Master Dissertation

Study on tensor calculus and CP-decomposition

NGUYEN VU LINH

Major in Knowledge Science
School of Knowledge Science
Japan Advanced Institute of Science and Technology

March 2015

Master Dissertation

Study on tensor calculus and CP-decomposition

By Nguyen Vu Linh (1350016)
Supervisor: Professor Ho Tu Bao

Major in Knowledge Science
School of Knowledge Science
Japan Advanced Institute of Science and Technology

Written under the direction of
Professor Ho Tu Bao

and approved by
Associate Professor Dam Hieu Chi
Associate Professor Huynh Van Nam
Professor Tsutomu Fujinami

February 2015 (Submitted)

Abstract

Study on tensor calculus and CP-decomposition

Nguyen Vu Linh (1350016)

School of Knowledge Science,
Japan Advanced Institute of Science and Technology

March 2015

Keywords: Tensor, CP-decomposition, Multi-way array, Temporal link prediction, Spectral clustering.

Tensor have been widely studied in mathematics and physics for along time and increasingly applied in many areas of data mining. There are two ways to think about tensors: tensors are representations of multilinear maps; tensors are elements of a tensor product of two or more vector spaces. For our purpose, "a N^{th}-order tensor" is defined as "an element of tensor product of N real vector spaces $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_N$, denoted by $\mathcal{V}_1 \otimes \mathcal{V}_2 \otimes \ldots \otimes \mathcal{V}_N$. When fixing the bases of $\mathcal{V}_1$, $\mathcal{V}_2$ and $\mathcal{V}_N$, a tensor can be represented by a N"-way array in the vector space $\mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$ and elements in $\mathcal{V}_n$ can be represented as vectors in $\mathbb{R}^{d_n}$, where d_n is the dimension of $\mathcal{V}_n$.

The equivalence between two vector space $\mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \ldots \otimes \mathbb{R}^{d_N}$ and $\mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$ allows us to not distinguish these two spaces. Such equivalence provides great advantages for data mining applications because N-way array may provide nature and compact representation for numerous complex kinds of data that the integrated result of several inter-related variables or they are combinations of underlying latent components or factors. Furthermore, when considering N-way array as element of $\mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$, powerful results related to tensor can be employed to construct tensor based methods to solve challenging problems. Especially, when working with challenging problems for big data related to capture, manage, search, visualize, cluster, classify, assimilate, merge, and process the data within a tolerable elapsed time. When working on tensor data, the large required storage memory and the inter-relation between variables often make the problems become more complicated. Many tensor-based models known as multiway models

have been constructed to deal with those challenges by exploring the meaningful hidden structure and to finding low-rank representation of data. Also, each kind of model have its own advantages and disadvantages which should be carefully considered based on the application context. For example, using CP-decomposition may lead to losing important data structure while Tucker decomposition may be problematic in high-dimensions with many irrelevant features.

One interesting tensor-based method is temporal link prediction method based on CP-decomposition constructed to do temporal link prediction task on bipartite networks whose links evolve over time and node set consists vertices of two types such that only vertices of diffent types can be linked. Such problem is important and has been studied in many researches because prediction is crucial tasks in real applications and bipartite networks can be used to represent various kinds of structures, dynamics, and interaction patterns found in social activities. Temporal link prediction method based on CP-decomposition have shown advantages comparing with others, such as its power in exploring the structure of data, requiring less memory and giving outperformed experimental results. Also, tensor based-methods can predict the links for times $(T+1)^{th}$, ... , $(T+L)^{th}$ while other models are limited to temporal prediction for a single time step. Motivating by these advantages, we extend tensor-based method on bipartite networks to do temporal link prediction problem on specific class of bipartite networks in which new verticies of one type may join networks at concerning time and may link to other vertices in the next time point. The key ideas of the proposed methods are to employ CP-decomposition to decompose weight data into factors of three separated kinds, each fluctuates independently from others and collect additional information of vertices of open type and learn a function to predict values of open type vertex factors from the additional information and use to predict values of those factors corresponding to new vertices.

Clustering plays an outstanding role in numerous data mining applications such as scientific data exploration, information retrieval and text mining, spatial database applications, Web analysis, CRM, marketing, medical diagnostics, computational biology, and many others. Intuitively, clustering aims to identify groups of "similar behavior" data considered as a first impression on data when dealing with the empirical data. Since tensor data have became popular in data mining, we focus on constructing a versatile clustering for tensor data. Considering clustering methods based on vector space model, spectral clustering have several attractive advantage such that it is versatile, easy to implement, often provide better performance comparing with traditional methods like K-mean. Furthermore, spectral clustering is easy to extend for tensor data since it work with only requirement about similarity while other methods often require more additional information. To handling the clustering task on tensor data, we construct a CP-decomposition based spectral clustering by constructing appropriate similarities and employing CP-decomposition to exploring the hidden structure of data and reducing the storage memory. The empirical results provide the evidence to conclude that the proposed models can give the acceptable accuracy and CP-decomposition may help to reduce the storage memory and improve the clustering accuracy by exploring the hidden structure of data.

Concerning temporal link prediction and clustering problems, we discuss about the

achievements and provide suggestions to and make plan to complete these objective as future works. For temporal link prediction problem, we plan to implement the method and evaluate using empirical results and extend method to do more general problem when vertices of two type join the concerning networks at the same time. Considering the clustering problem, we plan to construct several similarity measures and extend the available vector space model based multi-view spectral clustering for tensor data. We also give opportunity and suggestions to construct a tensor space model based clustering method which tensor data is transformed in to vector space data and spectral clustering methods are applied on the transformed data in order to cluster the data.

Acknowledgments

I would like to express my deep gratitude to my master thesis advisor Professor Ho Tu Bao of Japan Advanced Institute of Science and Technology. He did introduce me to the research field of machine learning and data mining and spend much time instructing me how to do research. Becoming his student is one of biggest chances for me.

I would like to show my gratitude to the committee members, Professor Dam Hieu Chi, Professor Katsuhiro Umemoto, Professor Huynh Van Nam and Professor Tsutomu Fujinami of Japan Advanced Institute of Science and Technology, for their insightful and constructive comments that helped to improve the presentation of this thesis.

I would like to thank my second supervisor Professor Yukio Hayashi and my supervisor for minor research Professor Mitsuru Ikeda of Japan Advanced of Science and Technology for their suggestions and continuous encouragements.

I am grateful to Japan Advanced Institute of Science and Technology for providing me finance support and a good environment to study.

I would also like to thank all members of Ho Laboratory who are great friends and helpful to color my life.

Finally, I owe more than thanks to my family for their loves and encouragement during the duration of Master's course.

Nguyen Vu Linh
JAIST, Japan
February, 2015

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The first part of this chapter provides an overview of tensor calculus and its role in data mining. In the second part, the research problems and objectives are presented. Finally, the content of this thesis is given in the third part.

1.1 Tensor calculus and its role in data mining

1.1.1 Tensor calculus

Tensor have been widely studied in mathematics and physics for along time and increasingly applied in many areas of data mining. There are two ways to think about tensors: tensors are representations of multilinear maps; tensors are elements of a tensor product of two or more vector spaces [21,45,60]. In this thesis, we use the definition of tensor that "a N^{th}-order tensor is an element of tensor product of N vector spaces $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_N$, denoted by $\mathcal{V}_1 \otimes \mathcal{V}_2 \otimes \ldots \otimes \mathcal{V}_N$, which is represented by a N"-way array or element of vector spaces $\mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$ when the bases of $\mathcal{V}_1$, $\mathcal{V}_2$ and $\mathcal{V}_N$ are fixed.

Tensor and tensor product of vector spaces are difficult objects in algebra. Those objects have been studied for along time in algebra areas like tensor algebra and multilinear algebra [11, 19, 24, 30, 49, 55, 66, 70]. To understand the properties of those objects, deep understanding about vector space, relation between vector space and other algebra objects including multi-linear function, quotient spaces, etc are required. There are some readable documents about tensor, tensor product of vector spaces and related results in general case, for example [49, 60].

In this thesis, we consider one specific case of 3^{rd}-order tensor as element of tensor product of 3 Euclidean vector spaces $\mathbb{R}^I$, $\mathbb{R}^J$ and $\mathbb{R}^T$, denoted by $\mathbb{R}^I \otimes \mathbb{R}^J \otimes \mathbb{R}^T$, which is not so complicated but have been widely applied in multi-way data analysis [3, 19, 38]. Also the definition of tensor product of vector spaces is not presented here because it is complicated and we only work on the vector space constructed on the set of 3 real way arrays, denoted by $\mathbb{R}^{I \times J \times T}$. Vector space $\mathbb{R}^I \otimes \mathbb{R}^J \otimes \mathbb{R}^T$, and the relation between $\mathbb{R}^I$, $\mathbb{R}^J$ and $\mathbb{R}^T$ have been well studied in algebra. Furthermore, as point out in [21], the structure of two vector spaces $\mathbb{R}^I \otimes \mathbb{R}^J \otimes \mathbb{R}^T$ and $\mathbb{R}^{I \times J \times T}$ are equivalent, it implies that results on

$\mathbb{R}^I \otimes \mathbb{R}^J \otimes \mathbb{R}^T$ can be easily proved on $\mathbb{R}^{I \times J \times T}$.

1.1.2 Tensor-based models in data mining

Data representation has been considered as one of the most important problem in machine learning and data mining [12, 33]. Many applications are based on vector space model in which data is represented as vectors $x \in \mathbb{R}^I$. In vector space model, the features are implicitly assumed to be independent [12]. Vector space model have been used to construct many models, for example, classification, clustering [54], support vector machine, etc. There are numerous readable documents about vector space based models [33, 54, 64].

However, in many situations, there are reasons to consider data as tensors. For example, multi-channel electroencephalogram (EEG) data are commonly represented by an $I \times T$ matrix which represents recorded signals of I electrodes at T times. In order to discover hidden brain dynamics, often frequency content of the signals like signal power at J particular frequencies, also needs to be considered. Then, EEG data can be arranged as an 3-way dataset of size $I \times J \times T$ [50]. Similarly, a 3-way tensor $\underline{\mathcal{X}}$ of size $I \times J \times T$ can be used to represent the publication information of I authors on J conferences over T years, where element x_{ijt} is 1 if authors i have publication on conference j at year t and is 0 if otherwise [2, 26].

Generally, tensors can be used to represent several complex kinds of data that the integrated result of several inter-related variables or they are combinations of underlying latent components or factors [18]. For instances, 3-order tensors can be used to represent multi-view data [48], time series data [2, 26], etc. When tensors are used to represent data, we work on tensor space model instead of vector space model [13, 34, 50, 56, 59, 69].

Note that, when working with vector space models $\mathbb{R}^I$, a $I \times J$ matrix is often used to represent a dataset consists of J samples. Further more, as matrix have been well studied in linear algebra, many interesting results can be employed when handling data. For example, one may apply matrix factorization models to explore the structure of data or find low-rank representation. Some popular matrix factorization methods are LU, QR, SVD, etc. The situation is similar when working with tensor space models. When tensors are used to represent data, tensor based methods allows us to discover meaningful hidden structures of complex data and to perform generalizations by capturing multi-linear and multi-aspect relationships [17].

Tensors and tensor based methods have recently became a promising direction in many areas, especially in data mining because it may provide a natural representation for Big Data which consists of multidimensional, multi-modal datasets which are so huge, complex and cannot be easily stored or processed by using standard computers. Many challenging problems for big data are related to capture, manage, search, visualize, cluster, classify, assimilate, merge, and process the data within a tolerable elapsed time. Such problems can be solved by employing tensor decomposition models (or multi-way models), which allow us to discover meaningful hidden structures of complex data and provide compact representation via suitable low-rank approximations [3, 17, 18, 38].

There many tensor decomposition models and each kind of model have its own advantages and disadvantages when comparing with others. For example, ones may lose

important structure in the data when using CP-decomposition while Tucker decomposition may be problematic in high-dimensions with many irrelevant features [6, 7]. Some popular multi-way models (tensor based models) are listed in Figure 1.1.

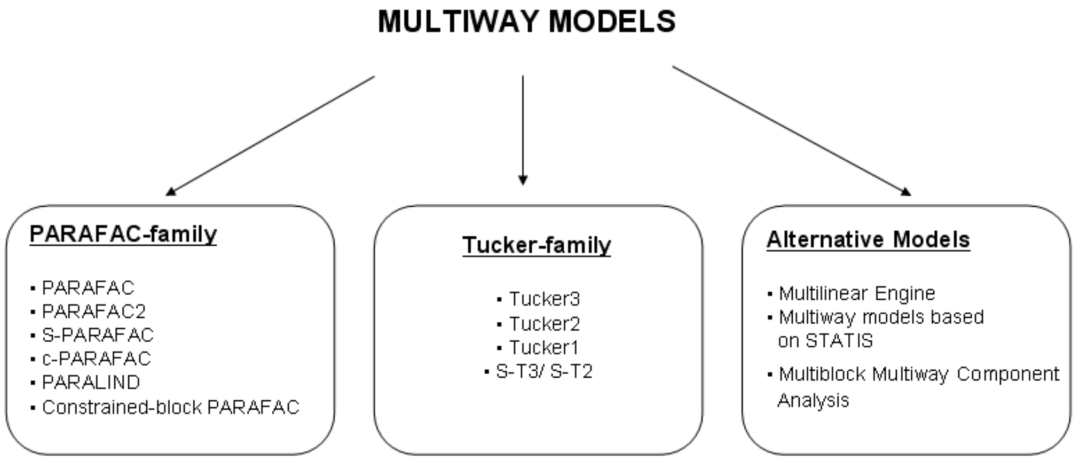

Figure 1.1: List of popular multi-way models (tensor based models) [3].

1.2 Problem formulating and objectives

1.2.1 Constructing a temporal link prediction method on open bipartite networks evolve over time

Temporal link prediction is a common problem for data which is assumed to have the underlying periodic structure. Formally, temporal link prediction is the problem of predicting the links at time $(T+1)^{th}$ given link data for times 1 through T^{th} [2, 26]. In other words, we consider the problem of predicting the change or apparition of new links in time-evolving networks. Temporal link prediction is different from simply predicting future links without considering the network evolution history because it exploits the past history of link evolution provided by the state of the network at successive time slices [27].

In many applications dealing with temporal link prediction, data can be represented in form of bipartite networks with two set of vertices and only vertices of different types can be connected by links [2, 26, 43]. Considering bipartite networks that evolve over time, two sets of vertices are fixed and only the link between different kind of vertices change. Bipartite networks can be used to represent various kinds of structures, dynamics, and interaction patterns found in social activities [8, 22, 52, 53, 67]. For example, bipartite network is used to represent the network consists of I authors and J conferences where each link represents the possibility that an author publishes on a conference and temporal link prediction is used to predict which authors will publish at which conferences in year $(T+1)^{th}$ given the publication data for the T previous years [2, 26]; I users/groups of user, J providers and their relations in a recommendation systems can be represented by

a bipartite network [67]; Another example is the bipartite network consists of I patient groups and J drugs with the links represent the adverse effect of drugs.

Many temporal link prediction methods are based on matrix calculus which collapse the data into a single matrix by summing (with and without weights) the matrices corresponding to the time slices and using the result matrix to predict the network links in the next time points. Such methods have been successfully applied in several applications. For example, data about co-authorship networks extracted from arXiv [3, 26, 27], etc. However, the matrix-based methods are limited to temporal prediction for a single time step while in many applications, we wish to predict the links for a period of time starting at $(T+1)^{th}$ or in other words, for times $(T+1)^{th}$, ... , $(T+L)^{th}$. Recently, tensor-based methods were proposed which can be used in solving both single step and periodic temporal link prediction problems [3, 26, 27, 61]. In other words, tensor-based methods can be used to extend the application context of matrix based methods. In tensor-based methods, tensor decomposition namely CP-decomposition (CANDECOMP/PARAFAC) is employed with several purposes such as to explore the natural three-dimensional structure, reduce tensor dimensionality or to retrieve latent trends [3, 61].

