satisfied simultaneously. Thus, at every optimization step, although the cost function L decreases monotonically, the auxiliary function h is not always optimal. After solving Equation (3.29), we obtain

$$\hat{\boldsymbol{\mathcal{A}}} = \left\{ \hat{\boldsymbol{\mathcal{Z}}}^* \circledast \hat{\boldsymbol{\mathcal{B}}} \right\} \oslash \left\{ \hat{\boldsymbol{\mathcal{B}}}^2 + \sum_n \lambda_{A,n} \hat{\boldsymbol{\mathcal{L}}}_n^2 + \eta_A \right\}, \tag{3.30}$$

$$\hat{\boldsymbol{\mathcal{B}}} = \left\{ \hat{\boldsymbol{\mathcal{Z}}}^* \circledast \hat{\boldsymbol{\mathcal{A}}} + \gamma \hat{\boldsymbol{\mathcal{W}}} \right\} \oslash \left\{ \hat{\boldsymbol{\mathcal{A}}}^2 + \sum_n \lambda_{B,n} \hat{\boldsymbol{\mathcal{L}}}_n^2 + \eta_B + \gamma \right\}, \tag{3.31}$$

where $\hat{\boldsymbol{\mathcal{Z}}}^*$ is the complex conjugate of $\hat{\boldsymbol{\mathcal{Z}}}$. In summary, Equations (3.26) and (3.27) correspond to Step 1 of the MM algorithm, and Equations (3.30) and (3.31) correspond to Step 2. Algorithm 1 summarizes the proposed method.

3.3.4 Computational complexity

The algorithm consists of updated Equations (3.26), (3.27), (3.30), and (3.31). The time complexity of (3.27), (3.30), and (3.31) is $\mathcal{O}(T^N)$, and that of (3.26) is $\mathcal{O}(NT^N \log T)$. Since (3.26) is derived from the cyclic convolution by $\boldsymbol{\mathcal{A}} * \boldsymbol{\mathcal{B}} = \mathrm{IFFT}(\mathrm{FFT}(\boldsymbol{\mathcal{A}}) \circledast \mathrm{FFT}(\boldsymbol{\mathcal{B}}))$, the time complexity $\mathcal{O}(NT^N \log T)$ of the FFT is dominant. Consequently, the overall time complexity of the update equation is $\mathcal{O}(NT^N \log T)$.

3.3.5 Extension to non-periodic signals

Since the proposed method uses FFT, we assume periodicity in the signal $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{T_1 \times \cdots \times T_N}$. Therefore, only when the first-order tensor (vector $\boldsymbol{y} \in \mathbb{R}^T$), the resulting observation signal may force the leading and trailing values to be equal, which may worsen the complete accuracy. We address this problem by concatenating zero vectors $\mathbf{0}_T \in \mathbb{R}^T$ of the same size in $\boldsymbol{y}$, as in

$$\boldsymbol{y}_{double} \coloneqq [\boldsymbol{y}, \mathbf{0}_T] \in \mathbb{R}^{2T}. \tag{3.32}$$

The projection vector $\boldsymbol{o}_\Omega \in \mathbb{R}^T$ ($\boldsymbol{\mathcal{O}}_\Omega \in \mathbb{R}^{T_1 \times \cdots \times T_N}$ in tensor) is also extended as in

$$\boldsymbol{o}_{\Omega,double} \coloneqq [\boldsymbol{o}_\Omega, \mathbf{0}_T] \in \mathbb{R}^{2T}. \tag{3.33}$$

That is, we extend any signal $\boldsymbol{y}$ to a periodic signal and set the extended portion as unobserved. Equations (3.32) and (3.33) are then reset as inputs to Algorithm 2 ($\boldsymbol{y} \leftarrow \boldsymbol{y}_{double}$, $\boldsymbol{o}_\Omega \leftarrow \boldsymbol{o}_{\Omega,double}$). The algorithm's estimation result $\boldsymbol{a} * \boldsymbol{b} \in \mathbb{R}^{2T}$ then uses only the values of the first half.

3.4 Experiment

The experiment was conducted in the following environments: CPU: Intel(R) Xeon(R) Silver 4214R CPU @ 2.40GHz, 12 cores, Memory: 512GByte, Software: Matlab R2021b.

3.4.1 Completion of clipped data

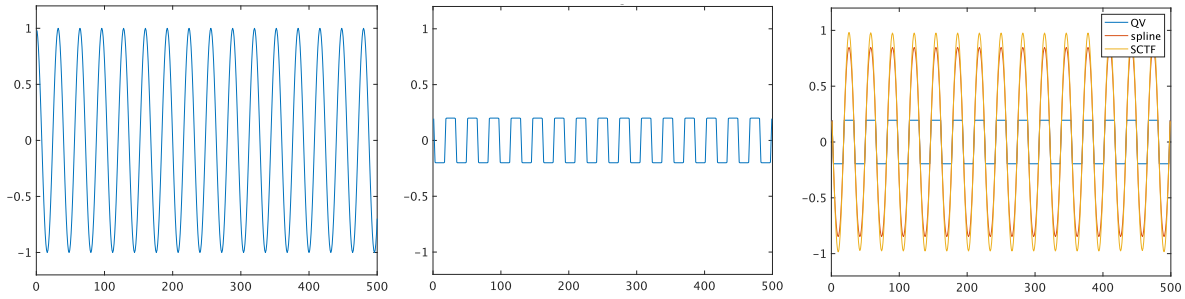
This section presents the evaluation of the proposed method (SCTF) for reconstructing clipped data (declipping) as a type of completion. Clipping is an operation that uses a certain clipping level $c > 0$ to replace entries above c and below $-c$ with c and $-c$. The clipping operation on the entry of a tensor is given by

$$\mathcal{X}(t_1, t_2, \ldots, t_N) = \min(c, \max(-c, \mathcal{X}_0(t_1, \ldots, t_N))). \tag{3.34}$$

The value range of the clipped data was $[-c, c]$.

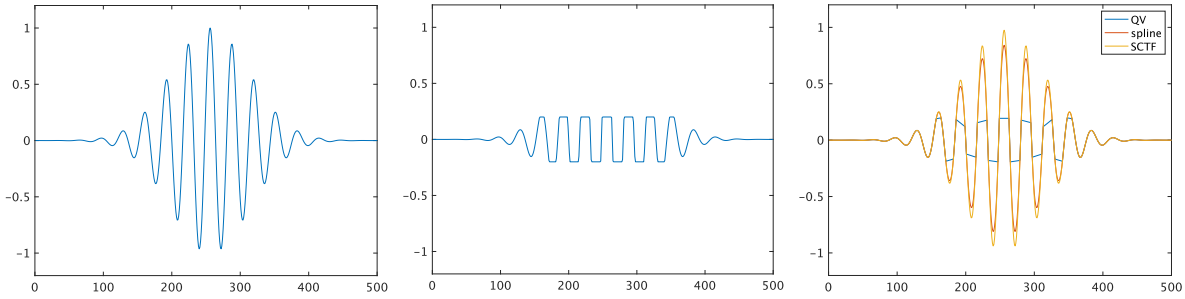
The indices of the clipped entries were recorded and treated as missing values for reconstruction. Thus, the set of observed entries can be expressed as

$$\mathcal{O}(t_1, \ldots, t_N) = \begin{cases} 1 & -c < \mathcal{X}(t_1, \ldots, t_N) < c \\ 0 & \text{otherwise} \end{cases}. \tag{3.35}$$



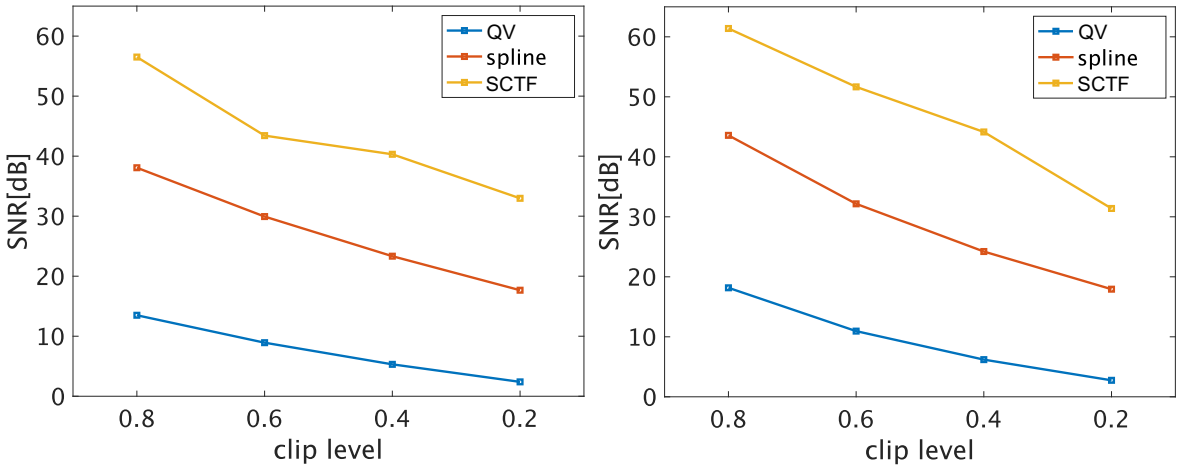
(a) Original signal (b) Clipped signal (c) De-clipped signals

Figure 3.7 Example of declipping experiment: (a) original signal of the sine function, (b) clipped signal with clipping level = 0.2, and (c) these reconstructed signals by using QV regularization, cubic spline interpolation, and SCTF



(a) Original signal (b) Clipped signal (c) De-clipped signals

Figure 3.8 Example of declipping experiment: (a) original signal of wavelet function, (b) clipped signal with clipping level = 0.2, and (c) these reconstructed signals by using QV regularization, cubic spline interpolation, and SCTF.



(a) sine (b) wavelet

Figure 3.9 Values of SNR in declipping experiments with various clipping levels.

Clipped signal completion (1st order tensor)

In this experiment, we evaluated SCTF using signal completion. The signals to be completed were the sine and wavelet functions with a clip level of 0.2, and both had a maximum amplitude of 1. Examples of clipping are presented in Figures 3.7 and 3.8. The original signals are shown in Figures 3.7a and 3.8a, and the signals after clipping are shown in Figures 3.7b and 3.8b. We compared SCTF with QV regularization and spline interpolation. In SCTF, we set $\tau_1 = 32$, $\gamma = 1.0 \times 10^8$, and $\lambda_{A,1} = \lambda_{B,1} = \eta_A = \eta_B = 0.1$.

Figures 3.7c and 3.8c show the signals reconstructed using QV regularization, spline interpolation, and SCTF. The signal completed by SCTF had a maximum amplitude of approximately 1, which was the best among the three methods. The spline method reconstructs the waveform signals; however, its amplitude is smaller than that of the original signal. However, the QV method failed to restore the signal.

Figure 3.9 shows the signal-to-noise ratio (SNR) values of the declipping experiment for various clipping levels. The clipping levels were 0.8, 0.6, 0.4, and 0.2, and the parameters (λ, τ) were adjusted for all levels. SCTF achieved significantly higher values of SNR than QV regularization and spline interpolation.

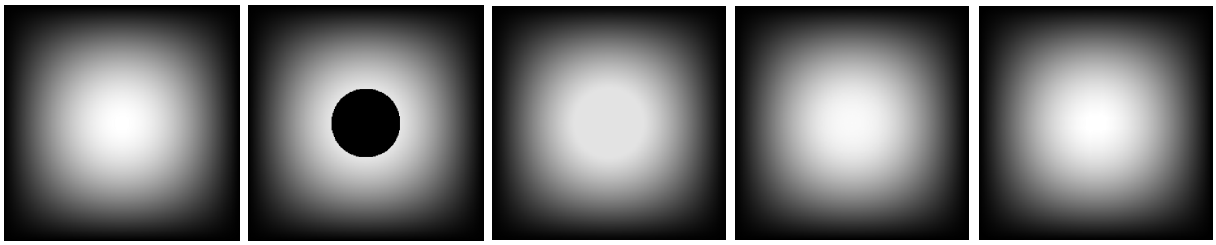
Completion of clipped data (2nd order tensor)

In this experiment, we evaluated SCTF by completing a clipped image of 2D-sin. The maximum value of the image (amplitude of 2D-sin) was 255, and the clip threshold $c = 230$. The original image is shown in Figure 3.10a, and the image after clipping is shown in Figure 3.10b. We compared SCTF with the QV model and Fast-MDT-Tucker [5]. The QV model is the model without convolution in Equation (3.21), and is formulated as

$$\hat{\boldsymbol{\mathcal{Z}}} \leftarrow \arg\min_{\boldsymbol{\mathcal{Z}}} \|\boldsymbol{\mathcal{O}} \circledast (\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{Z}})\|_F^2 + \sum_n \lambda_n \|\boldsymbol{\mathcal{L}}_n * \boldsymbol{\mathcal{Z}}\|_F^2,$$

and $\hat{\boldsymbol{\mathcal{Z}}}$ denotes the estimated completion tensor; In SCTF, we set $\tau_1 = \tau_2 = 151$, $\gamma = 1.0 \times 10^8$, $\lambda_{A,1} = \lambda_{A,2} = \lambda_{B,1} = \lambda_{B,1} = 200$, and $\eta_A = \eta_B = 400$.

Figures 3.10 and 3.11 show the images reconstructed using the QV model, Fast-MDT-Tucker, and the proposed SCTF. In addition, Table 3.1 shows the numerical evaluation of the completion accuracy. The completion image by SCTF has a maximum amplitude close to 255 and shows an improvement in the oscillations that occurred in Fast-MDT-Tucker, indicating the effect of the smoothing term. In fact, SCTF had the best value in PSNR. However, SCTF had a flat shape that differed from that of sine at a maximum amplitude of 255. This affects the numerical evaluation of the completion accuracy, and SCTF is worse in SSIM than in Fast-MDT-Tucker. On the other hand, the QV model fails to recover 2D-sin because it flatly completes the missing parts.



(a) Original image (b) Clipped image (c) QV (d) Fast-MDT (e) SCTF

Figure 3.10 Results of recoverd cliped image using QV (c), Fast-MDT-Tucker (d) and SCTF (e).

Table 3.1 Comparison of the peak signal-to-noise ratio (PSNR), the structural similarity (SSIM), and the computing time (sec) of recovery clipped images using TV, Fast-MDT–Tucker, and proposed method.

