KU LEUVEN

Tensor decompositions for latent dirichlet allocation in topic modeling

Nozomi Takemura

Thesis submitted for the degree of
Master of Science in Statistics, option
General Statistical Methodology

Thesis supervisors:
Prof. dr. ir. Karl Meerbergen
Prof. dr. ir. Johan Suykens
Dr. Nick Vannieuwenhoven

Assessors:
Prof. dr. ir. Luc De Raedt
Prof. dr. ir. Jan Aerts

Mentor:
Mr. Bruno Coussement

Academic year 2017 – 2018

Preface

First and foremost, I would like to thank Prof. Karl Meerbergen for giving me an opportunity to work on this thesis topic. I also would like to thank Prof. Johan Suykens for undertaking the role of the second supervisor. Without their willingness to having me as a Master thesis student, I could not have worked on this thesis topic.

I want to sincerely thank Dr. Nick Vannieuwenhoven for the constantly close guidance, feedback, and support throughout this thesis work, ranging from giving me invaluable advice on research methods to kindly explaining theories I did not understand. Without his supervision and help, I would not have been able to finish this thesis work.

I also thank Mr. Bruno Coussement for giving me support and feedback on this work; especially for the guidance on my presentation and thesis writing.

I thank Prof. Luc De Raedt and Prof. Jan Aerts for agreeing to be readers for this thesis work.

Finally, I also thank my family and friends in Master of Statistics for their incessant support and for having me experience a great and memorable two years in Leuven.

Nozomi Takemura

Abstract

Latent Dirichlet allocation (LDA) is one of the most popular topic models, which is used to learn unobserved topics from a collection of documents by assuming a specific data generation process specified with the dependency between latent and observed variables. One of the challenges of LDA is the inference of latent variables, which has been tackled with a variety of approaches but all of them exhibit some disadvantages.

Recently, however, tensor decomposition-based methods stemming from the method of moments have been proposed to recover a subset of LDA parameters, which are supported by the solidly theoretical framework and hugely researched computational methods advanced in the field of linear algebra or numerical analysis.

In this thesis, as one such moment-based parameter recovery methods for LDA, we propose symmetric Canonical Polyadic Decomposition. The main aim of this thesis is to empirically investigate if this method works since there is no previous research with empirical results. Furthermore, one of the biggest motivations for proposing this method is that it theoretically recovers the set of unique parameter vectors of topic-word distribution with probability 1, while most of other previously introduced methods, such as ones based on Markov chain Monte Carlo, do not have such strong theoretical guarantees for parameter estimation.

Although the basic concept of our proposed method has been presented earlier in literature, most of the proposed methods are dependent on the orthogonalization of an input tensor. This leads to the second motivation for proposing our method that does not involve such orthogonalization process. Specifically, the inclusion of such an orthogonalization step theoretically poses a much stricter limit on the maximum number topics that can be considered in an LDA model than our proposed method.

To further study if our proposed method actually works, we conduct experiments using synthetic data generated from LDA models with various parameter settings, subsequently comparing generated parameter with the recovered parameter. Moreover, parameters of the same LDA models are inferred from the same synthetic data using popularly used collapsed Gibbs sampling. By doing this, we compare the results from our method to the ones from collapsed Gibbs sampling. Finally, we apply our method to pieces of real datasets and extract hidden topics, which are again compared with the ones obtained from collapsed Gibbs sampling.

The main contribution of this thesis is that we experimentally determine if our proposed method does not correctly recover the parameter of topic-word distributions of LDA despite the existence of a theoretical guarantee.

Contents

List of Figures and Tables

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Topic modeling

Topic modeling is a kind of machine learning which has been frequently used especially in the research field of text mining. In topic modeling, it is aimed to discover a semantic meaning in a bunch of words of multiple documents by learning a pattern of word occurrences [41]. In other words, the goal of topic modeling is to find topics from a set of documents. To achieve this purpose, models assumed in topic modeling possess some specific characteristics. For example, a topic model consists a variable "topic", which is unobservable and called a *latent variable*. As another example, topic models often need to specify a semantic dependency between topics and words to extract those hidden topics. Such specification of dependencies among variables is actually one of the most distinctive characteristics of a *probabilistic graphical model*, in which *nodes* and *arcs* respectively indicate variables and dependencies among them [11]. Hence, topic models can be viewed as a probabilistic graphical model and their graphical representations determine joint distributions of them.

Latent Dirichlet allocation (LDA) [8] is one of the most popular models among various topic models [62]. It models how a collection of documents are probabilistically generated. The fundamental assumption of LDA is that each document is composed from a mixture of topics and each of those topics is characterized by a distribution of words [32]. By learning such distribution of words for each topic and figuring out words with higher probability mass functions (PMF), we can acquire a good insight about each topic. Figure 1.1 shows a toy example of a LDA model. In this case, it can be seen that "Money", "Bank", and "Loan" have the first, second, and third highest PMF. This, for example, allows us to give a label "Bank (finance)" to this topic. On the other hand, for the other topic (red), the words with higher PMF are "River", "Bank", and "Water", which would lead us to assign a topic name "River". The important thing to note is that LDA does not provide a topic label by itself and thus we have to do it by ourselves.

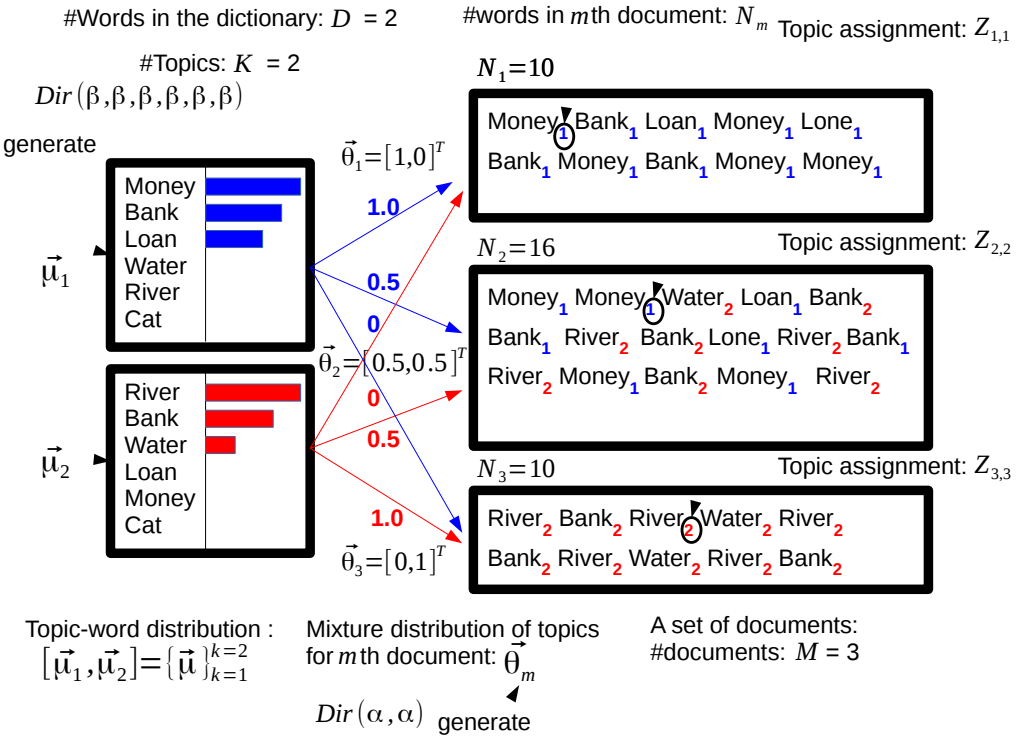

Figure 1.1: An example of a simple LDA model where there are two topics, each of which has a binomial distribution over words for each document. The example was inspired by Figure in p.40 [28].

1.2 Challenge

The main difficulty in LDA is in its learning phase; more specifically, our main interest, topics (distribution), is never observed and therefore the normal maximum likelihood estimation method cannot be directly applicable in parameter estimation. Hence a variety of other approaches to overcome this challenge have been proposed. As examples, expectation-maximization based approach [8], Gibbs sampling approach [22], or expectation-propagation [43]. Although these approaches have brought a success in training LDA models, it is also true that they are accompanied by some challenging aspects. For instance, expectation-maximization based procedures have a disadvantage that they could converge not only to a local optima, but also a saddle point, and Gibbs sampling-based approaches can result in slow mixing [62]. To tackle these challenges, the application of *tensor decompositions* for LDA has recently been proposed.

1.3 Tensor decomposition approach

The basic idea behind tensor decomposition based approaches is *method of moments* [47], in which the parameter of a model is estimated by solving a set of equations

representing relationship between observed empirical row moments and those derived from a model. Anandkumar et. al proved equations between symmetrized empirical lower order moments (upto third) and a set of parameter specifying distributions of words for topics and *Dirichlet prior* for the topic distributions [5]. This result, with the uniqueness of the lower order tensor decompositions, implies that the parameter of LDA can be recovered through the tensor decomposition approaches.

1.4 Motivation

Although various tensor decomposition methods have been examined to extract the parameter of LDA [4, 49], there is no preceding research trying to recover the parameter only by *symmetric canonical polyadic decomposition*, which would be a simpler approach than approaches examined so far with comparably rigorous support from theoretical perspective. Moreover, the basic theory for this method has been introduced in [5, 49]; however in these studies, no empirical results of parameter estimation for LDA model based on this method are not reported. Therefore, in this thesis, we implemented this tensor based method to estimate a subset of parameter of LDA model and then empirically verified if this method accurately recovers the parameter of synthetic data from LDA.

1.5 Outline

Before introducing our main result, the LDA is firstly explained in Chapter 2. In the following Chapter 3, the basic knowledges on tensor and tensor decomposition are provided. After that, the application of tensor decomposition for LDA is described in Chapter 4. Eventually, we presents our method and experimental results in Chapter 5.

Chapter 2

Latent Dirichlet Allocation

In this section, we briefly review the basic assumptions of topic models and introduce LDA as an example of topic models, which would help readers to intuitively understand the concept of LDA. Subsequently, we introduce the LDA models in slightly details, beginning from its assumptions, moving to its property as a *generative model*, followed by its parameter estimation method. As a general reference of the LDA, the following papers are recommendable: [8, 25, 6].

2.1 LDA as a topic model

As being briefly introduced in Chapter 1, the main goal of the application of topic model is finding hidden themes prevailing over a collection of documents, which is often called corpus, by analyzing the data of observed words ([39], p.4). These themes are called topics in the context of topic model, and probabilistic specification on them is one of the fundamental assumption in topic model as given below:

Assumption 1. *A topic is a distribution over certain set of distinct words.*

Apart from this assumption, there are two more core assumptions for the topic model. One is *bags of words*. In other words, this can be restated as follows [6, 58]:

Assumption 2. *A document is a just set of words without and hence it is order invariant.*

The other is related to a characteristic of words as observable variables in topic model and declared as follows:

Assumption 3. *Words are random variables which are independently, and identically distributed, given the topic.*

Assumption 3 implies that the structure of topic model consists the relationships of conditional independences among words conditioned on topics. This indicates that topic models can be often described as a (probabilistic) graphical model.

2.1.1 Example

Let us briefly check if a LDA surely satisfies the above three assumptions thinking the scenario considered in the example given in Figure 1.1. At first and foremost, it is important to note that Figure 1.1 actually gives information about "true" model we will never be able to observe. In practice, it is necessary to estimate all unobserved variables shown in Figure 1.1. Thus, we re-describe the same LDA model in a a slightly more realistic and abstract way in Figure 2.1.

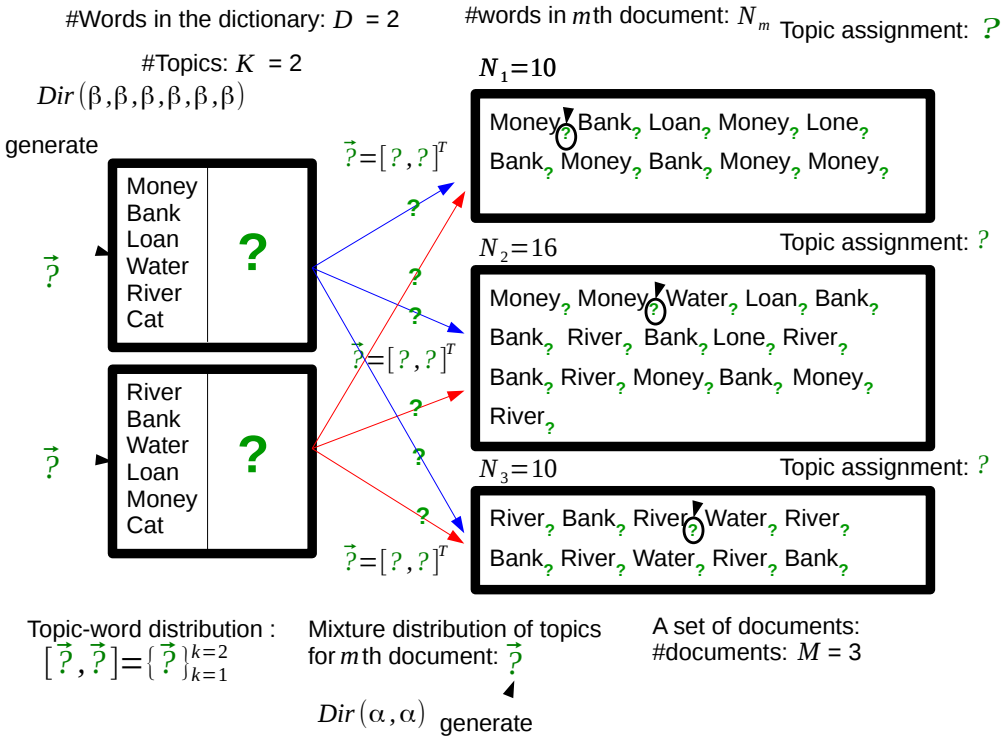

Figure 2.1: An example of a simple LDA model where there are two topics, each of which has a binomial distribution over words for each document. The figure is based on ([28], Figure in p.40).

As can be seen in Figure 2.1, only observed words are available as data and a lot of other hidden variables need to be estimated. However, it is important to note that the specific structure of this LDA model is actually satisfying the Assumption 1 to 3. For example, Figure 2.1 implies that two topics existence which are distribution over word, meaning that Assumption 1 is satisfied. Furthermore, the figure shows that each word is associated to its own topic assignment which is only dependent

on topic-distribution for that document, implying that Assumption 2 and 3 are also met. Hence, LDA is indeed a topic model.

As described in Figure 2.1, LDA models how each word in each document is generated over a corpus and such a specification of generating process with statistical assumption allow LDA to represent a collection of documents well. In the preceding sections, we will review LDA in details.

2.2 Model Assumption

Latent Dirichlet allocation (LDA) is a generative model. As briefly mentioned in the previous section, LDA has a few assumptions. Here, those are again reviewed in a clear way as blow: [15]:

1 A document is associated to multiple topics.

2 The word-order has semantically no influence on each document; in other words, there is the exchangeability of the words in a document.

3 The number of latent topics is known and constant.

4 LDA specifies certain dependency among hidden and observed variables with which the generating process of a set of documents is also specified (p.6 of [7].)

As 4th assumption claims, one of the key characteristics of LDA is that it is a generating model. To specify such a generating process of LDA, there exist several important distributions. As seen later in Section 2.5, it is important to note that the starting point of such a generating process of LDA is the assumption that there are multiples distributions over a set of words; in other words, topics. Such interpretation of topic is the core assumption of topic models. Hence, LDA surely satisfies the fundamental assumption in topic models. In the following section, two of those, *multinomial distribution* and *Dirichlet distribution* are firstly introduced, followed by a detailed description of LDA model.

2.3 Dirichlet distribution

To explain Dirichlet distribution, let us first recall binomial distribution. The binomial distribution ($B(n, p)$) is specified with parameters p and n, which respectively indicate the probability of success and total number of trials. With these parameters, the probability distribution of $B(n, p)$ can be given as follows [23]:

$$P(X = x|n, p) = \binom{n}{x} p^x (1-p)^{n-x}), \quad x \geq 0, \tag{2.1}$$

where x is the number of success and $(1 - p)$ means probability of fail.

The multinomial distribution is just an extension of the binomial distribution. More specifically, multinomial distribution differs from binomial distribution in a

way such that former allows an even take more than 2 possible outcomes, while the latter one only accepts binary outcomes. Thus, multinomial distribution's pdf is given as follows [23]:

$$P(x_1, \cdots, x_k | n, p_1, \cdots, p_k) = \frac{n!}{\prod_{i=1}^{k} x_i!} \cdot p_i^{x_i}, \quad \sum_{i=1}^{k} x_i = n, x_i \geq 0,$$

where $p_1, \cdots, p_k$ are probability of occurrence of each outcome, $x_1, \cdots, x_k$ are corresponding number of occurrence of each outcome, and n is the number of trials; in other words, total number of outcomes.