When using bipartite networks to represent data, we may face some special bipartite networks with common situation that new vertices may join network at time T^{th} and may link to other vertices at time $(T+1)^{th}$. For example, when a bipartite network is used to represent data in recommendation system, new users/groups of user appear and may interest or use service of providers in next time points; also in drug side effect example, new drugs are launched and may have adverse effect on patient groups in the future; etc. For simplicity, let call such networks by open bipartite networks to distinguish from networks without new added vertices.

Several temporal link prediction methods have been proposed for bipartite networks [2, 23, 26] but in our knowledge, no temporal link prediction method have been proposed to deal with problem on open bipartite networks. In our point of view, temporal link prediction is challenging because it is difficult to learn how new vertices will link to other vertices and how it effects links among previous vertices at time $(T+1)^{th}$. Our objective is to extend temporal link prediction method using CP-decomposition in [2, 26] to address temporal link prediction in open bipartite networks in which new vertices of type-1 may join networks at time T^{th}.

1.2.2 Constructing a versatile clustering method for tensor data

Clustering is an important subject of active research areas in several fields such as statistics, pattern recognition, machine learning and data mining which aims to identify groups of "similar behavior" data. In machine learning and data mining, clusters correspond to hidden patterns and is often referred as a first impression on data when dealing with the empirical data. Clustering plays an outstanding role in numerous data mining applications such as scientific data exploration, information retrieval and text mining, spatial database applications, Web analysis, CRM, marketing, medical diagnostics, computational biology, [9, 44, 57, 64], and many others.

Many clustering algorithms have been proposed and successfully applied to real-life data mining problems. A list of popular clustering methods and algorithms is given bellow [9].

1. Hierarchical methods: Agglomerative algorithms, divisive algorithms.
2. Partitioning methods: Relocation algorithm, probabilistic clustering, K-medoids methods, K-mean methods, density-based algorithms.
3. Grid-based methods.
4. Methods based on co-occurrence of categorical data.
5. Constraint-based clustering.
6. Clustering algorithms used in machine learning:Gradient descent and artificial neural networks, evolutionary methods.
7. Scalable clustering algorithms.
8. Algorithm for high dimensional data: Subspace clustering, projection techniques, co-clustering techniques.

A detail review on these clustering algorithms is given in [9]. Also, there are many readable survey papers on clustering algorithms such as [1, 63, 65]

Two general criterias used to compare clustering methods are "How easy to use the method?" and "How well the performance is when applying methods to analyze data?"[57]. The former is usually used to compare the execution time and storage requirements of computer-oriented methods. As complex kinds of data have became a challenge in data mining, we suggest to consider "How easy to interpret the result?" and "How easy to extend such methods for complex kinds of data?" as parts of the first criteria when comparing clustering methods.

Among numerous clustering methods, spectral clustering has recently became one of the most popular modern methods because it is simple to implement and often provide outperforms performance when comparing with traditional clustering algorithms such as the K-means [54, 64]. The intuition of spectral clustering is to representing the data of I data sample $x_1, x_2, \ldots, x_I$ is in form of the similarity graph $G = (V, E)$ which have following properties

- Each vertex $v_i \in V$ represents a data point x_i.
- Two vertices x_i and x_j are connected if their similarity s_{ij} is positive or larger than a certain threshold, and s_{ij} is weight of edge $E(x_i, x_j) \in E$.

The problem of finding "similar groups" among original data sample $x_1, x_2, \ldots, x_I$ now is equivalent to the problem of finding groups of vertices in the similarity graph such that the edges between different groups have very low weights and the edges within a group have high weights. The latter problem can be solve by employing results from

linear algebra and graph theory, and the result can be easily interpreted on the original data[64].

Note that spectral clustering is simple to implement and easily to extend for complex data because it only requires determined similarity measures while other methods often require more additional information. For example, even when extending simple method like K-mean, we have to consider two objects, centers/centroids and distance measure. Furthermore, as multi-view approach have been considered as a promising direction in data mining with the power to simultaneously treat heterogeneous kinds of data in order to improve the empirical results. Considering clustering task, up to now, we have not found any paper working under multi-view direction related to K-mean while several interesting multi-view spectral clustering methods were recently proposed with well empirical results and is theoretically guaranteed [48, 62].

Motivating by the above reasons, we plan to extend the spectral clustering for 3^{rd} order tensor data as the first attempt in constructing efficient and easy implement clustering methods for tensor data.

1.3 Thesis structure

This thesis is organized as follow

- **Chapter 1** presents the research problems, objectives of the thesis. An overview of tensor calculus and its role in data mining is also presented as a guide for further study in tensor calculus. Then two interseting problems in data mining, namely, temporal link prediction and clustering are introduced and the objectives related to constructing such methods for tensor data are given.

- **Chapter 2** firstly provides necessary notations to understand the content of thesis. Then, definition of tensor and an overview of a common used tensor decomposition, namely CP-decomposition, and related family of application multi-way models known as PARAFAC family. We also give discussions to avoid confusion when applying tensor calculus in data mining, though tensor is complicated objects.

- **Chapter 3** presents a quick survey on temporal link prediction and a proposed temporal link prediction method using CP-decomposition. The first part of this chapter presents necessary concepts related to temporal link prediction and open bipartite networks and an introduction to temporal link prediction on bipartite networks. Then the second part provide the related works which is needed to understand the proposed temporal link prediction method for open bipartite networks. In the third part, the proposed method is presented. Finally, discussions and future works is given in the fourth part.

- **Chapter 4** provides an overview of spectral clustering and suggestions to extend this method to deal with tensorial data. Firstly, we give an overview of spectral clustering including the general schema, opportunity to extend for tensorial data. Also,

the currents research on clustering methods for tensor data is given. Secondly, a proposed spectral clustering method namely CP-decomposition based spectral clustering is presented. Then, experiment results is analyzed to evaluate the proposed method. Finally, the suggestions for future research are given.

- **Chapter 5** provides the conclusion of thesis and the future works. In the first part, contributions are given and evaluated. Then, research plans and promising techniques might be used to complete or improve the considered research problems in this thesis are presented.

Chapter 2

Tensor and CP-decomposition

This chapter provides the list of notations which will be used through this thesis, definition of tensor and an overview of a common used tensor decomposition, namely CP-decomposition, and related family of application multi-way models known as PARAFAC family. We also give discussions to avoid confusion when applying tensor calculus in data mining, though tensor is complicated objects.

2.1 Preliminaries

In this section, we introduce the list of notations related to tensor which will be used through this thesis. We also recall the definition of some vector spaces which is necessary for understanding the later parts of this thesis.

2.1.1 Notations

The notations used in this thesis is very similar to that presented in [38]. The important notations are presented in Table 2.1 and other notations will be introduced where they are necessary.

2.1.2 Some notations related to 3^{rd} order tensor

We recall some notations related to 3^{rd} tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I \times J \times T}$ from [38] which are used in later parts.

1. **Fibers**

   Fibers are the higher order analogue of matrix rows and columns. A fiber is defined by fixing every index but one. A matrix column is a mode-1 fiber and a matrix row is a mode-2 fiber. Third-order tensors have column, row, and tube fibers, denoted by $\mathbf{x}_{:jt}$, $\mathbf{x}_{i:t}$, and $\mathbf{x}_{ij:}$, respectively. Fibers are always assumed to be column vectors. See Figure 2.1 for an illustration.

Table 2.1: List of important notations.

Symbol	Notation
$\mathbb{R}^I$	The Euclidean vector space of dimension I.
$\mathbf{A}$	The matrix $\mathbf{A}$.
a_{ij}	The (i,j)-th element of matrix $\mathbf{A} \in \mathbb{R}^{I \times J}$.
$\mathbf{a}$	The vector $\mathbf{a}$.
a_i	The i-th element of vector $\mathbf{a} \in \mathbb{R}^I$.
$\underline{\mathcal{X}}$	The tensor $\underline{\mathcal{X}}$
$x_{d_1 d_2 \ldots d_N}$	The $(i_{d_1}, i_{d_2}, \ldots, i_{d_N})$-th element of tensor $\underline{\mathcal{X}} \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$
$\mathbf{X}_{i::}, \mathbf{X}_{:j:}$	The i-th horizontal, j-th lateral slices of tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I \times J \times T}$
$\mathbf{X}_{::t}, \mathbf{X}_t$	The t-th frontal slices of tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I \times J \times T}$
$\mathbf{X}_{(n)}$	The mode-n of N-th order tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I \times J \times T}$
$\circ$	The outer product "$\circ$"
$\otimes$	The tensor product "$\otimes$"

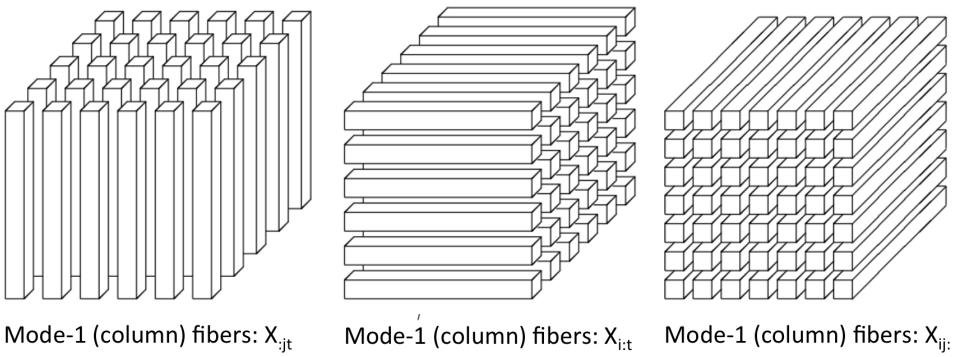

Figure 2.1: Fibers of a 3^{rd}-order tensor [38].

2. **Slices**

   Slices are two-dimensional sections of a tensor, defined by fixing all but two indices. Figure 2.2 shows the horizontal, lateral, and frontal slides of a third-order tensor $\underline{\mathcal{X}}$, denoted by $\mathbf{X}_{i::}$, $\mathbf{X}_{:j:}$, and $\mathbf{X}_{::t}$, respectively.

3. **Mode-n matricization**

   The mode-n matricization of a 3^{rd} tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I \times J \times T}$ is denoted by $\mathbf{X}_{(n)}$ and arranges the mode-n fibers to be the columns of the matrix. The following example of 3^{rd} tensor $\underline{\mathcal{X}} \in \mathbb{R}^{3 \times 4 \times 2}$ may help ones to have a better understanding about mode-n matricization and its relation to slices.

   Let the frontal slices $\mathbf{X}_1$ ($\mathbf{X}_{::1}$) and $\mathbf{X}_2$ ($\mathbf{X}_{::2}$) of $\underline{\mathcal{X}} \in \mathbb{R}^{3 \times 4 \times 2}$ be as in Figure 2.3.

   Then, the three mode-n ($n = 1, 2, 3$) unfoldings are given in Figure 2.4.

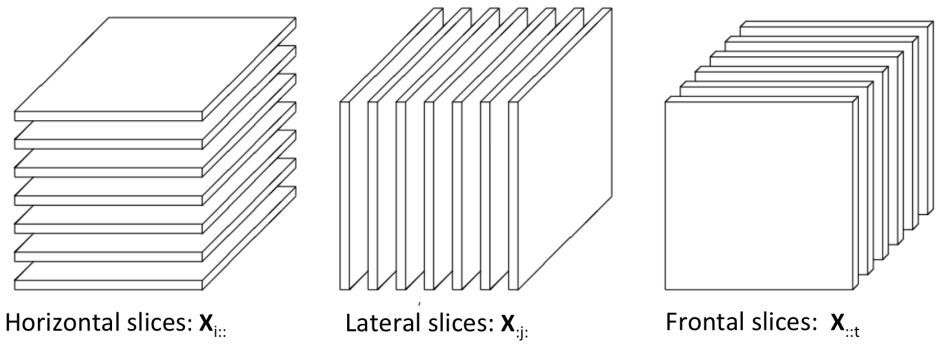

Figure 2.2: Slices of a 3^{rd}-order tensor [38].

$$\mathbf{X}_1 = \begin{bmatrix} 1 & 4 & 7 & 10 \\ 2 & 5 & 8 & 11 \\ 3 & 6 & 9 & 12 \end{bmatrix}, \quad \mathbf{X}_2 = \begin{bmatrix} 13 & 16 & 19 & 22 \\ 14 & 17 & 20 & 23 \\ 15 & 18 & 21 & 24 \end{bmatrix}.$$

Figure 2.3: The frontal slices of tensor [38].

$$\mathbf{X}_{(1)} = \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \end{bmatrix},$$

$$\mathbf{X}_{(2)} = \begin{bmatrix} 1 & 2 & 3 & 13 & 14 & 15 \\ 4 & 5 & 6 & 16 & 17 & 18 \\ 7 & 8 & 9 & 19 & 20 & 21 \\ 10 & 11 & 12 & 22 & 23 & 24 \end{bmatrix},$$

$$\mathbf{X}_{(3)} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & \cdots & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & \cdots & 21 & 22 & 23 & 24 \end{bmatrix}.$$

Figure 2.4: The three mode-n of tensor [38].

2.1.3 Real vector space

For the sake of simplicity, we introduce an intuitive version of real vector space from [70]. Ones whose interested in mathematical definition may referred to several stand algebra books, for example [4, 10], etc.

Definition 1 *A vector space $\mathcal{V}$ over $\mathbb{R}$, or a real vector space $\mathcal{V}$, is a set of objects, known as vectors, together with vector addition "+" and multiplication of vectors by element of $\mathbb{R}$, and satisfying the following properties:*

- *(VA1) For every $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{V}$, we have $\boldsymbol{x} + \boldsymbol{y} \in \mathcal{V}$.*
- *(VA2) For every $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z} \in \mathcal{V}$, we have $\boldsymbol{x} + (\boldsymbol{y} + \boldsymbol{z}) = (\boldsymbol{x} + \boldsymbol{y}) + \boldsymbol{z}$.*

- *(VA3) There exists an element* $\boldsymbol{0} \in \mathcal{V}$ *such that for every* $\boldsymbol{x} \in \mathcal{V}$*, we have* $\boldsymbol{x} + \boldsymbol{0} = \boldsymbol{0} + \boldsymbol{x} = \boldsymbol{x}$*.*
- *(VA4) For every* $\boldsymbol{x} \in \mathcal{V}$*, there exists* $-\boldsymbol{x} \in \mathcal{V}$ *such that* $\boldsymbol{x} + (-\boldsymbol{x}) = \boldsymbol{0}$*.*
- *(VA5) For every* $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{V}$*, we have* $\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{y} + \boldsymbol{x}$*.*
- *(SM1) For every* $\alpha \in \mathbb{R}$ *and* $\boldsymbol{x} \in \mathcal{V}$*, we have* $\alpha\boldsymbol{x} \in \mathcal{V}$*.*
- *(SM2) For every* $\alpha \in \mathbb{R}$ *and* $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{V}$*, we have* $\alpha(\boldsymbol{x} + \boldsymbol{y}) = \alpha\boldsymbol{x} + \alpha\boldsymbol{y}$*.*
- *(SM3) For every* $\alpha, \beta \in \mathbb{R}$ *and* $\boldsymbol{x} \in \mathbb{R}$*, we have* $(\alpha + \beta)\boldsymbol{x} = \alpha\boldsymbol{x} + \beta\boldsymbol{x}$*.*
- *(SM4) For every* $\alpha, \beta \in \mathbb{R}$ *and* $\boldsymbol{x} \in \mathbb{R}$*, we have* $(\alpha\beta)\boldsymbol{x} = \alpha(\beta\boldsymbol{x})$*.*
- *(SM5) For every* $\boldsymbol{x} \in \mathcal{V}$*, we have* $1\boldsymbol{x} = \boldsymbol{x}$*.*

Note that the elements $\alpha, \beta, \gamma \in \mathbb{R}$ discussed in (SM1)-(SM5) are known as scalars. Multiplication of vectors by elements of $\mathbb{R}$ is sometimes known as scalar multiplication.