	QV	Fast-MDT-Tucker	Proposed
PSNR	34.9	50.8	**55.1**
SSIM	0.910	**0.987**	0.974
computing time	**3.29**	8.74	72.46

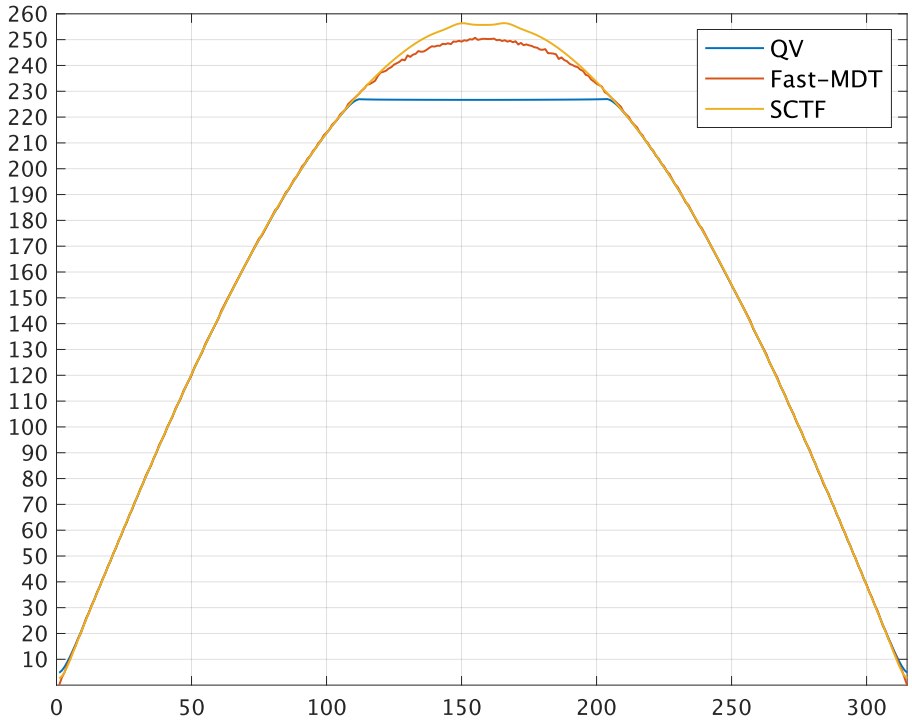


Figure 3.11 Cross-sectional view of the recoverd cliped image at height 157.

3.4.2 Completion of random missing data

This section presents the results of the completion of random missing data.

Completion of RGB images (3rd order tensor)

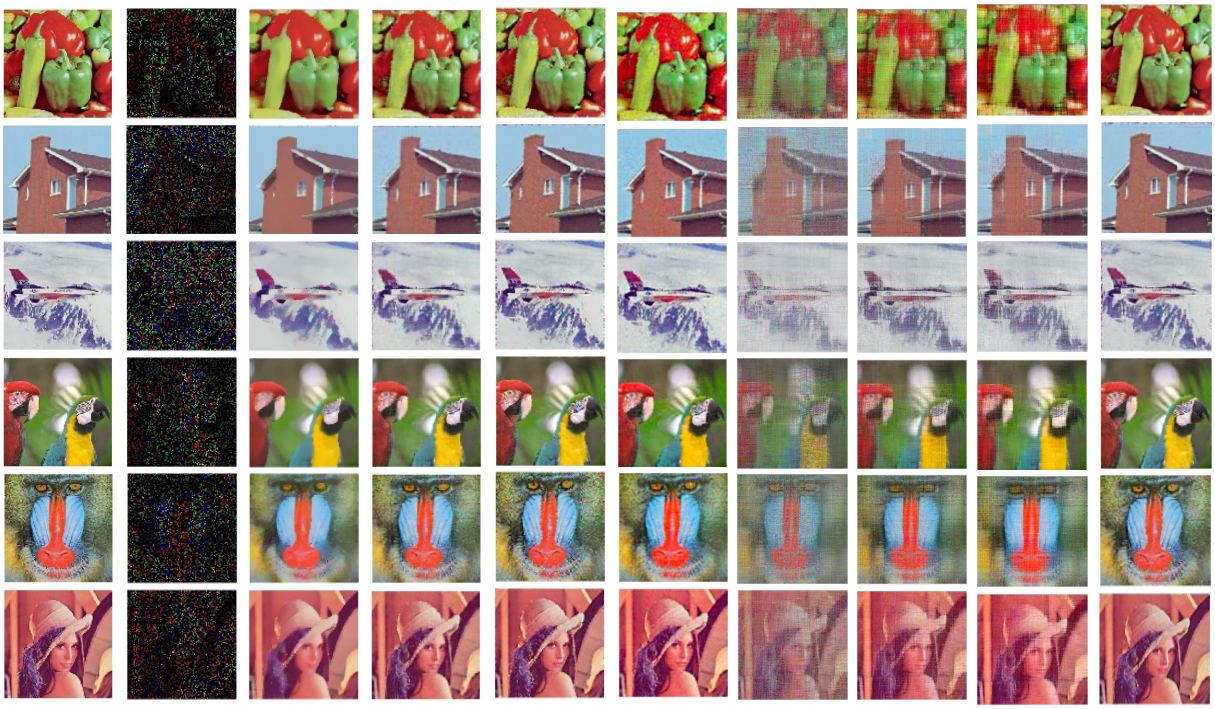
In this experiment, we evaluated SCTF using RGB images. Six images were tested, with a missing rate of 85%. Figures 3.12a and 3.12b show the original image and the image with missing data, respectively. We compared SCTF with TVLR [52], SPCQV [16], Fast-MDT-Tucker [5], BCPF [69], UTF [76], TNN [72], and PSTNN [103]. In SCTF, we set $\tau_1 = \tau_2 = \tau_3 = 9$, $\gamma = 1.0 \times 10^7$, $\lambda_{A,1} = \lambda_{A,2} = \lambda_{B,1} = \lambda_{B,2} = 50$, $\lambda_{A,3} = \lambda_{B,3} = 0$, and $\eta_A = \eta_B = 50$.

Figure 3.12 shows the experimental results. SCTF improves the image blur in TVLR, SPCQV, and BCPF and the jaggies in Fast-MDT-Tucker, UTF, TNN, and PSTNN. This may be because SCTF is based on the idea of both smoothness and MDT. Table 3.2 summarizes the recovery performance (PSNR and SSIM) and runtime. SCTF had the highest PSNR for five images and SSIM for four images. It is also slower than Fast-MDT-Tucker and UTF but has a faster computation time than SPCQV, which is the second most accurate method. In other words, SCTF is completed with high accuracy and at a modest computational cost.

Completion of MRI images (3rd order tensor)

In this experiment, we evaluated SCTF using MRI images. We prepared MRI images with sizes of $(100 \times 91 \times 91)$ with 70% and 95% of the random voxel missing. Figures 3.13a and 3.13b show the original image and the image with missing data, respectively. We compare SCTF with TVLR [52], SPCQV [16], Fast-MDT-Tucker [5], BCPF [69], UTF [76], TNN [72], and PSTNN [103]. In SCTF, we set $\tau_1 = \tau_2 = \tau_3 = 4$, $\gamma = 1.0 \times 10^7$, $\lambda_{A,1} = \lambda_{A,2} = \lambda_{B,1} = \lambda_{B,2} = 50$, $\lambda_{A,3} = \lambda_{B,3} = 0$, and $\eta_A = \eta_B = 1000$.

Figure 3.13 shows the experimental results. SCTF showed successful completion in both the 70% and 95% missing cases. In the 95% case, TVLR, Fast-MDT-Tucker, BCPF, and UTF fail to recover, and SPCQV smoothes the image excessively. In addition, TNN and PSTNN do not restore smoothly compared to SCTF. Table 3.3 summarizes the recovery performance (PSNR and SSIM) and runtime. We can confirm that SCTF has the best completion accuracy compared to the other methods; it takes longer to execute than Fast-MDT-Tucker and UTF but more than half the time of SPCQV.

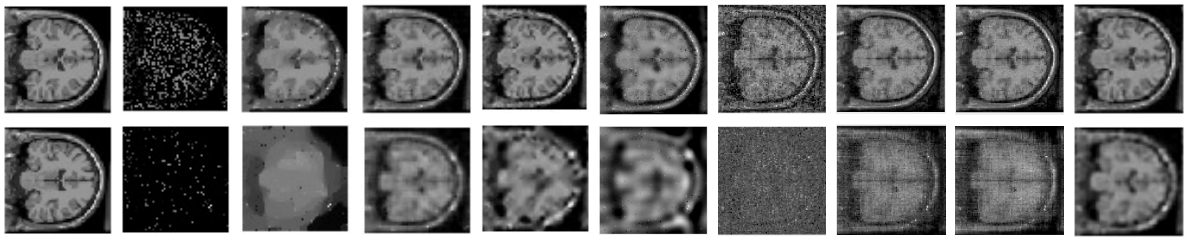


(a) Original (b) Missing (c) TVLR (d) SPCQV (e) Fast-MDT (f) BCPF (g) UTF (h) TNN (i) PSTNN (j) SCTF

Figure 3.12 Results of recoverd RGB images completion using TVLR (c), SPCQV (d), Fast-MDT-Tucker (e), BCPF (f), UTF (g), TNN (h), PSTNN (i), and SCTF (j). The image types are listed in order from the top row: *pappers*, *house*, *airplane*, *parrots*, *mandrill*, and *lena*.

Table 3.2 Comparison of the peak signal-to-noise ratio (PSNR), the mean absolute error (MAE), the structural similarity (SSIM) and the computing time (sec) of recovery RGB images. Missing ratio is 85%.

Image name	evaluation	TVLR	SPCQV	Fast-MDT	BCPF	UTF	TNN	PSTNN	SCTF
peppers	PSNR	23.5	25.5	25.3	23.5	14.8	18.3	19.2	**26.5**
	MAE	1.80×10^6	1.52×10^6	1.36×10^6	1.87×10^6	6.92×10^6	4.21×10^6	3.62×10^6	$\mathbf{1.32 \times 10^6}$
	SSIM	0.931	0.950	0.951	0.770	0.547	0.786	0.825	**0.961**
	rumtime	37.07	54.19	3.02	88.59	**0.61**	19.52	21.2	38.20
house	PSNR	24.3	26.8	25.4	25.5	17.3	22.4	22.5	**27.3**
	MAE	1.42×10^6	1.27×10^6	1.35×10^6	1.34×10^6	5.04×10^6	2.44×10^6	2.36×10^6	$\mathbf{1.24 \times 10^6}$
	SSIM	0.892	0.913	0.897	0.797	0.463	0.766	0.771	**0.923**
	runtime	35.20	35.62	2.66	105.94	**0.45**	18.37	20.93	25.07
airplane	PSNR	22.2	24.3	22.8	23.2	17.6	20.6	20.8	**24.3**
	MAE	1.92×10^6	$\mathbf{1.62 \times 10^6}$	1.67×10^6	1.73×10^6	3.91×10^6	2.89^6	2.71×10^6	1.73×10^6
	SSIM	0.655	0.669	**0.706**	0.797	0.225	0.364	0.378	0.685
	runtime	42.95	38.99	2.85	59.28	**0.40**	18.37	20.77	14.69
parrots	PSNR	24.6	25.4	24.5	24.5	15.6	20.4	21.2	**25.9**
	MAE	1.48×10^6	1.33×10^6	1.21×10^6	1.50×10^6	5.72×10^6	2.99×10^6	2.63×10^6	$\mathbf{1.52 \times 10^6}$
	SSIM	0.899	0.902	0.906	0.820	0.375	0.721	0.748	**0.913**
	runtime	40.63	51.44	2.73	55.44	**0.39**	17.80	20.48	49.53
mandrill	PSNR	21.1	**21.8**	19.4	21.3	16.1	18.6	19.0	21.5
	MAE	3.04×10^6	$\mathbf{2.72 \times 10^6}$	3.40×10^6	2.93×10^6	5.81×10^6	4.24×10^6	4.02×10^6	2.92×10^6
	SSIM	0.667	**0.720**	0.648	0.583	0.375	0.545	0.567	0.706
	runtime	31.75	63.97	4.87	99.38	**0.42**	17.51	20.35	18.97
lena	PSNR	24.3	26.0	25.0	25.8	16.2	20.9	19.9	**26.2**
	MAE	1.70×10^6	$\mathbf{1.43 \times 10^6}$	1.44×10^6	1.52×10^6	5.81×10^6	3.01×10^6	2.77×10^6	1.52×10^6
	SSIM	0.938	0.950	0.939	0.783	0.566	0.856	0.876	**0.952**
	runtimes	36.94	45.99	4.61	72.94	**0.41**	17.44	19.95	23.93



(a) Original (b) Missing (c) TVLR (d) SPCQV (e) F-MDT (f) BCPF (g) UTF (h) TNN (i) PSTNN (j) SCTF

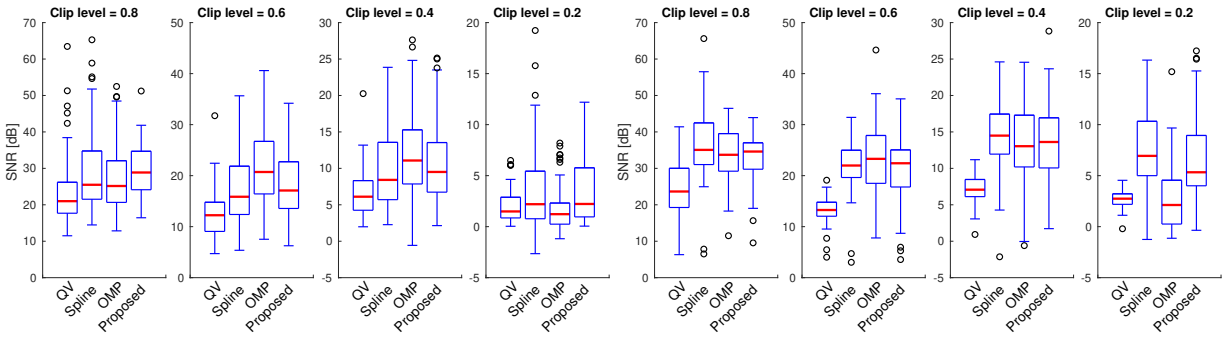
Figure 3.13 Results of recoverd MRI image completion using TVLR (c), SPCQV (d), Fast-MDT-Tucker (F-MDT) (e), BCPF (f), UTF (g), TNN (h), PSTNN (i), and SCTF (j). The time slice t is 50. The 1st column is the image with 70% missing, the 2nd row is the image with 95% missing.