In the Bayesian context, the parameter is considered to have certain distribution. Such distribution is called *prior*, and the prior distribution for parameter of multinomial distribution $p_1, \cdots, p_k$ is often specified by the *Dirichlet distribution* characterized by parameter $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_k]$ and often denoted as $Dir(\boldsymbol{\alpha})$. The Dirichlet distribution is then described in the following [23]:

$$p(P = \{p_i\} | \boldsymbol{\alpha}) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} p_i^{\alpha_i - 1} \coloneqq B(\boldsymbol{\alpha}) \prod_{i=1}^{k} p_i^{\alpha_i - 1}, \quad \boldsymbol{p} \in \Delta^{k-1}, \quad \alpha_0 \coloneqq \sum_{i=1}^{k} \alpha_i, \tag{2.2}$$

where Γ is the gamma function and $\Delta^{K-1} = \{p \in \mathbb{R}_+^K \mid ||p||_1 = 1\}$. Specifically, Δ^{K-1} implies $\sum_{i=1}^{k} p_i = 1$; in the LDA context, the pdf of topic proportion for each document sums up to one. Note that $\boldsymbol{p} \in \Delta^{K-1}$ in the above indicates that all possible k dimensional vectors $\boldsymbol{p}$ sampled from $Dir(\boldsymbol{\alpha})$ reside in a $k-1$ dimensional subspace called $(k-1)$-*simplex* Δ^{K-1}. This constraint is due to the fact that $\boldsymbol{p}$ is a probability vector and its sum has to be always 1. For example, in LDA context, if there are only two topics $\boldsymbol{p} = [p_x, p_y]$, then sampled topic distributions always need to meet $p_x + p_y = 1$, which is just a line. Thus, for 2 dimensional Dirichlet distribution, sampled $\boldsymbol{p}$ is in 1-simple: line. Similarly, when $K = 3$, it is easily seen that $\boldsymbol{p}$ lie in the 2-simplex: triangle.

One of the reason why this prior is often used with multinomial distribution is that the Dirichlet distribution is the *conjugate prior* of multinomial; in other words, *posterior*, which is a distribution of parameter after observing data, of Dirichlet prior and likelihood (data) based on multinomial distribution is again Dirichlet distribution. A few example of Dirichlet distributions and effect of $\boldsymbol{\alpha}$ are given in the Figure 2.2. As seen in later, multinomial and Dirichlet distribution play an important role in LDA; especially, their conjugacy is crucial in the context of Bayesian inference for LDA. In the following section, we will introduce LDA model in details.

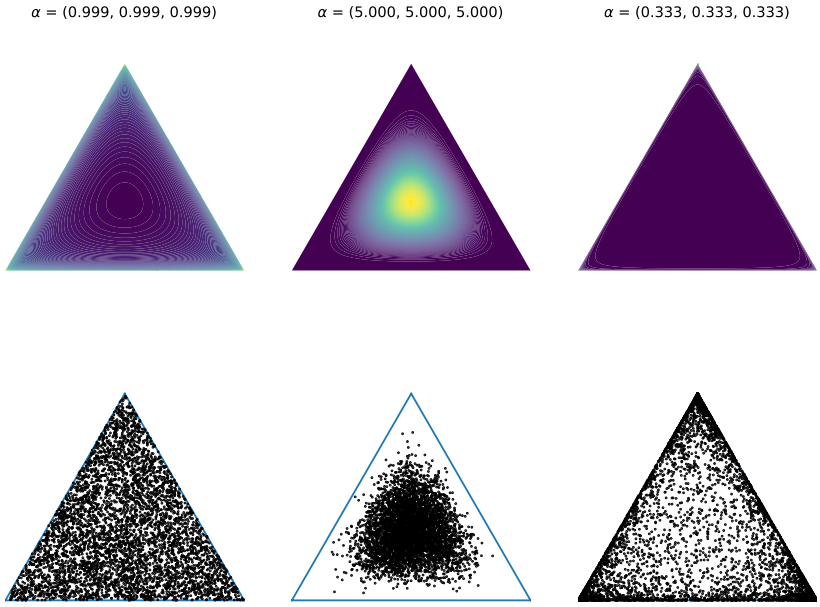

Figure 2.2: Examples of the Dirichlet distribution

Examples of 3-dimensional Dirichlet distributions with different parameter $\boldsymbol{\alpha}$. From three figures, the effect of $\boldsymbol{\alpha}$ values can be obsreved. Specifically, when all $\alpha_i \approx 1$, the distribution is almost uniform over the 2-dimensional simplex. When $= (5, 5, 5)$, corresponding distribution is symmetric; thus samples $\boldsymbol{p}$ are more likely to be drawn from around center. Note that it is very unlikely to draw a topic mixture such as $\boldsymbol{p} = [0, 0, 1]$, $\boldsymbol{p} = [0, 1, 0]$, or $\boldsymbol{p} = [1, 0, 0]$, $\boldsymbol{p} = [0, 1, 1]$, $\boldsymbol{p} = [1, 0, 1]$, or $\boldsymbol{p} = [1, 1, 0]$. This is not intuitively suitable for LDA because it is quite possible for a document which exhibits a few topics among all. When $\alpha_i < 1$ are taken, however, the corresponding distribution becomes sparse. For example, with the setting $\boldsymbol{\alpha} = [0.3333, 0.3333, 0.3333]$, it can be seen that most of samples $\boldsymbol{p}$ are either $\boldsymbol{p} = [0, 0, 1]$, $\boldsymbol{p} = [0, 1, 0]$, or $\boldsymbol{p} = [1, 0, 0]$, and it is less likely to draw a $\boldsymbol{p}$ all of whose entries are non-zero. In our experiments, based on the assumption that most of documents would be expressed with a few topics, symmetric Dirichlet distributions with $\alpha_i < 1$ are employed.

2.4 Model description

As mentioned in Section 2.2, in LDA, each document is modeled as mixture of multiple topics. The distribution from which such a topic proportion is drawn for each document is characterized by the Dirichlet distribution $Dir(\boldsymbol{\alpha})$ with parameter vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \cdots, \alpha_K]^T$. Here, K is the number of latent topics. Based on

Equation (2.2), the corresponding probability density function (pdf) for $Dir(\boldsymbol{\alpha})$ is given as follows [5]:

$$p_\alpha(h) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{K}\Gamma(\alpha_i)}\prod_{i=1}^{K}\theta_i^{\alpha_i-1} \coloneqq B(\boldsymbol{\alpha})\prod_{i=1}^{K}\theta_i^{\alpha_i-1}, \quad \boldsymbol{\theta}\in\Delta^{K-1}, \quad \alpha_0 \coloneqq \sum_{i=1}^{K}\alpha_i.$$

Note that Δ^{K-1} again suggests $\sum_{i=1}^{K}\theta_i = 1$. Let us consider Figure 1.1 as an example. In this example of LDA model, it is assumed that there are two topics, $\boldsymbol{\alpha}\in\mathbb{R}^2$ and $\boldsymbol{\theta} = [\theta_1, \theta_2]^T$. Indeed, for each of three documents, document-specific topic distributions, $\boldsymbol{\theta}_1 = [1,0]^T, \boldsymbol{\theta}_2 = [0.5, 0.5]^T, \boldsymbol{\theta}_3 = [0,1]$, are drawn from $Dir(\boldsymbol{\alpha})$.

Regarding the distribution of words given a topic (k), we assume it is given by the kth column vector of the $topics \times words$ probability matrix $M \coloneqq [\boldsymbol{\mu}_1|\boldsymbol{\mu}_2|\cdots|\boldsymbol{\mu}_K], \boldsymbol{\mu}_k \in \mathbb{R}^D$. Here, D is the number of words in a dictionary. In LDA, it is assumed that each distribution of words of kth topic $\boldsymbol{\mu}_k$ is independently generated from another Dirichlet distribution:

$$Dir(\boldsymbol{\beta}) = \frac{\Gamma(\beta_0)}{\prod_{d=1}^{D}\Gamma(\beta_i)}\prod_{d=1}^{D}\mu_i^{\beta_i-1} \coloneqq B(\boldsymbol{\beta})\prod_{d=1}^{D}\mu_i^{\beta_i-1}, \quad \boldsymbol{\mu}\in\Delta^{D-1}, \quad \beta_0 \coloneqq \sum_{d=1}^{D}\beta_i.$$

Again, let us have a look an example shown in Figure 1.1. As can be seen in this figure, there are two topics with 6 words in the dictionary and thus the distribution of words for each topic can be expressed as $\boldsymbol{\mu} = [Money, Bank, Loan, Water, River, Cat]^T$. More specifically, for topic 1, $\boldsymbol{\mu}_1 \approx [0.5, 0.3, 0.2, 0, 0, 0]^T$ and for topic 2, $\boldsymbol{\mu}_2 \approx [0, 0.3, 0, 0.1, 0.6, 0]^T$.

Furthermore, for the convenience of modeling, the l words appearing in a document are described using l unit-basis vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_l \in \mathbb{R}^D$. For example, $\boldsymbol{x}_l = \boldsymbol{e}_i$ if and only if lth word in the document in the corpus corresponds to ith word in the dictionary [5]. Let us again explain this concept using the example given in Figure 1.1. In this example LDA model, the distinct number of words appearing in the whole sets of documents, the dictionary size D in other words, is 6. Thus, any words in three documents are described by the unit-basis vector $\boldsymbol{x}\in\mathbb{R}^6$. For example, in the first document, there are 10 words and each word can be encoded into as follows:

$$\boldsymbol{x}_1 = [1,0,0,0,0,0]^T, \boldsymbol{x}_2 = [0,1,0,0,0,0]^T, \cdots, \boldsymbol{x}_{10} = [1,0,0,0,0,0]^T. \tag{2.3}$$

With these descriptions, the conditional expectation of the tth word in a document, $\boldsymbol{x}_t$, given jth topic is derived by marginalizing in d; namely,

$$\mathbb{E}(\boldsymbol{x}_t|\text{topic} = k) = \sum_{i=1}^{d}\boldsymbol{\mu}_{i,k}\boldsymbol{e}_i = \boldsymbol{\mu}_k.$$

2.5 Generating process

Here, the documents generating process assumed under the LDA model is briefly recalled; see [25, 8] for more details.

1 For each topic, a topic-word distribution $\boldsymbol{\mu}_k$ is drawn from $Dir(\boldsymbol{\beta})$ $(k = 1, \cdots, K)$.

2 For each document (a set of words $\boldsymbol{w}_m$) $(m = 1, \cdots, M)$,

   1 A document-specific topic proportion $\boldsymbol{\theta}_m$ (a multinomial distribution of topics) is drawn from $Dir(\boldsymbol{\alpha})$

   2 From the sampled document-specific distribution of topics, topic assignments (indexes) $z_{n,m} (1 \leq z_{n,m} \leq K)$ are individually drawn from the multinomial distribution $Mult(\boldsymbol{\theta}_m)$ for all N_m words in the mth document. Here, N_m is the number of words contained in the mth document.

   3 Finally, according to the sampled topic index $z_{n,m}$, nth word in the mth document $w_{n,m}$ is drawn from corresponding topic specific multinomial distribution of words whose pdf is characterized by $\boldsymbol{\mu}_{z_{n,m}}$.

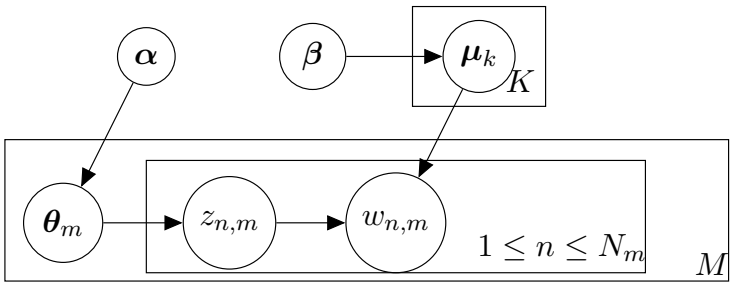

Figure 2.3: Graphical representation of the LDA mdoel.

The square is called *plate* and it indicates that the (latent) varialbes in a plate are repeatedly generated certain times specified with the index shown in the right corner of the plate. As mension earlier, in graphical model, the denpendency among latent and observed variables, which actually defines the LDA mdoel, is clearly depicted with nodes (circles) and arrows.

For a clear understanding the above process, let us explain using the example shown in Figure 1.1. In this example, the number of topics K is given as 2 and the number of words in the dictionary is 6. Hence, at first topic-words distributions $\boldsymbol{\mu}_i \in \mathbb{R}^6$ for each topic are drawn from $Dir(\boldsymbol{\beta})$ where $\boldsymbol{\beta} \in \mathbb{R}^6$. Note that the symmetric parameter is used for $Dir(\boldsymbol{\beta})$ with the same reason explained in Figure 2.2. This yields $[\boldsymbol{\mu}_1, \boldsymbol{\mu}_2]$. Then, for 1st document, a topic-proportion $\boldsymbol{\theta}_1$ is drawn from $Dir(\boldsymbol{\alpha})$ and obtained $[1, 0]$. Subsequently, for a first word, a topic assignment $z_{1,1} = 1$ is drawn from the multinomial (in this case actually binomial because we have only two topics) specified with $\boldsymbol{\theta} = [1, 0]$. Finally, a word "Money" is drawn from a topic-words distribution corresponding to the topic assignment; in the 1st word of 1st document case, $\boldsymbol{\mu}_{z_{1,1}} = \boldsymbol{\mu}_1$. This process continue for all words in the 1st document; namely until the 10th word "Money". After that, a new topic-proportion for the 2nd document $\boldsymbol{\theta}_2$ is drawn from $Dir(\boldsymbol{\alpha})$; in this case $Dir(\boldsymbol{\alpha}) = [0.5, 0.5]$. Then again

all 16 words are drawn in the same way explained above. The same procedure is done for third document, with which the data generating process from this LDA is completed. The generative assumption of LDA shown above can be well represented with a graphical representation; see Figure 2.3.

2.6 Parameter estimation

The goal of LDA is to estimate the parameter matrix, $M := [\boldsymbol{\mu}_1|\boldsymbol{\mu}_2|\cdots|\boldsymbol{\mu}_K]$ and $\boldsymbol{\theta} := (\theta_1, \theta_2, \cdots, \theta_K) = \mathbb{E}(\boldsymbol{\theta})$. To achieve this inference, we formulate the likelihood of (a generative) LDA model. Based on the generating process shown above, the likelihood of a document is given as follows [25]:

$$\begin{aligned} &p(\boldsymbol{w}_m, \boldsymbol{z}_m, \boldsymbol{\theta}_m, \{\boldsymbol{\mu}_k\}_{k=1}^{k=K}|\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &= p(\{\boldsymbol{\mu}_k\}_{k=1}^{k=K}|\boldsymbol{\beta}) \prod_{n=1}^{N_m} p(w_{m,n}|\boldsymbol{\mu}_{z_{m,n}}) p(z_{m,n}|\boldsymbol{\theta}_m) p(\boldsymbol{\theta}_m|\boldsymbol{\alpha}). \end{aligned}$$

For convenience of specifying the full-likelihood for the whole documents later, we consider marginalization over the hidden variables $\boldsymbol{z}_m, \theta_m, \{\boldsymbol{\mu}_k\}_{k=1}^{k=K}$ as in [25]:

$$\begin{aligned} &p(\boldsymbol{w}_m|\boldsymbol{\alpha}, \boldsymbol{\beta}) \\ &= \iint \sum_{\boldsymbol{Z}_m} p(\boldsymbol{w}_m, \boldsymbol{z}_m, \boldsymbol{\theta}_m, \{\boldsymbol{\mu}_k\}_{k=1}^{k=K}|\boldsymbol{\alpha}, \boldsymbol{\beta})\, d\{\boldsymbol{\mu}_k\}_{k=1}^{k=K}\, d\boldsymbol{\theta}_m \\ &= \iint p(\{\boldsymbol{\mu}_k\}_{k=1}^{k=K}|\boldsymbol{\beta}) p(\boldsymbol{\theta}_m|\boldsymbol{\alpha}) \prod_{n=1}^{N_m} \sum_{Z_{m,n}} p(w_{m,n}|\boldsymbol{\mu}_{z_{m,n}}) p(z_{m,n}|\boldsymbol{\theta}_m)\, d\{\boldsymbol{\mu}_k\}_{k=1}^{k=K}\, d\boldsymbol{\theta}_m \\ &= \iint p(\{\boldsymbol{\mu}_k\}_{k=1}^{k=K}|\boldsymbol{\beta}) p(\boldsymbol{\theta}_m|\boldsymbol{\alpha}) \prod_{n=1}^{N_m} p(w_{m,n}|\boldsymbol{\mu}_{k=1}^{k=K}, \boldsymbol{\theta}_m)\, d\{\boldsymbol{\mu}_k\}_{k=1}^{k=K}\, d\boldsymbol{\theta}_m. \end{aligned}$$

Taking product for whole documents, the full-likelihood is given as

$$p(\{\boldsymbol{w}\}_{m=1}^{M}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{m=1}^{M} p(\boldsymbol{w}_m|\boldsymbol{\alpha}, \boldsymbol{\beta}).$$

Note $\{\boldsymbol{w}\}_{m=1}^{M} = [\boldsymbol{w}_1, \cdots, \boldsymbol{w}_M]$ where $\boldsymbol{w}_m = [\boldsymbol{w}_{1,m}, \cdots, \boldsymbol{w}_{N_m,m}]$ indicates a set of all unit vectors corresponding to words of a corpus; namely the (observed) data we have.

This likelihood would be too complicated to simply apply traditional inference method such as the maximum likelihood approach. Furthermore, we arr often interested in information associated to hidden variables, for example knowing from which topic each word is drawn given a bunch of documents; hence, the posterior probability of all latent variables after observing a set of documents is often tried to be inferred. In other words, the distribution $P(\mathcal{Z}|\{\boldsymbol{w}\}_{m=1}^{M})$, where $\mathcal{Z}$ indicates a set of latent variables, is our main interest.