2.1.4 Vector spaces $\mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$

The following definition of vector spaces $\mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$ is cited from [21]

Definition 2 *Let* $\mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$ *be the set of all* N*-way array of size* $d_1 \times d_2 \times \ldots \times d_N$*,* $\mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$ *is a vector space of dimension* $d_1 d_2 \ldots d_N$ *together with the following operations*

- *Addition*

$$\underline{\boldsymbol{A}} + \underline{\boldsymbol{B}} := \underline{\boldsymbol{C}}, \quad \text{where} \quad c_{d_1 d_2 \ldots d_N} = a_{d_1 d_2 \ldots d_N} + b_{d_1 d_2 \ldots d_N} \tag{2.1}$$

  for all $\underline{\boldsymbol{A}}, \underline{\boldsymbol{B}} \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$*.*

- *Scalar multiplication*

$$\lambda\underline{\boldsymbol{A}} := \underline{\boldsymbol{B}}, \text{where} \quad b_{d_1 d_2 \ldots d_N} = \lambda a_{d_1 d_2 \ldots d_N}, \tag{2.2}$$

  for all $\underline{\boldsymbol{A}} \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$ *and* $\lambda \in \mathbb{R}$*.*

2.2 What is Tensor?

Tensor have been widely studied in various fields including mathematics, physics, engineering and have became a trend in data mining and data analytics as the rapidly increasing of big data. But when reviewing papers and application works, we saw that the definition of tensor in some papers are somewhat vague. For example the term "a N-way data" is often referred to "a N^{th}-order tensor" or these two terms are used exchangeably. Also, when looking for documents with key word "tensor", ones may see some terms related to "tensor", for instance "metric tensor", "stress tensor", "Riemann curvature tensor". To

answer the question "what is tensor?" and which "definition" we should be studied and applied in our research scope, we looking for the answer in the popular papers, books and research documents related to tensor.

Very clearly answers for the above questions can be found in [21]:

> "Even though tensors are well-studied objects in the standard graduate mathematics curriculum [4, 25, 36, 42, 58] and more specifically in multilinear algebra [10, 30, 49, 55, 66], a "tensor" continues to be viewed as a mysterious object by outsiders. We feel that we should say a few words to demystify the term.
>
> In mathematics, the question "what is a vector?" has the simple answer "a vector is an element of a vector space" in other words, a vector is characterized by the axioms that define the algebraic operations on a vector space. In physics, however, the question what is a vector? often means "what kinds of physical quantities can be represented by vectors?" The criterion has to do with the change of basis theorem: an n-dimensional vector is an "object" that is represented by n real numbers once a basis is chosen only if those real numbers transform themselves as expected when one changes the basis. For exactly the same reason, the meaning of a tensor is obscured by its more restrictive use in physics. In physics (and also engineering), a tensor is an "object" represented by a k-way array of real numbers that transforms according to certain rules (cf. (2.2)) under a change of basis. In mathematics, these "transformation rules" are simply consequences of the multilinearity of the tensor product and the change of basis theorem for vectors."

Similar discussions also can be found in [38, 46]. In many documents, term "3^{rd} order tensor" (or "3-way tensor") is identical with the 3-way array $\underline{\mathcal{A}}$ which is indeed a representation of tensor $\underline{\mathcal{X}}$ when the bases are fixed as point out in later section 2.3. In our understanding, the reasons of this situation are formal definition of tensor and related properties in multilinear algebra are complicated and many common used results in data mining can be stated without introducing arrays of coordinates as point out in [19,38,41].

2.3 Tensor and CP-decomposition

In this section, we give definition of tensor and summary some important results from [21,46]. Ones who interested in tensor, tensor product of vector spaces and related results may referred to several papers such as [19, 21, 47, 69] and standard algebra books [4, 10, 21, 25, 30, 36, 42, 49, 55, 58, 66].

Definition 3 *Let $\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_N$ be real vector spaces of dimensions $d_1, d_2, \ldots, d_N$, respectively, tensor product space $V_1 \otimes V_2 \otimes \ldots \otimes V_N$ is vector space of dimension $d_1 d_2 \ldots d_N$; element $\underline{\mathcal{X}} \in \mathcal{V}_1 \otimes \mathcal{V}_2 \otimes \ldots \otimes \mathcal{V}_N$ is called a tensor of order N.*

When we fix a choice of basis $\{\mathbf{e}^n_{d_{in}} | d_{i_n} = 1, 2, \ldots, d_n\}$, for $\mathcal{V}_n$, respectively, then $\underline{\mathcal{X}}$ has coordinate representation

$$\underline{\mathcal{X}} = \sum_{d_{i_1}=1}^{d_1} \ldots \sum_{d_{i_N}=1}^{d_N} x_{d_{i_1} \ldots d_{i_N}} \mathbf{e}^1_{d_{i_1}} \otimes \ldots \otimes \mathbf{e}^N_{d_{i_N}}. \tag{2.3}$$

The coefficients form a N-way array, $\underline{\mathcal{A}} \in \mathbb{R}^{d_1 \times \ldots \times d_N}$. Also, element of an d_n-dimensional vector space $\mathcal{V}_n$ may be represented by an d_n-tuple of numbers in $\mathbb{R}^{d_n}$ up to a choice of basis, for $n = 1, 2, \ldots, N$ [21].

Before giving the formulation of CP-decomposition, we summarize results from [21] which may give insight view about the relations among some related terms such as "tensor","N-way array", "tensor product", "outer product", etc., and the first imagination about CP-decomposition (or outer-product decomposition of a tensor).

Let $\mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \ldots \otimes \mathbb{R}^{d_N}$ be the tensor product of the vector spaces $\mathbb{R}^{d_1}, \mathbb{R}^{d_2}, \ldots, \mathbb{R}^{d_N}$ and let ϕ be the *Segre map* which is a multilinear such that

$$\begin{aligned} \phi : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \times \ldots \times \mathbb{R}^{d_N} &\longrightarrow \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N} \\ (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) &\longmapsto \phi(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) = \underline{\mathcal{X}}, \end{aligned} \tag{2.4}$$

where the $(i_{d_1}, i_{d_2}, \ldots, i_{d_1})$-th element of $\underline{\mathcal{X}}$ is defined as

$$x_{i_{d_1} i_{d_2} \ldots i_{d_1}} = x_{1 i_{d_1}} x_{2 i_{d_2}} \ldots x_{N i_{d_N}}. \tag{2.5}$$

From the universal property of tensor product we have a unique linear map θ such that the diagram in Figure 2.5 commutes, i.e,

$$\phi((\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)) = \theta(\otimes((\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N))) = \theta((\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \ldots \otimes \mathbf{x}_N)), \tag{2.6}$$

for all $(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \in \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \times \ldots \times \mathbb{R}^{d_N}$. In other word, we have $\phi = \theta \mathbf{o} \otimes$, where "$\theta \mathbf{o} \otimes$" is composite function.

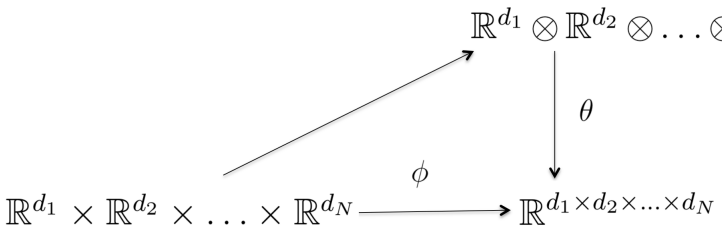

Figure 2.5: Commutative diagram [21].

Note that result of *Segre map* in Equation 2.5 can be write in form of outer product such that $\phi((\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)) = \mathbf{x}_1 \circ \mathbf{x}_2 \circ \ldots \circ \mathbf{x}_N$, where "$\circ$" is the outer product of vectors. Equation 2.6 implies that

$$\theta((\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \ldots \otimes \mathbf{x}_N)) = \mathbf{x}_1 \circ \mathbf{x}_2 \circ \ldots \circ \mathbf{x}_N. \tag{2.7}$$

Furthermore, θ is a vector space isomorphism since $dim(\mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \ldots \otimes \mathbb{R}^{d_N}) = dim(\mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}) = d_1 d_2 \ldots d_N$. It implies that tensor product of vector $\mathbf{x}_1 \otimes \mathbf{x}_2 \otimes \ldots \otimes \mathbf{x}_N$ and the N-way array $\underline{\mathcal{X}} = \phi((\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)) = \mathbf{x}_1 \circ \mathbf{x}_2 \circ \ldots \circ \mathbf{x}_N$ are equivalent objects which belong to equivalent vector spaces $\mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \ldots \otimes \mathbb{R}^{d_N}$ and $\mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$, respectively. Such equivalences allows us to not distinguish between these two spaces, and both $\underline{\mathcal{X}} \in \mathbb{R}^{d_1} \otimes \mathbb{R}^{d_2} \otimes \ldots \otimes \mathbb{R}^{d_N}$ and $\theta(\underline{\mathcal{X}}) \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$ can be called tensor.

Such results provide great advantages for data mining applications related to N-way array data because N-way array (consider as element of $\mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N}$) may provide nature and compact representation for data and target problem may be solve using tensor space model based methods [3, 13, 17, 34, 38, 50, 56, 59, 69]. From now on, we only focus on the special case of 3-way tensors which is indeed 3 way arrays in $\mathbb{R}^{I \times J \times T}$ because it is enough for understanding the content of later parts with notice that many important results are originally proved on $\mathbb{R}^I \otimes \mathbb{R}^J \otimes \mathbb{R}^T$.

Rank-1 tensor or decomposable tensor is a special kind of tensor and is necessary to understand CP-decomposition [21, 38].

Definition 4 *A 3^{rd}-order tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I \times J \times T}$ is of rank one if it can be written as the outer product of 3 vectors, i.e.,*

$$\underline{\mathcal{X}} = \boldsymbol{a} \circ \boldsymbol{b} \circ \boldsymbol{c}, \tag{2.8}$$

where $\boldsymbol{a} \in \mathbb{R}^I$, $\boldsymbol{b} \in \mathbb{R}^J$ and $\boldsymbol{c} \in \mathbb{R}^T$.

CP-decomposition is used to decompose a tensor into a sum of component rank-one tensors [19, 38]. Theoretically, any tensor $\underline{\mathcal{X}}$ can be decomposed (non uniquely) into a linear combination of decomposable tensors or in other words, given a tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I \times J \times T}$, we can find $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$ and $\mathbf{c}_r \in \mathbb{R}^T$ such that

$$\underline{\mathcal{X}} = \sum_{1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \tag{2.9}$$

where $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$ and $\mathbf{c}_r \in \mathbb{R}^T$.

Elementwise, equation 2.9 is written as

$$x_{ijt} = \sum_{r=1}^{R} \lambda_r a_{ri} b_{rj} c_{rt}, \tag{2.10}$$

for $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$, $t = 1, 2, \ldots, T$ and $r = 1, 2, \ldots, R$.

CP-decomposition of a 3^{rd}-order tensor is illustrated in Figure 2.6

Matrices $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_R] \in \mathbb{R}^{I \times R}$, $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_R] \in \mathbb{R}^{J \times R}$ and $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_R] \in \mathbb{R}^{T \times R}$ are called component matrices.

There are some kind of ranks defined on tensors, the most common ones called CP-rank, outer product rank or in short $rank(.)$ [21, 38].

Definition 5 *The rank of a tensor $\underline{\mathcal{X}}$, denoted $rank(\underline{\mathcal{X}})$, is defined as the smallest number R of rank-one tensors that generate $\underline{\mathcal{X}}$ as their sum.*

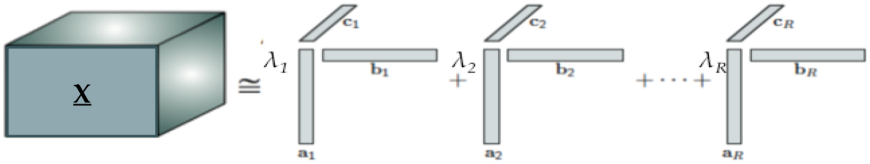

Figure 2.6: CP decomposition of a third order tensor [38].

For a 3^{rd} order tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I\times J\times T}$, the lower and upper bounds of number R are $max\{I, J, T\}$ and $min\{IJ, JT, IT\}$, respectively. Or in other words, we have the following inequality

$$max\{I, J, T\} \leq R \leq min\{IJ, JT, IT\} \tag{2.11}$$

For detail about the bounds of tensor rank, ones may refer to some documents [20, 35, 38, 40], etc.

Another important remark is that in Equation 2.9, we see that for any 3^{rd}-order tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I\times J\times T}$ can be decomposed into a linear combination of decomposable tensors but we did not see how to find such combination for a given tensor. In our point of view, there is no available model to find an exactly decomposition for a given tensor. Generally, CP-decomposition is used to determine the feasible solution region for the approximation problems. And based on the purpose of the approximation problems, the objective functions are constructed and are solved to find the optimal solution/solutions on the determined feasible solution region. The class of models follow this procedures is called PARAFAC family and is presented in Section 2.4.

2.4 PARAFAC family

PARAFAC family consists of PARAFAC model and other models, which have relaxed the restrictions enforced by a PARAFAC model to capture data-specific structures [3]. The simplest version of PARAFAC model is R-component PARAFAC model. Given a tensor $\underline{\mathcal{X}}$ and a number R, R-component PARAFAC model used to find a linear combination of R-decomposable tensors which is best approximates the given tensor $\underline{\mathcal{X}}$ [3, 14, 15, 20, 21, 32, 37, 38]. In other words, R-component PARAFAC model find three component matrices $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_R] \in \mathbb{R}^{I\times R}$, $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_R] \in \mathbb{R}^{J\times R}$ and $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_R] \in \mathbb{R}^{T\times R}$ by solving the following optimization problem

$$\min_{\mathbf{A}\in\mathbb{R}^{I\times R}, \mathbf{B}\in\mathbb{R}^{J\times R}, \mathbf{C}\in\mathbb{R}^{T\times R}} \|\underline{\mathcal{X}} - \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r\|_F, \tag{2.12}$$

where F is Frobenius norm.