Table 3.3 Comparison of the peak signal-to-noise ratio (PSNR), the mean absolute error (MAE), the structural similarity (SSIM) and the computing time (sec) of recovery MRI images. Missing ratio are 70% and 95%. Note that SSIM is averaged over time.

missingrate	evaluation	TVLR	SPCQV	Fast-MDT	BCPF	UTF	TNN	PSTNN	SCTF
70%	PSNR	23.8	26.5	23.8	20.1	14.4	22.7	23.4	**27.0**
	MAE	6.76×10^6	5.70×10^6	6.52×10^6	1.26×10^7	2.86×10^6	9.84×10^6	8.76×10^6	$\mathbf{5.21 \times 10^6}$
	SSIM	0.619	0.633	0.638	0.330	0.360	0.521	0.528	**0.700**
	rumtime	39.24	230.91	7.67	176.43	**7.30**	77.64	90.86	48.12
95%	PSNR	16.3	20.6	18.0	10.5	12.3	17.1	17.8	**21.1**
	MAE	1.94×10^7	1.29×10^7	1.58×10^7	3.24×10^7	4.40×10^6	2.17×10^6	1.90×10^7	$\mathbf{1.13 \times 10^7}$
	SSIM	0.139	0.391	0.312	0.207	0.071	0.203	0.220	**0.459**
	runtime	99.94	187.58	11.68	130.48	**6.54**	77.03	87.95	70.42

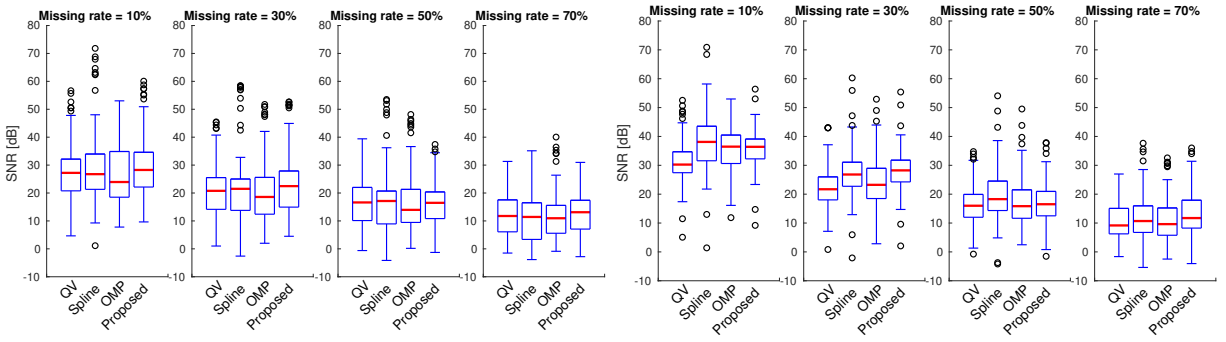
3.4.3 Applications to audio inpainting

In this section, we compare SCTF with existing methods for audio inpainting [120], [121]. Audio inpainting is a method to estimate missing entries of a single audio signal. In this study, we addressed two types of missing: clipping and random missing. Four levels of clipping $c \in \{0.8, 0.6, 0.4, 0.2\}$ and four rates of missing $\{10\%, 30\%, 50\%, 70\%\}$ are tested. Noise was not assumed. We apply the proposed method to the audio inpainting problem and compare it with QV regularization, spline interpolation, and orthogonal matching pursuit (OMP) [120], [121]. OMP is one method of sparse modeling.



(a) Music (b) Speech

Figure 3.14 Box-plot of SNR values in declipping experiments with clip levels 0.8, 0.6, 0.4, and 0.2. Segments of music and speech audio signals were recovered by quadratic variation regularization (QV), cubic spline interpolation (Spline), orthogonal matching pursuit (OMP), and SCTF (Proposed).



(a) Music (b) Speech

Figure 3.15 Box-plot of SNR values in completion experiments with random missing rates 10%, 30%, 50%, and 70%. Segments of music and speech audio signals were recovered by quadratic variation regularization (QV), cubic spline interpolation (Spline), orthogonal matching pursuit (OMP), and SCTF (Proposed)

Figures 3.14 and 3.15 and Tables 3.4 and 3.5 show the completion accuracy SNR [dB] for the proposed and existing methods, respectively. Table 3.4 shows that the proposed

Table 3.4 Average and standard deviation of SNR [dB] in declipping

clip level	QV	Spline	OMP	SCTF
Music(0.8)	23.3 ± 8.5	27.9 ± 15.0	27.4 ± 9.0	$\mathbf{28.9 \pm 7.0}$
Music(0.6)	13.0 ± 7.0	16.3 ± 12.1	$\mathbf{21.9 \pm 7.4}$	17.9 ± 6.1
Music(0.4)	6.4 ± 2.9	9.6 ± 4.9	$\mathbf{11.8 \pm 5.7}$	10.6 ± 5.3
Music(0.2)	2.0 ± 1.4	3.4 ± 3.7	1.6 ± 1.9	$\mathbf{3.5 \pm 3.1}$
Speech(0.8)	25.6 ± 10.0	33.6 ± 20.2	$\mathbf{33.6 \pm 7.5}$	32.8 ± 6.6
Speech(0.6)	13.3 ± 2.8	21.7 ± 5.0	$\mathbf{23.2 \pm 7.1}$	21.5 ± 6.2
Speech(0.4)	7.2 ± 1.8	14.5 ± 4.4	13.4 ± 6.2	$\mathbf{13.7 \pm 5.3}$
Speech(0.2)	2.7 ± 0.8	$\mathbf{7.5 \pm 3.8}$	2.8 ± 3.2	6.6 ± 4.0

Table 3.5 Average and standard deviation of SNR [dB] in completion

missing rate	QV	Spline	OMP	SCTF
Music(10%)	27.4 ± 10.5	28.8 ± 13.7	27.2 ± 11.6	$\mathbf{29.1 \pm 10.9}$
Music(30%)	20.8 ± 9.9	21.9 ± 13.1	20.8 ± 11.7	$\mathbf{23.0 \pm 11.1}$
Music(50%)	16.4 ± 8.7	$\mathbf{16.8 \pm 12.0}$	16.3 ± 10.8	16.1 ± 7.9
Music(70%)	12.2 ± 7.6	11.1 ± 9.1	11.5 ± 8.6	$\mathbf{12.9 \pm 8.1}$
Speech(10%)	31.2 ± 7.6	$\mathbf{37.8 \pm 9.9}$	35.2 ± 7.8	35.8 ± 6.7
Speech(30%)	22.7 ± 7.3	27.3 ± 8.7	24.4 ± 8.2	$\mathbf{28.2 \pm 7.4}$
Speech(50%)	16.7 ± 7.0	$\mathbf{19.6 \pm 9.2}$	17.3 ± 8.4	17.1 ± 7.0
Speech(70%)	10.7 ± 6.1	12.2 ± 8.3	11.1 ± 7.4	$\mathbf{13.3 \pm 7.9}$

Figure 3.16 Extrapolation of a typical wave signal. The completion data is the image of size 256 × 256 with 32 pixels missing from the periphery.

method performed the best in the clip levels 0.8 and 0.2 for Music and 0.4 for Speech in terms of completion of clipped signals, while OMP was more accurate in the other cases. For signals with large missing parts, such as clipped signals, a sparse modeling such as OMP is considered more effective than the proposed method. In contrast, the proposed method has high completion accuracy for random missing signals as shown in Table 3.5.

From the above experiments, the proposed method is highly accurate for Audio inpainting for some data. Still, other methods often shows better completion accuracy, which is a challenge for audio inpainting problems.

3.4.4 Signal extrapolation

Experiments are conducted to apply the tensor completion technique using the proposed method to the extrapolation of tensors. Here, experiments were conducted on artificially created periodic signals. The experimental results are shown in Figure 3.16. This signal is a sin function oscillating in the oblique direction. The size of the signal is 256 × 256, and it is missing 32 pixels on the top, bottom, left, and right sides. The experimental results show a very natural complement of the periodic function. This is considered to be because the signal is typical and compatible with the proposed method that focuses on similarity.

3.4.5 Analysis of algorithm

Convergence of algorithm

Monotonic convergence is expected because the proposed algorithm uses the MM algorithm. Figure 3.17 shows that the objective function converges monotonically and that the algorithm works correctly.

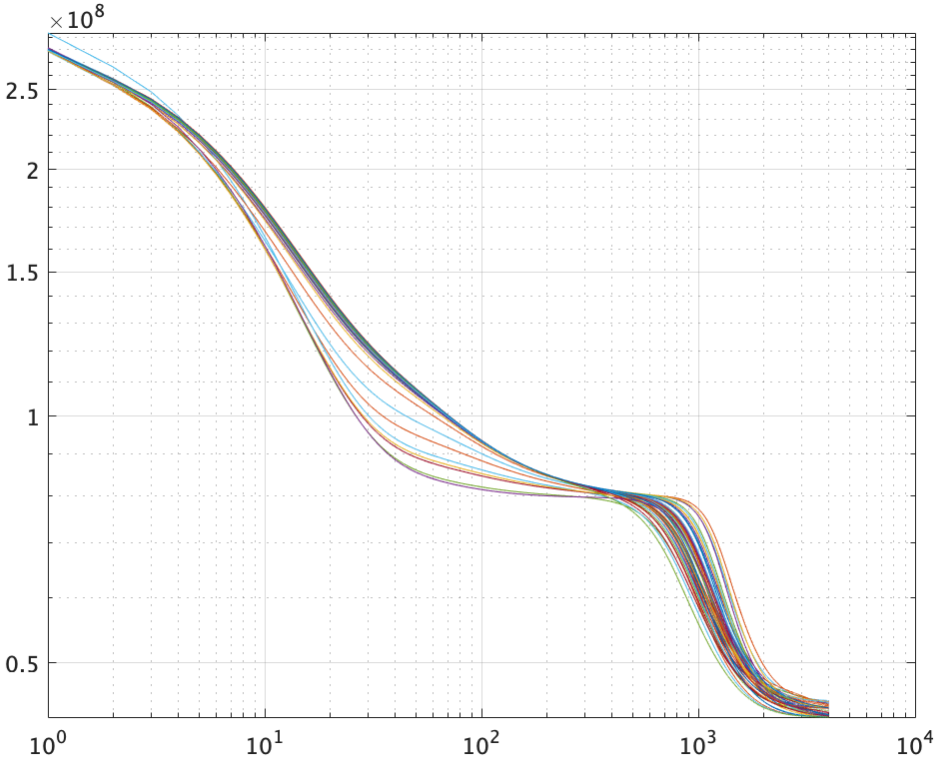


Figure 3.17 Optimization behavior: This figure shows 50 curves for 50 different initial values. The completion problem deals with the completion of clipped 2D-sin.

Hyper-parameter sensitivities

We investigated the effects of hyper-parameters of SCTF in the completion. SCTF has three hyperparameters: the delay window size $\tau_1, \ldots, \tau_N$, smoothing level $\lambda_{A,1}, \ldots, \lambda_{A,N}$, $\lambda_{B,1}, \ldots, \lambda_{B,N}$ and scale adjustment η_A, η_B (see Equations (3.22)). We redefine each of the three parameter types as $\tau \coloneqq \tau_1 = \ldots = \tau_N$, $\lambda \coloneqq \lambda_{A,1} = \ldots = \lambda_{A,N} = \lambda_{B,1} = \ldots = \lambda_{B,N}$, and $\eta \coloneqq \eta_A = \eta_B$.

The experimental setup was the same as that in Section 3.4.1; we recovered the clipped 2D-sin image. Three hyper-parameters were varied in the range $\tau \in \{1, 9, 25, 81, 151, 315\}$ and $\lambda \in \{0, 200, 400, 600, 800, 1000\}$ and $\eta \in \{0, 200, 400, 600\ , 800, 1000\}$. A declipping experiment was performed for all the combinations to calculate the recovery accuracy.

Figure 3.18 shows the experiment's five-time average of the PSNR. Increasing τ improves the accuracy, whereas making it too large worsens the accuracy. For example, when $\tau = 1$, the algorithm matches the QV regularization and recovers smoothly; however, when the delay window is smaller than the clip range, such as when $\tau = 9$ or $\tau = 25$, it cannot recover at all. Therefore, it is important to set τ appropriately.

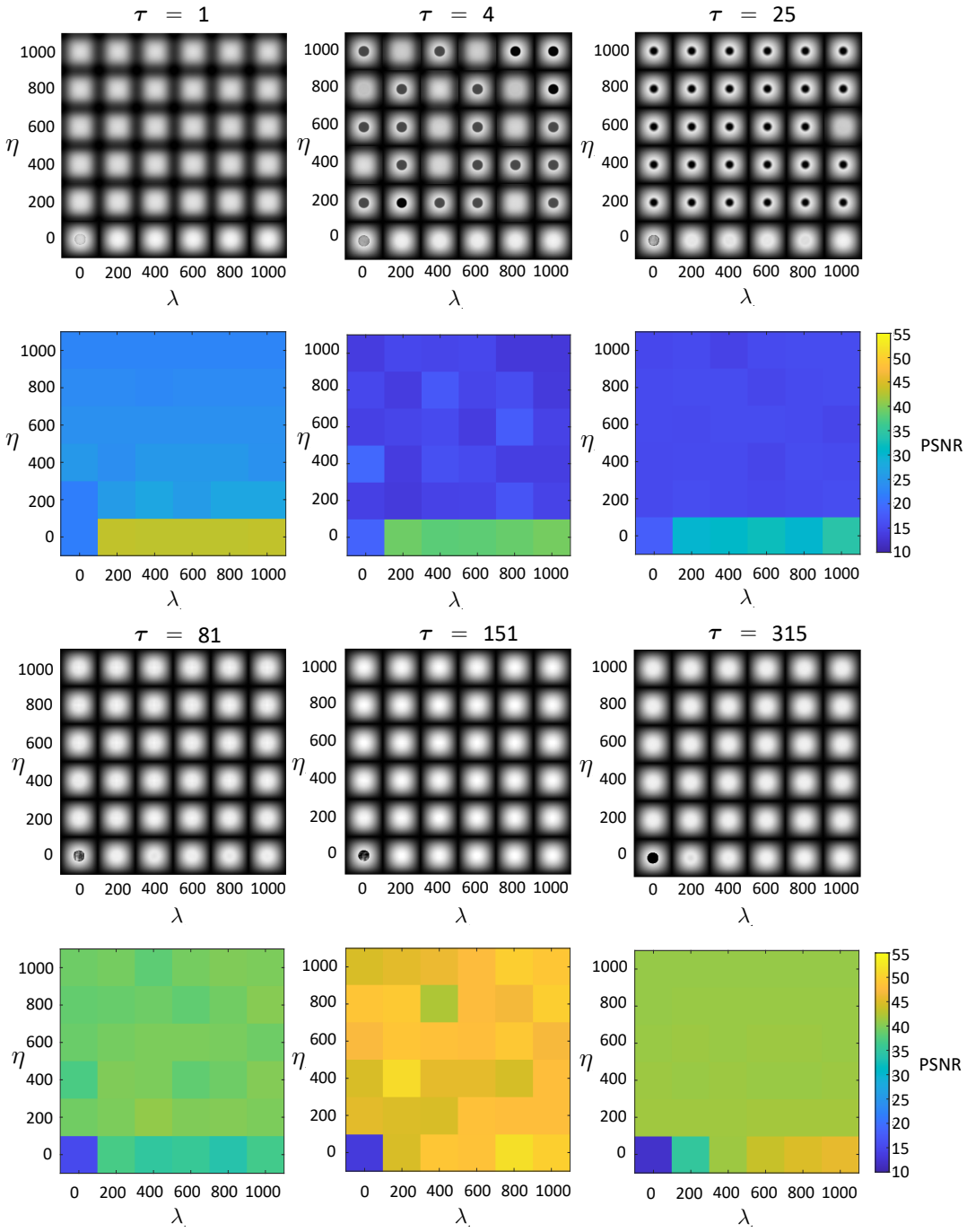


Figure 3.18 Results of extrapolation using the proposed method for periodic and simple signals.