It is known that such exact inference is intractable for the LDA model, and therefore various methods for approximation, such as *Gibbs sampling* or *variational*

inference with expectation-maximization approach [8], are used according to [25]. In the following sections, we briefly recall *Gibbs sampling* and *collapsed Gibbs sampling* which have been frequently employed to achieve an approximation of inference in the LDA models.

2.6.1 Gibbs sampling

Gibbs sampling [21] is a kind of Markov Chain Monte Carlo (MCMC) simulation technique with which Bayesian inference can be performed. The core idea of MCMC is that any statistics of a posterior distribution can be estimable if we can draw a sufficiently large number of samples from that distribution [61]. Although this may sound easy, it is generally not straightforward to take a sample from posterior distribution, especially in a high dimensional situation such as the LDA case.

Gibbs sampling is a very useful sampling procedure when the structure of conditional distributions of a target distribution is available. This method can be regarded as a Metropolis-Hastings method where a series of proposal distributions are specified by conditional distributions of a target distribution [40]. One of the remarkable characteristics of Gibbs sampling is that it allows us to consider only an univariate conditional distribution [60]. For example, let us assume that we want to take a sample from a three dimensional target distribution $P(x_1, x_2, x_3)$. Then what we have to do is taking a sample from conditional distributions of one variable given the others:

$$
\begin{aligned}
x_1^t &\sim P(x_1|x_2^{t-1}, x_3^{t-1}) \\
x_2^t &\sim P(x_2|x_3^{t-1}, x_1^t) \\
x_3^t &\sim P(x_3|x_1^t, x_2^t) \\
x_1^{t+1} &\sim P(x_1|x_2^t, x_3^t) \\
&\vdots
\end{aligned}
$$

After the above Markov chain seems to have reached convergence at some iterations, which means that the samples taken later than that iterations can be thought to have the same distribution as the target posterior distribution [61], a summary statistics such as mean or mode are computed from those samples. It is important that samples taken in the beginning part of iterations do not represent the true posterior distribution well. Thus, those samples are often thrown away as the *burn in* period and are not used for calculating statistics.

2.6.2 Collapsed Gibbs sampling

In this section, we introduce one of the most popular inference methods for LDA: collapsed Gibbs sampling. The following discussion is based mainly on Section 5 of [25].

As mentioned before, in the LDA model, $P(\mathcal{Z}|\{\boldsymbol{w}\}_{m=1}^M)$ needs to be computed. Although it could be possible to directly work on this joint distribution, it would

be a good idea to first marginalize out $\{\boldsymbol{\theta}\}_{m=1}^{M}$ and $\{\boldsymbol{\mu}_k\}_{k=1}^{k=K}$ within $\mathcal{Z}$ and perform Gibbs sampling from $P(\{\boldsymbol{z}\}_{m=1}^{M}|\{\boldsymbol{w}\}_{m=1}^{M})$. This procedure is called collapsed Gibbs sampling. Note that this kind of marginalization is not always possible. As shown later, integrating over $\{\boldsymbol{\theta}\}_{m=1}^{M}$, $\{\boldsymbol{\mu}_k\}_{k=1}^{k=K}$ in the original target distribution of the LDA model leads to an elegant description, which allows us to draw its sample much efficiently. Note that $\{\boldsymbol{z}\}_{m=1}^{M}$ indicates a set of topic assignments for all words in a corpus; namely, $\{\boldsymbol{z}\}_{m=1}^{M} = (z_{1,1}, \ldots, z_{N_1,1}, z_{1,2}, \ldots, z_{N_2,2}, \cdots, z_{1,M}, \ldots, z_{N_M,M})$.

As the first step of collapsed Gibbs sampling, let us consider the conditional distribution of the topic assignment for ith word among whole words over M documents z_i given the other topic assignments $\{\boldsymbol{z}\}_{m=1}^{M-i}$ and all words $\{\boldsymbol{w}\}_{m=1}^{M}$: $p(z_i|\{\boldsymbol{z}\}_{m=1}^{M-i}, \{\boldsymbol{w}\}_{m=1}^{M})$. Using a basic probabilistic rule, this probability distribution can be written as follows [25]:

$$p(z_i|\{\boldsymbol{z}\}_{m=1}^{M-i}, \{\boldsymbol{w}\}_{m=1}^{M}) = \frac{p(\{\boldsymbol{z}\}_{m=1}^{M}, \{\boldsymbol{w}\}_{m=1}^{M})}{p(\{\boldsymbol{z}\}_{m=1}^{M-i}, \{\boldsymbol{w}\}_{m=1}^{M})} = \frac{p(\{\boldsymbol{w}\}_{m=1}^{M}|\{\boldsymbol{z}\}_{m=1}^{M})}{p(\boldsymbol{w}_{m=1}^{M-i}|\{\boldsymbol{z}\}_{m=1}^{M-i})p(w_i)} \frac{p(\{\boldsymbol{z}\}_{m=1}^{M})}{p(\{\boldsymbol{z}\}_{m=1}^{M-i})}. \tag{2.4}$$

Since $p(\boldsymbol{w}|\boldsymbol{z})$ and $p(\boldsymbol{z})$ respectively have $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ priors, each distribution can be dealt with separately. Recalling the graph structure, we can rewrite each distribution in the form of integration over its corresponding related latent variables $\{\boldsymbol{\mu}\}_{k=1}^{K}, \{\boldsymbol{\theta}\}_{m=1}^{M}$:

$$p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\beta}) = \int p(\boldsymbol{w}|\{\boldsymbol{\mu}\}_{k=1}^{K}, \boldsymbol{z}) p(\{\boldsymbol{\mu}\}_{k=1}^{K}|\boldsymbol{\beta}) \, \mathrm{d}\{\boldsymbol{\mu}_k\}_{k=1}^{K}, \tag{2.5}$$

$$p(\{\boldsymbol{z}\}_{m=1}^{M}|\boldsymbol{\alpha}) = \int p(\{\boldsymbol{z}\}_{m=1}^{M}|\{\boldsymbol{\theta}\}_{m=1}^{M}) p(\{\boldsymbol{\theta}\}_{m=1}^{M}|\boldsymbol{\alpha}) \, \mathrm{d}\{\boldsymbol{\theta}\}_{m=1}^{M}. \tag{2.6}$$

Under the LDA model's assumption, each word w_i and $\boldsymbol{w}_{m=1}^{M-i}$ are at least conditionally independent given $\{\boldsymbol{\mu}\}_{k=1}^{K}$. Similarly, z_i and $\boldsymbol{z}_{m=1}^{M-i}$ are at least conditionally independent given $\{\boldsymbol{\theta}\}_{m=1}^{M}$. Hence, we can describe $p(\boldsymbol{w}|\{\boldsymbol{\mu}\}_{k=1}^{K}, \boldsymbol{z}) p(\{\boldsymbol{\mu}\}_{k=1}^{K}|\boldsymbol{\beta})$ and $p(\{\boldsymbol{z}\}_{m=1}^{M}|\{\boldsymbol{\theta}\}_{m=1}^{M})$ as follows [25]:

$$p(\boldsymbol{w}|\{\boldsymbol{\mu}\}_{k=1}^{K}, \boldsymbol{\beta}) = \prod_{voc=1}^{D} \prod_{k=1}^{K} \mu_{voc,k}^{n_{d,k}}, \quad p(\{\boldsymbol{z}\}_{m=1}^{M}|\{\boldsymbol{\theta}\}_{m=1}^{M}) = \prod_{m=1}^{M} \prod_{k=1}^{K} \theta_{k,m}^{n_{k,m}}, \tag{2.7}$$

where $n_{d,k}$ and $n_{k,m}$ are respectively the total number of observed counts for the dth word of the dictionary belonging to kth topic over the whole documents and the total number of counts of topic assignments to the kth topic in the mth document. Inserting Equation (2.7) into Equations (2.5) and (2.6) as well as specifying the $Dir(\boldsymbol{\beta})$ and $Dir(\alpha)$ prior, we obtain the following [25]:

$$p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\beta}) = B(\boldsymbol{\beta}) \int \prod_{k=1}^{K} \prod_{voc=1}^{D} \mu_{voc,k}^{n_{d,k}+\beta_d-1} \, \mathrm{d}\boldsymbol{\mu}_k, \tag{2.8}$$

$$p(\{\boldsymbol{z}\}_{m=1}^{M}|\{\boldsymbol{\theta}\}_{m=1}^{M}) = B(\boldsymbol{\alpha}) \int \prod_{m=1}^{M} \prod_{k=1}^{K} \theta_{k,m}^{n_{k,m}+\beta_k-1} \, \mathrm{d}\boldsymbol{\theta}_m. \tag{2.9}$$

Noting that the Equation (2.8) except for $\prod_{k=1}^{K}$ and Equation (2.9) except for $\prod_{m=1}^{M}$ can be respectively regarded to be an unnormalized posterior composed from a multinomial likelihood function and conjugate Dirichlet prior. Therefore, those probabilities are easily written as [25]:

$$p(\boldsymbol{w}|\boldsymbol{z},\boldsymbol{\beta}) = \prod_{k=1}^{K} \frac{B(\boldsymbol{\beta})}{B(\boldsymbol{\beta}+\boldsymbol{n}_k)}, \quad \boldsymbol{n}_k = [n_{1,k}, n_{2,k}, \cdots, n_{D,k}]^T, \tag{2.10}$$

$$p(\{\boldsymbol{z}\}_{m=1}^{M}|\{\boldsymbol{\theta}\}_{m=1}^{M}) = \prod_{m=1}^{M} \frac{B(\boldsymbol{\alpha})}{B(\boldsymbol{\alpha}+\boldsymbol{n}_m)}, \quad \boldsymbol{n}_m = [n_{1,m}, n_{2,m}, \cdots, n_{K,m}]^T. \tag{2.11}$$

Substituting Equations (2.10) and (2.11) into Equation (2.4), the conditional distribution necessary for collapsed Gibbs sampling can be further simplified. For example, the conditional probability of a topic assignment of a word in mth document with dth dictionary index being kth topic can be given as:

$$\begin{aligned}
&p(z_i = k|\{\boldsymbol{z}\}_{m=1}^{M-i}, \{\boldsymbol{w}\}_{m=1}^{M}) \\
&= \frac{\prod_{k=1}^{K} \frac{B(\boldsymbol{\beta})}{B(\boldsymbol{\beta}+\boldsymbol{n}_k)}}{\frac{B(\boldsymbol{\beta})}{B(\boldsymbol{\beta}+\boldsymbol{n}_1)} \cdot \frac{B(\boldsymbol{\beta})}{B(\boldsymbol{\beta}+\boldsymbol{n}_2)} \cdot \cdots \cdot \frac{B(\boldsymbol{\beta})}{B(\boldsymbol{\beta}+\boldsymbol{n}_{d,k}^{(-1)})} \cdot \cdots \cdot \frac{B(\boldsymbol{\beta})}{B(\boldsymbol{\beta}+\boldsymbol{n}_K)}} \\
&\cdot \frac{\prod_{m=1}^{M} \frac{B(\boldsymbol{\alpha})}{B(\boldsymbol{\alpha}+\boldsymbol{n}_m)}}{\frac{B(\boldsymbol{\alpha})}{B(\boldsymbol{\alpha}+\boldsymbol{n}_1)} \cdot \cdots \cdot \frac{B(\boldsymbol{\alpha})}{B(\boldsymbol{\alpha}+\boldsymbol{n}_{k,m}^{(-1)})} \cdot \cdots \cdot \frac{B(\boldsymbol{\alpha})}{B(\boldsymbol{\alpha}+\boldsymbol{n}_M)}} \cdot \frac{1}{p(w_i)} \\
&= \frac{B(\boldsymbol{\beta}+\boldsymbol{n}_{d,k}^{(-1)})}{B(\boldsymbol{\beta}+\boldsymbol{n}_k)} \frac{B(\boldsymbol{\alpha}+\boldsymbol{n}_{k,m}^{(-1)})}{B(\boldsymbol{\alpha}+\boldsymbol{n}_m)} \frac{1}{c}, \quad c : \text{constant term} \\
&\propto \frac{\frac{\Gamma(\sum^{D} \beta_d + n_{d,k}^{(-1)})}{\Gamma(\beta_1 + n_{1,k}) \cdot \cdots \cdot \Gamma(\beta_d + n_{d,k} - 1) \cdot \cdots \cdot \Gamma(\beta_D + n_{D,k})}}{\frac{\Gamma(\sum^{D} \beta_d + n_{d,k})}{\prod_{d=1}^{D} \Gamma(\beta_d + n_{d,k})}} \\
&\cdot \frac{\frac{\Gamma(\sum^{K} \alpha_d + n_{k,m}^{(-1)})}{\Gamma(\alpha_1 + n_{1,m}) \cdot \cdots \cdot \Gamma(\alpha_k + n_{k,m} - 1) \cdot \cdots \cdot \Gamma(\alpha_K + n_{K,m})}}{\frac{\Gamma(\sum^{K} \alpha_k + n_{k,m})}{\prod_{k=1}^{K} \Gamma(\alpha_k + n_{k,m})}} \\
&\propto \frac{\Gamma(\sum^{D} \beta_d + n_{d,k}^{(-1)})\Gamma(\beta_d + n_{d,k})}{\Gamma(\sum^{D} \beta_d + n_{d,k})\Gamma(\beta_d + n_{d,k} - 1)} \cdot \frac{\Gamma(\sum^{K} \alpha_k + n_{k,m}^{(-1)})\Gamma(\alpha_k + n_{k,m})}{\Gamma(\sum^{K} \alpha_k + n_{k,m})\Gamma(\alpha_k + n_{k,m} - 1)},
\end{aligned}$$

where, $\boldsymbol{n}_{d,k}^{(-1)}, \boldsymbol{n}_{k,m}^{(-1)}$ are individually sets of word counts $n_{d,k}$ and topic assignment counts $n_{k,m}$ after ith word and topic assignment w_i, z_i are removed. Noting that

$\sum^D(\beta_d + n_{d,k}^{(-1)}) = \sum^D(\beta_d + n_{d,k}) - 1, \sum^K(\alpha_k + n_{k,m}^{(-1)}) = \sum^K(\alpha_k + n_{k,m}) - 1$, we can further simplify the probability $p(z_i = k|\{\boldsymbol{z}\}_{m=1}^{M-i}, \{\boldsymbol{w}\}_{m=1}^{M})$ using the key property of the Gamma function, namely $\Gamma(x+1) = x\Gamma(x)$, to that we get

$$
\begin{aligned}
& p(z_i = k|\{\boldsymbol{z}\}_{m=1}^{M-i}, \{\boldsymbol{w}\}_{m=1}^{M}) \\
& \propto \frac{\Gamma(\sum^D \beta_d + n_{d,k}^{-1})(\beta_d + n_{d,k} - 1))\Gamma(\beta_d + n_{d,k} - 1)}{(\sum^D(\beta_d + n_{d,k}) - 1)\Gamma(\sum^D \beta_d + n_{d,k} - 1)\Gamma(\beta_d + n_{d,k} - 1)} \\
& \cdot \frac{\Gamma(\sum^K \alpha_k + n_{k,m}^{-1})(\alpha_k + n_{k,m} - 1))\Gamma(\alpha_k + n_{k,m} - 1)}{(\sum^K(\alpha_k + n_{k,m}) - 1)\Gamma(\sum^K \alpha_k + n_{k,m} - 1)\Gamma(\alpha_k + n_{k,m} - 1)} \\
& \propto \frac{\beta_d + n_{d,k} - 1}{\sum^D(\beta_d + n_{d,k}) - 1} \cdot \frac{\alpha_k + n_{k,m} - 1}{\sum^K(\alpha_k + n_{k,m}) - 1}.
\end{aligned}
$$

The above result implies that the discrete conditional distribution from which collapsed Gibbs samplers are drawn can be characterized with only hyperprior for Dirichlet priors $\boldsymbol{\beta}, \boldsymbol{\alpha}$, word counts $n_{d,k}$, and topic counts $n_{k,m}$, which is much easy to compute.

After sufficient iterations enough to reach convergence, not only topic assignments are gained but $\{\boldsymbol{\theta}\}_{m=1}^{M}$ and $\{\boldsymbol{\mu}\}_{k=1}^{K}$ can be eventually computable using the final state of a Markov chain as given in the following [25]:

$$
[\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K]_{d,k} = \frac{\beta_d + n_{d,k}}{\sum_{d=1}^{D} \beta_d + \beta_d}, \quad [\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_M]_{k,m} = \frac{\alpha_k + n_{k,m}}{\sum_{k=1}^{K} \alpha_k + \alpha_k}. \tag{2.12}
$$

Equation (2.12) is derived from Bayesian inference. More specifically, with the Markov chain of updated topics assignments and observed words at hand $\mathcal{M} = [\boldsymbol{z}, \boldsymbol{w}]$, posteriors, $p(\boldsymbol{\theta}_m|\mathcal{M}, \boldsymbol{\alpha})$ for each document and $p(\boldsymbol{\mu}_k|\mathcal{M}, \boldsymbol{\beta})$ for each topic, can be estimated as given below [25]:

$$
p(\boldsymbol{\theta}_m|\mathcal{M}, \boldsymbol{\alpha}) = \frac{1}{c_1} \prod_{n=1}^{N_m} p(z_{m,n}|\boldsymbol{\theta}_m) p(\boldsymbol{\theta}_m|\boldsymbol{\alpha}) = Dir(\boldsymbol{\alpha} + \boldsymbol{n}_m), \tag{2.13}
$$

$$
p(\boldsymbol{\mu}_k|\mathcal{M}, \boldsymbol{\beta}) = \frac{1}{c_2} \prod_{z_i=k} p(w_i|\boldsymbol{\mu}_k)(\boldsymbol{\mu}_k|\boldsymbol{\beta}) = Dir(\boldsymbol{\beta} + \boldsymbol{n}_k), \tag{2.14}
$$

where c_1, c_2 are normalization constants. Note that the conjugacy between a Dirichlet prior and a multinomial is again used to obtain the last equivalence in both Equations (2.13) and (2.14).