Note that component matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are determined uniquely up to a permutation and scaling of columns [3]. Permutation of column mean that if

$$\{\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_R] \in \mathbb{R}^{I\times R}, \mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_R] \in \mathbb{R}^{J\times R}, \mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_R] \in \mathbb{R}^{T\times R}\}$$

is an optimal solution of optimization 2.12 and σ is a permutation on $\{1, 2, \ldots, R\}$ such that

$$\sigma(1, 2, \ldots, R) = (\sigma(1), \sigma(2), \ldots, \sigma(R)) \tag{2.13}$$

then

$$\{\mathbf{A}^{'} = [\mathbf{a}_{\sigma(1)}, \ldots, \mathbf{a}_{\sigma(R)}], \mathbf{B}^{'} = [\mathbf{b}_{\sigma(1)}, \ldots, \mathbf{b}_{\sigma(R)}], \mathbf{C} = [\mathbf{c}_{\sigma(1)}, \ldots, \mathbf{c}_{\sigma(R)}]\}$$

is also an optimal solution. And scaling of columns mean that

$$\{\mathbf{A} = [\alpha_1\mathbf{a}_1, \ldots, \alpha_R\mathbf{a}_R], B = [\beta_1\mathbf{b}_1, \ldots, \beta_R\mathbf{b}_R], \mathbf{C} = [\gamma_1\mathbf{c}_1, \ldots, \gamma_R\mathbf{c}_R]\}$$

with $\{\alpha_r, \beta_r, \gamma_r \in \mathbb{R} | \alpha_r\beta_r\gamma_r = 1, r = 1, 2, \ldots, R\}$ is also an optimal solution of optimization problem 2.12.

To avoid the scaling solutions which may make the problem become more complicated, the result of optimization problem 2.12 is usually normalized and the optimization problem is reformed as follows

$$\min_{\mathbf{A}\in\mathbb{R}^{I\times R}, \mathbf{B}\in\mathbb{R}^{J\times R}, \mathbf{C}\in\mathbb{R}^{T\times R}} \|\underline{\mathcal{X}} - \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r\|_F, \tag{2.14}$$

where $\{\|\mathbf{a}_r\| = \|\mathbf{b}_r\| = \|\mathbf{c}_r\| = 1 | r = 1, 2, \ldots, R\}$.

Although there is no model can find the exactly number R for a given tensor, several extension models of PARAFAC can be used to determine an appropriate number R by capturing data-specific structures [3]. Because extensions of PARAFAC are complicated and determining an appropriate number R for tensor data requires additional background, we leave those models as a future research and just list some popular models and their important characteristics in Table 2.2.

Table 2.2: Models in PARAFAC family [3].

Model name	Mathematical formulation	Handles rank-efficiency	Extend to N-way array
PARAFAC	$x_{ijt} = \sum_{r=1}^{R} a_{ri}b_{rj}c_{rt} + e_{ijt}$	No	Yes
PARAFAC2	$\mathbf{X}_t = \mathbf{A}_t\mathbf{D}_t\mathbf{B}^T + \mathbf{E}_t$	No	Yes
S-PARAFAC	$x_{ijt} = \sum_{r=1}^{R} a_{r(i+s_{jr})}b_{rj}c_{rt} + e_{ijt}$	No	Yes
PARALIND	$\mathbf{X}_t = \mathbf{A}\mathbf{H}\mathbf{D}_t\mathbf{B}^T + \mathbf{E}_t$	Yes	Yes
cPARAFAC	$x_{ijt} = \sum_{r=1}^{R} a_{ri}b^{r(j-\theta)}c_{rt}^{\theta} + e_{ijt}$	No	Yes

From literature reviews, we see that applications of PARAFAC model on 3^{rd}-order tensor usually consist of two stages: firstly, assuming that data have multiway structure and representing the data in form of 3^{rd}-order tensors; then, PARAFAC model is employed to carry out three component matrices which can be understood as a new representation with certain multiway structure [3, 11, 20, 38]. Note that certain multiway structure in

this case mean that data can represented in form of a 3-order tensor which is then can be represented in CP-decomposition form.

Furthermore, when working with 3^{rd}-order tensor, the number of elements that we have to store and manipulate is larger and rapidly increasing when the size of ways increase. It is also important to find a new representation which requires less storage memory like CP-decomposition form. For example, when working with a 3^{rd}-order tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I \times J \times T}$, we have to store and manipulate totally IJT elements while working with R-component CP-decomposition, ones only need to concern $R(I+J+T)$ elements. As point out in literature reviews [20,29,38], finding good approximation with small R have been widely studied and numerous fast and efficient algorithms have been proposed, improved and implemented on popular softwares (for example MATLAB), tensor decomposition models (in particularly, PARAFAC model) have become promising solution to handle the multiway array data.

Chapter 3

CP-decomposition based temporal link prediction on open bipartite networks evolve over time

The first part of this chapter present necessary concepts related to temporal link prediction, open bipartite networks and an introduction to temporal link prediction on bipartite networks. Then the second part provides the related works need to understand the proposed temporal link prediction method for open bipartite networks. In the third part, the proposed method is presented. Finally, discussions and future works is given in the fourth part.

3.1 Introduction

The temporal link prediction is a common problem for data which is assumed to have the underlying periodic structure. The problem of temporal link prediction can be summarized as follows

Definition 6 *Given link data for times* 1 *through* T^{th}*, temporal link prediction is the problem of predicting the links at time* $(T+1)^{th}$*.*

Of course this is a general problem and can be found in many kinds of data. In this chapter, we restrict the problem on one special kind of data which can be represented by bipartite networks whose links evolve over time.

Definition 7 *Bipartite networks are a common type of network data in which there are two types of vertices, and only vertices of different types can be connected by links [2,26,43].*

In a bipartite network that evolve over time, two sets of vertices are fixed and only the links between different kind of vertices change. Figure 3.1 is an example of bipartite networks with 5 vertices of each type.

Bipartite networks can be used to represent various kinds of structures, dynamics, and interaction patterns found in social activities [8, 22, 52, 53, 67]. For example, bipartite

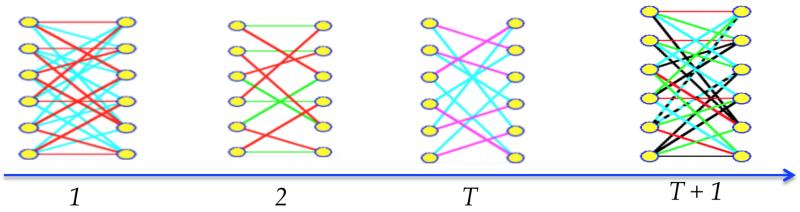

Figure 3.1: An example of bipartite network that evolve over time.

network is used to represent the network consists of I authors and J conferences where each link represents the possibility that an author publishes on a conference and temporal link prediction is used to predict which authors will publish at which conferences in year $(T+1)^{th}$ given the publication data for the T previous years [2,26]; I users/group of users and J providers and their relations in a recommendation systems can be represented by a bipartite network [67]; Another example is the bipartite network consists of I patient groups and J drugs with the links represent the adverse effect of drugs.

In real applications, we can face some special bipartite networks with common situation that new vertices may join network at time T^{th} and may link to other vertices at time $(T+1)^{th}$. There are numerous examples about such kind of networks. For example, when consider a recommendation system, new users/group of users appear and may interest or use service of providers in next time points; also in drug side effect example, new drugs are launched and may have adverse effect on patient groups in the future; etc.

For simplicity, let call such networks by open bipartite networks to distinguish from bipartite networks without new added vertices. We define such kind of networks as in Definition 8 and give illustration in Figure 3.2.

Definition 8 *An open bipartite network is a bipartite network in which new subjects may join the network at time T and may link to objects at time $(T+1)^{th}$.*

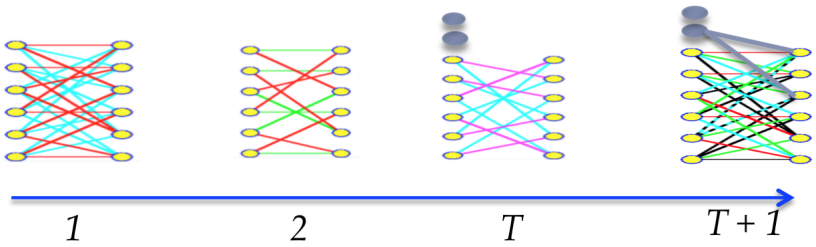

Figure 3.2: An example of open bipartite network that evolve over time.

Several temporal link prediction methods have been proposed for bipartite networks [2, 26] but in our knowledge, no temporal link prediction method proposed to deal with problem on open bipartite networks. In our point of view, temporal link prediction on open bipartite networks is challenging because it is difficult to learn how new vertices will link to

other vertices and how new added vertices effect links among previous vertices at time $(T+1)^{th}$. To do temporal link prediction on open bipartite networks, we extend temporal link prediction method using CP-decomposition in [2, 26] to address temporal link prediction in open bipartite networks in which new vertices of type-1 may join networks at time T^{th}.

3.2 Related works

Temporal link prediction methods for bipartite networks assume that link data have an underlying periodic structure and apply time series techniques to predict future links [2, 5, 23, 26, 67]. Almost traditional methods apply time series techniques directly on the link-data to predict future links. This schema is simple in implementation but have limitations such as high complexity, memory consuming and difficulty in exploiting the natural three-dimensional structure of temporal link data. Recently, temporal link prediction method using CP-decomposition were proposed and shown its power in exploring the structure of data, requiring less memory and giving outperformed experimental results comparing with the traditional methods [2, 26]. Because the proposed method for open bipartite networks can be seen as an extension of the method in [2, 26], understanding the key ideas of the method proposed in [2, 26] is necessary.

3.2.1 Temporal link prediction using CP-decomposition

In the method proposed in [2, 26], a 3^{rd}-order tensor is used to represent the link information data; CP-decomposition employed to explore the three dimensional structure of data; and the link information in the next time point is predicted using temporal forecasting techniques.

Figure 3.3 gives an illustration of consisting steps in this method.

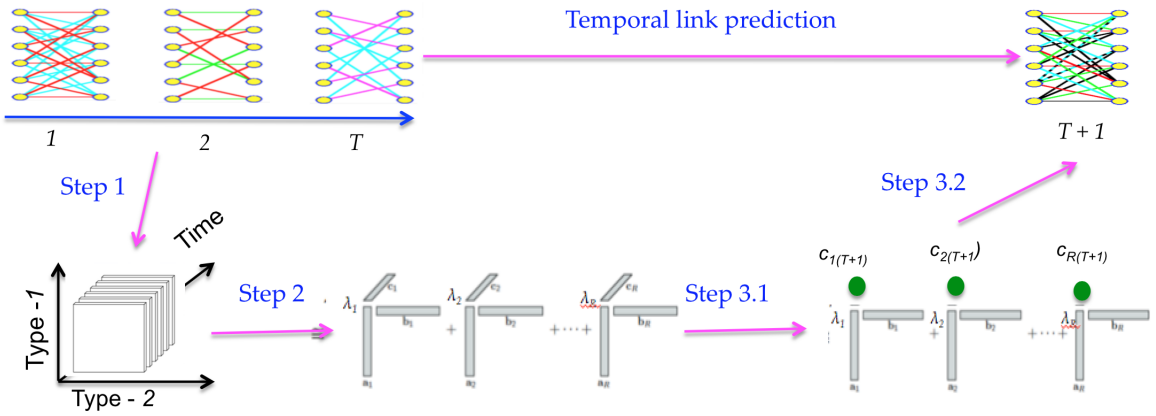

Figure 3.3: Illustration of temporal link prediction method proposed in [2, 26].

The method can be summarized as follows

- **Step 1:** *Data representation*

  The data is organized as a 3^{rd}-order tensor $\underline{\mathcal{X}}$ of size $I \times J \times T$, where x_{ijt} represents weight of the link between type-1 vertex i^{th} and type-2 vertex j^{th} at time t^{th}.

- **Step 2:** *CP-decomposition*

  Carry out CP-decomposition method to get three-component matrices:

$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_R] \in \mathbb{R}^{I \times R}, \tag{3.1}$$

$$\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_R] \in \mathbb{R}^{J \times R}, \tag{3.2}$$

$$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_R] \in \mathbb{R}^{T \times R}. \tag{3.3}$$

  where $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$ and $\mathbf{c}_r \in \mathbb{R}^T$, such that

$$\underline{\mathcal{X}} = \sum_{1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r. \tag{3.4}$$

  In Equation 3.4, $\{\mathbf{a}_r | r = 1, 2, \ldots, R\}$, $\{\mathbf{b}_r | r = 1, 2, \ldots, R\}$ and $\{\mathbf{c}_r | r = 1, 2, \ldots, R\}$ are the type-1 vertex, type-2 vertex and time factors, respectively. Furthermore, $\|\mathbf{a}_r\| = \|\mathbf{c}_r\| = \|\mathbf{c}_r\| = 1$ and are obtained by normalizing the result from the following optimization problem

$$\min_{\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r} = \|\underline{\mathcal{X}} - \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r\|_2^2, \tag{3.5}$$

  and $\lambda_r = \|\mathbf{a}_r\| \|\mathbf{b}_r\| \|\mathbf{c}_r\|$, for $r = 1, 2, \ldots, R$.

- **Step 3:** *Temporal forecasting*

  Using assumption that time factors inherits the underlying periodic properties of data, we can predict value of time factors at time $(T+1)^{th}$, denoted by γ_r, from elements of vector $\mathbf{c}_r = (c_{r1}, c_{r2}, \ldots, c_{rT})$ and the links of network at time $(T+1)^{th}$ by using temporal forecasting method as follows

  Step 3.1: *Predict value of time factors at time $(T+1)^{th}$ from its values at T previous time.*

  The temporal profiles are captured in the vectors $\mathbf{c}_r$, $r = 1, 2, \ldots, R$. Different components may have different trends, for example, they may have increasing, decreasing, or steady profiles. For each time factor vector r, its value at time $(T+1)^{th}$ is estimated by

$$\gamma_r = \frac{1}{T_0} \sum_{t=T-T_0-1}^{T} c_{rt}, \tag{3.6}$$

  where K_0 is prior number.

Alternatively, we can use the temporal profiles computed by CP-decomposition as a basis for predicting the scores in future [16].

Step 3.2: *Predict the links of network at time* $(T+1)^{th}$.

We define the similarity score for type-1 vertex i^{th} and type-2 vertex j^{th} using results from R-component CP model in Equation 3.4 and results in Step 3.1 as the (i,j) entry of the following matrix

$$\mathbf{X}_{T+1} = \sum_{r=1}^{R} \lambda_r \gamma_1 \mathbf{a}_r \circ \mathbf{b}_r, \tag{3.7}$$

or in other words, weight of links between type-1 vertex i^{th} and type-2 vertex j^{th} at time $(T+1)^{th}$ is

$$x_{(T+1)ij} = \sum_{r=1}^{R} \lambda_r \gamma_r a_{ri} b_{rj}, \tag{3.8}$$

for $i = 1, 2, \ldots, I$ and $i = 1, 2, \ldots, J$.

3.3 The proposed method

In this section, we present a proposed method to predict the weights of links at time $(T+1)^{th}$ of an open bipartite network whose the node set consists of I type-1 vertices and J type-2 vertices, and N new vertices of type-1 join network at time T. The proposed method was presented at ACIS2014 in December 2014. For the sake of an easy understanding and clear presentation, we revised some parts from the paper at ACIS2014 where the key ideas and contributions are the same.

3.3.1 Problem formulating

The method proposed in [2, 26] work for bipartite networks when the weights of all links through T previous time points are given. The key idea is to employ CP-decomposition the weight into three separated factors, each fluctuates independently from others. From this assumption, it is clear that if we known the value of time factors at time $(T+1)^{th}$, then the weights of links at time $(T+1)^{th}$ can be found by combining the values of time factors at time $(T+1)^{th}$ with values of vertex factors of type-1 and type-2. Furthermore, as time factors are assumed to be inherited the underlying structure of weight data, it allows to predict the values of time factors at time $(T+1)^{th}$ from values of time factors in T previous time points. The method works without any addition information than weights of links in T previous time points.