Chapter 4

Conclusions

Tensor completion by Automatic Rank Determination with Multiplicative Gamma Process (ARD-MGP)

We proposed a method to avoid the model redundancy in ARD, an original Bayesian CP decomposition, and achieved more accurate and efficient tensor completion and rank estimation. A proposed method is called MGP-ARD, in which the MGP prior distribution is set such that the core tensor is decayed. The redundancy of the model described here refers to the overlap of the column vector of the factor matrices, which causes the original ARD method to overestimate the rank. In the proposed method, MGP-ARD sets an ordinal order to the factor matrix, eliminating the duplication of the column vectors of the factor matrix. The avoidance of model redundancy leads to an improvement in sensitivity to noise and estimation time.

The effectiveness of the proposed method is confirmed by experiments on synthetic data and real data. In the experiment of tensor completion on synthetic data, we confirmed that the rank estimation accuracy is improved compared to the original method by removing the duplication of the column vectors of the factor matrices. This also ensured that sensitivity to noise was avoided. In the experiments of tensor completion on real data, we mainly investigated the accuracy and estimation time of completion estimation for image inpainting. We confirmed that the estimation time was reduced while maintaining high estimation accuracy. In addition, because of its robustness to noise, MGP-ARD is a very good technique that provides not only rank estimation and tensor completion but also tensor decomposition.

Tensor completion by Smooth Convolution Tensor Factorization (SCTF)

We proposed a new model and algorithm for tensor completion using a convolution of smooth-factor tensors. Because the proposed method corresponds to a rank-1 decomposition in the delay-embedded space, it can achieve high completion accuracy in a short

computation time. In the optimization formulation, we extended the existing mathematical model based on the inverse MDT by adding a penalty term for the factor tensor corresponding to the delay-embedding width. In addition, we set smoothing constraints for the factor to narrow down the candidate solutions. As for the algorithm for solving the optimization, we employed the MM algorithm with the expectation of monotonic convergence. Our experiments mainly completed clipped and random missing image data and confirmed that the proposed method achieves high completion accuracy with low computational cost. In the experiment, we also confirmed the effect of the completion accuracy on the variation in the delay-embedding width and monotone convergence of the algorithm.

Overall summary and future outlook on research

In this study, we proposed two methods: *Automatic Rank Determination with Multiplicative Gamma Process (ARD-MGP)* and *Smooth Convolution Tensor Factorization (SCTF)* for the purpose of accurate and efficient tensor completion.

Concerning ARD-MGP, we will attempt to expand on the aspects related to a rank determination conducted concurrently with the tensor completion. The proposed method works well for rank estimation when the true rank is small, such as 3 or 5. In the future, we aim to develop a method that can accurately estimate ranks, even for larger ranks. Also, future works include the extension of this study to other tensor decomposition models, such as tensor train and ring decompositions [122], [123], [122], [124].

As for SCTF, we would like to extend a discussion on the *convolution*. In recent years, Deep Image prior (DIP) [125] has attracted attention as an image completion technique related to convolution. DIP uses Convolutional Neural Network (CNN) as its architecture. While image completion using CNN often requires a large amount of training data [126], [127], DIP is an optimization method that minimizes the difference between the completed image obtained from the untrained CNN and the known image to be completed. In other words, DIP does not need training data. In considering the completion performance of DIP, it is necessary to evaluate whether the *convolution architecture itself* has completion capability. Based on that, we would like to attempt to theoretically and experimentally prove the validity of the completion capability of CNN by considering multi-stage tensor convolution operations.

Appendices

Derivation of approximate posterior distribution

This section describes the derivation of the approximate posterior distribution q of parameter $\boldsymbol{\Theta} = \{\boldsymbol{A}^{(1)}, \ldots, \boldsymbol{A}^{(N)}, \boldsymbol{\lambda}, \boldsymbol{\delta}, \tau_c\}$ in Chapter 2.

Derivation of $q(\boldsymbol{A}^{(n)})$

$$
\begin{aligned}
\ln q_n(\boldsymbol{A}^{(n)}) &= \mathbb{E}_{q(\boldsymbol{\Theta}\setminus\boldsymbol{A}^{(n)})}[\ln p(\boldsymbol{\mathcal{Y}}_\Omega, \boldsymbol{\Theta})] + \text{const} \\
&= \mathbb{E}_{q(\boldsymbol{\Theta}\setminus\boldsymbol{A}^{(n)})}\left[\ln\left\{p(\boldsymbol{\mathcal{Y}}_\Omega|\{\boldsymbol{A}^{(n)}\}_{n=1}^N, \tau_c)p(\tau_c)\prod_{n=1}^N p(\boldsymbol{A}^{(n)}|\boldsymbol{\lambda})p(\boldsymbol{\lambda})\right\}\right] + \text{const} \\
&\propto \mathbb{E}_{q(\boldsymbol{\Theta}\setminus\boldsymbol{A}^{(n)})}\left[\ln p(\boldsymbol{\mathcal{Y}}_\Omega|\{\boldsymbol{A}^{(n)}\}_{n=1}^N, \tau_c)\right] + \mathbb{E}_{q(\boldsymbol{\Theta}\setminus\boldsymbol{A}^{(n)})}\left[\ln p(\boldsymbol{A}^{(n)}|\boldsymbol{\lambda})\right] \\
&= \mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^N, \tau_c)}\left[\ln\left\{\prod_{i_1=1}^{I_1}\cdots\prod_{i_N=1}^{I_N}\mathcal{N}\left(\mathcal{Y}_{i_1,i_2,\ldots,i_N}\middle|\langle \boldsymbol{a}_{i_1,:}^{(1)}, \ldots, \boldsymbol{a}_{i_N,:}^{(N)}\rangle\right.\right.\right. \\
&\quad \left.\left.\left., \tau_c^{-1}\right)^{\mathcal{O}_{i_1,\ldots,i_N}}\right\}\right] + \mathbb{E}_{q(\boldsymbol{\Lambda})}\left[\ln\prod_{i_n=1}^{I_n}\mathcal{N}(\boldsymbol{a}_{i_n,:}^{(n)}|\boldsymbol{0}, \boldsymbol{\Lambda})\right] \\
&\propto \mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^N, \tau_c)}\left[\sum_{(i_1,\ldots,i_N)\in\Omega}\left\{-\frac{\tau}{2}\left(\mathcal{Y}_{i_1,\ldots,i_N} - \left\langle \boldsymbol{a}_{i_1,:}^{(1)}, \ldots, \boldsymbol{a}_{i_N,:}^{(N)}\right\rangle\right)^2\right\}\right] \\
&\quad + \mathbb{E}_{q(\boldsymbol{\Lambda})}\left[\sum_{i_n=1}^{I_n}\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{a}_{i_n,:}^{(n)}\right] \\
&\propto \mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^N, \tau_c)}\left[-\frac{\tau}{2}\sum_{(i_1,\ldots,i_N)\in\Omega}\left\{\left\langle \boldsymbol{a}_{i_1,:}^{(1)}, \ldots, \boldsymbol{a}_{i_N,:}^{(N)}\right\rangle^2 - \mathcal{Y}_{i_1,\ldots,i_N}\left\langle \boldsymbol{a}_{i_1,:}^{(1)}, \ldots\right.\right.\right. \\
&\quad \left.\left.\left., \boldsymbol{a}_{i_N,:}^{(N)}\right\rangle\right\}\right] + \sum_{i_n=1}^{I_n}\mathbb{E}_{q(\boldsymbol{\Lambda})}\left[\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{a}_{i_n,:}^{(n)}\right]
\end{aligned}
$$

$$
\begin{aligned}
&= -\frac{\mathbb{E}_{q(\tau_c)}\left[\tau_c\right]}{2}\mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\sum_{(i_1,\ldots,i_N)\in\Omega}\left\langle \boldsymbol{a}_{i_1,:}^{(1)},\ldots,\boldsymbol{a}_{i_N,:}^{(N)}\right\rangle^2\right] \\
&- \mathbb{E}_{q(\tau_c)}\left[\tau_c\right]\mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\sum_{(i_1,\ldots,i_N)\in\Omega}\mathcal{Y}_{i_1,\ldots,i_N}\left\langle \boldsymbol{a}_{i_1,:}^{(1)},\ldots,\boldsymbol{a}_{i_N,:}^{(N)}\right\rangle\right] \\
&+ \sum_{i_n=1}^{I_n}\mathbb{E}_{q(\boldsymbol{\Lambda})}\left[\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{a}_{i_n,:}^{(n)}\right] \\
&= -\frac{\mathbb{E}_{q(\tau_c)}\left[\tau_c\right]}{2}\mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\sum_{(i_1,\ldots,i_N)\in\Omega}\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\left(\circledast_{k\neq n}\boldsymbol{a}_{i_k,:}^{(k)}\right)\left(\circledast_{k\neq n}\boldsymbol{a}_{i_k,:}^{(k)}\right)^{\mathrm{T}}\boldsymbol{a}_{i_n,:}^{(n)}\right] \\
&- \mathbb{E}_{q(\tau_c)}\left[\tau_c\right]\mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\sum_{(i_1,\ldots,i_N)\in\Omega}\mathcal{Y}_{i_1,\ldots,i_N}\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\left(\circledast_{k\neq n}\boldsymbol{a}_{i_k,:}^{(k)}\right)\right] \\
&+ \sum_{i_n=1}^{I_n}\mathbb{E}_{q(\boldsymbol{\Lambda})}\left[\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{a}_{i_n,:}^{(n)}\right] \\
&= -\frac{\mathbb{E}_{q(\tau_c)}\left[\tau_c\right]}{2}\sum_{i_n=1}^{I_n}\mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\left(\bigodot_{k\neq n}\boldsymbol{A}^{(k)}\right)^{\mathrm{T}}\boldsymbol{O}_{i_n}\left(\bigodot_{k\neq n}\boldsymbol{A}^{(k)}\right)\boldsymbol{a}_{i_n,:}^{(n)}\right] \\
&- \sum_{i_n=1}^{I_n}\mathbb{E}_{q(\tau_c)}\left[\tau_c\right]\mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\left(\bigodot_{k\neq n}\boldsymbol{A}^{(k)}\right)^{\mathrm{T}}\boldsymbol{O}_{i_n}\mathrm{vec}(\boldsymbol{\mathcal{Y}}_\Omega)\right] \\
&+ \sum_{i_n=1}^{I_n}\mathbb{E}_{q(\boldsymbol{\Lambda})}\left[\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{a}_{i_n,:}^{(n)}\right] \\
&= -\frac{\mathbb{E}_{q(\tau_c)}\left[\tau_c\right]}{2}\sum_{i_n=1}^{I_n}\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\boldsymbol{A}^{(\backslash n)\mathrm{T}}\boldsymbol{O}_{i_n}\boldsymbol{A}^{(\backslash n)}\right]\boldsymbol{a}_{i_n,:}^{(n)} \\
&- \sum_{i_n=1}^{I_n}\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\mathbb{E}_{q(\tau_c)}\left[\tau_c\right]\mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\boldsymbol{A}^{(\backslash n)\mathrm{T}}\right]\boldsymbol{O}_{i_n}\boldsymbol{y}_{i_n} + \sum_{i_n=1}^{I_n}\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\mathbb{E}_{q(\boldsymbol{\Lambda})}\left[\boldsymbol{\Lambda}\right]\boldsymbol{a}_{i_n,:}^{(n)} \\
&= -\frac{1}{2}\sum_{i_n=1}^{I_n}\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\left\{\mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\boldsymbol{A}^{(\backslash n)\mathrm{T}}\boldsymbol{O}_{i_n}\boldsymbol{A}^{(\backslash n)}\right]\mathbb{E}_{q(\tau_c)}\left[\tau_c\right] + \mathbb{E}_{q(\boldsymbol{\Lambda})}\left[\boldsymbol{\Lambda}\right]\right\}\boldsymbol{a}_{i_n,:}^{(n)} \\
&- \sum_{i_n=1}^{I_n}\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\mathbb{E}_{q(\tau_c)}\left[\tau_c\right]\mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\boldsymbol{A}^{(\backslash n)\mathrm{T}}\right]\boldsymbol{O}_{i_n}\boldsymbol{y}_{i_n}
\end{aligned}
$$

$$= \ln \left\{ \prod_{i_n=1}^{I_n} \mathcal{N}(\boldsymbol{a}_{i_n,:} | \tilde{\boldsymbol{a}}_{i_n,:}^{(n)}, \boldsymbol{V}_{i_n}^{(n)}) \right\},$$

where parameters are defined as

$$\begin{aligned} \tilde{\boldsymbol{a}}_{i_n,:}^{(n)} &= \mathbb{E}_q[\tau_c] \boldsymbol{V}_{i_n}^{(n)} \mathbb{E}_q \left[\boldsymbol{A}^{(\backslash n)\mathrm{T}} \right] \boldsymbol{O}_{i_n} \boldsymbol{y}_{i_n} \\ \boldsymbol{V}_{i_n}^{(n)} &= \left(\mathbb{E}_q \left[\boldsymbol{A}^{(\backslash n)\mathrm{T}} \boldsymbol{O}_{i_n} \boldsymbol{A}^{(\backslash n)} \right] \mathbb{E}_q[\tau_c] + \mathbb{E}_q \left[\boldsymbol{\Lambda} \right] \right)^{-1}. \end{aligned}$$