In this way, the posterior inference for the parameter of a LDA model based on collapsed Gibbs sampling is completed.

2.7 Limitation of collapsed Gibbs sampling

2.7.1 Convergence

As mentioned before, one of common weaknesses in the collapsed Gibbs sampling and Gibbs sampling is slow mixing [63]. It is a condition of MCMC simulation

where the less frequent changes in the value for each hidden variable happen, which results in difficulty to vanish autocorrelation in the Markov chain [38]. In other word, it takes a significantly long time for the Markov chain to reach a state stable enough to be regarded as a sample from the target distribution. Indeed, in the case of application to the data of mixture distribution, Celeux, Hurn, and Robert demonstrate how rarely swapping in the values of latent variables for each iteration occur using an example of a Gaussian mixture of three components [13]. Since LDA is also ground from the mixture distribution assumption, it would encounter the same issue. To overcome such a challenge in collapsed Gibbs sampling procedure, it is probably important to try a lot of initial states and see if the results are stable or take large enough iterations. However, it is important to note that there is probably no guarantee that you have found the globally optimal solution even if the method converges.

2.7.2 Computational complexity

The computational complexity of collapsed Gibbs sampling for a LDA model is $O(MNK)$ [42] if we assume that each document has the same number of words N. Recall that M is the number of documents and K is the number of topics in a given corpus for the LDA model respectively‘. This implies that it could be difficult to apply this method when each document is assumed to be mixture of a lot of topics or data are from numerous number of documents or from documents with huge number of words.

2.8 Conclusion

Although a lot of studies have been done to break such a computational limitation, Zhang and Sisson point out that precedingly proposed inference method based on MCMC algorithm tend to suffer from a trade-off between the loss of efficient mixing in Markov chain and improvement computational time complexity [63].

On the other hand, as one of the promising methods for estimating the parameters of the LDA model, spectral methods were introduced in the papers [4, 52], which can basically be seen as computing an approximate *tensor decomposition*. In the next chapter, we will review some basic concept of tensors and their decompositions, and subsequently discuss their application for LDA.

Chapter 3

Tensors

3.1 Tensors

A tensor is a generalization of a matrix. It is namely a multidimensional array. The dimension of such an array is often called *order* or *mode*; see Figure 3.1. For example, a first-order tensor is a vector, a second-order tensor is a matrix, and tensors with orders greater than three are often called higher-order tensors [33]. Figure 3.1 displays a few examples of tensors.

As an element of matrix is often specified as $\boldsymbol{X}_{i,j}$ with subscription for its row

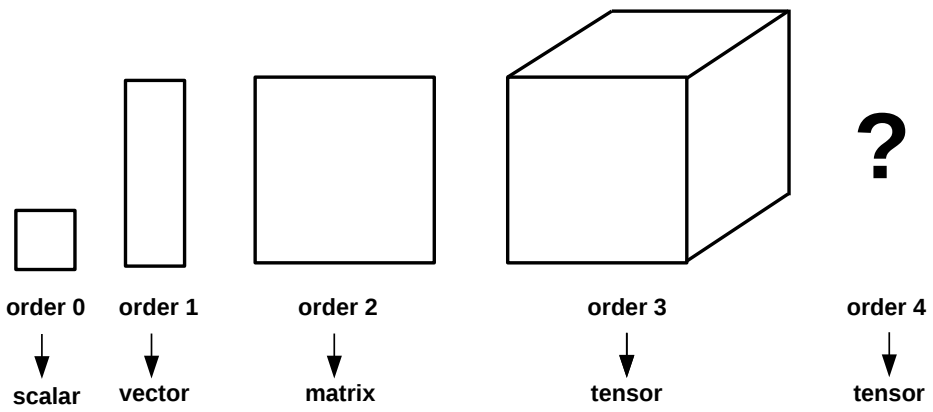

Figure 3.1: Examples of tensors.

From left to right, 0-way tensor (scalar), 1-way tensor (vector), 2- way tensor (matrix), 3-way tensor, and 4-way tensor.

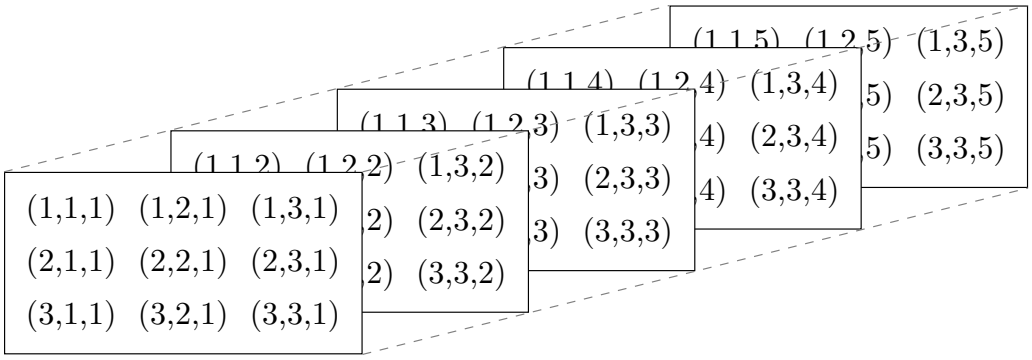

Figure 3.2: An example of indexing for a $3 \times 3 \times 5$ tensor $\mathcal{X}$: $\mathcal{X}_{i,j,k} = (i, j, k)$

and column number i, j, an element of tensor is described with subscription such as $\mathcal{X}_{i,j,k,\ldots}$. For example, Figure 3.2 shows the index system (i, j, k) for an $3 \times 3 \times 5$ tensor. Even though it can not draw a image for a tensor with its order more than 3, the way of indexing an element does not change.

Furthermore, as shown in Figure 3.2, we can even consider that the given tensor is composed from 5 different *slices*. For the three way tensors, a slice based on the same k, namely a matrix of $(:, :, k)$, is called *frontal slice*, a slice based on j is called *lateral slice*, and a slice based on i is called *horizontal slice*. These three types of slices are sometimes denoted as $\mathcal{X}_{:,:,k}, \mathcal{X}_{:,j,:}$, and $\mathcal{X}_{i,:,:}$ respectively. The idea of slices even leads to the concept called *matrcization* or *unfolding* of a tensor. This is a procedure to systematically transform a tensor into matrix. For example, given following a 3-way tensor $\mathcal{X} \in \mathbb{R}^{3\times3\times3}$ shown via frontal slices

$$\mathcal{X}_{:,:,1} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \quad \mathcal{X}_{:,:,2} = \begin{bmatrix} 10 & 11 & 12 \\ 13 & 14 & 15 \\ 16 & 17 & 18 \end{bmatrix} \quad \mathcal{X}_{:,:,3} = \begin{bmatrix} 19 & 20 & 21 \\ 22 & 23 & 24 \\ 25 & 26 & 27 \end{bmatrix}, \tag{3.1}$$

the mode-n matricizations are provided in the following way:

$$\begin{aligned} \mathcal{X}_{(1)} &= \begin{bmatrix} 1 & 2 & 3 & 10 & 11 & 12 & 19 & 20 & 21 \\ 4 & 5 & 6 & 13 & 14 & 15 & 22 & 23 & 24 \\ 7 & 8 & 9 & 16 & 17 & 18 & 25 & 26 & 27 \end{bmatrix} \\ \mathcal{X}_{(2)} &= \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 & 19 & 22 & 25 \\ 2 & 5 & 8 & 11 & 14 & 17 & 20 & 23 & 26 \\ 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 & 27 \end{bmatrix} \\ \mathcal{X}_{(3)} &= \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 10 & 13 & 16 & 11 & 14 & 17 & 12 & 15 & 18 \\ 19 & 22 & 25 & 20 & 23 & 26 & 21 & 24 & 27 \end{bmatrix}. \end{aligned} \tag{3.2}$$

Rank-one tensor is one of important types of tensors. An order-N tensor $\mathcal{X} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$ is called rank-one if $\mathcal{X}$ is expressed as outer products of N vectors; in

other words, $\mathcal{X}$ can be written down in the following form:

$$\mathcal{X} = \boldsymbol{v}_1 \circ \boldsymbol{v}_2 \circ \cdots \circ \boldsymbol{v}_N, \quad \boldsymbol{v}_1 \in \mathbb{R}^{I_1}, \boldsymbol{v}_2 \in \mathbb{R}^2, \cdots, \boldsymbol{v}_N \in \mathbb{R}^{I_N}. \tag{3.3}$$

Recall that an outer product of two vectors generates a matrix whose ith row and jth column entry is the product of ith row element of a vector and jth row of the other vector. The same concept can be extended to higher-order tensors. In other words, each element of the rank-one tensor $\mathcal{X}$ given above can be written as follows:

$$\mathcal{X}_{i_1,i_2,\cdots,i_N} = v_{1i_1} \cdot v_{2i_2} \cdots \cdot v_{Ni_N}, \tag{3.4}$$

where v_{ni_n} indicates the i_nth entry of the vector $\boldsymbol{v}_n$.

The outer product briefly introduced above is sometimes also called *tensor product*.

Tensor product is defined on two vector spaces. Specifically, if we suppose $\mathbb{V} \in \mathbb{R}^m$ and $\mathbb{W} \in \mathbb{R}^n$ be vector spaces, the tensor product between $\mathbb{V}$ and $\mathbb{W}$ is defined as a vector space $\mathbb{V} \otimes \mathbb{W}$, which is the image of a bilinear map shown below (Definition 3 of [24]):

$$\otimes : \mathbb{V} \times \mathbb{W} \longrightarrow \mathbb{V} \otimes \mathbb{W}.$$

Since a tensor product is itself a vector space, thus a basis of $\mathbb{V} \otimes \mathbb{W}$ can be considered. Such basis is defined by the all possible tensor products of basis vectors of $\mathbb{V}$ and basis vectors of $\mathbb{W}$. Let basis vectors of $\mathbb{V}$ be $\{\boldsymbol{e}_i\}_{i=1}^m$ and basis vectors of $\mathbb{W}$ be $\{\boldsymbol{f}_i\}_{j=1}^n$. Then, a basis of $\mathbb{V} \otimes \mathbb{W}$ is given as follows (Definition 4 of [24]):

$$\{\boldsymbol{e}_i \otimes \boldsymbol{f}_j\}_{i,j=1}^{m,n}, \tag{3.5}$$

where $\{\cdot\}_{i,j=1}^{m,n}$ indicates that basis are composed from $\boldsymbol{e}_i \otimes \boldsymbol{f}_j$ where all possible $i.j$ combinations of bases of two vector spaces are considered.

Let us consider an example to understand this concept easily. Let two of 1-way tensors $\boldsymbol{a}$ be $\boldsymbol{a} = [1, 2]^T$ and $\boldsymbol{b}$ be $\boldsymbol{b} = [3, 4]^T$. Then, the tensor product $\boldsymbol{a} \otimes \boldsymbol{b}$ is given as follows:

$$\begin{aligned}\boldsymbol{a} \otimes \boldsymbol{b} = \begin{bmatrix}1\\2\end{bmatrix} \otimes \begin{bmatrix}3\\4\end{bmatrix} &= \left(1\begin{bmatrix}1\\0\end{bmatrix} + 2\begin{bmatrix}0\\1\end{bmatrix}\right) \otimes \left(3\begin{bmatrix}1\\0\end{bmatrix} + 4\begin{bmatrix}0\\1\end{bmatrix}\right) \\ &= 1 \cdot 3\begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} + 1 \cdot 4\begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} + 2 \cdot 3\begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} + 2 \cdot 4\begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix}.\end{aligned}$$

It is easy to find that the tensor product $\boldsymbol{a} \otimes \boldsymbol{b}$ has four basis vectors; and its representation using matrix would be naturally introduced as follows:

$$\boldsymbol{a} \otimes \boldsymbol{b} = \begin{bmatrix}1 \cdot 3 & 1 \cdot 4\\ 2 \cdot 3 & 2 \cdot 4\end{bmatrix},$$

which is exactly corresponding to the outer product of $\boldsymbol{a}$ and $\boldsymbol{b}$: $\boldsymbol{a} \circ \boldsymbol{b}$.

As low rank decompositions of a matrix, such as the eigenvalue decomposition or the singular value decomposition are useful for real data analysis, it is natural

to extend such a concept to tensors. However, it is widely known that computing many types of tensor decompositions is a NP-hard problem [26]. Moreover, there might not exist a best approximation of a higher-order tensor [51]. The application of tensor decompositions to learning the parameter of latent variable models has recently been studied. [2, 4, 5].

3.2 The Tensor rank decomposition

In this section the tensor rank decomposition is briefly reviewed. The following sections are mainly based on [33, 19, 53, 17, 10].

3.2.1 CP-decomposition

To illustrate tensor decompositions, the concept of *rank* is essential. Sidiropoulos et al. explained the *rank* of tensor $\mathcal{X}$ as follows (Section 3 of [50]):

> "the minimum number of rank-one tensors needed to produce $\mathcal{X}$ as their sum."

Here, a rank-one tensor is an order-N tensor which is given by the outer product of N vectors [33]: $\mathcal{X} = \boldsymbol{a}^{(1)} \circ \boldsymbol{a}^{(2)} \circ \cdots \circ \boldsymbol{a}^{(N)}$. As an example, a 3-way rank one tensor is given in the bottom-left of Figure 3.3.

One of the popular tensor decompositions is the *canonical polyadic decomposition (CPD)*, which was originally introduced by Hitchcock [27], being discovered again later in the field of psychometrics [20]. It is defined as a decomposition of a tensor into a sum of rank-one tensors [12, 33]. For instance, CPD of a third-order tensor $\mathcal{X}$ is written as

$$\mathcal{X} = \sum_{r=1}^{R} \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r, \tag{3.6}$$

with a positive integer R, $\boldsymbol{a}_r \in \mathbb{R}^I$, $\boldsymbol{b}_r \in \mathbb{R}^J$, and $\boldsymbol{c}_r \in \mathbb{R}^K$ [33]. This means that an element of the tensor $\mathcal{X}_{i,j,k}$ can be expressed as following:

$$\mathcal{X}_{i,j,k} \equiv \sum_{r=1}^{R} a_{i,r} \cdot b_{j,r} \cdot c_{k,r}, \tag{3.7}$$

where $i = 1, \cdots, I, j = 1, \cdots, J, k = 1, \cdots, K$ [33]. Note $a_{i,r}$ means ith element of the vector $\boldsymbol{a}_r$. The top of Figure 3.3, for instance, shows rank-R CP decomposition for a 3-way tensor.

Furthermore, the vectors $\boldsymbol{a}_r, \boldsymbol{b}_r, \boldsymbol{c}_r, (r = 1, \cdots, R)$ are conventionally assumed to be normalized, meaning we can re-write the tensor as $\mathcal{X} \approx \sum_{r=1}^{R} \lambda_r \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r$ with normalization vector $\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_R) \in \mathbb{R}^R$ and it gives $||\boldsymbol{a}_r||^2 = ||\boldsymbol{b}_r||^2 = ||\boldsymbol{c}_r||^2 = 1$ in same norm [33].

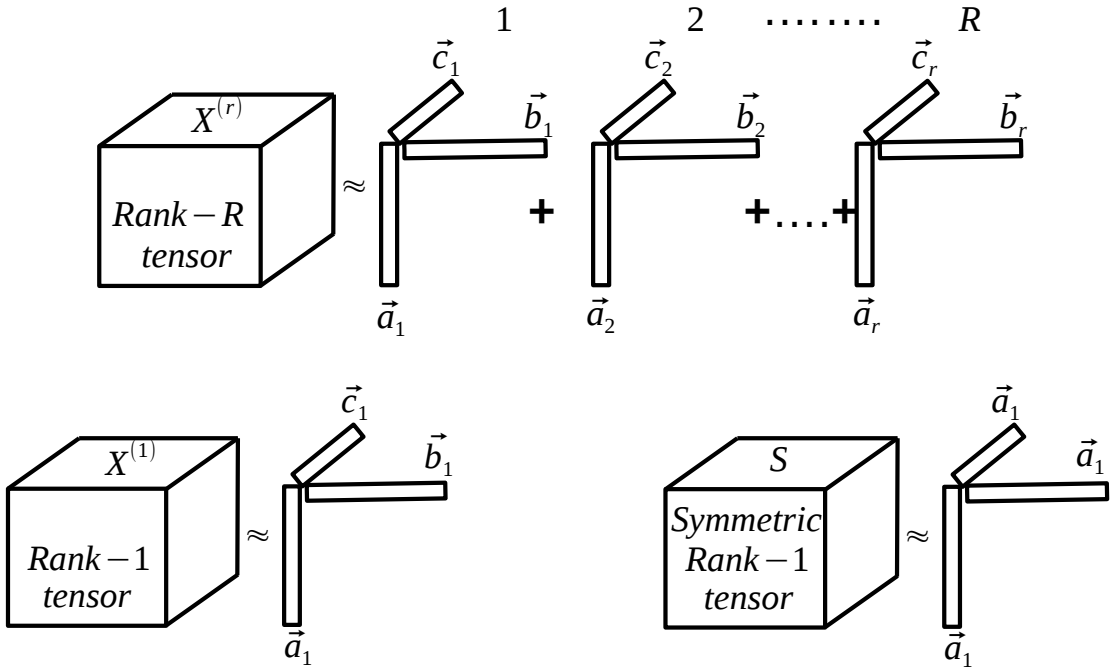

Figure 3.3: Examples of CP decompositions. The example was inspired by Figure 3.1 of [33].