Considering the problem of predicting the weights of links at time $(T+1)^{th}$ of an open bipartite networks whose a new vertex of type-1 join networks at time T and may link to vertices of type-2 at time $(T+1)$. Furthermore, it may also have effect on the fluctuations of weights on the whole networks though the structure of networks is changed. Then, it is easy to see that the main challenges/tasks in such problem are

- to learn the effect of new vertex of type-1 on the fluctuations of weights on the whole networks,
- and to predict the weights at time $(T+1)^{th}$.

If we employ the same assumptions, apply CP-decomposition to separate weights of links into 3 factors and predict the values of time factors as method in [2, 26], then we only have to predict the values of type-1 vertices factors corresponding to new vertex of type-1 in order to predict the weights at time $(T+1)^{th}$. Without additional information about or related to type-1 vertex factors, we can not predict the values corresponding to new vertex. Our solution for this task is to collect additional information of all type-1 vertices and learn a function which can predict values of type-1 vertex factors from the additional information and use to predict values of type-1 factors corresponding to new vertex. In case N new vertices of type-1 join network at time T, we use the function to predict all corresponding values of type-1 vertex factors. In other words, values of type-1 factors corresponding to new vertices are learned form additional information.

This schema implies that the values of factors corresponding to new vertices do not effect on learning the values of time factors corresponding to time $(T+1)^{th}$, then they do not effect on the weights of links between other vertices of type-1 and vertices of type-2. Also the proposed method is in fact the same with the method in [2, 26] when no new vertex join the networks.

The input and output of the proposed temporal link prediction method can be summarized as in Algorithm 1.

Algorithm 1: Temporal link prediction on an open bipartite network

Input:

1. A set of information of I type-1 vertices $S = \{\mathbf{s}_i | \mathbf{s}_i \in \mathbb{R}^P, i = 1, 2, \ldots, I\}$.
2. A set of information of N new vertices of type-1 $Q = \{\mathbf{q}_n | \mathbf{q}_n \in \mathbb{R}^P, n = 1, 2, \ldots, N\}$.
3. A weight tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I \times J \times T}$, where the (i, j, t)-th element x_{ijt} is the weight corresponding to link between type-1 vertex i^{th} and type-2 vertex j^{th} at time t^{th}, for $i = 1, 2, \ldots, I$, $j = 1, 2, \ldots, J$ and $t = 1, 2, \ldots, T$.

Output: A matrix $\mathbf{X}_{T+1} \in \mathbb{R}^{(I+N) \times J}$, with $x_{(T+1)ij}$ is the predicted information of link between type-1 vertex i^{th} and type-2 vertex j^{th} at time $(T+1)^{th}$, for $i = 1, 2, \ldots, I, I+1, \ldots, I+N$ and $i = 1, 2, \ldots, J$.

3.3.2 Assumptions

The proposed methods work under some assumptions as follows

- **Assumption 1:** When no vertices join network, weights data have underlying periodic structure or in other words, the weights at time $(T+1)^{th}$ can be predict from weights through T previous time points.
- **Assumption 2:** Information of vertices of type-1 is available and encoded in form of vectors in $\mathbb{R}^P$, P is a non-negative integer number.
- **Assumption 3:** For each type-1 vertex factor $\mathbf{a}_r$, we can learn a function f_r such that $a_{ri} = f_r(s_i)$, where s_i is information of type-1 vertex i^{th}.

3.3.3 Method

The assumptions in Section 3.3.2 are employed to construct a temporal link prediction method as follows

- **Step 1:** *Data representation*

  The weights of links through T time points are organized as a 3^{rd}-order tensor $\underline{\mathcal{X}}$ of size $I \times J \times T$, where x_{ijt} represents weight of the link between type-1 vertex i^{th} and type-2 vertex j^{th} at time t^{th}.

  Information of I type-1 vertices is represented in form of a matrix $\mathbf{S} \in \mathbb{R}^{I\times(P+1)}$, where the i^{th} row is the row vector $(1, s_{i1}, \ldots, s_{iP})$, for $i = 1, 2, \ldots, I$.

  Information of N new vertices of type-1 is represented in form of a matrix $\mathbf{Q} \in \mathbb{R}^{N\times(P+1)}$, where the n^{th} row is the row vector $(1, q_{n1}, \ldots, q_{nP})$, for $n = 1, 2, \ldots, N$.

- **Step 2:** *CP-decomposition*

  Carry out CP-decomposition method to get three-component matrices:

  $$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_R] \in \mathbb{R}^{I\times R}, \tag{3.9}$$
  $$\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_R] \in \mathbb{R}^{J\times R}, \tag{3.10}$$
  $$\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_R] \in \mathbb{R}^{T\times R}. \tag{3.11}$$

  where $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$ and $\mathbf{c}_r \in \mathbb{R}^T$, such that

  $$\underline{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r. \tag{3.12}$$

- **Step 3:** *Predict the values of type-1 vertex factors corresponding to new vertices*

  In this works, we assume that functions f_r is linear or in other words, for each r, we have

  $$f_r(s_i) = \alpha_{r0} + \sum_{p=1}^{P} \alpha_{rp} s_{ip}, \tag{3.13}$$

  for $i = 1, 2, \ldots, I$.

For each function f_r, we use least square method [33] to estimate vector $\alpha_r = (\alpha_{r0}, \alpha_{r1}, \ldots, \alpha_{rP})^T \in \mathbb{R}^{P+1}$ as follows

$$\alpha_r = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\mathbf{a}_r, \tag{3.14}$$

where $\mathbf{S}^T$ is transpose matrix of $\mathbf{S}$. For each new type-1 vertex n and each type-1 vertex factor, we find the corresponding value

$$s_{rn}^{'} = \alpha_{r0} + \sum_{p=1}^{P} \alpha_{rp} s_{ip}, \tag{3.15}$$

for $n = 1, 2, \ldots, N$ and $r = 1, 2, \ldots, R$.

Concisely, we can found matrix $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_R] \in \mathbb{R}^{R\times(P+1)}$ such that

$$\alpha = (\mathbf{S}^T\mathbf{S})^{-1}\mathbf{S}^T\mathbf{A}, \tag{3.16}$$

and then find a matrix $\mathbf{S}^{'}$ such that

$$\mathbf{S}^{'} = \mathbf{Q}\alpha, \tag{3.17}$$

where the $(r, n)^{th}$ element of $\mathbf{S}^{'}$ is indeed $s_{rn}^{'}$ in Equation 3.15.

For each r, we form column vector $\mathbf{d}_r \in \mathbb{R}^D$, where $D = I + N$, such that

$$\mathbf{d}_r = (a_{r1}, \ldots, a_{rI}, s_{r1}^{'}, \ldots, s_{rN}^{'})^T. \tag{3.18}$$

Note that forming vectors $\mathbf{d}_r$ only help to make the computational procedure in later steps more concisely.

- **Step 4:** *Temporal forecasting*

  Using assumption that time factors are inherited the underlying periodic properties of data, we can predict value of time factors at time $(T+1)^{th}$, denoted by γ_r, from elements of vector $\mathbf{c}_r = (c_{r1}, c_{r2}, \ldots, c_{rT})$. Then the links of network at time $(T+1)^{th}$ by combining γ_r, $\mathbf{b}_r$ and $\mathbf{d}_r$ estimated from Step 3 by following procedures

  Step 4.1: *Predict values of time factors at time $(T+1)^{th}$ from its values at T previous time.*

  The temporal profiles are captured in the vectors $\mathbf{c}_r$. Different components may have different trends, for example, they may have increasing, decreasing, or steady profiles. For each time factor vector $\mathbf{c}_r$, its value at time $(T+1)^{th}$ is estimated by

$$\gamma_r = \frac{1}{T_0} \sum_{t=T-T_0-1}^{T} c_{rt}, \tag{3.19}$$

  where T_0 is a prior number.

  Step 4.2: *Predict the links of network at time $(T+1)^{th}$.*

We define the similarity score for type-1 vertex d^{th} and type-2 vertex j^{th} using result from R-component CP model in Equation 3.12, and result from Steps 3 and Step 4.1 as the (d, j) entry of the following matrix

$$\mathbf{X}_{T+1} = \sum_{r=1}^{R} \lambda_r \gamma_r \mathbf{d}_r \circ \mathbf{b}_r, \tag{3.20}$$

or in other words, weight of link between type-1 vertex d^{th} and type 2 vertex j^{th} at time $(T+1)^{th}$ is

$$X_{(T+1)dj} = \sum_{r=1}^{R} \lambda_r \gamma_r d_{rd} b_{rj}, \tag{3.21}$$

for $d = 1, 2, \ldots, D$ and $i = 1, 2, \ldots, J$.

Note that, for $d = 1, 2, \ldots, I$, type-1 vertex d^{th} is the previous vertex $i = d$, and for $d = I+1, I+2, \ldots, I+N$, type-1 vertex d^{th} is the new type-1 vertex $n = d - I$.

3.4 Discussions and future works

The proposed method can be applied to do temporal link prediction on open bipartite networks whose N new vertices of type-2 join networks instead of type-1 vertices since the role of type-1 vertices and type-2 vertices are the same in the proposed method.

We also point out that in the proposed method, new vertices do not effect the fluctuation of weights corresponding to links among previous vertices at time $(T+1)^{th}$. Furthermore, the proposed method is an extension of method in [2, 26], then it can be used to extend the application context in that papers.

The proposed method is an intuitive method, it should be applied to analyze datasets in order to evaluate the performance. Unfortunately, because of limited time, we have not implemented the program and tested on real datasets. We leave this task as a future work of this thesis.

In the later research, we plan to focus on the following tasks

1. Collect data and run experiments on real datasets in order to evaluate performance of the proposed method.

2. Extend the proposed method to predict the links for a period of time starting at $(T+1)^{th}$ or in other words, for times $(T+1)^{th}$, ... , $(T+L)^{th}$.

3. Construct temporal link prediction for open bipartite networks when the new vertices of type-1 and type-2 join the concerned networks at the same time.

Chapter 4

CP-decomposition based spectral clustering

This section provides an overview of spectral clustering and suggestions to extend this method to deal with tensorial data. Firstly, we give an overview of spectral clustering including the general schema, opportunity to extend for tensor data. Also, the currents research on clustering methods for tensor data is given. Secondly, a proposed spectral clustering method namely CP-decomposition based spectral clustering is presented. Then, experiment results are analyzed to evaluate performance of the proposed method. Finally, the future plan to improve the proposed methods and suggestion to construct a versatile clustering method based on tensor space model are given.

4.1 Spectral clustering

Spectral clustering has became one of the most popular clustering algorithms [54, 64]. In this section, we present the general schema of spectral clustering and discussions that motivate us focus on studying spectral clustering and extend such methods for tensor data, though there many other clustering methods, for example, K-mean clustering, hierarchical clustering, etc.

4.1.1 General Schema of spectral clustering

Before going to present the general schema of spectral clustering, we provide an intuitive overview about spectral clustering which is summarized from technique papers [54, 64].

In applications deal with empirical data, clustering is use to identify the groups of data which have "similar behavior" and can be seen as the first attempt to learn about the structure of data. It is clear that different ways to define "similar behavior" will motivate people to construct different classes of clustering method.

One intuitive way to define the similar behavior is to assume that the T considered data points, denoted by $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_T$, are embedded into a weighted similarity graph $G = (V, E)$ where each vertex $\mathbf{v}_t$ represent data point $\mathbf{s}_t$ and for each edge $(\mathbf{v}_i, \mathbf{v}_j) \in E$,

its weight $w_{i,j}$ represent the "similarity" between vertex $\mathbf{v}_i$ and $\mathbf{v}_j$ or equivalently the "similarity" between data point $\mathbf{v}_i$ and $\mathbf{s}_j$. Then, the problem of find groups of similar data points change to a relaxation problem that finding groups of similarity vertices in the similarity graph. The relaxation problem is easier to solve because graph and similarity graph have been studied for a long time and many powerful results have been found. By embedding the data points into a similarity graph, we can employ any techniques and results related to similarity graph to do clustering on the embed similarity graph. Finally, the result on the similarity graph can be converted to the original data space though the embedded function is clear.

The clustering methods that follow the above procedure is called spectral clustering methods. Note that "similarity" is a general notation and also there are several techniques to do clustering on embed similarity graph. Spectral clustering methods are distinguished mainly by which similarity measures and clustering techniques on similarity graph are employed. In this thesis, we focus on symmetric similarity measure which used to constructed the embedded undirected graph with the properties that $w_{ij} = w_{ji}$ which is used to distinguished with other kind of graph whose w_{ij} may be different from w_{ji}.

The general schema of spectral clustering used in this thesis can be summarize as in Algorithm 2.

This schema is similar to schema in other papers [54, 64]. The only difference here is that we do not mention about any certain similarity measure as it should be chosen appropriately for each kind of data.

The matrix $\mathbf{L}$ in step 2 is called normalized Laplacian matrix. For other kinds of Laplacian and their applications in spectral clustering, ones are suggested to refer technical documents and survey papers [31, 64], etc.

4.1.2 Motivation and opportunity for tensor data

Before presenting the reasons that motivate us to study and extend spectral clustering methods for tensor data, we summarize two general criterias which have been used to compare clustering methods from [57] as follows

1. How easy to use the method. This criteria is usually used to compare the execution time and storage requirements of computer-oriented methods.

2. How well the performance is when applying methods to analyze data.

In our point of view, the second criteria can be used only when the evaluation criteria and purpose of the application problems are determined, and considered methods are used under same restricted conditions. Also, we suggest to consider "how easy to interpret the result" and "how easy to extend method for complex kinds of data" as parts of the first criteria when applying clustering method in data mining applications because interpret the result is a crucial steps and complex data has became a challenge in numerous real applications related to data mining.

Concerning the above criteria, we have seen reason that motivate us to focus on spectral clustering as follows

Algorithm 2: General schema of spectral clustering

Input:

1. A set of T data point $P = \{\mathbf{s}_t | t = 1, 2, \ldots, T\}$, where $\mathbf{s}_t$ represents information of data point t^{th}.

2. A similarity function S

$$\begin{aligned} S : & P \times P \longrightarrow \mathbb{R} \backslash (-\infty, 0) \\ & (\mathbf{s}_i, \mathbf{s}_j) \longmapsto S((\mathbf{s}_i, \mathbf{s}_j)) = w_{ij}. \end{aligned}$$

3. A number K of data groups.

Output: K groups of data points $G_1, G_2, \ldots, G_K$ such each data point s_t belong to one group G_k, for $t = 1, 2, \ldots, T$ and $k = 1, 2, \ldots, K$.

---

1. Construct an embedded undirected similarity graph $G = (V, E)$ using similarity measure S.

2. Compute the similarity matrix $\mathbf{W}$, where the (i, j)-th element of $\mathbf{W}$ is $w_{ij} = S(v_i, v_j)$.

3. Define $\mathbf{D}$ to be the orthogonal matrix whose (i, i)-th element is the sum of elements in $\mathbf{W}$'s i-th row, and compute Laplacian matrix $\mathbf{L} = \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2}$.

4. Compute the first K largest eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K$ of $\mathbf{L}$ and form the matrix $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_K] \in \mathbb{R}^{T \times K}]$ by stacking the eigenvectors in columns.

5. Form a matrix $\mathbf{Y}$ from $\mathbf{U}$ by renormalizing each of $\mathbf{U}$'s row to have unit norm.

6. Treating each row of $\mathbf{Y}$ as a vector in $\mathbb{R}^K$ and cluster them via K-mean method.

7. Finally, assign the original point $\mathbf{s}_t$ to the the cluster k if row t^{th} of $\mathbf{Y}$ was assigned to the cluster k.