Derivation of $q(\boldsymbol{\lambda})$

$$
\begin{aligned}
\ln q_n(\boldsymbol{\lambda}) &= \mathbb{E}_{q(\boldsymbol{\Theta}\setminus\boldsymbol{\lambda})}[\ln p(\boldsymbol{\mathcal{Y}}_\Omega, \boldsymbol{\Theta})] + \text{const} \\
&\propto \mathbb{E}_{q(\boldsymbol{\Theta}\setminus\boldsymbol{\lambda})}\left[\ln\left\{\prod_{n=1}^{N} p(\boldsymbol{A}^{(n)}|\boldsymbol{\lambda})\right\}\right] + \ln p(\boldsymbol{\lambda}) \\
&= \mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\ln\left\{\prod_{n=1}^{N}\prod_{i_n=1}^{I_n}\mathcal{N}(\boldsymbol{a}_{i_n,:}^{(n)}|\boldsymbol{0},\boldsymbol{\Lambda})\right\}\right] + \ln\prod_{r=1}^{R}\mathrm{Ga}(\lambda_r|c_0^r,d_0^r) \\
&= \mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\sum_{n=1}^{N}\sum_{i_n=1}^{I_n}\ln\mathcal{N}(\boldsymbol{a}_{i_n,:}^{(n)}|\boldsymbol{0},\boldsymbol{\Lambda})\right] + \sum_{r=1}^{R}\mathrm{Ga}\ln(\lambda_r|c_0^r,d_0^r) \\
&\propto \mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\sum_{n=1}^{N}\sum_{i_n=1}^{I_n}\left\{\frac{1}{2}\ln|\boldsymbol{\Lambda}| - \frac{1}{2}\left(\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{a}_{i_n,:}\right)\right\}\right] + \sum_{r=1}^{R}\{(c_0-1)\ln\lambda_r \\
&\quad -d_0\lambda_r\} \\
&= \mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\frac{\sum_{n=1}^{N}I_n}{2}\sum_{r=1}^{R}\ln\lambda_r - \frac{1}{2}\sum_{n=1}^{N}\sum_{i_n=1}^{I_n}\sum_{r=1}^{R}\left(\lambda_r a_{i_n,r}^{(n)}a_{i_n,r}^{(n)}\right)\right] \\
&\quad + \sum_{r=1}^{R}\{(c_0-1)\ln\lambda_r - d_0\lambda_r\} \\
&= \sum_{r=1}^{R}\left\{\frac{\sum_{n=1}^{N}I_n}{2}\ln\lambda_r - \frac{1}{2}\mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\sum_{n=1}^{N}\sum_{i_n=1}^{I_n}\left(\lambda_r a_{i_n,r}^{(n)}a_{i_n,r}^{(n)}\right)\right]\right. \\
&\quad \left. + (c_0-1)\ln\lambda_r - d_0\lambda_r\right\} \\
&= \sum_{r=1}^{R}\left\{\left(\frac{\sum_{n=1}^{N}I_n}{2} + c_0 - 1\right)\ln\lambda_r - \left(\frac{1}{2}\mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\sum_{n=1}^{N}\boldsymbol{a}_{:,r}^{(n)\mathrm{T}}\boldsymbol{a}_{:,r}^{(n)}\right] + d_0\right)\lambda_r\right\} \\
&= \sum_{r=1}^{R}\left\{\left(\frac{\sum_{n=1}^{N}I_n}{2} + c_0 - 1\right)\ln\lambda_r - \left(\frac{1}{2}\sum_{n=1}^{N}\mathbb{E}_{q(\{\boldsymbol{A}^{(n)}\}_{n=1,\neq n}^{N})}\left[\boldsymbol{a}_{:,r}^{(n)\mathrm{T}}\boldsymbol{a}_{:,r}^{(n)}\right] + d_0\right)\lambda_r\right\} \\
&= \ln\left\{\prod_{r=1}^{R}\mathrm{Ga}(\lambda_r|c_M^r,d_M^r)\right\},
\end{aligned}
$$

where parameters are defined as

$$
\begin{aligned}
c_M^r &= c_0 + \frac{1}{2}\sum_{n=1}^{N}I_n \\
d_M^r &= b_0 + \frac{1}{2}\sum_{n=1}^{N}\mathbb{E}_q[\boldsymbol{a}_{:,r}^{(n)\mathrm{T}}\boldsymbol{a}_{:,r}^{(n)}].
\end{aligned}
$$

Derivation of $q(\delta_r)$

$$
\begin{aligned}
\ln q_n(\delta_r) &= \mathbb{E}_{q(\boldsymbol{\Theta}\setminus\delta_r)}[\ln p(\boldsymbol{\mathcal{Y}}_\Omega, \boldsymbol{\Theta})] + \text{const} \\
&\propto \sum_{h=r}^{R} \left\{ \mathbb{E}_{q(\boldsymbol{\Theta}\setminus\delta_h)} \left[\ln p(\lambda_h|\tau_h)\right] + \ln p(\delta_h) \right\} \\
&= \sum_{h=r}^{R} \left\{ \mathbb{E}_{q(\boldsymbol{\lambda},\{\delta\}_{\neq r})} \left[\ln \text{Ga}(\lambda_h|c_0,\tau_h)\right] + \ln \text{Ga}(\delta_h|e_0,f_0) \right\} \\
&\propto \sum_{h=r}^{R} \left\{ \mathbb{E}_{q(\boldsymbol{\lambda},\{\delta\}_{\neq r})} \left[(c_0-1)\ln\left(\prod_{l=1}^{h}\delta_h\right) - \left(\prod_{l=1}^{h}\delta_h\right)\lambda_h + (e_0-1)\ln\delta_h - f_0\delta_h \right] \right\} \\
&= \sum_{h=r}^{R} \left\{ (c_0-1)\sum_{l=1}^{h} \mathbb{E}_{q(\{\delta\}_{\neq r})}\left[\ln\delta_h\right] - \mathbb{E}_{q(\boldsymbol{\lambda},\{\delta\}_{\neq r})} \left[\left(\prod_{l=1}^{h}\delta_h\right)\lambda_h \right] \right\} \\
&\quad + (e_0-1)\ln\delta_r - f_0\delta_r \\
&\propto (c_0-1)(R-r+1)\ln\delta_r - \sum_{h=r}^{R} \mathbb{E}_{q(\boldsymbol{\lambda})}\left[\lambda_h\right] \left(\prod_{l=1,l\neq r}^{h} \mathbb{E}_{q(\delta_l)}\left[\delta_l\right] \right) \delta_r \\
&\quad + (e_0-1)\ln\delta_r - f_0\delta_r \\
&= \left\{(c_0-1)(R-r+1) + e_0 - 1\right\}\ln\delta_r - \left\{ \sum_{h=r}^{R} \mathbb{E}_{q(\boldsymbol{\lambda})}\left[\lambda_h\right] \left(\prod_{l=1,l\neq r}^{h} \mathbb{E}_{q(\delta_l)}\left[\delta_l\right] \right) + f_0 \right\} \delta_r \\
&= \text{Ga}(\delta_r|e_M^r, f_M^r),
\end{aligned}
$$

where parameters are defined as

$$
\begin{aligned}
e_M^r &= (R-r+1)(c_0-1) + e_0 \\
f_M^r &= \sum_{h=r}^{R} \mathbb{E}_q[\lambda_r] \prod_{l=1,l\neq r}^{h} \mathbb{E}_q[\delta_l] + f_0.
\end{aligned}
$$

Derivation of $q(\tau_c)$

$$
\begin{aligned}
\ln q_n(\tau_c) &= \mathbb{E}_{q(\boldsymbol{\Theta}\setminus\boldsymbol{\tau}_c)}[\ln p(\boldsymbol{\mathcal{Y}}_\Omega, \boldsymbol{\Theta})] + \text{const} \\
&\propto \mathbb{E}_{q(\boldsymbol{\Theta}\setminus\boldsymbol{\tau}_c)}\left[\ln p(\boldsymbol{\mathcal{Y}}_\Omega | \{\boldsymbol{A}^{(n)}\}_{n=1}^N, \tau_c)\right] + \ln p(\tau_c) \\
&= -\frac{\tau_c}{2}\mathbb{E}_{q(\boldsymbol{A}^{(n)})}\left[\left\|\mathcal{O}\left(\mathcal{Y} - \sum_{r=1}^R \boldsymbol{a}_{:,r}^{(1)} \circ \cdots \circ \boldsymbol{a}_{:,r}^{(N)}\right)\right\|_F^2\right] + \frac{1}{2}\sum_{i_1,\ldots,i_N} \mathcal{O}_{i_1,\ldots,i_N} \ln \tau_c \\
&+ (a_0 - 1)\ln \tau_c - b_0 \tau_c \\
&= \left(a_0 + \frac{1}{2}\sum_{i_1,\ldots,i_N} \mathcal{O}_{i_1,\ldots,i_N} - 1\right)\ln \tau_c \\
&- \left(b_0 + \frac{1}{2}\mathbb{E}_{q(\boldsymbol{A}^{(n)})}\left[\left\|\mathcal{O}\left(\boldsymbol{\mathcal{Y}} - \sum_{r=1}^R \boldsymbol{a}_{:,r}^{(1)} \circ \cdots \circ \boldsymbol{a}_{:,r}^{(N)}\right)\right\|_F^2\right]\right)\tau_c \\
&= \ln \text{Ga}(\tau_c | a_M, b_M),
\end{aligned}
$$

where parameters are defined as

$$
\begin{aligned}
a_M &= a_0 + \frac{1}{2}\sum_{i_1,\ldots,i_N} \mathcal{O}_{i_1,\ldots,i_N} \\
b_M &= b_0 + \frac{1}{2}\mathbb{E}_q\left[\|\boldsymbol{\mathcal{O}} \circledast (\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}})\|_F^2\right].
\end{aligned}
$$

Derivation of approximate posterior distribution

$\mathcal{L}$ is calculated as follows

$$
\begin{aligned}
\mathcal{L}(q) &= \int q(\boldsymbol{\Theta}) \ln \left\{ \frac{p(\boldsymbol{\mathcal{Y}}_\Omega, \boldsymbol{\Theta})}{q(\boldsymbol{\Theta})} \right\} d\boldsymbol{\Theta} \\
&= \int q(\boldsymbol{\Theta}) \ln p(\boldsymbol{\mathcal{Y}}_\Omega, \boldsymbol{\Theta}) d\boldsymbol{\Theta} - \int q(\boldsymbol{\Theta}) \ln q(\boldsymbol{\Theta}) d\boldsymbol{\Theta} \\
&= \mathbb{E}_{q(\boldsymbol{\Theta})} \left[\ln p(\boldsymbol{\mathcal{Y}}_\Omega, \boldsymbol{\Theta}) \right] - \mathbb{E}_{q(\boldsymbol{\Theta})} \left[\ln q(\boldsymbol{\Theta}) \right] \\
&= \mathbb{E}_{q\left(\left\{\boldsymbol{A}^{(n)}\right\}, \tau_c\right)} \left[\ln p(\boldsymbol{\mathcal{Y}}_\Omega | \left\{\boldsymbol{A}^{(n)}\right\}, \tau_c^{-1}) \right] + \mathbb{E}_{q(\left\{\boldsymbol{A}^{(n)}\right\}, \boldsymbol{\lambda})} \left[\sum_{n=1}^{N} \ln p(\boldsymbol{A}^{(n)} | \boldsymbol{\lambda}) \right] \\
&+ \mathbb{E}_{q(\boldsymbol{\lambda})} \left[\ln p(\boldsymbol{\lambda}) \right] + \mathbb{E}_{q(\boldsymbol{\delta})} \left[\ln p(\boldsymbol{\delta}) \right] + \mathbb{E}_{q(\tau_c)} \left[\ln p(\tau_c) \right] - \mathbb{E}_{q(\left\{\boldsymbol{A}^{(n)}\right\})} \left[\sum_{n=1}^{N} \ln q(\boldsymbol{A}^{(n)}) \right] \\
&- \mathbb{E}_{q(\boldsymbol{\lambda})} \left[\ln q\left(\boldsymbol{\lambda}\right) \right] - \mathbb{E}_{q(\boldsymbol{\delta})} \left[\ln q(\boldsymbol{\delta}) \right] - \mathbb{E}_{q(\tau_c)} \left[\ln q\left(\tau_c\right) \right].
\end{aligned}
$$

Next, the expected value $\mathbb{E}$ of each term is computed.

$$
\begin{aligned}
\mathbb{E}_{q\left(\left\{\boldsymbol{A}^{(n)}\right\}, \tau_c\right)} \left[\ln p(\boldsymbol{\mathcal{Y}}_\Omega | \left\{\boldsymbol{A}^{(n)}\right\}, \tau_c^{-1}) \right] &= \frac{M}{2} \ln(2\pi) + \frac{M}{2} (\mathbb{E}_q \left[\ln \tau_c \right]) \\
&- \frac{1}{2} \mathbb{E}_q[\tau_c] \mathbb{E}_q \left[\left\| \boldsymbol{\mathcal{O}} \circledast \left(\boldsymbol{\mathcal{Y}} - \sum_{r=1}^{R} \boldsymbol{a}_{:,r}^{(1)} \circ \cdots \circ \boldsymbol{a}_{:,r}^{(N)} \right) \right\| \right] \\
&= \frac{M}{2} \ln(2\pi) + \frac{M}{2} (\psi(a_M) - \ln b_M) \\
&- \frac{a_M}{2b_M} \mathbb{E}_q \left[\left\| \boldsymbol{\mathcal{O}} \circledast \left(\boldsymbol{\mathcal{Y}} - \sum_{r=1}^{R} \boldsymbol{a}_{:,r}^{(1)} \circ \cdots \circ \boldsymbol{a}_{:,r}^{(N)} \right) \right\| \right].
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}_{q(\left\{\boldsymbol{A}^{(n)}\right\},\boldsymbol{\lambda})}\left[\sum_{n=1}^{N}\ln p(\boldsymbol{A}^{(n)}|\boldsymbol{\lambda})\right] &= \mathbb{E}_q\left[\sum_n\sum_{i_n}\left\{-\frac{R}{2}\ln(2\pi)+\frac{1}{2}\ln|\boldsymbol{\Lambda}|-\frac{1}{2}\left(\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{a}_{i_n,:}^{(n)}\right)\right\}\right] \\
&= \sum_n\left\{-\frac{RI_n}{2}\ln(2\pi)+\frac{I_n}{2}\sum_r\mathbb{E}_q[\ln\lambda_r]\right. \\
&\left.-\frac{1}{2}\sum_{i_n}\mathbb{E}_q\left[\boldsymbol{a}_{i_n,:}^{(n)\mathrm{T}}\boldsymbol{\Lambda}\boldsymbol{a}_{i_n,:}^{(n)}\right]\right\} \\
&= -\frac{R\sum_n I_n}{2\pi}\ln(2\pi)+\frac{\sum_n I_n}{2}\sum_r\mathbb{E}_q[\ln\lambda_r] \\
&-\frac{1}{2}\sum_n\sum_{i_n}\left\{\mathrm{Tr}\left(\mathbb{E}_q[\boldsymbol{\Lambda}]\mathrm{Var}\left(\boldsymbol{a}_{i_n,:}^{(n)}\right)\right)\right. \\
&\left.+\mathbb{E}_q\left[\boldsymbol{a}_{i_n}^{(n)\mathrm{T}}\right]\mathbb{E}_q[\boldsymbol{\Lambda}]\mathbb{E}_q\left[\boldsymbol{a}_{i_n}^{(n)}\right]\right\} \\
&= -\frac{R\sum_n I_n}{2\pi}\ln(2\pi)+\frac{\sum_n I_n}{2}\sum_r\left(\psi(c_M^r)-\ln d_M^r\right) \\
&-\frac{1}{2}\sum_n\sum_{i_n}\left\{\mathrm{Tr}\left(\tilde{\boldsymbol{\Lambda}}\boldsymbol{V}_{i_n}^{(n)}\right)\right\}-\frac{1}{2}\sum_n\left\{\mathrm{Tr}\left(\tilde{\boldsymbol{A}}^{(n)}\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{A}}^{(n)\mathrm{T}}\right)\right\} \\
&= -\frac{R\sum_n I_n}{2\pi}\ln(2\pi)+\frac{\sum_n I_n}{2}\sum_r\left(\psi(c_M^r)-\ln d_M^r\right) \\
&-\frac{1}{2}\mathrm{Tr}\left(\tilde{\boldsymbol{\Lambda}}\sum_n\sum_{i_n}\boldsymbol{V}_{i_n}^{(n)}\right)-\frac{1}{2}\mathrm{Tr}\left\{\tilde{\boldsymbol{\Lambda}}\sum_n\left(\tilde{\boldsymbol{A}}^{(n)\mathrm{T}}\tilde{\boldsymbol{A}}^{(n)}\right)\right\} \\
&= -\frac{R\sum_n I_n}{2\pi}\ln(2\pi)+\frac{\sum_n I_n}{2}\sum_r\left(\psi(c_M^r)-\ln d_M^r\right) \\
&-\frac{1}{2}\mathrm{Tr}\left\{\tilde{\boldsymbol{\Lambda}}\sum_n\left(\tilde{\boldsymbol{A}}^{(n)\mathrm{T}}\tilde{\boldsymbol{A}}^{(n)}+\sum_{i_n}\boldsymbol{V}_{i_n}^{(n)}\right)\right\}.
\end{aligned}
$$