Top: rank-R CP-decompositions on a 3-way tensor $X^{(r)} = \sum_{r=1}^{R} \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r$,
Bottom left: rank one tensors $X^{(1)} = \boldsymbol{a}_1 \circ \boldsymbol{b}_1 \circ \boldsymbol{c}_1$,
Bottom right: rank one symmetric tensor $S = \boldsymbol{a}_1 \circ \boldsymbol{a}_1 \circ \boldsymbol{a}_1$.

When a tensor has a CPD, it is also often written as [33]:

$$\mathcal{X} \equiv [\![A, B, C]\!] = \sum_{r=1}^{R} \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r, \tag{3.8}$$
$$A = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_R], B = [\boldsymbol{b}_1, \cdots, \boldsymbol{b}_R], C = [\boldsymbol{c}_1, \cdots, \boldsymbol{c}_R],$$

where A, B, C are called *factor matrices*. Using these factor matrices, we can actually obtain the matricizations (Equation (3.2)) of $\mathcal{X}$ as shown below (Equation 3.2 of [33]):

$$X_1 = A(C \odot B)^T, \quad X_2 = B(C \odot A)^T, \quad X_1 = C(B \odot A)^T, \tag{3.9}$$

where $\odot$ means *Khatri-Rao product* which is defined for two same column size matrices. For example, given $A := \{\boldsymbol{a}_i\}_{i=1}^{R}, A \in \mathbb{R}^{I \times R}$ and $B := \{\boldsymbol{b}_i\}_{i=1}^{R}, B \in \mathbb{R}^{J \times R}$, the Khatri-Rao product is defined as follows (Section 2.6 of [33]):

$$A \odot B := [\boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \;\; \boldsymbol{a}_2 \otimes \boldsymbol{b}_2 \;\; \cdots \boldsymbol{a}_R \otimes \boldsymbol{b}_R]. \tag{3.10}$$

Here, we emphasize that $\otimes$ appearing in the definition of Khatri-Rao product is *Kronecker product* of matrices. Kronecker product can be applied to matrices with different size of columns. For example, given two matrices $A := \{\boldsymbol{a}_i\}_{i=1}^{L}, A \in \mathbb{R}^{I \times L}$ and $B := \{\boldsymbol{b}_i\}_{i=1}^{M}, B \in \mathbb{R}^{J \times M}$, the Kronecker product of A and B is defined as follows (Section 2.6 of [33]):

$$A \otimes B = [\boldsymbol{a}_1 \otimes \boldsymbol{b}_1 \;\; \boldsymbol{a}_1 \otimes \boldsymbol{b}_2 \cdots \boldsymbol{a}_1 \otimes \boldsymbol{b}_M \;\; \boldsymbol{a}_2 \otimes \boldsymbol{b}_1 \cdots \boldsymbol{a}_L \otimes \boldsymbol{b}_M], \quad A \otimes B \in \mathbb{R}^{IJ \times LM}. \tag{3.11}$$

3.3 Algorithm for CP-decomposition

To achieve CP decomposition described above, a variety of algorithms have been proposed. In this section, we introduce a few of those algorithms. We emphasize that it is assumed that the rank of CP decomposition is known. As a main resource of Section 3.3.1, we consulted Section 3.4 of [33]. Regarding Section 3.3.2, Section 2.1 of [37] and Section 4.1 of [18] are main references. Finally, the description of *Structured Data fusion* given in Section 3.3.3, which we use in our experiments, is a summary of Sections 2 and 3 of [53].

3.3.1 Alternating least square method

Alternating least square method (ALS) is an optimization method and a family of least square methods. This method can be useful when we have to optimize an objective function with respect to a lot of variables. As Takane, Young, and De Leeuw explain in Section 3 of [55], the basic idea and flow of ALS are summarized in the Figure 3.4.

(1) Separate the whole set of variables into several subsets.
(2) Solve a least square problem regarding only one of such subsets of variables with remaining variables held constant and update variables of that subset.
(3) Repeat (2) until all subsets of variables are updated.
(4) Repeat (2) and (3) until certain convergence criteria are satisfied.

Figure 3.4: The basic flow of ALS

As an example, in the scenario of a CP-decomposition of a 3-way tensor, as shown by Kolda and Bader in Equation 3.7 of [33], the optimization problem can be formulated as

$$\min_{\tilde{\mathcal{X}}} ||\mathcal{X} - \tilde{\mathcal{X}}||, \quad \tilde{\mathcal{X}} = \sum_{r=1}^{R} \lambda_r \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r, \tag{3.12}$$

where $\mathcal{X}$ is a target tensor given in the form shown in Equation (3.8) and $\tilde{\mathcal{X}}$ is its estimate. Since Equation (3.12) implies that each element-wise difference between $\mathcal{X}$

and $\tilde{\mathcal{X}}$ needs to be minimized, this problem can be reformulated in the form of a matrix using the concept of slices and matricization of a tensor, with which the step (2) of Figure 3.4 for A is done by solving the following optimization problem [33]:

$$\min_{\tilde{A}} ||\mathcal{X}_{(1)} - \tilde{A}(C \odot B)^T||_F, \quad \tilde{A} = A \cdot diag(\boldsymbol{\lambda}), \tag{3.13}$$

where $||\cdot||_F$ indicates the Frobenius norm, which is sum of squares of the elements of the tensor, and $\odot$ indicates Khatri-Rao product. The corresponding solution is given as

$$\tilde{A} = \mathcal{X}_{(1)}[(C \odot B)^T]^\dagger = \mathcal{X}_{(1)}(C \odot B)(C^T C * B^T B))^\dagger, \tag{3.14}$$

where $\dagger$ means the pseudoinverse and $*$ indicates the element-wise product (*Hadamard product*). Note that the final equality in Equation (3.14) is obtained by the property of the pseudoinverse of the Khatri-Rao product and the final formulation of $\tilde{A}$ allows us to calculate $\tilde{A}$ more efficiently than the previous one [33]. Based on Equation (3.14), $\tilde{A}$ can be updated, with which $\tilde{B}$ is subsequently updated in the similar way. Finally, using those updated $\tilde{A}, \tilde{B}, \tilde{C}$ is updated. This cycle is continued until a convergence criterion is met. In this way, the CP decomposition of 3-way tensor is done.

3.3.2 Direct decomposition based on the generalized eigenvalue decomposition (GDVD) method

Generalized eigenvalue decomposition

The generalized eigenvalue decomposition is a procedure to discover so called *eigen-pairs of pencils of matrices (e.g. X, Y)* that are a set of (λ, v) such that

$$Xv = \lambda Y v \tag{3.15}$$

is satisfied for given square matrices of equal size X, Y (p.8 of [1]).

Example

A core concept of application of the generalized eigenvalue decomposition for CP decomposition is to simultaneously perform diagonalization on slices of a tensor. Let us consider a CP decomposition of a tensor $\mathcal{X} \in \mathbb{R}^{R \times R \times 2}$, assuming that its rank is known as R. As described below, under the rank-R CP decomposition (Equation (3.8)), each frontal slice for each node can be diagonalized with factor matrices A, B, C.

$$\mathcal{X}_{:,:,k} = \sum_{r=1}^{R} c_{k,r} \cdot \boldsymbol{a}_r \circ \boldsymbol{b}_r = A \cdot \Lambda_{\boldsymbol{C}_k} \cdot B^T (k = 1, 2), \tag{3.16}$$

$$C = \begin{bmatrix} c_{1,1} & \cdots & c_{1,R} \\ c_{2,1} & \cdots & c_{2,R} \end{bmatrix}, \quad \Lambda_{\boldsymbol{C}_k} = \begin{bmatrix} c_{k,1} & & \\ & \ddots & \\ & & c_{k,R} \end{bmatrix}.$$

Note that the notation for matricized tensor is corresponding to the one introduced in Chapter 3. If it is assumed that factor matrices A, B are of rank equal to R, $\mathcal{X}_{:,:,1}, \mathcal{X}_{:,:,2}$ are invertible. Thus, equations containing only A or B can be created as following:

$$\begin{aligned} \mathcal{X}_{:,:,1}\mathcal{X}_{:,:,2}^{-1} &= A\Lambda_{\boldsymbol{C}_1}B^T(A\Lambda_{\boldsymbol{C}_2}B^T)^{-1} = A\Lambda_{\boldsymbol{C}_1}\Lambda_{\boldsymbol{C}_2}^{-1}A^{-1} := A\Lambda_{\boldsymbol{C}}A^{-1}, \\ (\mathcal{X}_{:,:,2}^{-1}\mathcal{X}_{:,:,1})^T &= ((A\Lambda_{\boldsymbol{C}_2}B^T)^{-1}A\Lambda_{\boldsymbol{C}_1}B^T)^T = ((B^T)^{-1}\Lambda_{\boldsymbol{C}_2}^{-1}\Lambda_{\boldsymbol{C}_1}B^T)^T = B\Lambda_{\boldsymbol{C}}B^{-1}, \end{aligned} \tag{3.17}$$

where diagonal elements of $\Lambda_{\boldsymbol{C}}$ are distinct. Note that Equation (3.17) is actually a reformulation of Equation (3.15) where $X = \mathcal{X}_{:,:,1}, Y = \mathcal{X}_{:,:2}$ and hence this example can be interpreted as a generalized eigenvalue problem. By performing the eigenvalue decomposition on $\mathcal{X}_{:,:,1}\mathcal{X}_{:,:,2}^{-1}$ and $(\mathcal{X}_{:,:,2}^{-1}\mathcal{X}_{:,:,1})^T$ respectively, we can extract corresponding eigenvectors $[\boldsymbol{a}_1, \cdots \boldsymbol{a}_R]$ and $[\boldsymbol{b}_1, \cdots \boldsymbol{b}_R]$. Then by substituting obtained $\boldsymbol{a}_r$ and $\boldsymbol{b}_r$ into Equation (3.16), C is also calculated.

This is a intuitively brief illustration how CP decomposition can be done with generalized eigenvalue decomposition approach. In fact, above example is too restrictive, and a more relaxed approach based on simultaneous eigenvalue decomposition and detailed discussion can be found in the work [37].

3.3.3 Structured Data fusion based on nonlinear least square method

Data Fusion is one of the methods with which a more insightful analysis is aimed to be done through combining multiple data resources. As mentioned in Section 1 of [53, 36], one of classical examples of data fusion is Canonical correlation analysis proposed by Hotelling [30]. This method can quantitatively provide us with how much association exists between two different datasets. Such a new insight obtained with a joint analysis of multiple datasets would be a preferable aspect of data fusion. Interestingly, the concept of data fusion smoothly leads to the application of tensor. For example, let us recall the example discussed in Section 3.3.2. In this example, if we assume the $\mathcal{X} \in \mathbb{R}^{R\times R\times 2}$ is composed from two sets of datasets for different trials each of which is respectively summarized in each frontal slice $\mathcal{X}_{:,:,1}$ and $\mathcal{X}_{:,:,2}$, we can consider this CP decomposition problem as a data fusion.

Structured Data Fusion was considered by Sorber, Van Barel, and De Lathauwer [53]. This procedure can be applicable for a variety of tensor decompositions including CP decomposition. There are two remarkable advantages of this method. One is capability to jointly decompose the multiple tensors, namely multiple datasets, and the other is availability to specify a variety of structural constrains on factor matrices such as orthogonality or nonnegativity. Even when we need not to jointly analyze multiple sources of tensor-type data, it is still very powerful to be able to specify a structure of factor matrices. For instance, when a model in which a factor matrix is associated to the probability, posing non-negativity restriction on it would be reasonable.

The brief overview of such a structurally constrained tensor decomposition is presented in the below. According to Section 2 of [53], an underlying variable z

associated to a specific model is firstly considered, subsequently a factor matrix with certain structural restriction is generated by transforming z. For example, in a scenario of a non-negative decomposition of a single tensor $\mathcal{X}$, we would need to consider a model $\mathcal{M}(\mathcal{X}(z))$ to provide a tensor as a mapping of a factor matrix $\mathcal{X}(z)$, which is actually a function of z and characterized with non-negativity of its element. In this case, our target decomposition is thus achieved by solving the following optimization problem with respect to z.

$$\min_{z} \frac{1}{2}||\mathcal{M}(\mathcal{X}(z)) - \mathcal{X}||_F^2, \tag{3.18}$$

where $||\cdot||_F$ indicates the Frobenius norm. Note that Equation (3.18) is slightly different from the original formula given in Equation 1 of [53]. This is because data fusion and data incompleteness need not to be considered in this example case.

The above optimization problem 3.18 is tackled with nonlinear least squares methods. In this procedure, an initial solution z is firstly fed with either outputs from generalized eigenvalue decomposition on $\mathcal{X}$ or random values. Such initial solution is additively updated per each iteration through minimizing the second order approximation of the objective function given in Equation (3.18) with only first order derivative of the cost function calculated (Section 3 of [53]). The iteration stops when a preset convergence criterion is met and final solution is obtained. In this way, the structurally restricted tensor decomposition with nonlinear least square method is completed. We refer the reader to [44, 53] for more details.

3.4 Symmetric tensors

A *symmetric tensor* is one important type of tensor. To explain what *symmetric tensors* are, let us introduce the concept of *cubical* array. An N-way array $\mathcal{Y} \in \mathbb{R}^{n_1 \times \cdots \times n_N}$ is called cubical array if it satisfies $n_1 = \cdots = n_N$. When elements of a cubical array $\mathcal{X}$ are invariant under permutations of the indexes, namely $\mathcal{X}_{i_1,\cdots,i_N} = \mathcal{X}_{\sigma(i_1,\cdots,i_N)}$, such cubical array is called symmetric tensor; see Section 3 of [17]. Here, $\sigma(\cdot)$ means all possible permutation of an inputted indexes. For instance, the 3-way tensors $\mathcal{X} \in \mathbb{R}^{2\times 2\times 2}$ are symmetric if $\mathcal{X}_{1,1,2} = \mathcal{X}_{2,1,1} = \mathcal{X}_{1,2,1}$ and $\mathcal{X}_{1,2,2} = \mathcal{X}_{2,1,2} = \mathcal{X}_{2,2,1}$ (e.g. Section 3.4.1).

Given a cubical tensor, it can be symmetrized through a symmetrization map conventionally denoted as $\mathcal{S}$. For example, for a 3-way cubical tensor, such map $\mathcal{S}$ is given as

$$S : \mathbb{R}^{n\times n\times n} \longrightarrow \mathcal{S}^3\mathbb{R}^n \quad \boldsymbol{a}_1 \otimes \boldsymbol{a}_2 \otimes \boldsymbol{a}_3, \longmapsto \frac{1}{|\mathcal{S}|} \sum_{\sigma\in\mathcal{S}} \boldsymbol{a}_{\sigma(1)} \otimes \boldsymbol{a}_{\sigma(2)} \otimes \boldsymbol{a}_{\sigma(3)}. \tag{3.19}$$

Let us consider an example of symmetrization of a 3-way tensor given as $\boldsymbol{x}\otimes\boldsymbol{x}\otimes\boldsymbol{y}, \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{y} \in \mathbb{R}^n$. The symmetrized tensor can be written as image of above map: $\mathcal{S}(\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{y})$. Based on Equation (3.19), we can draw following relationship:

$$\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{y} + \boldsymbol{y} \otimes \boldsymbol{x} \otimes \boldsymbol{x} + \boldsymbol{x} \otimes \boldsymbol{y} \otimes \boldsymbol{x} = \frac{3!}{2} \cdot \mathcal{S}(\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{y}) = 3 \cdot \mathcal{S}(\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{y}). \tag{3.20}$$

Many researchers have found that the connection between the symmetric tensors and polynomials exists: for instance Section 3.1 of [17]. For an intuitive understanding, let us consider an example. Suppose we have a N-way symmetric tensor $\mathcal{X} = [\![U]\!], \quad \boldsymbol{u} \in \mathbb{R}^n$. Then, the bijective relationship between the tensor $\mathcal{X}$ and a homogeneous polynomial is provided as following (Section IV of [16]):

$$p(\boldsymbol{x}) = \sum_{i_1,\cdots,i_n}^{n} \mathcal{X}_{i_1,\cdots,i_n} \times x_{i_1} \times \cdots \times x_{i_n}, \tag{3.21}$$

where $\boldsymbol{x}$ is a set of variables $[x_1, \cdots, x_n]^T$. The same author also mentions that the above bijective relationship implies that a CP decomposition of $\mathcal{X}$ can be converted into a linear combination of a product of n linear forms (Equation 7 of [16]):

$$p(\boldsymbol{x}) = \sum_{i=1}^{R_s} (\boldsymbol{u}_i^T \boldsymbol{x})^N, \tag{3.22}$$

where R_s implicitly indicates the possible existence of different decomposition. The decomposition given in Equation (3.22) is called a *Waring decomposition* with respect to a polynomial (Section 6 of [46]) and the smallest R_s in Equation (3.22) is referred as *symmetric rank* of $\mathcal{X}$ (Section IV of [16]).