---

1. Considering the first criteria, K-mean and spectral clustering are two good methods as point out in [57, 64]. More precisely, K-mean clustering methods work when a distance measure and centroids are defined while spectral clustering work when a similarity measure is given. As distance and similarity measure have very close properties, and it is difficult to define centroids for complex kind of data as tensorial data, we think that spectral clustering is more versatile and somewhat easier to extend for complex data.

2. On the second criteria, in many application on data in $\mathbb{R}^T$, the empirical results of spectral clustering methods are better comparing with K-mean methods [54, 64]. We hope to achieve the similar results when working on complex data.

Another motivated reason comes when we study about the multi-view approach which have been considered as a promising direction in data mining with the power to simultaneously treat heterogeneous kind data in order to improve the empirical results. Up to now, we have not found any paper working under multi-view direction related to K-mean while several interesting multi-view spectral clustering methods were recently proposed with well empirical results and is theoretically guaranteed [48, 62].

Motivating by the above reasons, we plan to firstly extend the spectral clustering for 3^{rd} order tensor data. After that we aim to construct appropriate similarity measure in order to extend multi-view spectral clustering.

4.2 CP-decomposition based spectral clustering

4.2.1 Challenges and tasks

When constructing spectral clustering method for 3^{rd} tensor data using the general schema in 4.1.1, we faced two following challenges

1. When working with data in form of a 3-order tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I\times J\times T}$, we have to store and manipulate IJT elements which will rapidly increase when I, J or T increase.

2. What is an appropriate similarity measure should be constructed because when the whole data of T point is store in form of a 3-order tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I\times J\times T}$, data of each data point t^{th} is represented in form of a 2-way tensor $\mathbf{X}_t \in \mathbb{R}^{I\times J}$.

To deal with the first challenge, we employed CP-decomposition with a small number R of components to find a new representation of data. Note that when working on new representation, the number of elements that we have store and manipulate is $(I+J+T)R$ which is small comparing with IJT when R is small.

The second challenge seems more complicated and we decided to construct difference similarity measures and empirically comparing them on real data.

Combining the two challenges, our tasks now change to construct a CP-decomposition spectral clustering method which consists of two following smaller tasks: to find a similarity measure; and determine the smallest number of component in CP-decomposition that

help to reduce the storage memory while keep the good performance result though we may loose important information when applying CP-decomposition with small number of component. The answer for this question will be given in Section 4.3.2 after doing experiments on real data and analyzing the results.

4.2.2 The proposed method

The key ideas of the proposed method are to carry out CP-decomposition method to get a three-component matrices which is the new representation of original data in form of a 3^{rd} order tensor; and apply the spectral clustering method in Algorithm 2 on the new representation with an appropriate similarity measure. A summary of the proposed spectral clustering method is given in Algorithm 3.

Note that to apply the proposed method we have to determine which similarity measure will be employed in Step 1. In this thesis, we firstly extend the similarity measure used in [54] to measure the similarity between 2^{rd}-order tensors as follows

$$\begin{aligned} S_1 :&\mathbb{R}^{I\times J} \times \mathbb{R}^{I\times J} \longrightarrow \mathbb{R}\backslash(-\infty, 0) \\ &(\mathbf{s}_i, \mathbf{s}_j) \longmapsto S((\mathbf{s}_i, \mathbf{s}_j)) = exp(\parallel \mathbf{s}_i - \mathbf{s}_j \parallel^2_{fro} /2\sigma^2), \end{aligned} \tag{4.3}$$

where σ^2 is scaling parameter and should be chosen appropriately for each special kind of data as pointed out in [54].

We also construct another similarity measuare by appropriately modifying 2-dimensional (2D) correlation which has been used in applications related to 2-way data [51,68]. Given two 2-way arrays $X, Y \in \mathbb{R}^{I\times J}$, their 2D correlation is defined as

$$corr2(\mathbf{A}, \mathbf{B}) = \frac{\sum_{i=1}^{I}\sum_{j=1}^{J}(a_{ij} - \overline{A})(b_{ij} - \overline{B})}{\sqrt{(\sum_{i=1}^{I}\sum_{j=1}^{J}(a_{ij} - \overline{A}))^2(\sum_{i=1}^{I}\sum_{j=1}^{J}(b_{ij} - \overline{B}))^2}}, \tag{4.4}$$

where $\overline{A} = \frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J} a_{i,j}$ and called mean of $\mathbf{A}$.

Note that given two 2-way arrays, their 2D correlation may be positive or negative and a high negative correlation indicates a high correlation as high positive does. If we consider the similarity between two 2-way arrays as how high their correlation is, we can construct a similarity measure S_2 such that

$$\begin{aligned} S_2 :&\mathbb{R}^{I\times J} \times \mathbb{R}^{I\times J} \longrightarrow \mathbb{R}\backslash(-\infty, 0) \\ &(\mathbf{s}_i, \mathbf{s}_j) \longmapsto ((\mathbf{s}_i, \mathbf{s}_j)) =\mid corr2(\mathbf{s}_i, \mathbf{s}_j) \mid, \end{aligned} \tag{4.5}$$

where $\mid x \mid$ is the absolute value of $x \in \mathbb{R}$, and $corr2(\mathbf{s}_i, \mathbf{s}_j)$ is the 2D correlation of $\mathbf{s}_i, \mathbf{s}_j \in \mathbb{R}^{I\times J}$.

4.3 Experiment

In this section, we formulate the problem of retrieving the nature label of data using Algorithm 3 and set up 4 experiment on Blue Crabs dataset which is a 3-way data set

Algorithm 3: CP-DECOMPOSITION BASED SPECTRAL CLUSTERING

Input:

1. A set of T data point $P = \{\mathbf{s}_t | t = 1, 2, \ldots, T\}$, we $\mathbf{s}_t \in \mathbb{R}^{I \times J}$ represents information of data t^{th}.

2. A similarity function S

$$\begin{aligned} S : & P \times P \longrightarrow \mathbb{R} \backslash (-\infty, 0) \\ & (\mathbf{s}_i, \mathbf{s}_j) \longmapsto S((\mathbf{s}_i, \mathbf{s}_j)) = w_{ij}. \end{aligned}$$

3. A number K of data groups.

4. A number R of components in CP-decomposition.

Output: k groups of data points $G_1, G_2, \ldots, G_K$ such each data point $\mathbf{s}_t$ belong to one group G_k, for $t = 1, 2, \ldots, T$ and $k = 1, 2, \ldots, K$.

---

1. Form a 3^{rd} order tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I \times J \times T}$ such that the frontal slices $\mathbf{X}_{::t} = \mathbf{s}_t$, for $t = 1, 2, \ldots, T$.

2. Carry out CP-decomposition to get three-component matrices

$$\begin{aligned} \mathbf{A} &= [\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_R] \in \mathbb{R}^{I \times R}, \\ \mathbf{B} &= [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_R] \in \mathbb{R}^{J \times R}, \\ \mathbf{C} &= [\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_R] \in \mathbb{R}^{T \times R}. \end{aligned} \tag{4.1}$$

   where $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$ and $\mathbf{c}_r \in \mathbb{R}^T$.

3. Form new representation $s_t^{'} \in \mathbb{R}^{I \times J}$ for each data point t^{th} in the following form

$$\mathbf{s}_t^{'} = \sum_{r=1}^{R} c_{rt} \mathbf{a}_r \circ \mathbf{b}_r, \tag{4.2}$$

   for $t = 1, 2, \ldots, T$.

4. Apply Algorithm 2 on new representation $S^{'} = \{s_t^{'} | t = 1, 2, \ldots, T\}$.

5. Assign the original point $\mathbf{s}_t$ to the cluster k if $\mathbf{s}_t^{'}$ was assigned to the cluster k.

---

with the natural label. Detail information of Blue Crabs Data dataset can be found in the website of The Three-Mode Company (`http://three-mode.leidenuniv.nl`) or in the related papers [28, 39].

4.3.1 Data description and experiment setup

The Blue Crabs dataset contains information of 16 healthy Blue Crabs originating from the Albermarle Sound, 16 healthy Blue Crabs from Pamlico River and 16 diseased Blue Crabs from Pamlico River.

Figure 4.1: Image of a Blue Crab.

For each Blue Crabs, information consist of levels of 25 trace elements in 3 tissues namely gill, hepatopancreas, and muscle tissue. The information of each Blue Crab is represent in form of a 25×3 matrix, let denoted by $\mathbf{X}_t \in \mathbb{R}^{3\times 25}$, where x_{ijt} is the level of trace element i in tissue j consider Crab t, for $i = 1, 2, \ldots, 25$, $j = 1, 2, 3$ and $t = 1, 2, \ldots, 48$. The whole data set is represented in form of a 3^{rd}-order tensor $\underline{\mathcal{X}} \in \mathbb{R}^{25\times 3\times 48}$.

From this dataset, we extract some datasets and design some experiments as described in Table 4.1.

For Algorithm 2, when the dataset, number of cluster and similarity are determined, we repeat the experiment 100 times and calculate the mean of accuracy of 100 repeated times.

For Algorithm 3, when the dataset, number of cluster, similarity and number of component in CP-decomposition R are determined, we repeat the experiment 100 times and also calculate the mean of accuracy of 100 repeated times. We run this procedure for $R = 1, 2, \ldots, 75$ because we want to test whether we can find a new representation with small number of component which give good clustering results and if yes, how small R can be.

To access the accuracy of a determined experiment, we use the criteria given in [57] which can be summaries as follow

- For each data set, we form $L = \{L_1, L_2, \ldots, L_K\}$ is a disjoint partition of Crabs such that all Cabs in group L_k have the same properties and Crabs with different

Table 4.1: Description of experiments.

Input	Number of Cluster	Algorithm	Similarity measure
48 Crabs	3	Algorithm 2	S_1
	3	Algorithm 2	S_2
	3	Algorithm 3	S_1
	3	Algorithm 3	S_2
16 healthy Crabs from Albemarle Sound and 16 healthy Crabs from Pamlico River	2	Algorithm 2	S_1
	2	Algorithm 2	S_2
	2	Algorithm 3	S_1
	2	Algorithm 3	S_2
16 healthy Crabs from Pamlico River and 16 disease Crabs from Pamlico River	2	Algorithm 2	S_1
	2	Algorithm 2	S_2
	2	Algorithm 3	S_1
	2	Algorithm 3	S_2
16 healthy Crabs from Albemarle Sound and 16 disease Crabs from Pamlico River	2	Algorithm 2	S_1
	2	Algorithm 2	S_2
	2	Algorithm 3	S_1
	2	Algorithm 3	S_2

properties are assign in different groups. There are 3 properties namely healthy from Albemarle Sound, healthy from Pamlico River and disease from Pamlico River.

- $L^{'} = \{L_1^{'}, L_2^{'}, \ldots, L_K^{'}\}$ is another disjoint partition of Crabs that we find after run experiment.
- The accuracy is defined as the similarity between two partitions L and $L^{'}$, denoted by $c(L, L^{'})$, and is calculated as follows

$$c(L, L^{'}) = (\sum_{t=1}^{T} \sum_{j=i+1}^{T} \gamma_{ij}) / \binom{T}{2}, \tag{4.6}$$

where $\binom{T}{2} = \frac{T!}{2!(T-2)!}$ and

$$\gamma_{ij} = \begin{cases} 1 & \text{if there exist } k \text{ and } k^{'} \text{ such that } s_i \text{ and } s_j \text{ are in} \\ & \text{both } L_k \text{ and } L_{k'}^{'}, \\ 1 & \text{if there exist } k \text{ and } k^{'} \text{ such that } s_i \text{ is in both of} \\ & L_k \text{ and } L_{k'}^{'} \text{ while } s_j \text{ is neither in } L_k \text{ nor } L_{k'}^{'}, \\ 0 & \text{otherwise.} \end{cases} \tag{4.7}$$

The following example from [57] may help to a better understanding about the criteria.

Example 1 *Let $S = \{a, b, c, d, e, f\}$ be a dataset consists of six data point which have disjoint partition $L = \{L_1, L_2\}$ such that $L_1 = \{a, b, c\}$ and $L_2 = \{d, e, f\}$. Assuming that we apply a clustering method with the priori number of clusters are 3 and obtain the results $L^{'} = \{L_1^{'}, L_2^{'}, L_3^{'}\}$ such that $L_1^{'} = \{a, b\}$, $L_2^{'} = \{c, d, e\}$ and $L_3^{'} = \{f\}$. Then the point-pairs are tabulated as in Figure 4.2.*

Point-pair	*ab*	*ac*	*ad*	*ae*	*af*	*bc*	*bd*	*be*	*bf*	*cd*	*ce*	*cf*	*de*	*df*	*ef*	*Total*
Together in both	*												*			2
Separate in both			*	*	*		*	*	*			*				7
Mixed		*				*				*	*			*	*	6

Figure 4.2: The point-pairs result [57].

Using criteria in Equation 4.6, it is easy to see that

$$c(L, L^{'}) = 9/\binom{6}{2} = 0.6,$$

or in other words, the accuracy of the considering clustering method is 60% with respect to criteria given in Equation 4.6.

4.3.2 Experimental result and discussion

In this section, we present results of experiments setup in Table 4.1. The results are also analyzed and visualized, and used to evaluate Algorithm 2 and Algorithm 3 presented in Section 4.1.1 and Section 4.2.2, respectively. Note that when applying Algorithm 2, the scaling parameter σ^2 should be determined as a turning parameter. In this section, we fix $\sigma^2 = 0.001$ since our main purpose is to test the ability to find a compact representation (i.e, a low-rank representation with acceptable accuracy) of CP-decomposition in Algorithm 3 and leave turning parameter problem as a future work of this thesis. For short, we denote that

- **Dataset** 1 contains information of all 48 Crabs which belong to three clusters namely healthy Crabs from Albemarle Sound, healthy Crabs from Pamlico River and disease Crabs from Pamlico River.
- **Dataset** 2 contains information of 16 healthy Crabs from Albemarle Sound and 16 healthy Crabs from Pamlico River.
- **Dataset** 3 contains information of 16 healthy Crabs from Pamlico River and 16 disease Crabs from Pamlico River.
- **Dataset** 4 contains information of 16 healthy Crabs from Albemarle Sound and 16 disease Crabs from Pamlico River.