$$\begin{aligned}\mathbb{E}_q[\ln p(\boldsymbol{\lambda})] &= \sum_{r=1}^{R}\left\{-\ln\Gamma(c_0)+c_0\sum_{l=1}^{r}\mathbb{E}_q[\ln\delta_l]+(c_0-1)\mathbb{E}_q[\ln\lambda_r]-\prod_{l=1}^{r}\mathbb{E}_q[\tau_l]\mathbb{E}_q[\lambda_r]\right\}\\ &= \sum_{r=1}^{R}\left\{-\ln\Gamma(c_0)+c_0\sum_{l=1}^{r}\left(\psi(e_M^l)-\ln f_M^l\right)+(c_0-1)(\psi(c_M^r)-\ln d_M^r)\right.\\ &\quad\left.-\frac{c_M^r}{d_M^r}\prod_{l=1}^{r}\frac{e_M^l}{f_M^l}\right\}.\end{aligned}$$

$$\begin{aligned}\mathbb{E}_q[\ln p(\boldsymbol{\delta})] &= \sum_{r=1}^{R}\{-\ln\Gamma(f_0)+e_0\ln f_0+(e_0-1)\mathbb{E}_q[\ln\delta_r]-f_0\mathbb{E}_q[\delta_r]\}\\ &= \sum_{r=1}^{R}\left\{-\ln\Gamma(f_0)+e_0\ln f_0+(e_0-1)\left(\psi(e_M^r)-\ln f_M^r\right)-f_0\frac{e_M^r}{f_M^r}\right\}.\end{aligned}$$

$$\begin{aligned}\mathbb{E}_q[\ln p(\tau_c)] &= -\ln\Gamma(a_0)+a_0\ln b_0+(a_0-1)\mathbb{E}_q[\ln\tau_c]-b_0\mathbb{E}_q[\tau_c]\\ &= -\ln\Gamma(a_0)+a_0\ln b_0+(a_0-1)(\psi(a_M)-\ln b_M)-b_0\frac{a_M}{b_M}.\end{aligned}$$

$$\begin{aligned}-\mathbb{E}_q\left[\sum_{n=1}^{N}\ln q(\boldsymbol{A}^{(n)})\right] &= -\sum_n\sum_{i_n}\mathrm{E}_q\left[\ln q(\boldsymbol{a}_{i_n,:}^{(n)})\right]=\sum_n\sum_{i_n}\left\{\frac{1}{2}\ln\left|\boldsymbol{V}_{i_n}^{(n)}\right|\right\}\\ &\quad+\frac{R\sum_n I_n}{2}[1+\ln(2\pi)].\\ -\mathbb{E}_q\left[\ln q(\boldsymbol{\lambda})\right] &= \sum_r\left\{\ln\Gamma(c_M^r)-c_M^r\ln d_M^r-(c_M^r-1)\mathrm{E}_q\left[\ln\lambda_r\right]+d_M^r\mathrm{E}_q\left[\lambda_r\right]\right\}\\ &= \sum_r\left\{\ln\Gamma(c_M^r)-c_M^r\ln d_M^r-(c_M^r-1)(\psi(c_M^r)-\ln d_M^r)+d_M^r\frac{c_M^r}{d_M^r}\right\}\\ &= \sum_r\left\{\ln\Gamma(c_M^r)-\ln d_M^r-(c_M^r-1)\psi(c_M^r)+c_M^r\right\}.\\ -\mathbb{E}_q\left[\ln q(\boldsymbol{\delta})\right] &= \left\{\ln\Gamma(e_M^r)-e_M^r\ln f_M^r-(e_M^r-1)(\psi(e_M^r)-\ln f_M^r)+f_M^r\frac{e_M^r}{f_M^r}\right\}\\ &= \sum_r\left\{\ln\Gamma(e_M^r)-\ln f_M^r-(e_M^r-1)\psi(e_M^r)+e_M^r\right\}.\\ -\mathbb{E}_q\left[\ln q(\tau_c)\right] &= \ln\Gamma(a_M)-\ln b_M-(a_M-1)\psi(a_M)+a_M.\end{aligned}$$

The above is the calculation of the expected value of each term, $\mathbb{E}$. Substitute these equations into the expression for the variational lower bound $\mathcal{L}(q)$.

$$
\begin{aligned}
\mathcal{L}(q) = & \frac{M}{2}(\psi(a_M) - \ln b_M) - \frac{a_M}{2b_M}\mathbb{E}_q\left[\left\|\boldsymbol{\mathcal{O}} \circledast \left(\boldsymbol{\mathcal{Y}} - \sum_{r=1}^{R} \boldsymbol{a}_{:,r}^{(1)} \circ \cdots \circ \boldsymbol{a}_{:,r}^{(N)}\right)\right\|_F^2\right] \\
& + \frac{\sum_n I_n}{2}\sum_r (\psi(c_M^r) - \ln d_M^r) - \frac{1}{2}\mathrm{Tr}\left\{\tilde{\boldsymbol{\Lambda}}\sum_n\left(\tilde{\boldsymbol{A}}^{(n)\mathrm{T}}\tilde{\boldsymbol{A}}^{(n)} + \sum_{i_n}\boldsymbol{V}_{i_n}^{(n)}\right)\right\} \\
& + \sum_{r=1}^{R}\left\{c_0\sum_{l=1}^{r}\left(\psi(e_M^l) - \ln f_M^l\right) + (c_0-1)(\psi(c_M^r) - \ln d_M^r) - \frac{c_M^r}{d_M^r}\prod_{l=1}^{r}\frac{e_M^l}{f_M^l}\right\} \\
& + \sum_{r=1}^{R}\left\{(e_0-1)\left(\psi(e_M^r) - \ln f_M^r\right) - f_0\frac{e_M^r}{f_M^r}\right\} + (a_0-1)(\psi(a_M) - \ln b_M) - b_0\frac{a_M}{b_M} \\
& + \sum_n\sum_{i_n}\left\{\frac{1}{2}\ln\left|\boldsymbol{V}_{i_n}^{(n)}\right|\right\} + \sum_r\left\{\ln\Gamma(c_M^r) - \ln d_M^r - (c_M^r-1)\psi(c_M^r) + c_M^r\right\} \\
& + \sum_r\left\{\ln\Gamma(e_M^r) - \ln f_M^r - (e_M^r-1)\psi(e_M^r) + e_M^r\right\} \\
& + \ln\Gamma(a_M) - \ln b_M - (a_M-1)\psi(a_M) + a_M + \mathrm{const} \\
= & \left(\frac{M}{2} + a_0 - a_M\right)\psi(a_M) - \left(\frac{M}{2} + a_0\right)\ln b_M \\
& - \frac{a_M}{2b_M}\mathbb{E}_q\left[\left\|\boldsymbol{\mathcal{O}} \circledast \left(\boldsymbol{\mathcal{Y}} - \sum_{r=1}^{R} \boldsymbol{a}_{:,r}^{(1)} \circ \cdots \circ \boldsymbol{a}_{:,r}^{(N)}\right)\right\|_F^2\right] \\
& - \frac{1}{2}\mathrm{Tr}\left\{\tilde{\boldsymbol{\Lambda}}\sum_n\left(\tilde{\boldsymbol{A}}^{(n)\mathrm{T}}\tilde{\boldsymbol{A}}^{(n)} + \sum_{i_n}\boldsymbol{V}_{i_n}^{(n)}\right)\right\} + \sum_n\sum_{i_n}\left\{\frac{1}{2}\ln\left|\boldsymbol{V}_{i_n}^{(n)}\right|\right\} \\
& + \sum_{r=1}^{R}\left\{\left((R-r+1)c_0 + e_0 - e_M^r\right)\psi(e_M^r) - \left((R-r+1) + e_0\right)\ln f_M^r\right\} \\
& + \sum_{r=1}^{R}\left\{\ln\Gamma(c_M^r) + c_M^r\right\} \\
& + \sum_{r=1}^{R}\left\{\left(\frac{\sum_n I_n}{2} + c_0 - c_M^r\right)\psi(c_M^r) - \left(\frac{\sum_n I_n}{2} + c_0\right)\ln b_M\right\} \\
& + \sum_{r=1}^{R}\left\{\ln\Gamma(e_M^r) + e_M^r\left(1 - \frac{1}{f_M^r}\left(\frac{c_M^r}{d_M^r}\prod_{l=1,l\neq r}^{r}\frac{e_M^l}{f_M^l} + f_0\right)\right)\right\} \\
& + \ln\Gamma(a_M) + a_M - b_0\frac{a_M}{b_M}
\end{aligned}
$$

$$
\begin{aligned}
&= -\frac{a_M}{2b_M}\mathbb{E}_q\left[\left\|\boldsymbol{\mathcal{O}} \circledast \left(\boldsymbol{\mathcal{Y}} - \sum_{r=1}^{R} \boldsymbol{a}_{:,r}^{(1)} \circ \cdots \circ \boldsymbol{a}_{:,r}^{(N)}\right)\right\|_F^2\right] \\
&- \frac{1}{2}\mathrm{Tr}\left\{\tilde{\boldsymbol{\Lambda}} \sum_n \left(\tilde{\boldsymbol{A}}^{(n)\mathrm{T}}\tilde{\boldsymbol{A}}^{(n)} + \sum_{i_n} \boldsymbol{V}_{i_n}^{(n)}\right)\right\} + \frac{1}{2}\sum_n \sum_{i_n} \ln\left|\boldsymbol{V}_{i_n}^{(n)}\right| \\
&- a_M \ln b_M - \sum_{r=1}^{R} e_M^r \ln f_M^r + \sum_{r=1}^{R}\left\{\ln\Gamma(c_M^r) + c_M^r\right\} - \sum_{r=1}^{R} c_M^r \ln b_M \\
&+ \sum_{r=1}^{R}\left\{\ln\Gamma(e_M^r) + e_M^r\left(1 - \frac{1}{f_M^r}\left(\frac{c_M^r}{d_M^r}\prod_{l=1,l\neq r}^{r}\frac{e_M^l}{f_M^l} + f_0\right)\right)\right\} \\
&+ \ln\Gamma(a_M) + a_M - b_0\frac{a_M}{b_M} \\
&= -\frac{a_M}{2b_M}\mathbb{E}_q\left[\|\boldsymbol{\mathcal{O}} \circledast (\boldsymbol{\mathcal{Y}} - \boldsymbol{\mathcal{X}})\|_F^2\right] \\
&- \frac{1}{2}\mathrm{Tr}\left\{\tilde{\boldsymbol{\Lambda}} \sum_n \left(\tilde{\boldsymbol{A}}^{(n)\mathrm{T}}\tilde{\boldsymbol{A}}^{(n)} + \sum_{i_n} \boldsymbol{V}_{i_n}^{(n)}\right)\right\} + \frac{1}{2}\sum_n \sum_{i_n} \ln\left|\boldsymbol{V}_{i_n}^{(n)}\right| \\
&+ \sum_{r=1}^{R}\left\{\ln\Gamma(c_M^r) + \ln\Gamma(e_M^r) + c_M^r\left(1 - \ln b_M\right)\right. \\
&\left.+ e_M^r\left(1 - e_M^r \ln f_M^r - \frac{1}{f_M^r}\left(\frac{c_M^r}{d_M^r}\prod_{l=1,l\neq r}^{r}\frac{e_M^l}{f_M^l} + f_0\right)\right)\right\} \\
&+ \ln\Gamma(a_M) + a_M\left(1 - \ln b_M - \frac{b_0}{b_M}\right).
\end{aligned}
$$