3.4.1 Example: Waring decomposition

Let us consider an simple example inspired by what Comon demonstrated (Example 16 of [16].) Suppose that we have a 3-way symmetric tensor $\mathcal{X} \in \mathbb{R}^{2\times2\times2}$ whose two frontal slices are individually given as

$$\mathcal{X}_{:,:,1} = \begin{bmatrix} 4 & 0 \\ 0 & -4 \end{bmatrix}, \quad \mathcal{X}_{:,:,2} = \begin{bmatrix} 0 & -4 \\ -4 & 0 \end{bmatrix}.$$

Based on Equation (3.21), the polynomial is given in the following:

$$p(\boldsymbol{x}) = p(x_1, x_2) = 4x_1 \cdot x_1 \cdot x_1 - 4x_2 \cdot x_2 \cdot x_1 - 4x_2 \cdot x_1 \cdot x_2 - 4x_1 \cdot x_2 \cdot x_2 = 4x_1^3 - 12x_1x_2^2.$$

Subsequently, we reformulate the above polynomial into the form of Equation (3.22):

$$\begin{aligned} p(x_1, x_2) &= 4x_1^3 - 12x_1x_2^2 = 8x_1^3 + (-x_1 - x_2)^3 + (-x_1 + x_2)^3 \\ &= 8\left(\begin{bmatrix} 1 \\ 0 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right)^3 + 2\left(\begin{bmatrix} -1 \\ -1 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right)^3 + 2\left(\begin{bmatrix} -1 \\ 1 \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right)^3. \end{aligned}$$

On the other hand, if we extend to complex field the polynomial can be written as

$$\begin{aligned} p(x_1, x_2) &= 4x_1^3 - 12x_1x_2^2 = 2(x_1 + ix_2)^3 + 2(x_1 - ix_2)^3 \\ &= 2\left(\begin{bmatrix} 1 \\ i \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right)^3 + 2\left(\begin{bmatrix} 1 \\ -i \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right)^3. \end{aligned}$$

Thus, the CP decomposition of $\mathcal{X}$ in $\mathbb{R}$ and $\mathbb{C}$ are respectively obtained as:

$$\begin{aligned}\mathcal{X} &= 8 \cdot \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} + 2 \cdot \begin{bmatrix}-1\\-1\end{bmatrix} \otimes \begin{bmatrix}-1\\-1\end{bmatrix} \otimes \begin{bmatrix}-1\\-1\end{bmatrix} + 2 \cdot \begin{bmatrix}-1\\1\end{bmatrix} \otimes \begin{bmatrix}-1\\1\end{bmatrix} \otimes \begin{bmatrix}-1\\1\end{bmatrix} \\ &= 2 \cdot \begin{bmatrix}1\\i\end{bmatrix} \otimes \begin{bmatrix}1\\i\end{bmatrix} \otimes \begin{bmatrix}1\\i\end{bmatrix} + 2 \cdot \begin{bmatrix}1\\-i\end{bmatrix} \otimes \begin{bmatrix}1\\-i\end{bmatrix} \otimes \begin{bmatrix}1\\-i\end{bmatrix}. \end{aligned} \tag{3.23}$$

Equation (3.23) suggests that the (symmetric) rank of a CP decomposition is not always the same in $\mathbb{R}$ and $\mathbb{C}$.

3.5 Uniqueness or identifiability

One of the most notable properties of CPDs of higher-order tensors is their uniqueness, which is not generally guaranteed for the rank decomposition of order-2 tensors [50]. For instance, a decomposition of a matrix $X = UV$, where $X \in \mathbb{R}^{m\times n}, U \in \mathbb{R}^{m\times r}, V \in \mathbb{R}^{r\times n}$, can be rewritten as $X = UPP^{-1}V = (UP)(P^{-1}V)$ with an invertible $P \in \mathbb{R}^{r\times r}$, indicating that it is not unique. On the other hand, for a CP-decompositions of the order-3 tensors, one fundamental concept of their uniqueness is based on the *Kruskal ranks* of the factor matrices A, B, C, which was introduced by Kruskal [35, 34].

Kruskal introduced a new concept of rank as following:

Theorem 1 (Kruskal [34]). *The Kruskal rank of a matrix X is written as k_X and defined as the largest number i such that any i sets of columns of X is linearly independent .*

To understand Theorem 1, let us consider a few example matrices and their Kruskal ranks as follows:

$$A = \begin{bmatrix}1 & 0 & 1\\0 & 1 & 1\\0 & 0 & 0\end{bmatrix}, \quad B = \begin{bmatrix}1 & 1 & 0\\0 & 0 & 1\\0 & 0 & 0\end{bmatrix}, \quad C = \begin{bmatrix}1 & 0 & 1\\0 & 1 & 1\\0 & 0 & 1\end{bmatrix}. \tag{3.24}$$

As mentioned in Theorem 1, we need to study linear dependency among all possible single-column sets, column-pair sets, or column-triplet sets in these examples to know the Kruscal ranks. Let us firstly have a look on single column sets. Since any columns of A, B, C are non-zero vectors, it is found that all corresponding Kruskal ranks are at least ≥ 1. When we see the column-pairs, on the other hand, it is revealed that only column-pairs extracted from A and C are linearly independent but those extracted from B are not. In fact, if we look at the pair of first and second columns of B, they are exactly same vector, meaning that they are linearly dependent. Thus, it is determined that Kruscal rank of B is 1 and Kruskal ranks of A, C are at least ≥ 2. Finally, when we study the linear independence among column-triplets of A and C, we observe that all columns of C are linearly independent while those of A are not; indeed, the third column of A is just a sum of first and second columns. Hence, the Kruscal ranks of A and C are respectively 2 and 3. Note there is no guarantee that

the Kruscal rank of a matrix match its matrix rank. In fact, the obtained Kruscal ranks of A, B, C are 2, 1, 3, whereas their matrix ranks are individually 2, 2, 3.

Using this concept, Kruskal proposed one of fundamental statements on uniqueness of CP decomposition as follows:

Theorem 2 (Kruskal [35, 34]). *$k_A + k_B + k_C \geq 2R + 2$ is a sufficient condition for the uniqueness of the CP-decomposition of $\mathcal{X}$ given in (3.8) .*

Furthermore, the uniqueness of *generic* low-rank *symmetric CP-decompositions*, e.g. $\mathcal{X} = \sum_{r=1}^{R} \boldsymbol{a}_r \circ \boldsymbol{a}_r \circ \boldsymbol{a}_r$, has recently been proved [14]. Here, "generic" means that if you choose the parameters of the model from any continuous probability density function with nonlocal support, then the probability that a sample from this distribution is unique is equal to 1.

Note, uniqueness mentioned here means there is only one set of rank-one tensors which satisfies Equation (3.8)); in other words, non-uniqueness coming from permutation of $\sum_{r=1}^{R} \boldsymbol{a}_r \circ \boldsymbol{b}_r \circ \boldsymbol{c}_r$ and scaling is not considered [2, 33]. These results of uniqueness for order-3 tensors give an insight on the identifiability of parameter in statistical inference for some types of latent variable models, especially when there are three observable features.

3.6 Sensitivity

Once the tensor decomposition is done, the sensitivity analysis of an obtained solution of approximation should be carried out to investigate if it is still stable when the problem slightly changes. For this reason, in the research field of numerical linear algebra, *condition number* has been employed. In the following section, we first illustrate the classical concept about condition number, secondly introducing a recently proposed concept *geometric condition number*, which we use for our experiment. As main references for Section 3.6.1 and Section 3.6.2, Lecture 12 of [57] and [10] are used.

3.6.1 Absolute condition number

The first step to establish the concept of condition number is to regard a problem as a function f mapping a vector space $\mathbb{V}$, which is a set of data, to another vector space $\mathbb{X}$, which is a set of corresponding solutions: $f : \mathbb{V} \longrightarrow \mathbb{X}$ (p.89 of [57]). From such $\mathbb{V}$, a point (data) $\boldsymbol{x}$ is chosen and a slight perturbation from that point $\delta\boldsymbol{v} := \mathbf{v} - \boldsymbol{v}$ is secondly considered to quantitatively evaluate the shift in corresponding function value: $\delta f := f(\mathbf{v}) - f(\boldsymbol{v})$. For certain data at hand, which is again a specific point in $\mathbb{V}$, the problem is said to be *well-conditioned* when any tiny permutations from that point result in a small δf. Vice versa, when such kinds of permutations end up with a large value δf, the problems are called *ill-conditioned*. This suggests that considering ratio between δf and $\delta\boldsymbol{v}$ enables us to assess a problem-specific characteristic. The ratio when $||\delta\boldsymbol{v}||$ is infinitely approaching 0 is called *absolute condition number* and

defined in the following (Equation 12.1 of [57] and 1.2 of [10]):

$$\kappa[f](\boldsymbol{v}) = \lim_{\epsilon \to 0} \sup_{||\delta \boldsymbol{v}|| \leqq \epsilon} \frac{||\delta f||}{||\delta \boldsymbol{v}||}. \tag{3.25}$$

An absolute condition number provides us with an quantitative insight about the complexity of a problem; here, complexity might be interpretable as difficulty of obtaining a robust solution given similar data. As explained in [10], this information is practically of great importance. For instance, in a tensor decomposition context, although a pure tensor free from noise is rarely accessible in practice as data, it could be still possible to attain a reasonable solution if condition number is around 1. This is because such condition number implies that a small degree of noise on data has a small effect on the solution.

Example

To get a more intuitive understanding of the condition numbers, let us consider an example of a simple linear equation problem using MATLAB as Vannieuwenhoven did in [59]. Given a matrix A and a vector $\boldsymbol{b}$, our goal is to find the solution vector $\boldsymbol{x}$; namely $A\boldsymbol{x} = \boldsymbol{b}$ needs to be solved in terms of $\boldsymbol{x}$. Suppose A and two slightly different $\boldsymbol{b}$, $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$, are explicitly provided as follow:

```
>> A = [3,7;3,7.0001], b1 = [10;10.0001], b2 = [10;10.0000]
```

$$A = \begin{bmatrix} 3.0000 & 7.0000 \\ 3.0000 & 7.0001 \end{bmatrix} \quad \boldsymbol{b}_1 = \begin{bmatrix} 10.0000 \\ 10.0001 \end{bmatrix} \quad \boldsymbol{b}_2 = \begin{bmatrix} 10.0000 \\ 10.0000 \end{bmatrix}.$$

Since A is invertible, we obtain two solution vectors $\boldsymbol{x}_1, \boldsymbol{x}_2$ for $\boldsymbol{b}_1, \boldsymbol{b}_2$ cases respectively:

```
>> x1 = inv(A)*b1, x2 = inv(A)*b2, Ainv = inv(A)
```

$$\boldsymbol{x}_1 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad \boldsymbol{x}_2 = \begin{bmatrix} 3.3333 \\ 0 \end{bmatrix} \quad A^{-1} = 10^4 \begin{bmatrix} 2.3334 & -2.3333 \\ -1.0000 & 1.0000 \end{bmatrix}.$$

Here, we observe that two solution appear to be different. To investigate in details, the distance between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ is compared with the one between $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$:

```
>> difx = norm(x1-x2), difb = norm(b1 -b2)
```

$$\texttt{difx} = ||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2 = 2.5386, \quad \texttt{difb} = ||\boldsymbol{b}_1 - \boldsymbol{b}_2||_2 = 10^{-4}.$$

This result might be surprising since a seemingly tiny perturbation of $\boldsymbol{b}$ of size 10^{-4} leads to obtaining a different solution $\boldsymbol{x}$ which is apart from the original solution in Euclidean distance of order $2.5 \approx 10^0$. However, this kinds of perturbation effect was actually expected by calculating the condition number even before an actual calculation is done.

Let us suppose that $||\delta \boldsymbol{b}||= 10^{-4}$ is small enough and thus consider only $\frac{||\delta f||}{||\delta \boldsymbol{v}||}$ term of Equation (3.25) to calculate $\hat{\kappa}$. Under this assumption, the condition number is calculated as

```
>> kappa = norm(inv(A)*b1-inv(A)*b2)/norm(b1-b2)
```

$$\hat{\kappa} \approx 2.5386 \cdot 10^4. \tag{3.26}$$

Since the approximated $\hat{\kappa}$ appears to be large, this example problem would be regarded to be ill-conditioned.

Now let us think about the condition number of applying A^{-1} to a vector. As will be shown later, this condition number is actually equal to $||A^{-1}||_2$. The difference from Equation (3.26) is that the maximum over all $||\boldsymbol{b}_1 - \boldsymbol{b}_2||$ within ϵ should be taken at this moment as shown in Equation (3.25). Hence, the condition number of A^{-1} $\kappa_{\hat{A^{-1}}}$ is calculated as follows:

$$\begin{aligned}
&\hat{\kappa}_{A^{-1}} \\
&= \max_{||\boldsymbol{b}_1-\boldsymbol{b}_2||_2} \frac{||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2}{||\boldsymbol{b}_1 - \boldsymbol{b}_2||_2} = \max_{||\boldsymbol{b}_1-\boldsymbol{b}_2||_2} \frac{||A^{-1}\boldsymbol{b}_1 - A^{-1}\boldsymbol{b}_2||_2}{||\boldsymbol{b}_1 - \boldsymbol{b}_2||_2} = \max_{||\boldsymbol{b}_1-\boldsymbol{b}_2||_2} \frac{||A^{-1}(\boldsymbol{b}_1 - \boldsymbol{b}_2)||_2}{||\boldsymbol{b}_1 - \boldsymbol{b}_2||_2}.
\end{aligned} \tag{3.27}$$

Note that the Equation (3.27) is exactly the definition of the spectral norm of A^{-1}. In our example,

```
>> kappa_upper = norm(Ai nv)
```

$$\hat{\kappa}_{A^{-1}} \approx 3.5901 \cdot 10^4. \tag{3.28}$$

Using $\hat{\kappa}_{upper}$, we may be able to expect a perturbation effect as

```
>> kappa_upper*norm(b1-b2)
```

$$\widehat{||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2} = 3.5901. \tag{3.29}$$

In this way, some expectation of $||\boldsymbol{x}_1 - \boldsymbol{x}_2||_2$ corresponding to a small perturbation can be drawn beforehand.

3.6.2 Geometric condition number

It is proved in [10] that the function

$$\begin{aligned}
\Phi :& S \times S \cdots \times S \to \mathbb{R}^N, \\
& (\boldsymbol{a}_1 \otimes \boldsymbol{a}_2 \otimes \boldsymbol{a}_3, \cdots, \boldsymbol{a}_r \otimes \boldsymbol{b}_r \otimes \boldsymbol{c}_r) \mapsto [\boldsymbol{a}_1 + \boldsymbol{a}_2 + \boldsymbol{a}_3 + \cdots + \boldsymbol{a}_r + \boldsymbol{b}_r + \boldsymbol{c}_r],
\end{aligned}$$

where S is the set of rank-1 tensors, is a function that is locally invertible under some advanced conditions (Lemma 2.1 in [10]). This local inverse function Φ_p^{-1} takes

as input a rank-r tensor $p1 + \cdots + p_r$ and sends it back to the tuple $(p_1, \cdots, p_r)$. So this is a function that solves the tensor decomposition problem. It would be immediately noticed that Φ cannot be an invertible function in the classic sense, because $\Phi^{-1}(p)$ has a multiple of $r!$ preimages, because $\Phi^{-1}(p) = (p_1, \cdots, p_r)$, but also $\Phi^{-1}(p) = (p_r, \cdots, p_1)$. Indeed, $p_r + \cdots + p_1 = p = p_r + \cdots + p_1$. This happens for all of the permutations.

[10] was able to derive a condition number for these local inverse functions Φ_p^{-1}; see Theorem 1.1 in the reference. Let us denote this condition number by $\kappa_{geo}(\mathfrak{p}, \mathbf{p})$. The authors of [10] also found that $\kappa_{geo}(\mathfrak{p}, \mathbf{p})$ satisfies an following conventional rule in the field of numerical analysis (Equation 1.4 of [10])

$$||\mathfrak{p} - \Phi_{\mathfrak{p}}^{-1}(\mathbf{w})|| \lesssim \kappa_{geo}(\mathfrak{p}, \boldsymbol{p}) \cdot ||\Phi(\mathfrak{p}) - \mathbf{w}||, \tag{3.30}$$

where $||\mathfrak{p} - \Phi_{\mathfrak{p}}^{-1}(\mathbf{w})||$ and $||\Phi(\mathfrak{p}) - \mathbf{w}||$ are respectively forward error and backward error. Equation (3.30) implies that $\kappa_{geo}(\mathfrak{p}, \mathbf{p})$ can be easily computed, as explained in Section 5.1 of [10]. For example, Using Equation (3.30), we could make a guess on the $\kappa_{geo}(\mathfrak{p}, \boldsymbol{p})$ with synthetic data. For example, let us consider the geometric condition number for a rank r symmetric CP decomposition of a 3-way tensor $X = [\![\mathrm{U,U,U}]\!]$ with its observed tensor $\tilde{\mathcal{X}}$ and corresponding calculated factor matrix $\hat{U}$. In this scenario, Equation (3.30) can be written as follow:

$$\begin{aligned} &||K - \tilde{K}||_F \leq \kappa_{geo} \cdot ||\mathcal{X} - \tilde{\mathcal{X}}||_F, \\ &K := kr(U) = [\boldsymbol{p}_1, \cdots, \boldsymbol{p}_r], \quad K := kr(\tilde{U}) = [\boldsymbol{q}_1, \cdots, \boldsymbol{q}_r], \\ &\mathcal{X} = \boldsymbol{p}_1 + \cdots \boldsymbol{p}_r, \quad \tilde{\mathcal{X}} = \boldsymbol{q}_1 + \cdots \boldsymbol{q}_r, \end{aligned} \tag{3.31}$$

where, $kr(\cdot)$ indicate the KhatriRao product; e.g. $kr(U) = U \odot U \odot U$. If we simulate a synthetic data, the true T, U are known and hence K_{geo} can be roughly approximated. Note that we have just briefly introduced the basic idea of geometric condition number. For more elaborated explanation or discussion on it, [10, 9, 59] are recommendable.