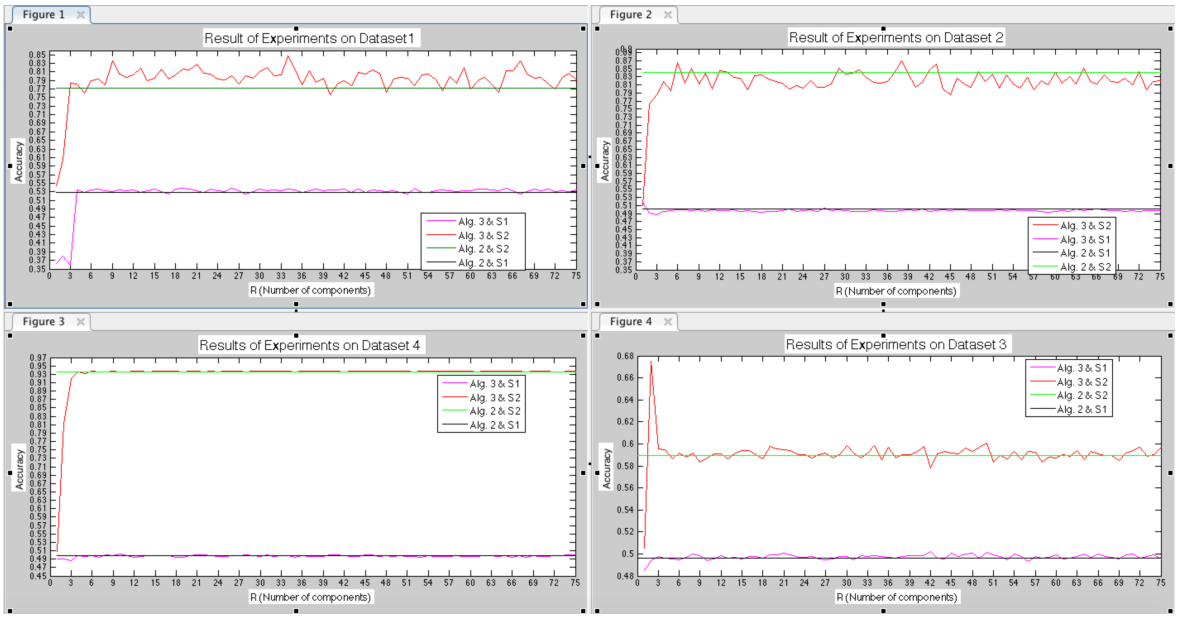

Figure 4.3: Experimental result on 4 datasets.

We implement the Algorithm 2 and Algorithm 3 and run experiments on MATLAB (R2013b). The experiment results are visualized in Figure 4.3.

From the experimental results, we see that for both similarity measurement S_1 and S_2, we can choose a small number of component in CP-decomposition which give the acceptable result comparing with K-mean. In other words, CP-decomposition can be used to find low-rank representation of the original data which may help to reduce the storage memory while still give acceptable accuracy. Furthermore, when analyzing the experiments results corresponding to dataset 1 and 4, results of Algorithm 3 are slightly higher than that of Algorithm 2. Such results are matched with discussions from some papers that CP-decomposition also help to explore the hidden structure of data which give advantages for CP-decomposition based methods when comparing with other methods [3, 17, 18].

4.4 Future works

In later research, we plan to focus on following tasks

1. Construct several similarity measures for tensor data.

2. Extend the multi-view spectral clustering methods proposed in [48] which is illustrated in Figure 4.4 for tensor data.

3. Construct a tensor space model based spectral clustering method.

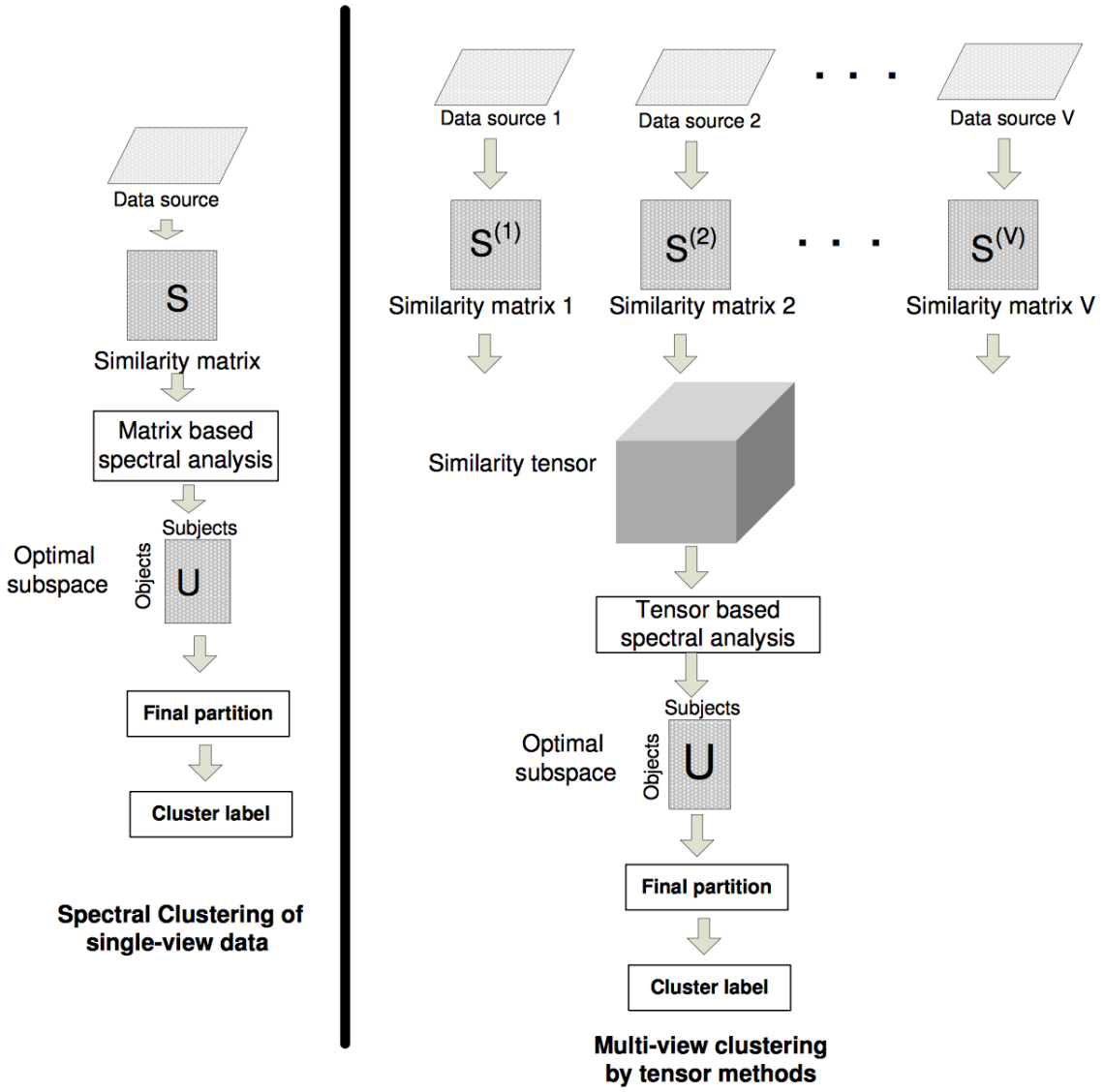

Figure 4.4: Comparison between single view (left) and multi-view (right) spectral clustering [48].

4.4.1 Single-view spectral clustering vs multi-view spectral clustering

Considering multi-view spectral clustering method in Figure 4.4, a data source can be considered as the combination of original data and a similarity measure. In case V similarity measure are constructed and dataset consist of I data sample, we can form a similarity tensor $\underline{\mathcal{X}} \in \mathbb{R}^{I \times I \times V}$ and applying tensor based methods proposed in [48] to cluster data.

In order to see the relation between single view spectral clustering and multi-view

spectral clustering proposed in [48], it is necessary to know about the formulations of these two approaches.

Let $\mathbf{U} \in \mathbb{R}^{I \times K}$ is the relaxed assignment matrix, where I is the number of data points and K is the number of clusters, the single-view spectral clustering problem can be expressed as follows

$$\begin{aligned} \min_{\mathbf{U}} \quad & \text{trace}(\mathbf{U}^T \mathbf{L}_{Ncut} \mathbf{U}), \\ \text{s.t.} \quad & \mathbf{U}^T \mathbf{U} = \mathbf{I}, \end{aligned} \tag{4.8}$$

where $\mathbf{I}$ is the $K \times K$ identity matrix and $\mathbf{L}_{Ncut}$ is the normalized Laplacian matrix corresponding to the normalized cuts (Ncut) which is defined as $\mathbf{L}_{Ncut} = \mathbf{I} - \mathbf{L}$.

As point out in [48], the single-view spectral clustering can also be formulated as the following Frobenius norm optimization problem

$$\begin{aligned} \min_{\mathbf{U}} \quad & \|\mathbf{U}^T \mathbf{L} \mathbf{U}\|_F, \\ \text{s.t.} \quad & \mathbf{U}^T \mathbf{U} = \mathbf{I}. \end{aligned} \tag{4.9}$$

Results of Problem 4.8 and 4.9 are the set of K dominant eigenvectors of $\mathbf{L}_{Ncut}$ and the set of K dominant eigenvectors of $\mathbf{L}$. These two sets are equivalent in the sense that they both span the dominant eigenspace of $\mathbf{L}$ [48].

When V-similarity measures are determined, we plan to construct V similarity matrices $\mathbf{L}^1, \mathbf{L}^2, \ldots, \mathbf{L}^V$ and cluster the data by applying multi-view spectral clustering methods proposed in [48] whose formulations are as follows

1. *Multi-view clustering by trace maximization*

$$\begin{aligned} \max_{\mathbf{U},\mathbf{w}} \quad & \sum_{v=1}^{V} \mathbf{w}_v \text{trace}(\mathbf{U}^T \mathbf{L}^v \mathbf{U}) = \sum_{v=1}^{V} \text{trace}(\mathbf{U}^T \mathbf{w}_v \mathbf{L}^v \mathbf{U}), \\ \text{s.t.} \quad & \mathbf{U}^T \mathbf{U} = \mathbf{I}, \quad \mathbf{w} \geq 0 \quad \text{and} \quad \|\mathbf{w}\|_F = 1. \end{aligned} \tag{4.10}$$

2. *Multi-view clustering by integration of the Frobenius- norm objective function*

$$\begin{aligned} \max_{\mathbf{U}} \quad & = \sum_{v=1}^{V} \|\mathbf{U}^T \mathbf{L}^v \mathbf{U}\|_F^2, \\ \text{s.t.} \quad & \|\mathbf{w}\|_F = 1. \end{aligned} \tag{4.11}$$

3. *Multi-view clustering by matrix integration in the Frobenius-norm objective function*

$$\begin{aligned} \max_{\mathbf{U},\mathbf{w}} \quad & = \sum_{v=1}^{V} \|\mathbf{U}^T \mathbf{w}_v \mathbf{L}^v \mathbf{U}\|_F^2, \\ \text{s.t.} \quad & \mathbf{U}^T \mathbf{U} = \mathbf{I}, \quad \mathbf{w} \geq 0 \quad \text{and} \quad \|\mathbf{w}\|_F = 1. \end{aligned} \tag{4.12}$$

All of three optimization problems 4.10, 4.11 and 4.12 using tensor methods as proposed in [48].

4.4.2 A tensor space model based clustering method

As tensor data requires large storage memory, we also try to construct a tensor space model based clustering method in which tensor data in $\mathbb{R}^{I\times J}$ is transformed in to vector space $\mathbb{R}^T$ and spectral clustering methods are applied on the transformed data in order to cluster the data. Figure 4.5 gives an illustration of the proposed method and Algorithm 4 provides a summary.

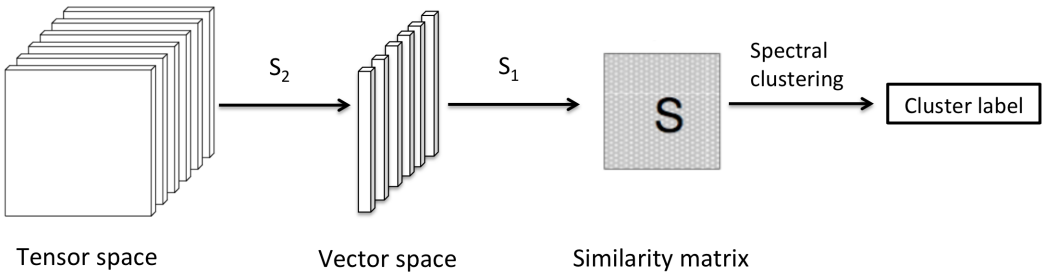

Figure 4.5: A tensor space model based clustering method.

Algorithm 4: A TENSOR SPACE MODEL BASED CLUSTERING METHOD

Input: A set of point $P = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_T | \mathbf{s}_t \in \mathbb{R}^{I\times J}, t = 1, 2, \ldots, T\}$ and number K of clusters.

Output: K groups of data points $G_1, G_2, \ldots, G_K$ such each data point $\mathbf{s}_t$ belong to one group G_k, for $t = 1, 2, \ldots, T$ and $k = 1, 2, \ldots, K$.

1. For each data point $\mathbf{s}_t$, construct a new representation $\mathbf{x}_t \in \mathbb{R}^T$ such that, $x_{tj} = corr2(\mathbf{s}_t, \mathbf{s}_j)$, for $j = 1, 2, \ldots, T$.
2. Apply a clustering method on new representation $P^{'} = \{\mathbf{x}_t | t = 1, 2, \ldots, T\}$.
3. Assign the original point $\mathbf{s}_t$ to the cluster k if $\mathbf{x}_t$ was assign to the cluster k.

Note that Method 4 is very versatile. Given a dataset consists of T point, each is represented inform of second order tensor in $\mathbb{R}^{I\times J}$, the key ideas are to learn a new representation in Euclidean vector space $\mathbb{R}^T$ and apply a clustering to do clustering task on the new representation.

Since the clustering algorithm in step 2 should be chosen appropriately by studying the structure of data in new presentation space, we have to carefully analyze the relation between S and $S^{'}$ and the transformation function in step 1 in order to fully complete the clustering task. In this section, we employ the spectral clustering proposed algorithm proposed in [54] to do clustering task in steps 2 and implement the Algorithm 4 to cluster data in 4 datasets described in Table 4.1 .

Because of limited time, we skip the procedure to choose the scaling parameter σ^2 automatically as suggested in [54] because it should be adjusted appropriately for each kind of data. The main purposes here are to check whether method given in Algorithm 4 can give acceptable results and whether the scaling parameter σ^2 have effect on the cluster results. The former is done by comparing the results with results given by Algorithm 2 together with similarity measure S_2 as given in Figure 4.3 while the later was done by analyzing the results of 200 experiments when the value of scaling parameter σ^2 is fixed as $\sigma^2 \in \{0.01n | n = 1, 2, \ldots, 200\} \in [0.01, 2]$.

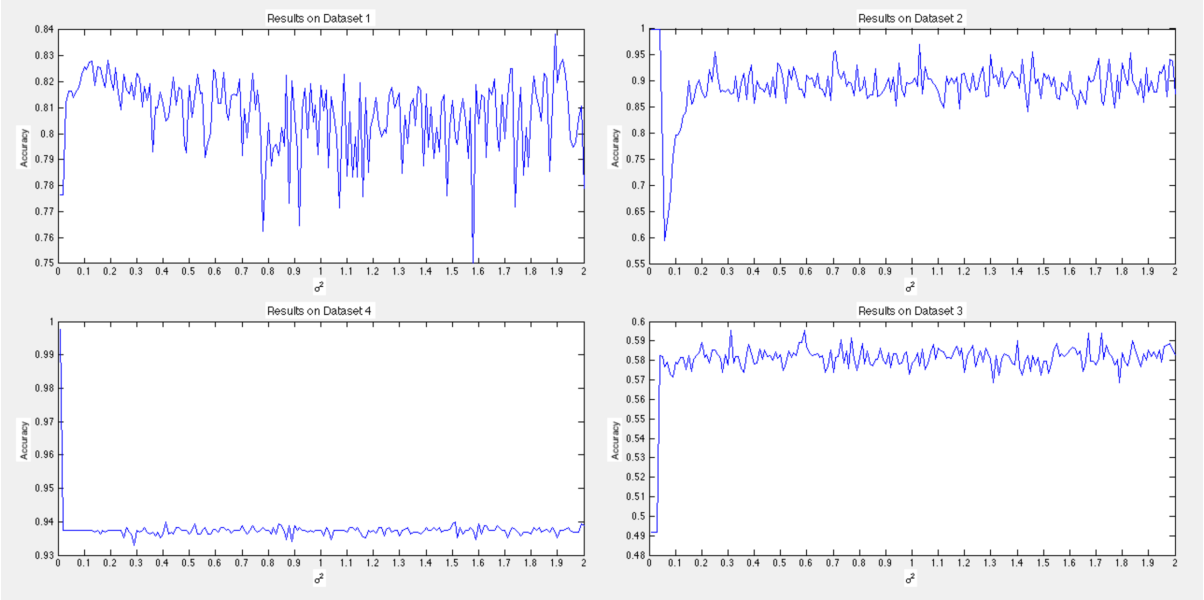

Figure 4.6: Experiment results of Algorithm 4 on 4 datasets.