Derivation of approximate posterior distribution

$$
\begin{aligned}
p(\mathcal{Y}_{i_1,\ldots,i_N}|\boldsymbol{\mathcal{Y}}_\Omega) &= \int p(\mathcal{Y}_{i_1,\ldots,i_N}|\boldsymbol{\Theta})p(\boldsymbol{\Theta}|\boldsymbol{\mathcal{Y}}_\Omega)d\boldsymbol{\Theta} \\
&\simeq \int p\left(\mathcal{Y}_{i_1,\ldots,i_N}\middle|\left\{\boldsymbol{a}^{(n)}_{i_n,:}\right\},\tau^{-1}\right) q\left(\left\{\boldsymbol{a}^{(n)}_{i_n,:}\right\}\right) q(\tau) d\left\{\boldsymbol{a}^{(n)}_{i_n,:}\right\} d\tau \\
&= \int \mathcal{N}\left(\mathcal{Y}_{i_1,\ldots,i_N}\middle|\langle \boldsymbol{a}^{(1)}_{i_1,:},\ldots,\boldsymbol{a}^{(N)}_{i_N,:}\rangle,\tau_c^{-1}\right)\prod_n \mathcal{N}(\boldsymbol{a}_{i_n,:}|\tilde{\boldsymbol{a}}^{(n)}_{i_n,:},\boldsymbol{V}^{(n)}_{i_n}) q(\tau) d\left\{\boldsymbol{a}^{(n)}_{i_n,:}\right\} d\tau \\
&= \int \mathcal{N}\left(\mathcal{Y}_{i_1,\ldots,i_N}\middle|\left(\circledast_{n\neq 1}\boldsymbol{a}^{(n)}_{i_n,:}\right)^{\mathrm{T}}\boldsymbol{a}^{(n)}_{i_n,:},\tau_c^{-1}\right)\mathcal{N}(\boldsymbol{a}_{i_1,:}|\tilde{\boldsymbol{a}}^{(1)}_{i_1,:},\boldsymbol{V}^{(1)}_{i_1}) d\boldsymbol{a}^{(1)}_{i_1,:} \\
&\prod_{n\neq 1}\mathcal{N}(\boldsymbol{a}_{i_n,:}|\tilde{\boldsymbol{a}}^{(n)}_{i_n,:},\boldsymbol{V}^{(n)}_{i_n}) q(\tau) d\left\{\boldsymbol{a}^{(n)}_{i_n,:}\right\}_{n\neq 1} d\tau \\
&= \int \mathcal{N}\left(\mathcal{Y}_{i_1,\ldots,i_N}\middle|\left(\circledast_{n\neq 1}\boldsymbol{a}^{(n)}_{i_n,:}\right)^{\mathrm{T}}\tilde{\boldsymbol{a}}^{(n)}_{i_n,:},\tau_c^{-1}+\left(\circledast_{n\neq 1}\boldsymbol{a}^{(n)}_{i_n,:}\right)^{\mathrm{T}}\boldsymbol{V}^{(1)}_{i_1}\left(\circledast_{n\neq 1}\boldsymbol{a}^{(n)}_{i_n,:}\right)\right) \\
&\prod_{n\neq 1}\mathcal{N}(\boldsymbol{a}_{i_n,:}|\tilde{\boldsymbol{a}}^{(n)}_{i_n,:},\boldsymbol{V}^{(n)}_{i_n}) q(\tau) d\left\{\boldsymbol{a}^{(n)}_{i_n,:}\right\}_{n\neq 1} d\tau \\
&\simeq \int \mathcal{N}\left(\mathcal{Y}_{i_1,\ldots,i_N}\middle|\left(\circledast_{n\neq 1}\boldsymbol{a}^{(n)}_{i_n,:}\right)^{\mathrm{T}}\tilde{\boldsymbol{a}}^{(n)}_{i_n,:},\tau_c^{-1}+\left(\circledast_{n\neq 1}\tilde{\boldsymbol{a}}^{(n)}_{i_n,:}\right)^{\mathrm{T}}\boldsymbol{V}^{(1)}_{i_1}\left(\circledast_{n\neq 1}\tilde{\boldsymbol{a}}^{(n)}_{i_n,:}\right)\right) \\
&\prod_{n\neq 1}\mathcal{N}(\boldsymbol{a}_{i_n,:}|\tilde{\boldsymbol{a}}^{(n)}_{i_n,:},\boldsymbol{V}^{(n)}_{i_n}) q(\tau) d\left\{\boldsymbol{a}^{(n)}_{i_n,:}\right\}_{n\neq 1} d\tau \\
&\vdots \\
&\simeq \int \mathcal{N}\left(\mathcal{Y}_{i_1,\ldots,i_N}\middle|\langle \tilde{\boldsymbol{a}}^{(1)}_{i_1,:},\ldots,\tilde{\boldsymbol{a}}^{(N)}_{i_N,:}\rangle,\tau_c^{-1}+\sum_n\left\{\left(\circledast_{k\neq n}\tilde{\boldsymbol{a}}^{(n)}_{i_n,:}\right)^{\mathrm{T}}\boldsymbol{V}^{(n)}_{i_n}\left(\circledast_{k\neq n}\tilde{\boldsymbol{a}}^{(n)}_{i_n,:}\right)\right\}\right) \\
&\mathrm{Ga}(\tau_c|a_M,b_M)d\tau \\
&\simeq \mathcal{T}\left(\mathcal{Y}_{i_1,\ldots,i_N}\middle|\langle \tilde{\boldsymbol{a}}^{(1)}_{i_1,:},\ldots,\tilde{\boldsymbol{a}}^{(N)}_{i_N,:}\rangle,\left\{\frac{b_M}{a_M}+\sum_n\left\{\left(\circledast_{k\neq n}\tilde{\boldsymbol{a}}^{(k)}_{i_k,:}\right)^{\mathrm{T}}\boldsymbol{V}^{(n)}_{i_n}\left(\circledast_{k\neq n}\tilde{\boldsymbol{a}}^{(k)}_{i_k,:}\right)\right\}\right\}^{-1}\right. \\
&\left.,2a_M\right).
\end{aligned}
$$

The following formula for the marginal Gaussian distribution is used in the transformation of the formula in the fourth line. The process of deriving the formula is described in [99].

Formula for Gaussian Peripheral Distribution

Suppose that the Gaussian distribution around $\boldsymbol{x}$ and the conditional Gaussian distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$ are given by

$$\begin{aligned} p(\boldsymbol{x}) &= \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}, \boldsymbol{\Lambda}^{-1}) \\ p(\boldsymbol{y}|\boldsymbol{x}) &= \mathcal{N}(\boldsymbol{y}|\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}, \boldsymbol{L}^{-1}). \end{aligned}$$

The marginal distribution of $\boldsymbol{y}$ at this time is

$$p(\boldsymbol{y}) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \boldsymbol{L}^{-1} + \boldsymbol{A}\boldsymbol{\Lambda}^{-1}\boldsymbol{A}^{\mathrm{T}}).$$

Acknowledgement

本研究は，名古屋工業大学横田達也准教授のもとで博士後期課程の３年間で行われた研究成果をまとめたものです．横田達也准教授には，その間の熱心なご指導ならびに御助言を賜りました事を心から感謝いたします．研究活動を通して,「テンソル分解・行列因子分解」,「数理最適化」,「画像解析」などの技術を学ぶこともでき，博士後期課程の３年間は非常に有意義な時間でありました．

本論文の審査をして頂いている，本学の本谷秀堅教授，和田山正教授，烏山昌幸准教授，名古屋大学の竹内一郎教授にお礼を申し上げます．特に，私が学部生だった当時に，本谷教授，和田山教授，竹内教授が担当していた「統計的機械学習」に関する幾つかの講義は，本研究に関心を持ったきっかけになっています．

横田達也研究室のメンバーには非常に深く感謝いたします．研究に関する議論や就職活動を始めとする日頃の何気ない雑談は自分にとって非常に大事な時間でした．特に，私と同じく「遅延埋めこみ」の研究に従事していた山本龍宣氏に関しては，研究活動・研究室生活共に非常にお世話になりました．

また，横田研と共同運営をしている本谷研究室のメンバー及び本谷秀堅教授にも感謝をしております．特に，本谷秀堅教授に関しては，研究に関する助言や応用数理分野に関する知見を数多くいただき深く感謝をしております．

旧竹内・烏山研究室の方々にも，学会参加（情報論的学習理論ワークショップ）に際して，研究に関する数多くの議論や学会中の交流などで大変お世話になり，非常に感謝をしております．

学外からでは，研究に関していくつか助言を頂いた，理研 AIP テンソル学習チームの Qibin Zhao チームリーダーにも大変感謝をしております．

博士課程の期間中，研究の周辺分野（主に数学）をテーマに幾つか自主勉強会を開催していました．そのメンバーであった，旧竹内・烏山研究室の杉山一弥氏（当時），金沢工業大学中沢実研究室の渡辺魁氏，早稲田大学豊田秀樹研究室の泉荘太郎氏（当時），東京工業大学渡辺澄夫研究室の小林文太郎氏（当時）には多大なる感謝をしております．杉山氏とは「関数解析」,「確率論」，渡辺氏とは「測度論」,「信号処理」を開催し,「データ科学」と「信号処理」を統一的な視点で俯瞰できる数理背景を知れたと同時に,「信号処理」を用いる本研究においてネックになっていた部分の解決にも繋がりました．泉氏とは「主観ベイズ」を開催し,「ベイズ推論」を用いる本研究の立ち位置を再確認できたと同時に,「意思決定論」に基づくベイズ統計学を認知できたことは，自分にとって大きな学びがあ

りました．小林氏とは「位相空間論」を開催すると同時に，ご好意により他３名との勉強会にも不定期に参加をしていただきました．元々，彼の数学への姿勢に影響を受けて，いくつかの勉強会を実施する運びとなったため，私にとって非常に大きな出会いでした．

最後に，私をここまで常に温かく見守って下さった両親に深く感謝をいたします．

Achievements

Journal Papers

- H. Takayama, Q. Zhao, H. Hontani, and T. Yokota, “ Bayesian tensor completion and decomposition with automatic CP rank determination using MGP shrinkage prior,” *SN Computer Science*, vol. 3, no. 225, pp. 1 - 17, 2022.
- H. Takayama, and T. Yokota, “ A New Model for Tensor Completion: Smooth Convolutional Tensor Factorization,” *IEEE Access*, vol. 11, pp. 67526-67539, 2023.

Conference Proceedings

- H. Takayama, and T. Yokota, “Fast Signal Completion Algorithm With Cyclic Convolutional Smoothing,” in *Proceedings of the Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, 2022.
- 高山拓夢，Qibin Zhao，本谷秀堅，横田達也，“ MGP 縮退事前分布を用いたテンソル補完及びランク決定法”，情報論的学習理論ワークショップ（IBIS2022），2022.
- 高山拓夢，横田達也，“ 平滑畳み込みテンソル分解によるテンソル補完”，情報論的学習理論ワークショップ（IBIS2023），2023.

References

[1] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of ACM Conference on Recommender Systems*, pp. 79–86, 2010.

[2] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 208–220, 2012.

[3] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Problems*, vol. 27, no. 2, pp. 025010.1–025010.19, 2011.

[4] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, "Tensor robust principal component analysis with a new tensor nuclear norm," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 4, pp. 925–938, 2019.

[5] R. Yamamoto, H. Hontani, A. Imakura, and T. Yokota, "Fast algorithm for low-rank tensor completion in delay-embedded space," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2058–2066, 2022.

[6] I. Balazevic, C. Allen, and T. Hospedales, "TuckER: Tensor factorization for knowledge graph completion," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 5185–5194, 2019.

[7] H. Lee, Y.-D. Kim, A. Cichocki, and S. Choi, "Nonnegative tensor factorization for continuous EEG classification," *International Journal of Neural Systems*, vol. 17, no. 04, pp. 305–317, 2007.

[8] F. Cong, Q.-H. Lin, L.-D. Kuang, X.-F. Gong, P. Astikainen, and T. Ristaniemi, "Tensor decomposition of EEG signals: a brief review," *Journal of Neuroscience Methods*, vol. 248, pp. 59–69, 2015.

[9] H. Becker, L. Albera, P. Comon, M. Haardt, G. Birot, F. Wendling, M. Gavaret, C. G. Bénar, and I. Merlet, "EEG extended source localization: tensor-based vs. conventional methods," *NeuroImage*, vol. 96, pp. 143–157, 2014.

[10] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, 2009.

[11] E. Acar, D. M. Dunlavy, T. G. Kolda, and M. Mørup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41–56, 2011.

[12] E. Acar, T. G. Kolda, and D. M. Dunlavy, "All-at-once optimization for coupled matrix and tensor factorizations," *arXiv preprint arXiv:1105.3422*, 2011.

[13] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.

[14] N. Kreimer and M. D. Sacchi, "A tensor higher-order singular value decomposition for prestack seismic data noise reduction and interpolation," *Geophysics*, vol. 77, no. 3, pp. V113–V122, 2012.

[15] A. Krishnamurthy and A. Singh, "Low-rank matrix and tensor completion via adaptive sampling," in *Proceedings of Advances in Neural Information Processing Systems*, pp. 836–844, 2013.

[16] T. Yokota, Q. Zhao, and A. Cichocki, "Smooth PARAFAC decomposition for tensor completion," *IEEE Transactions on Signal Processing*, vol. 64, no. 20, pp. 5423–5436, 2016.

[17] Q. Song, H. Ge, J. Caverlee, and X. Hu, "Tensor completion algorithms in big data analytics," *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 1, pp. 1–48, 2019.

[18] P. Zhou, C. Lu, Z. Lin, and C. Zhang, "Tensor factorization for low-rank tensor completion," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1152–1163, 2017.

[19] J. Hadamard, "Sur les problèmes aux dérivés partielles et leur signification physique," *Princeton University Bulletin*, vol. 13, pp. 49–52, 1902.

[20] X. Guo and Y. Ma, "Generalized tensor total variation minimization for visual data recovery," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3603–3611, 2015.

[21] M. Zhu and T. Chan, "An efficient primal-dual hybrid gradient algorithm for total variation image restoration," *UCLA Cam Report*, vol. 34, pp. 8–34, 2008.

[22] K. Huang, N. D. Sidiropoulos, and A. P. Liavas, "A flexible and efficient algorithmic framework for constrained matrix and tensor factorization," *IEEE Transactions on Signal Processing*, vol. 64, no. 19, pp. 5052–5065, 2016.

[23] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.

[24] K. Takeuchi, Y. Kawahara, and T. Iwata, "Structurally regularized non-negative tensor factorization for spatio-temporal pattern discoveries," in *Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference*, pp. 582–598, 2017.

[25] H. Liu, Y. Li, M. Tsang, and Y. Liu, "Costco: A neural tensor completion model for sparse tensors," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 324–334, 2019.

[26] J. D. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proceedings of the 22nd international conference on Machine learning*, pp. 713–719, 2005.

[27] N. Srebro, J. Rennie, and T. Jaakkola, "Maximum-margin matrix factorization," *Advances in neural information processing systems*, vol. 17, 2004.

[28] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, 2009.

[29] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, vol. 35, no. 12, pp. 61–70, 1992.

[30] X. Han, J. Wu, L. Wang, Y. Chen, L. Senhadji, H. Shu, *et al.*, "Linear total variation approximate regularized nuclear norm optimization for matrix completion," in *Abstract and Applied Analysis*, vol. 2014, Hindawi, 2014.

[31] S. L. Brunton and J. N. Kutz, *Data-driven science and engineering: Machine learning, dynamical systems, and control*. Cambridge University Press, 2022.

[32] P. Paatero and U. Tapper, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, pp. 111–126, 1994.

[33] D. Lee and H. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–91, 11 1999.

[34] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *Journal of Mathematics and Physics*, vol. 6, no. 1-4, pp. 164–189, 1927.

[35] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an n-way generalization of " eckart-young " decomposition," *Psychometrika*, vol. 35, no. 3, pp. 283–319, 1970.