Chapter 4

Tensor Decomposition for LDA

4.1 Related work

As mentioned in Section 2.6, under the LDA model, the parameter $M := [\boldsymbol{\mu}_1|\boldsymbol{\mu}_2|\cdots|\boldsymbol{\mu}_K]$ can be estimated by using a tensor decomposition approach. Anandkumar et al. [4] show that parameter M and Dirichlet prior $\boldsymbol{\alpha}$ can be recovered by applying *Excess Correlation Analysis (ECA)* based on multiple implementation of SVD to a modified third-order cross moment $\mathbb{E}(\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x})$. They also mention that observing at least three words in each document is enough to estimate $\mathbb{E}(\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x})$. As an alternative approach, the direct decomposition relying on *orthogonal tensor decomposition* and *tensor power method* is presented in the study [5], which was found to be remarkable in its performance with respect to robustness and speed [4].

4.2 Parameter estimation

From hereon, an approach of parameter estimation on LDA which will be adopted in this thesis is described. The fundamentals idea of this method stem from the following theorem, which was proved in the studies [5, 4].

Theorem 3 (Anandkumar et al. [4, 5])**.** *Define*

$$\begin{aligned} M_1 :=& \mathbb{E}[\boldsymbol{x}_1], \quad M_2 := \mathbb{E}[\boldsymbol{x}_1 \otimes \boldsymbol{x}_2] - \frac{\alpha_0}{\alpha_0+1} M_1 \otimes M_1, \\ M_3 :=& \mathbb{E}[\boldsymbol{x}_1 \otimes \boldsymbol{x}_2 \otimes \boldsymbol{x}_3] - \frac{\alpha_0}{\alpha_0+2}\Big(\mathbb{E}[\boldsymbol{x}_1 \otimes \boldsymbol{x}_2 \otimes M_1] + \mathbb{E}[\boldsymbol{x}_1 \otimes M_1 \otimes \boldsymbol{x}_2] + \\ & \mathbb{E}[M_1 \otimes \boldsymbol{x}_1 \otimes \boldsymbol{x}_2]\Big) + \frac{2\alpha_0^2}{(\alpha_0+2)(\alpha_0+1)} M_1 \otimes M_1 \otimes M_1. \end{aligned} \tag{4.1}$$

Then

$$M_2 = \sum_{k=1}^{K} \frac{\alpha_k}{(\alpha_0+1)\alpha_0} \boldsymbol{\mu}_k \otimes \boldsymbol{\mu}_k, \tag{4.2}$$

$$M_3 = \sum_{k=1}^{K} \frac{2\alpha_k}{(\alpha_0+2)(\alpha_0+1)\alpha_0} \boldsymbol{\mu}_k \otimes \boldsymbol{\mu}_k \otimes \boldsymbol{\mu}_k. \tag{4.3}$$

Recall that $\boldsymbol{x}_i$ appearing in the above are variables recorded as a unit basis vector explained in Section 2.4, respectively corresponding to ith observed word. Hence, $\mathbb{E}(\boldsymbol{x}_1)$ is expected to asymptotically give information on probability of occurrence of each word in the dictionary based on the given corpus. $\boldsymbol{x}_1 \otimes \boldsymbol{x}_2$ are variables corresponding to pair-words occurrence in each document; thus it is a matrix of size $D \times D$. $\boldsymbol{x}_1 \otimes \boldsymbol{x}_2 \otimes \boldsymbol{x}_3$ are variables corresponding to triple-word occurrence in each document; hence, it is a cubical tensor in $R^{D \times D \times D}$. The $\alpha_0 = \sum_{k=1}^{K} \alpha_k$ is sum of parameter of the Dirichlet distribution from which topic-distributions are generated.

However, Ruffini, Casanellas, and Gavaldvariation [49] point out the importance to take the length of each document into consideration to attain higher reliability when calculating observed moment, and proposed the following theorem as an extension of previous work [64]:

Theorem 4 (Ruffini, Casanellas, and Gavald [49]). *Define*

$$\tilde{M}_{1h} := \frac{\sum_{i=1}^{M} (X^{(i)})_h}{\sum_{i=1}^{M} N_i}, \tag{4.4}$$

$$\tilde{M}_{2h,l,} := \frac{\sum_{i=1}^{M} (X^{(i)})_h (X^{(i)})_l}{\sum_{i=1}^{M} (N_i - 1) N_i}, \tag{4.5}$$

$$\tilde{M}_{2h,h} := \frac{\sum_{i=1}^{M} (X^{(i)})_h ((X^{(i)})_h - 1)}{\sum_{i=1}^{M} (N_i - 1) N_i}, \tag{4.6}$$

$$\tilde{M}_{3h,l,m} := \frac{\sum_{i=1}^{M} (X^{(i)})_h (X^{(i)})_l (X^{(i)})_m}{\sum_{i=1}^{M} (N_i - 2)(N_i - 1) N_i}, \tag{4.7}$$

$$\tilde{M}_{3h,l,l} := \frac{\sum_{i=1}^{M} (X^{(i)})_h (X^{(i)})_l ((X^{(i)})_l - 1)}{\sum_{i=1}^{M} (N_i - 2)(N_i - 1) N_i}, \tag{4.8}$$

$$\tilde{M}_{3l,l,l} := \frac{\sum_{i=1}^{M} (X^{(i)})_l ((X^{(i)})_l - 1)((X^{(i)})_l - 2)}{\sum_{i=1}^{M} (N_i - 2)(N_i - 1) N_i}, \tag{4.9}$$

$$M_2^{\alpha} := \tilde{M}_2 - \frac{\alpha_0}{\alpha_0 + 1} \tilde{M}_1 \otimes \tilde{M}_1 \tag{4.10}$$

$$\tilde{M}_3^{\alpha} := \tilde{M}_3 - \frac{\alpha_0}{\alpha_0 + 2} (M_{1,2}) + \frac{2\alpha_0^2}{(\alpha_0 + 2)(\alpha_0 + 1)} \tilde{M}_1 \otimes \tilde{M}_1 \otimes \tilde{M}_1, \tag{4.11}$$

$$(M_{1,2})_{h,l,m} = ((\tilde{M}_2)_{h,l} (\tilde{M}_1)_m + (\tilde{M}_2)_{l,m} (\tilde{M}_1)_h + (\tilde{M}_2)_{m,h} (\tilde{M}_1)_l) \tag{4.12}$$

Then

$$E[\tilde{M}_2^{\alpha}] = \sum_{k=1}^{K} \frac{\alpha_k}{(\alpha_0 + 1)\alpha_0} \boldsymbol{\mu}_k \otimes \boldsymbol{\mu}_k = M_2^{\alpha}, \tag{4.13}$$

$$E[\tilde{M}_3^{\alpha}] = \sum_{k=1}^{K} \frac{2\alpha_k}{(\alpha_0 + 2)(\alpha_0 + 1)\alpha_0} \boldsymbol{\mu}_k \otimes \boldsymbol{\mu}_k \otimes \boldsymbol{\mu}_k = M_3^{\alpha}. \tag{4.14}$$

Recall that N_i indicates the number of words in ith document, M indicates the total number of documents. $(X^{(i)})_h$ is hth entry of D-dimensional vector in which the observed counts of hth word of the dictionary in ith document are stored. In other

words, $X^{(i)}$ is regarded as the sum of observed variables realized as unit-basis vectors corresponding to all words in a document; recall examples shown in Equation (2.3). For example, if the input data are three documents shown in Figure 1.1, the observed count data $X^{(i)}$ are given as follows:

$$X^{(1)} = [5, 3, 2, 0, 0, 0], \quad X^{(2)} = [4, 5, 2, 1, 4, 0], \quad X^{(3)} = [0, 3, 0, 2, 5, 0]. \tag{4.15}$$

Note that we presume the vector of words of the dictionary is specified in the following way: [“Money”,“Bank”,“Loan”,“Water”,“River”,“Cat”].

The above theorem implies that the exact first order cross moment M_1 can be estimated from the weighted sample average of realizations of $\boldsymbol{x}$ of one word. Using such $\tilde{M}_1$ and the $\tilde{\mathbb{E}}(\boldsymbol{x} \otimes \boldsymbol{x}) = \tilde{M}_2$ that is the weighted sample average of realizations of word pairs $\boldsymbol{x} \otimes \boldsymbol{x}$, which can be regarded as the estimation of two-way contingency table for co-occurrence of 2 words, the symmetrized cross moment M_2^{α} can be estimated.

Here, α_0, which has an influence on the concentration of Dirichlet distribution, needs not be estimated to get $\tilde{M}_2^{\alpha}$ since it is assumed to be known for this method [5]. This assumption would be more reasonable in the situation in which you have little ideas about the topic mixture proportion of each document in your data. The reason for this is that you only have to specify $\sum_{k=1}^{K} \alpha_k$, which would be much easier than specifying all α_i in the sense that Dirichlet distribution is sensitive to the setting of each α as shown in Figure 2.2 and thus more attention needs to be paid. Furthermore, considering the fact that the relatively large number of topics, e.g. $K \approx 100$, is sometimes adopted in LDA in practice, it would be very difficult to make a reasonable choice of all α_k setting; thus, specifying only sum of α_i could mitigate such a difficulty. Hence, this characteristics would one of the advantages of this parameter recovery method.

In terms of M_3^{α}, $\tilde{M}_3^{\alpha}$ would be attained from $\tilde{M}_1$ and $\tilde{\mathbb{E}}(\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x}) = \tilde{M}_3$ which is the sample average of realizations of word triplets $\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x}$. This is the estimate of the three-way contingency table for co-occurrence of three words.

Noting that $M_{1,2}$ is symmetric from a reasoning similar to Equation (3.20), $\mathbb{E}[\tilde{M}_3^{\alpha}]$ is indeed a symmetric tensor, and thus its CP decomposition and symmetric CP decomposition (Equation (4.14)) are unique with probability 1 based on the study of Chiantini et al [14], as discussed in Section 3.5. This means that, based on Theorem 1.1 of [14], the topic-specific distribution of words, i.e. the conditional distribution of words given each topic, $[\boldsymbol{\mu}_1 | \boldsymbol{\mu}_2 | \cdots | \boldsymbol{\mu}_K]$ can be uniquely recovered with probability one, provided that

$$K < \frac{\binom{D+2}{3}}{D}, \quad \text{where, } (D, K) \neq (6, 9). \tag{4.16}$$

Here, K is the number of topics and D the number of vocabulary of the dictionary.

Again, we emphasize that these parameters can be recovered upto the permutation and scaling effect; namely, the output of the decomposition given in the eq.4.14 takes the the following form in practice: $E[\tilde{M}_3^{\alpha}] = \sum_{k=1}^{K} \lambda_k \cdot \boldsymbol{\mu}_k^{\mathrm{row}} \otimes \boldsymbol{\mu}_k^{\mathrm{row}} \otimes \boldsymbol{\mu}_k^{\mathrm{row}}$, where $\boldsymbol{\mu}_k^{\mathrm{row}}$ is an unnormalized $\boldsymbol{\mu}_k$ and λ_k is the corresponding coefficient. Nevertheless, this

limitation with respect permutation and scaling has no adverse effect on recovering parameter $\boldsymbol{\mu}_k, \alpha_k$ of LDA model. One reason is that $\boldsymbol{\mu}_k$ has a constraint of $||\boldsymbol{\mu}_k||_1 = 1$, with which the scaling effect on each $\boldsymbol{\mu}_k$ is able to be fixed as well as α_k. The other reason is that our goal is just extracting each column $\boldsymbol{\mu}_k$, which is still the same even after columns of $[\boldsymbol{\mu}_1|\cdots|\boldsymbol{\mu}_K]$ are permuted. With these reasons, even though limitation regarding scaling and permutation exist for this parameter recovery method for LDA, it is still valid as long as those two effects are correctly dealt with. However, if we assume a situation where a set of true topic-words distribution vectors $\{\boldsymbol{\mu}\}_{k=1}^K$ are not known but their labels are known, the tensor method cannot assign each of $\{\hat{\boldsymbol{\mu}}\}_{k=1}^K$ to corresponding labels indicated by index k. This is because original $\boldsymbol{\mu}_k$ and extracted row $\hat{\boldsymbol{\mu}}_k$ are no longer guaranteed to belong to the same label because of permutation effect. Note that the above situation would rarely occur in practice since LDA is an unsupervised learning and topic labels are generally given by the users based on the words with higher probability of each $\{\hat{\boldsymbol{\mu}}\}_{k=1}^K$.

4.3 The proposed algorithm

In this section, we briefly explain our tensor method algorithm based the symmetric CP decomposition. It is fundamentally based on the theorems introduced earlier [5, 48]. The basic flow of our algorithm is that a matrix of counts for each word in the dictionary per document W $(M \times D)$ is firstly created from input data to compute the $\tilde{M}_1, \tilde{M}_2, \tilde{M}_3$ in Theorem 4. Using these empirical moments, $\tilde{M}_3$ is secondly symmetrized, to which eventually symmetric CP decomposition is applied to recover the parameter $[\boldsymbol{\mu}_1 \cdots \boldsymbol{\mu}_K]$ and $\boldsymbol{\alpha}$. Although this operation may seem to be unnecessary since $\mathbb{E}(\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x})$ looks symmetric even before this operation by its construction, this operation is actually required to be done. As Anandkumar points out (Section 3.2 of [3]), this is because the probabilities of latent topics to be drawn for each word are not mutually independent because they are generated from Dirichlet distribution, leading the original $\mathbb{E}(\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x})$ to be slightly unsymmetric and thus the specific operation needs to made to symmetrize $\mathbb{E}(\boldsymbol{x} \otimes \boldsymbol{x} \otimes \boldsymbol{x})$ as shown in Equation (4.11).

The main flow of our algorithm is as follows. To begin with, creation of input data of a matrix recorded with the unit-basis system is done. Computing diagonal elements of lower-order moments given in Theorem 4 is then performed, followed by the calculation of completely non diagonal and semi-diagonal elements of those moments. The detailed flow of our proposed algorithm is shown in Figure 4.1. Let us explain our algorithm in a little bit more.

The line (1) feeds input based on the assumption of our method and LDA model; namely, the number of topics K and sum of parameters of Dirichlet prior for topic distribution α_0 are known. ith row of X is corresponding to ith observed word among all words in a corpus recorded as a unit row vector of the size D similar to Equation (2.3). The line (2) computes the quantity of Equation (4.4). The (3) creates the quantity explained in Equation (4.15) with which all of following computations are proceeded. The line (4) calculates the quantities appearing in the

denominators of Equation (4.4) to Equation (4.9). After this, diagonal elements of $\tilde{M}_2$ defined in Equation (4.6) and those of $\tilde{M}_3$ defined in Equation (4.9) are computed using W. Subsequently, the lines (7)-(9) computes the quantities of nominators in Equations (4.5) and (4.8) respectively. The line (10) computes the quantity shown in Equation (4.5) and lines (11)-(12) calculates the quantities expressed in Equation (4.8). The line (15) computes quantity of nominator shown in Equation (4.7) and the line (16) computes the quantity provided by Equation (4.7). This procedure is followed by the lines (17)-(21) which compute all the quantities given in Equations (4.10) to (4.12). Finally, the line (22)-(24) compute the $\boldsymbol{\mu}_k, \alpha_k (k = 1, \cdots, K)$ given in the left side of Equation (4.14). In this way, our algorithm completes the task of recovering $\boldsymbol{\mu}_k, \alpha_k (k = 1, \cdots, K)$.

Regarding the algorithm for a symmetric tensor decomposition itself, structured data fusion by nonlinear least squares [53] (sdf_nls) explained in Section 3.3.3 is utilized. This is because, as mentioned before, this algorithm enables us to impose structural constrains on decomposition. In fact, the symmetric constrain that is equality of three factor matrices for 3-way CP decomposition and non-negativity constrain on each element of such factor matrix are imposed.

Our method differs from the ones employed in [49] and [4] or proposed in [5] especially in the sense that our method does not include orthogonalization process because it is not theoretically needed. Such avoidance of orthogonalization process brings about an advantage to our proposed method compared with those presented in [49] and [4].

More specifically, in our proposed method, the larger number of topics can be assumed when the dictionary size is larger than 2, which is almost always true in practical application. Let us explain in details. The preceding methods are based on the orthogonalization; namely, modifying $\tilde{M}_3$ to be expressed as orthogonal factor matrix of the size $D \times K$. This means that the maximum number of topics K_{max} that can be accommodated with these method is restricted by the dictionary size:

$$K_{max} \leq D. \tag{4.17}$$

On the other hand, our proposed method is not dependent on orthogonalization and the factor matrix $[\boldsymbol{\mu}_1 | \boldsymbol{\mu}_2 | \cdots | \boldsymbol{\mu}_K]$ needs not to be orthogonal. Hence, the number of topics K can be freely selected as long as K, D satisfies Equation (4.16), which can be written down in a little bit more details:

$$K_{max} < \frac{(D+2)(D+1)}{6}, \quad (D, K) \neq (6, 9). \tag{4.18}$$

It is found that the maximum number of topics being able to be applied to LDA model is larger for our proposed method than for preceding methods when the number of words in the dictionary is larger than 2 by comparing Equations (4.17) and (4.18) as follows:

$$\frac{(D+2)(D+1)}{6} - D = 0 \quad = \quad D^2 - 3D + 2 = (D-1)(D-2) = 0.$$

Therefore, our proposed method is superior than the ones presented in [49, 4] with respect to flexibility of the maximum number of topics considered in LDA.