From the Figure 4.6, we see that different value of scaling parameter σ^2 may give different accuracy. In the Table 4.2, we compare the highest results from those experiment and results given by Algorithm 2 together with similarity measure S_2.

Table 4.2: Comparing results of Algorithm 2 (S_2) and highest result of Algorithm 4.

Algorithm	Dataset 1	Dataset 2	Dataset 3	Dataset 4	Results
Algorithm 2 and S_2	0.7721	0.8413	0.5897	0.9363	Accuracy
Algorithm 4	0.8384	1.0000	0.5954	0.9975	Accuracy
	1.89	$0.01 - 0.04$	0.31	0.01	σ^2

The result from Table 4.2 show that if we can find an appropriate scaling parameter σ^2, we can get acceptable accuracy when applying Algorithm 4. Considering later works related to Algorithm 4, we plan to focus on two following tasks: construct method to learn the scaling parameter σ^2 automatically; and construct other clustering methods to do clustering task in step 2.

Chapter 5

Thesis conclusion and future works

5.1 Thesis conclusion

We have presented our research focusing on tensor calculus, some tensor based methods and their applications in data mining. As "tensor" is a complicated subject and some times is defined in different way, it is difficult for one who starts to learn about tensor and apply tensor-based methods. In chapter 2, we present some discussions which are helpful to obtain a clear understanding about tensor, and how to avoid confuse when applying tensor and tensor-based methods in machine learning and data mining. Concerning the applications of tensor-based methods, in Chapter 3 and Chapter 4, we present two interesting problems in machine learning namely temporal link prediction and spectral clustering, and employ CP-decomposition models to extend the available methods for tensor data. The main contributions of this thesis can be summarized as follows

1. *Temporal link prediction:* We propose a temporal link prediction method for open bipartite networks by extending the method proposed in [2, 26] for bipartite network. The key ideas of the proposed method are: considering weight of link as a multivariate function of three variables which represent time and vertices; carrying out CP-decomposition to extract the hidden structure of data represented by three component matrices where each matrix is corresponding to one variable and columns of the corresponding matrix are considered as separated factors; employing linear regression to learn the relations between vertices' information and the corresponding component matrices and using those relations to estimate the value of separated factors corresponding to new vertices; finally, all the results are combined in order to predict the weights of links at time $(T+1)^{th}$.

2. *Spectral clustering:* We present an intuitive overview about spectral clustering and a general schema of spectral clustering. We also provide discussions about advantages of spectral clustering over other clustering methods and opportunity to extend spectral clustering methods for data tensor. Then, we propose a CP-decomposition based spectral clustering method, implement the method and run the program on 4 experiments. After analyzing the results, we make conclusion that, for small number

of component R in CP-decomposition, the proposed method may give the acceptable results. In other word, CP-decomposition may help to reduce the storage memory. Furthermore, employing CP-decomposition may help to improve the accuracy of spectral clustering by explore the hidden structure of data.

5.2 Future works

Because of the limited time, we can not solve completely tasks discussed in this thesis and we plan to do them as future works as discussion in Section 3.4 and Section 4.4. Some important points can be summarized as follows

1. *Temporal link prediction:* We plan to focus on several tasks: collecting data and run experiments on real datasets in order to evaluated the proposed method; extending the proposed method in Section 3.3.3 to predict the links for a period of time starting at $(T+1)^{th}$ or in other words, for times $(T+1)^{th}$, ... , $(T+L)^{th}$; constructing temporal link prediction for open bipartite networks when the new vertices of type-1 and type-2 join the concerned networks at the same time.

2. *Spectral clustering:* We will focus on constructing several similarity measures for tensor data and extending the multi-view spectral clustering methods proposed in [48] as discussed in Section 4.4.1. Another important tasks related to spectral clustering/clustering is to construct a tensor space model based clustering method using suggestions presented in Section 4.4.2.

Of course, the above tasks are challenging and require time focusing on both theoretical discussion and experiment works. But as tensor data has been increasingly considered in data mining, and the presented results and suggestions are reasonable, the works presented in this thesis are worth for us to focusing on.

Bibliography

[1] Abbasi, A. A., and Younis, M., A survey on clustering algorithms for wireless sensor networks. *Computer communications*, 30(14), 2826-2841, 2007.

[2] Acar, E., Dunlavy, D. M., and Kolda, T. G., Link prediction on evolving data using matrix and tensor factorizations. *IEEE International Conference on Data Mining Workshops, ICDMW'09*, 2009.

[3] Acar, E., and Yener, B., Unsupervised multiway data analysis: A literature survey. *IEEE Transactions on Knowledge and Data Engineering*, 21(1), 6-20, 2009.

[4] Adkins, W.A. and Weintraub, S.H., Algebra: an approach via module theory. *Graduate Texts in Mathematics*, 136, Springer-Verlag, New York, 1992.

[5] Al Hasan, M., and Zaki, M. J., A survey of link prediction in social networks. *Social network data analytics*, Springer US, 243-275, 2011.

[6] Allen, G. I., Regularized tensor factorizations and higher-order principal components analysis. *arXiv preprint arXiv:1202.2476*, 2012.

[7] Allen, G., Sparse higher-order principal components analysis. *In International Conference on Artificial Intelligence and Statistics*, pp. 27-36, 2012.

[8] Barber, M. J., Faria, M., Streit, L., and Strogan, O., Searching for communities in bipartite networks. *Workshop on Stochastic and Quantum Dynamics of Biomolecular Systems*, 171182, 2008.

[9] Berkhin, P. A., Survey of clustering data mining techniques. *In Grouping multidimensional data*, Springer Berlin Heidelberg, pp. 25-71, 2006.

[10] Bourbaki, N., Algebra I: Chapters 13, Elements of Mathematics. *Springer-Verlag*, Berlin, 1998.

[11] Burdick, D. S., An introduction to tensor products with applications to multiway data analysis. *Chemometrics and intelligent laboratory systems*, 28(2), 229-237, 1995.

[12] Cai, D., He, X., and Han, J., Learning with tensor representation. 2006.

[13] Cai, D., He, X., and Han, J., Tensor space model for document analysis. *In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 625-626, 2006.

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

[15] Carroll, J. D., Pruzansky, S., and Kruskal, J. B., CANDELINC: A general approach to multidimensional analysis of many-way arrays with linear constraints on parameters. *Psychometrika*, 45(1), 3-24, 1980.

[16] Chatfield, C., and Yar, M., Holt-Winters forecasting: some practical issues. *The Statistician*, 129-140, 1988.

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

[18] Cichocki, A., Zdunek, R., Phan, A. H., and Amari, S. I., Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation. *John Wiley & Sons*, 2009.

[19] Comon, P., Tensors: a partial survey. *Signal Processing Magazine*, 2014.

[20] De Lathauwer, L., De Moor, B., and Vandewalle, J., On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4), 1324-1342, 2000.

[21] De Silva, V., and Lim, L. H., Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084-1127, 2008.

[22] Dhillon, Inderjit S., Co-clustering documents and words using bipartite spectral graph partitioning. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2001.

[23] Dhote, Y., Mishra, N., and Sharma, S., Survey and analysis of temporal link prediction in online social networks. *Advances in Computing*, International Conference on Communications and Informatics, ICACCI, 2013.

[24] Dullemond, K. and Peeters, K., Introduction to Tensor Calculus. 1991.

[25] Dummit, D.S. and Foote, R.M., Abstract algebra, 3rd Ed. *John Wiley and Son*, Hoboken, NJ, 2003.

[26] Dunlavy, D. M., Kolda, T. G., and Acar, E., Temporal link prediction using matrix and tensor factorizations. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 5(2):10, 2011.

[27] Gao, S., Denoyer, L., and Gallinari, P., Temporal link prediction by integrating content and structure information. *In Proceedings of the 20th ACM international conference on Information and knowledge management*, pp. 1169-1174, 2011.

[28] Gemperline, P. J., Miller, K. H., West, T. L., Weinstein, J. E., Hamilton, J. C., and Bray, J. T., Principal component analysis, trace elements, and blue crab shell disease. *Analytical Chemistry*, 64, 523-531, 1992.

[29] Grasedyck, L., Kressner, D., and Tobler, C. A, Literature survey of low rank tensor approximation techniques. *GAMM Mitteilungen*, 36(1):53-78, 2013.

[30] Greub, W., Multilinear algebra, 2nd Ed. *Springer-Verlag*, New York, NY, 1978.

[31] Guan-zhong, C. X. Y. D., and Li-bin, Y. A. N. G., Survey on Spectral Clustering Algorithms [J]. *Computer Science*, 7(005), 2008.

[32] Harshman, R. A., Foundations of the parafac procedure: models and conditions for an "explanatory" multimodal factor analysis. 1970.

[33] Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Friedman, J., and Tibshirani, R., The elements of statistical learning. *New York: Springer*, 2(1), 2009.

[34] He, X., Cai, D., Liu, H., and Han, J., Image clustering with tensor representation. *In Proceedings of the 13th annual ACM international conference on Multimedia*, pp. 132- 140, 2005.

[35] Howell, T. D., Global properties of tensor rank. *Linear Algebra and its Applications*, 22, 9-23, 1978.

[36] Hungerford, T.W., Algebra. *Graduate Texts in Mathematics*, 73, Springer-Verlag, New York, NY, 1980.

[37] Kilmer, M. E., and Martin, C. D., Factorization strategies for third-order tensors. *Linear Algebra and its Applications*, 435(3):641-658, 2011.

[38] Kolda, T. G., and Bader, B. W., Tensor decompositions and applications. *SIAM review*, 51(3), 455-500, 2009.

[39] Kroonenberg, P. M., Basford, K. E., and Gemperline, P. J., Grouping three-mode data with mixture methods: the case of the diseased blue crabs. Journal of Chemometrics, 18, 508-518, 2004.

[40] Kruskal, J. B., Rank, decomposition, and uniqueness for 3-way and N-way arrays. *Multiway data analysis*, 33, 1989.

[41] Landsberg, J. M., Tensors: geometry and applications. *American Mathematical Society*, 128, 2012.

[42] Lang, S., Algebra, Rev. 3rd Ed. *Graduate Texts in Mathematics*, Springer-Verlag, New York, NY, 211, 2002.

[43] Larremore, D. B., Clauset, A., and Jacobs, A. Z., Efficiently inferring community structure in bipartite networks. *arXiv preprint*, arXiv:1403.2933, 2014.

[44] Lauer, F., and Schnorr, C., Spectral clustering of linear subspaces for motion segmentation. *IEEE 12th International Conference on Computer Vision*, pp. 678-685, 2009.

[45] Lerman, E., Multilinear algebra notes. 2011.

[46] Lim, L. H., Multilinear Algebra in Data Analysis: tensors, symmetric tensors, nonnegative tensors. *Workshop on Algorithms for Modern Massive Datasets*, Stanford, 2006.

[47] Lim, L. H., Whats possible and whats not possible in tensor decompositionsa freshmans view. *In Workshop on Tensor Decompositions*, American Institue of Mathematics, 2014.

[48] Liu, X., Ji, S., Glanzel, W., and De Moor, B., Multiview partitioning via tensor methods. *IEEE Transactions on Knowledge and Data Engineering*, 25(5), 1056-1069, 2013.

[49] Marcus, M., Finite Dimensional Multilinear Algebra, Parts I and II, *Series of Monographs and Textbooks in Pure and Applied Mathematics*, 23, Marcel Dekker, New York, NY, 1973 and 1975.

[50] Miwakeichi, F., Martnez-Montes, E., Valds-Sosa, P. A., Nishiyama, N., Mizuhara, H., and Yamaguchi, Y., Decomposing EEG data into spacetimefrequency components using parallel factor analysis. *NeuroImage*, 22(3), 1035-1045, 2004.

[51] Morita, S., Shinzawa, H., Noda, I., and Ozaki, Y., Perturbation-correlation moving-window two-dimensional correlation spectroscopy. *Applied spectroscopy*, 60(4), 398-406, 2006.

[52] Murata T., Community division of heterogeneous networks. *Complex Sciences*, Springer Berlin Heidelberg, 1011-1022, 2009.

[53] Newman, M. E. J., and M. Girvan., Finding and evaluating community structure in networks. *Physical review E*, 69(2). 2004.

[54] Ng, A. Y., Jordan, M. I., and Weiss, Y., On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2, 849-856. 2002.

[55] Northcott, D.G., Multilinear algebra. *Cambridge University Press*, Cambridge, UK, 1984.

[56] Plakias, S., and Stamatatos, E., Tensor space models for authorship identification. *In Artificial Intelligence: Theories, Models and Applications*, Springer Berlin Heidelberg, pp. 239-249, 2008.

[57] Rand, W. M., Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336), 846-850, 1971.

[58] Rotman, J.J., Advanced modern algebra. *Prentice Hall*, Upper Saddle River, NJ, 2002.

[59] Saha, S., Murthy, C. A., and Pal, S. K., Classification of web services using tensor space model and rough ensemble classifier. *In Foundations of Intelligent Systems*, Springer Berlin Heidelberg, pp. 508-513, 2008.

[60] Smolinsky, L., Chapter 9: Multi-linear algebra. 2002.

[61] Spiegel, S., Clausen, J., Albayrak, S., and Kunegis, J., Link prediction on evolving data using tensor factorization. *In New Frontiers in Applied Data Mining*, Springer Berlin Heidelberg, pp. 100-110, 2012.

[62] Tao, H., Hou, C., and Yi, D., Multiple-View Spectral Embedded Clustering Using a Co-training Approach. *In Computer Engineering and Networking*, Springer International Publishing, pp. 979-987, 2014.

[63] Vega-Pons, S., and Ruiz-Shulcloper, J., A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03), 337-372, 2011.

[64] Von Luxburg, U., A tutorial on spectral clustering. *Statistics and computing*, 17(4), 395-416, 2007.

[65] Xu, R., and Wunsch, D., Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3), 645-678, 2005.

[66] Yokonuma, T., Tensor spaces and exterior algebra. *Translations of Mathematical Monographs*, 108, AMS, Providence, RI, 1992.

[67] Wakita, K., and K. Suzuki., Extracting multi-facet community structure from bipartite networks. *IEEE International Conference on Computational Science and Engineering*, 4, 2009.

[68] Wang, G., Karnes, J., Bunker, C. E., and Lei Geng, M., Two-dimensional correlation coefficient mapping in gas chromatography: Jet fuel classification for environmental analysis. *Journal of molecular structure*, 799(1), 247-252, 2006.

[69] Westwick, R., Transformations on tensor spaces. *Pacific Journal of Mathematics*, 23(3), 613-620, 1967.

[70] WWL Chen, Linear algebra: Chapter 5, Linear algebra lecture notes. *Macquarie University*, 2008.

Publications

[1] Nguyen V.L., and Ho T.B., Temporal link prediction using CP-decomposition. *Asia conference on information systems*, ACIS, 1-3 December, Nha Trang, 2014.