[36] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.

[37] L. R. Tucker, "Implications of factor analysis of three-way matrices for measurement of change," *Problems in Measuring Change*, vol. 15, pp. 122–137, 1963.

[38] H. A. L. Kiers, "Hierarchical relations among three-way methods," *Psychometrika*, vol. 56, no. 3, pp. 449–470, 1991.

[39] C. Eckart and G. M. Young, "A principal axis transformation for non-hermitian matrices," *Bulletin of the American Mathematical Society*, vol. 45, pp. 118–121, 1939.

[40] J. B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear algebra and its applications*, vol. 18, no. 2, pp. 95–138, 1977.

[41] J. Håstad, "Tensor rank is NP-complete," *Journal of Algorithms*, vol. 11, no. 4, pp. 644–654, 1990.

[42] P. Paatero, "Construction and analysis of degenerate PARAFAC models," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 14, no. 3, pp. 285–299, 2000.

[43] V. De Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 3, pp. 1084–1127, 2008.

[44] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[45] J. B. Kruskal, *Rank, decomposition, and uniqueness for 3-way and n-way arrays*, pp. 7–18. North-Holland Publishing Co., 1989.

[46] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Review*, vol. 52, pp. 471–501, 2007.

[47] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717–772, 2009.

[48] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," in *Proceedings of the IEEE 12th International Conference on Computer Vision*, pp. 2114–2121, 2009.

[49] N. Komodakis, "Image completion using global optimization," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 442–452, 2006.

[50] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 417–424, 2000.

[51] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[52] T. Yokota and H. Hontani, "Simultaneous visual data completion and denoising based on tensor rank and total variation minimization and its primal-dual splitting algorithm," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3732–3740, 2017.

[53] B. Madathil and S. N. George, "Twist tensor total variation regularized-reweighted nuclear norm based tensor completion for video missing area recovery," *Information Sciences*, vol. 423, pp. 376–397, 2018.

[54] R. Tomioka and T. Suzuki, "Convex tensor decomposition via structured schatten norm regularization," *Advances in neural information processing systems*, vol. 26, 2013.

[55] B. Romera-Paredes and M. Pontil, "A new convex relaxation for tensor completion," *Advances in neural information processing systems*, vol. 26, 2013.

[56] B. Huang, C. Mu, D. Goldfarb, and J. Wright, "Provable models for robust low-rank tensor completion," *Pacific Journal of Optimization*, vol. 11, no. 2, pp. 339–364, 2015.

[57] J. Xue, Y.-Q. Zhao, Y. Bu, W. Liao, J. C.-W. Chan, and W. Philips, "Spatial-spectral structured sparse low-rank representation for hyperspectral image super-resolution," *IEEE Transactions on Image Processing*, vol. 30, pp. 3084–3097, 2021.

[58] R. A. Harshman *et al.*, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis," *UCLA Working Papers Phonetics*, vol. 16, pp. 1–84, 1970.

[59] M. Filipović and A. Jukić, "Tucker factorization with missing data with application to low-n-rank tensor completion," *Multidimensional Systems and Signal Processing*, vol. 26, no. 3, pp. 677–692, 2015.

[60] D. R. Hunter and K. Lange, "A tutorial on MM algorithms," *The American Statistician*, vol. 58, no. 1, pp. 30–37, 2004.

[61] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. SIAM, 2000.

[62] Y. Xu, R. Hao, W. Yin, and Z. Su, "Parallel matrix factorization for low-rank tensor completion," *Inverse Problems and Imaging*, vol. 9, no. 2, pp. 601–624, 2015.

[63] A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, and S.-I. Amari, "Non-negative tensor factorization using alpha and beta divergences," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 3, pp. III–1393–III–1396, 2007.

[64] A. Cichocki and A.-H. Phan, "Fast local algorithms for large scale nonnegative matrix and tensor factorizations," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 92, no. 3, pp. 708–721, 2009.

[65] A. Shashua and T. Hazan, "Non-negative tensor factorization with applications to statistics and computer vision," in *Proceedings of the 22nd International Conference on Machine Learning*, pp. 792–799, 2005.

[66] Z. Fan, X. Song, and R. Shibasaki, "CitySpectrum: A non-negative tensor factorization approach," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 213–223, 2014.

[67] K. Zhang, M. Wang, S. Yang, and L. Jiao, "Spatial–spectral-graph-regularized low-rank tensor decomposition for multispectral and hyperspectral image fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 4, pp. 1030–1040, 2018.

[68] Y.-L. Chen, C.-T. Hsu, and H.-Y. M. Liao, "Simultaneous tensor decomposition and completion using factor priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 577–591, 2013.

[69] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian CP factorization of incomplete tensors with automatic rank determination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1751–1763, 2015.

[70] W.-J. Li and D. Y. Yeung, "Relation regularized matrix factorization," in *Proceedings of the 21st International Joint Conference on Artificial Inteligence*, pp. 1126–1131, 2009.

[71] A. L. Multipliers, "Bilinear modeling via augmented lagrange multipliers (BALM)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 8, 2012.

[72] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Transactions on Signal Processing*, vol. 65, no. 6, pp. 1511–1526, 2016.

[73] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra and its Applications*, vol. 435, no. 3, pp. 641–658, 2011.

[74] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. 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*, pp. 3842–3849, 2014.

[75] S. Du, Y. Shi, W. Hu, W. Wang, and J. Lian, "Robust tensor factorization for color image and grayscale video recovery," *IEEE Access*, vol. 8, pp. 174410–174423, 2020.

[76] S. Du, Q. Xiao, Y. Shi, R. Cucchiara, and Y. Ma, "Unifying tensor factorization and tensor nuclear norm approaches for low-rank tensor completion," *Neurocomputing*, vol. 458, pp. 204–218, 2021.

[77] T.-X. Jiang, M. K. P. Ng, X. Zhao, and T. Huang, "Framelet representation of tensor nuclear norm for third-order tensor completion," *IEEE Transactions on Image Processing*, vol. 29, pp. 7233–7244, 2019.

[78] C. Lu, X. Peng, and Y. Wei, "Low-rank tensor completion with a new tensor nuclear norm induced by invertible linear transforms," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5996–6004, 2019.

[79] T. Yokota, B. Erem, S. Guler, S. K. Warfield, and H. 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*, pp. 8251–8259, 2018.

[80] A. Muhammad, M. Nikola, D. Justin, and J. Patrick, "Matrix and tensor based methods for missing data estimation in large traffic networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, pp. 1816–1825, 2016.

[81] G. Sheng, L. Denoyer, P. Gallinari, and G. Jun, "Probabilistic latent tensor factorization model for link pattern prediction in multi-relational networks," *The Journal of China Universities of Posts and Telecommunications*, vol. 19, pp. 172–181, 2012.

[82] L. Xiong, X. Chen, T.-K. Huang, J. Schneider, and J. G. Carbonell, "Temporal collaborative filtering with Bayesian probabilistic tensor factorization," in *Proceedings of the 2010 SIAM International Conference on Data Mining*, pp. 211–222, SIAM, 2010.

[83] A. Bhattacharya and D. B. Dunson, "Sparse Bayesian infinite factor models," *Biometrika*, vol. 98, pp. 291–306, 2011.

[84] P. Rai, Y. Wang, S. Guo, G. Chen, D. Dunson, and L. Carin, "Scalable Bayesian low-rank decomposition of incomplete multiway tensors," in *Proceedings of International Conference on Machine Learning*, pp. 1800–1808, 2014.

[85] M. Zhou, Y. Liu, Z. Long, L. Chen, and C. Zhu, "Tensor rank learning in CP decomposition via convolutional neural network," *Signal Processing: Image Communication*, vol. 73, pp. 12–21, 2019.

[86] K. Hosono, S. Ono, and T. Miyata, "Weighted tensor nuclear norm minimization for color image denoising," in *Proceedings of IEEE International Conference on Image Processing*, pp. 3081–3085, IEEE, 2016.

[87] J. A. Bazerque, G. Mateos, and G. B. Giannakis, "Rank regularization and Bayesian inference for tensor completion and extrapolation," *IEEE Transactions on Signal Processing*, vol. 61, no. 22, pp. 5689–5703, 2013.

[88] Z. Zhou, J. Fang, L. Yang, H. Li, Z. Chen, and R. S. Blum, "Low-rank tensor decomposition-aided channel estimation for millimeter wave MIMO-OFDM systems," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 7, pp. 1524–1538, 2017.

[89] K. Wei and Y. Fu, "Low-rank Bayesian tensor factorization for hyperspectral image denoising," *Neurocomputing*, vol. 331, pp. 412–423, 2019.

[90] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian sparse Tucker models for dimension reduction and tensor completion," *arXiv preprint arXiv:1505.02343*, 2015.

[91] X. Chen, Z. He, and L. Sun, "A Bayesian tensor decomposition approach for spatiotemporal traffic data imputation," *Transportation Research Part C: Emerging Technologies*, vol. 98, pp. 73–84, 2019.

[92] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-hard," *Journal of the ACM*, vol. 60, no. 6, pp. 1–39, 2013.

[93] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *Journal of Machine Learning Research*, vol. 1, no. Jun, pp. 211–244, 2001.

[94] D. P. Wipf, S. S. Nagarajan, J. Platt, D. Koller, and Y. Singer, "A new view of automatic relevance determination.," in *Proceedings of Advanced in Neural Information Processing Systems*, pp. 1625–1632, Citeseer, 2007.

[95] D. J. MacKay, "A practical bayesian framework for backpropagation networks," *Neural Computation*, vol. 4, no. 3, pp. 448–472, 1992.

[96] R. M. Neal, *Bayesian Learning for Neural Networks*, vol. 118. Springer-Verlag, 1996.

[97] D. J. MacKay, "Bayesian methods for backpropagation networks," in *Models of Neural Networks III*, pp. 211–254, Springer, 1996.

[98] A. Shapiro, "Identifiability of factor analysis: Some results and open problems," *Linear Algebra and its Applications*, vol. 70, pp. 1–7, 1985.

[99] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[100] F. Sedighin, A. Cichocki, T. Yokota, and Q. Shi, "Matrix and tensor completion in multiway delay embedded space using tensor train, with application to signal reconstruction," *IEEE Signal Processing Letters*, vol. 27, pp. 810–814, 2020.

[101] Z. Long, Y. Liu, L. Chen, and C. Zhu, "Low rank tensor completion for multiway visual data," *Signal Processing*, vol. 155, pp. 301–316, 2019.

[102] F. Sedighin and A. Cichocki, "Image completion in embedded space using multistage tensor ring decomposition," *Frontiers in Artificial Intelligence*, vol. 4, p. 687176, 2021.

[103] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, and L.-J. Deng, "Multi-dimensional imaging data recovery via minimizing the partial sum of tubal nuclear norm," *Journal of Computational and Applied Mathematics*, vol. 372, p. 112680, 2020.

[104] G. Liu and W. Zhang, "Recovery of future data via convolution nuclear norm minimization," *IEEE Transactions on Information Theory*, vol. 69, no. 1, pp. 650–665, 2022.

[105] G. Liu, "Time series forecasting via learning convolutionally low-rank models," *IEEE Transactions on Information Theory*, vol. 68, no. 5, pp. 3362–3380, 2022.

[106] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, "Spectral analysis of nonlinear flows," *Journal of Fluid Mechanics*, vol. 641, pp. 115–127, 2009.

[107] S. L. Brunton, B. W. Brunton, J. L. Proctor, E. Kaiser, and J. N. Kutz, "Chaos as an intermittently forced linear system," *Nature Communications*, vol. 8, no. 1, p. 19, 2017.

[108] E. N. Lorenz, "Deterministic nonperiodic flow," *Journal of Atmospheric Sciences*, vol. 20, no. 2, pp. 130–141, 1963.

[109] I. Markovsky, "Structured low-rank approximation and its applications," *Automatica*, vol. 44, no. 4, pp. 891–909, 2008.

[110] P. Van Overschee and B. De Moor, "Subspace algorithms for the stochastic identification problem," *Automatica*, vol. 29, no. 3, pp. 649–660, 1993.

[111] B. Erem, R. M. Orellana, D. E. Hyde, J. M. Peters, F. H. Duffy, P. Stovicek, S. K. Warfield, R. S. MacLeod, G. Tadmor, and D. H. Brooks, "Extensions to a manifold learning framework for time-series analysis on dynamic manifolds in bioelectric signals," *Physical Review E*, vol. 93, no. 4, p. 042218, 2016.

[112] Q. Shi, J. Yin, J. Cai, A. Cichocki, T. Yokota, L. Chen, M. Yuan, and J. Zeng, "Block Hankel tensor ARIMA for multiple short time series forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 5758–5766, 2020.

[113] X. Wang, L. Miranda-Moreno, and L. Sun, "Hankel-structured tensor robust PCA for multivariate traffic time series anomaly detection," *arXiv preprint arXiv:2110.04352*, 2021.

[114] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 60–65, 2005.

[115] P. Coupé, P. Yger, S. Prima, P. Hellier, C. Kervrann, and C. Barillot, "An optimized blockwise nonlocal means denoising filter for 3-d magnetic resonance images," *IEEE Transactions on Medical Imaging*, vol. 27, no. 4, pp. 425–441, 2008.

[116] B. B. Mandelbrot and B. B. Mandelbrot, *The fractal geometry of nature*, vol. 1. WH freeman New York, 1982.

[117] A. P. Pentland, "Fractal-based description of natural scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 661–674, 1984.

[118] T. Yokota and H. Hontani, "Tensor completion with shift-invariant cosine bases," in *proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pp. 1325–1333, 2018.

[119] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, "Parallel factor analysis in sensor array processing," *IEEE transactions on Signal Processing*, vol. 48, no. 8, pp. 2377–2388, 2000.

[120] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, "Audio inpainting," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 922–932, 2011.

[121] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, "A constrained matching pursuit approach to audio declipping," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 329–332, 2011.

[122] Y.-B. Zheng, T.-Z. Huang, X.-L. Zhao, Q. Zhao, and T.-X. Jiang, "Fully-connected tensor network decomposition and its application to higher-order tensor completion," in *Proceedings of AAAI Conference on Artificial Intelligence*, vol. 35, pp. 11071–11078, 2021.

[123] Q. Zhao, G. Zhou, S. Xie, L. Zhang, and A. Cichocki, "Tensor ring decomposition," *arXiv preprint arXiv:1606.05535*, 2016.

[124] J. L. Hinrich and M. Mørup, "Probabilistic tensor train decomposition," in *Proceedings of European Signal Processing Conference*, pp. 1–5, 2019.

[125] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9446–9454, 2018.

[126] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–14, 2017.

[127] Y. Li, S. Liu, J. Yang, and M.-H. Yang, "Generative face completion," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5892–5900, 2017.