(1) input a word index matrix X of size $N \times D$, K, $\alpha_0 = \sum_{k=1}^{K} \alpha_k$, $[N_1, \cdots, N_m]$

(2) calculate $\tilde{M}_1$ by taking the mean of each column of X

(3) create a matrix of the word-count per document $W \in \mathbb{R}^{M \times D}$

(4) calculate $\sum_{m=1}^{M} N_m$, $\sum_{m=1}^{M}(N_m - 1)N_m$, $\sum_{m=1}^{M}(N_m - 2)(N_m - 1)N_m$

(5) **for** $i = 1$ **to** $\binom{D}{2}$

(6) **for** $m = 1$ **to** M

(7) $(X^{(m)})_{h_i}(X^{(m)})_{l_i} + +$,

(8) $(X^{(m)})_{h_i}((X^{(m)})_{h_i} - 1)(X^{(m)})_{l_i} + +$

(9) $(X^{(m)})_{h_i}(X^{(m)})_{l_i}((X^{(m)})_{l_i} - 1) + +$

(10) $\tilde{M}_{2h_i,l_i} = \frac{(X^{(m)})_{h_i}(X^{(m)})_{l_i}}{\sum_{m=1}^{M}(N_m - 1)N_m}$

(11) $\tilde{M}_{3h_i,h_i,l_i} = \tilde{M}_{3h_i,l_i,h_i} = \tilde{M}_{3l_i,h_i,h_i} = \frac{(X^{(m)})_{h_i}((X^{(m)})_{h_i} - 1)(X^{(m)})_{l_i}}{\sum_{m=1}^{M}(N_m - 2)(N_m - 1)N_m}$

(12) $\tilde{M}_{3l_i,l_i,h_i} = \tilde{M}_{3l_i,h_i,l_i} = \tilde{M}_{3h_i,l_i,l_i} = \frac{(X^{(m)})_{h_i}(X^{(m)})_{l_i}((X^{(m)})_{l_i} - 1)}{\sum_{m=1}^{M}(N_m - 2)(N_m - 1)N_m}$

(13) **for** $i = 1$ **to** $\binom{D}{3}$

(14) **for** $m = 1$ **to** M

(15) $(X^{(m)})_{h_i}(X^{(m)})_{l_i}(X^{(m)})_{s_i} + +$

(16) $\tilde{M}_{3h,l,s} = \frac{(X^{(m)})_{h_i}(X^{(m)})_{l_i}(X^{(m)})_{s_i}}{\sum_{m=1}^{M}(N_m - 2)(N_m - 1)N_m}$

(17) calculate $\tilde{M}_1 \otimes \tilde{M}_1$

(18) calculate $\tilde{M}_2^{\alpha}$

(19) calculate $M_{1,2}$

(20) calculate $\tilde{M}_1 \otimes \tilde{M}_1 \otimes \tilde{M}_1$

(21) calculate $\tilde{M}_3^{\alpha}$

(22) perform the symmetric CP-decomposition of $\tilde{M}_3^{\alpha}$

(23) normalize the extracted raw $\boldsymbol{\mu}_k (1 \leq k \leq K)$

(24) calculate α_k by $\lambda_k \cdot \frac{(\alpha_0 * (\alpha_0 + 1) * (\alpha_0 + 2))}{2} \cdot c_k^3$, where $c_1 \cdots c_K$ are normalization constants for $\boldsymbol{\mu}_1 \cdots \boldsymbol{\mu}_K$.

Figure 4.1: Overview of our symmetric CP decomposition based parameter estimation algorithm for LDA.

h_i, l_i: ith word-pair indexies among all possible word-pair combinations. h_i, l_i, s_i: ith word triplets' indexies among all possible combinations of them.

Chapter 5

Experiments

In this section, we provide our hyperthesis and corresponding experimental method and result.

5.1 Problem statement

Before stating our hypothesis, method, and result, we again restate our problem statement in this thesis in the following:

- Can we estimate the topic-word distribution $[\boldsymbol{\mu}_1|\boldsymbol{\mu}_2|\cdots|\boldsymbol{\mu}_K]$ of an LDA model with a symmetric CP decomposition?

The main reason why we are interested in investigating this hypothesis is that there are very limited experiments in the theoretical papers promoting these tensor algorithms. Thus, the empirical investigation of these algorithms will be first to make a decent comparison.

To investigate the above problem, we have made several hypothesis and corresponding research methods, which will be introduced in the following section.

5.2 Hypothesis and our method

Hypothesis 1. *The parameter estimation method based on the symmetric CP decomposition method theoretically works. Thus the error of our method would be as small as the one from collapsed Gibbs sampling method.*

Note that we measure error as the relative error between recovered $[\hat{\boldsymbol{\mu}}_1, \cdots, \hat{\boldsymbol{\mu}}_K] = \{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}$ and true $[\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K] = \{\boldsymbol{\mu}\}_{k=1}^{k=K}$ of LDA model for synthetic data in this section. The relative error is calculated in the following way:

$$\sqrt{3 \cdot ||\{\boldsymbol{\mu}\}_{k=1}^{k=K} - \{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K} * (\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K} \backslash \{\boldsymbol{\mu}\}_{k=1}^{k=K})||_F^2}, \tag{5.1}$$

where $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K} \backslash \{\boldsymbol{\mu}\}_{k=1}^{k=K}$ indicates the solution to the equation $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K} X = \{\boldsymbol{\mu}\}_{k=1}^{k=K}$ and $||\cdot||_F$ means frobenius norm.

Since Hypothesis 1 may be too general, we firstly aim to verify a more concrete hypothesis as follows.

Hypothesis 2. *When synthetic data is generated with an orthogonality constrain imposed on* $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ *where* $K \geq 2$*, our symmetric CP decomposition based approach should correctly recover* $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ *up to permutation and scaling. Thus, the error between* $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ *and* $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}$ *after consideration of permutation is expected to be at least as small as the one from collapsed Gibbs sampling.*

Note that the Hypothesis 2 is based on the Theorem 2. More specifically, the CP decomposition of $\tilde{M}_3^{\alpha}$ calculated from the data which are generated from mutually orthogonal $\boldsymbol{\mu}_k$ is unique since the Kruskal rank of corresponding factor matrix$\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ is K and thus Theorem 2 is fulfilled with

$$K + K + K \geq 2K + 2, \quad \text{namely,} \quad K \geq 2.$$

Hypothesis 3. *The condition number of the tensor corresponding to the computed parameters is a good indicator of the true relative error.*

The above hypothesis is stemmed from Equation (3.30).

5.3 Experimental environment

All of experiments are done using the NALAG server at the computer science department of KU Leuven. The computer system contains two Intel Xeon E5-2697 v3 CPUs (14 cores, each 2.6GHz), and 128GB of system memory. Synthetic data are generated with Python 3.6.4 and the parameter is recovered with MATLAB 9.3. To implement our algorithm based on tensor operation, we used the MATLAB package `Tensorlab` [45], which contains a lot of function tools for tensor operations.

‘

5.4 Experiment procedure

To study Hypothesis 2, we first generated synthetic data from an LDA model. Each dataset was generated from LDA models with different number of topics K and corresponding Dirichlet prior $\boldsymbol{\alpha}$ to additionally observe if our method is robust to the assumption of K of the LDA models. More specifically, K was changed between 3 and 10 as Dirichlet priors $Dir(\boldsymbol{\alpha})$ for topic distributions and $Dir(\boldsymbol{\beta})$ for topic-word distributions, symmetric priors: $\boldsymbol{\alpha} = [\frac{1}{K}_1, \cdots, \frac{1}{K}_K]$, $\boldsymbol{\beta} = [\frac{1}{D}_1, \cdots, \frac{1}{D}_D]$ were selected. We modified the Python code shown in [29] to generate the synthetic data.

After synthetic data generation, our parameter estimation method based on the symmetric CP decomposition was applied to each dataset. Using recovered $[\hat{\boldsymbol{\mu}}_1, \cdots, \hat{\boldsymbol{\mu}}_K]$, we calculated relative errors for evaluation of our hypothesis as well as measuring computational times and computing condition numbers to evaluate our method.

Finally, parameter estimation was done for each dataset with collapsed Gibbs sampling (Section 2.6.2.) This is because the result from the collapsed Gibbs sampling approach is employed as benchmark, allowing us to evaluate our tensor method approach. Specifically, relative error, geometric condition number, and computational time were calculated and compared with the ones of our tensor approach. For the implementation of the collapsed Gibbs sampling, the MATLAB function `LDA_Gibbs` was used [31].

To strengthen the reliability of the measured computational time and computed relative error and geometric condition number, five different random trials were generated for as certain parameter setting. Hence, for each parameter setup, mean and median of condition number, relative error, and computational time are calculated when the tensor method is utilized.

On the other hand, the collapsed Gibbs sampling was applied to only one set of synthetic data generated from each parameter setting because it turned out that it took a long computational time and several trials of running showed seemingly stable results. Therefore, neither mean nor median values are not reported.

5.5 Data generation

We generated multiple data from the LDA model with a set of symmetric priors for $\boldsymbol{\alpha}, \boldsymbol{\beta}$. Several different trials for the number of topics, the dictionary size, and orthogonality of $[\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K]$ were generated to investigate their influence on the parameter recovery via the tensor method. The orthogonality constraint was imposed by specifying a range of the smallest singular value for a matrix $[\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K]$. Here, we again stress that $\boldsymbol{\mu}_k$s are stochastic vectors.

To generate linearly independent $\boldsymbol{\mu}_k$s, the accept-reject method was used. More specifically, as explained in Section 2.5, each $\boldsymbol{\mu}_k$ was sampled from $Dir(\boldsymbol{\beta}), \boldsymbol{\beta} \in \mathbb{R}^D$ and put into a matrix to form $[\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K]$. If the smallest singular value σ of the sampled $[\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K]$ was less than 0.01, we regarded it to be far less orthogonal and sampled a new $[\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K]$ again. In contrast, if $1.1 < \sigma < 0.01$ was satisfied, $[\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K]$ was stored. The above range of σ seemed to be too wide, it was further divided into following five cases of narrower ranges: $1.1 < \sigma < 0.7, 0.7 < \sigma < 0.5, 0.5 < \sigma < 0.1, 0.1 < \sigma < 0.05, 0.05 < \sigma < 0.01$; then, the remaining data generating process was completed.

This orthogonality on $[\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K]$ is important since it has a significant influence on the condition number. Specifically, Breiding and Vannieuwenhoven proved that the condition numbers of completely orthogonal CP decomposable tensors are 1; see Proposition 5.1 of [10]. This means that the completely orthogonal CP decomposition is a well-conditioned problem. Therefore, application of our symmetric CP decomposition approach to synthetic data generated with $[\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K]$ constrained to be orthogonal is expected to produce a correct solution even if input data are slightly contaminated for some reasons such as a finite sample size from which M_3^{α} is estimated.

5.6 Results: synthetic data

For readability, the following abbreviations are used in this section:

- TOP: the number of topics K.
- VOC: the number of vocabularies in the dictionary D or the number of distinct words in a corpus.
- DN: the number of documents.
- NT: the number of terms in each document.
- σ: the smallest singular value of $[\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_K]$, which determines its orthogonality.
- TolX: the lower bound of the step size in Tensorlab.
- TolFun: the lower bound of the decrease of the objective function in Tensorlab.
- MaxIte: the maximum iterations in Tensorlab.
- ITE: the number of iterations for collapsed Gibbs sampling.
- BI: the number of burn-in iterations for collapsed Gibbs sampling.
- $\mathrm{E}_{\mathrm{rel}}$: the relative error between recovered (estimated) $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}$ calculated as in Equation (5.1).
- $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$: empirical geometric condition number computed from $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}$.
- Time: computational time for each algorithm (sec).
- $\kappa_{\{\boldsymbol{\mu}\}_{k=1}^{k=K}}$: theoretical geometric condition number computed from $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$.
- $\mathrm{BE}_{\mathrm{rel}}$: relative backward error between the input tensor $\tilde{M}_3^{\alpha}$ and the reconstructed tensor from $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}$.
- GEVD initialization: generalized eigenvalue decomposition based initialization.

5.6.1 Effect of the number of topics

Method

To empirically verify Hypothesis 2, we first examined if our approach based on symmetric CP decomposition works over different number of topics by comparing it to the collapsed Gibbs sampling approach. To investigate this, we generated multiple datasets with the following setting: VOC=20, DN=1000, NT=100, MaxIte=300, $1.1 > \sigma > 0.7$.

Especially, the constraint of $1.1 > \sigma > 0.7$ was used with the expectation of imposing the sufficient orthogonality constraint on $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$. This constraint was

used to allow us to observe the effect of the number of topics without considering perturbation effect. As discussed in Section 5.5, with this restriction, it is anticipated that $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}} \approx 1$, suggesting that the solution obtained from each trial is robust to perturbation and thus reliable. If we had not utilized this condition, it would have been difficult to look at only the effect of topics because of the lack of guarantee that $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}} \approx 1$. Note that even a narrower range for σ around 1 was tried, for example $1.05 > \sigma > 0.9$, but it turned out that it took extremely long to generate such data. Furthermore, to ensure $K \geq 2$, TOP was changed between 3 and 10.

Regarding the parameter setting of `sdf_nls`, the default following setting is adopted: TolFun = TolX = 10^{-14}, MaxIte = 300, and the initialization method based on a generalized eigenvalue decomposition.

Results

The Tables 5.1 and 5.2 show the experimental results for respectively the tensor-based method and the collapsed Gibbs sampling approach. The tables suggest that our method does not seem to work at all. As a matter of fact, if we focus on E_{rel} of our algorithm, the orders of magnitudes of median and mean of E_{rel} for different number of topics are almost 10^1. This is much larger than the orders of magnitudes of E_{rel} for collapsed Gibbs sampling's 10^{-2}. This result does not support our Hypothesis 2.

Table 5.1: Result from the tensor decomposition approach with GEVD initialization for Hypothesis 2

	MEAN			MEDIAN		
TOP	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME
3	0.9712	1.2735	0.8791	1.2805	1.1949	0.6774
4	0.8114	1.9794	0.4623	0.0167	1.3529	0.4343
5	1.0801	1.6663	0.6117	1.2719	1.4628	0.5882
6	1.7295	36.1209	0.4815	1.8248	1.4886	0.4593
7	1.6828	13.2919	0.5556	1.6680	3.2055	0.4347
8	1.6338	2.5565	0.4148	1.7682	1.9112	0.3586
9	1.8274	3.1886	1.7801	1.8331	2.8969	1.4552
10	1.6910	13.6184	2.0236	1.7029	6.7570	1.7490

Setting of data generation: (VOC=20, DN=1000, NT=100, $1.1 < \sigma < 0.7$.)
Setting of `sdf_nls`: (MaxIte=300, TolX=10^{-14}, TolFun=10^{-14}.)

Table 5.2: Collapsed Gibbs sampling approach.

	ITE=300, BI=150			ITE=200, BI=100		
TOP	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME
3	0.0171	1.1293	908.7232	0.0174	1.1373	332.4693
4	0.0100	1.0411	713.5551	0.0094	1.0406	322.8017
5	0.0158	1.2922	779.3240	0.0174	1.2857	338.8986
6	0.0216	1.2394	746.4717	0.0214	1.2941	324.0987
7	0.0179	1.2275	522.8715	0.0199	1.3163	532.0985
8	0.0261	1.1257	554.9420	0.0254	2.0639	561.8462
9	0.0274	1.4411	546.5860	0.0312	1.3517	569.4557
10	0.0346	1.2093	904.2192	0.0364	1.2865	403.8918

Setting of data generation: (VOC=20, DN=1000, NT=100, $1.1 < \sigma < 0.7$.)
Setting of `sdf_nls`: (MaxIte=300, TolX=10^{-14}, TolFun=10^{-14}.)

To further investigate why the parameter is not well recovered by our method, we newly made following hypothesis:

Hypothesis 4. *The generalized eigenvalue decomposition based initialization method may have an adverse effect on correct parameter recovery but the psudorandom initialization method may not.*

Hypothesis 5. *The setting (MaxIte, TolX, TolFun) = (300,* 10^{-14}, 10^{-14}*) may not be appropriate to obtain a plausibly correct optimal solution.*

Effect of the initialization

To study Hypothesis 4, we re-ran our algorithm switching the initialization of `sdf_nls` from the generalized eigenvalue decomposition based initialization method to pseudorandom one. The results do not support Hypothesis 4 (Table 5.3.) In fact, if we look at E_{rel}, the means and medians of E_{rel} across different TOP obtained with GEVD initialization are rather larger than those obtained with pseudorandom initialization. Hence, it is concluded that Hypothesis 4 is not valid.

Table 5.3: Result from the tensor decomposition approach with pseudorandom initialization for Hypothesis 4

	MEAN			MEDIAN		
TOP	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME
3	1.2309	1.3632	0.4483	1.2757	1.1729	0.3309
4	1.9609	2.1870	0.3199	1.8143	1.6791	0.3096
5	1.6943	11.9027	0.3244	1.5216	1.8279	0.3133
6	1.7177	4.1642	0.3494	1.9036	2.0709	0.3376
7	1.9547	2.1793	0.3370	1.8985	1.6822	0.3479
8	2.1581	8.9509	0.3683	1.9783	2.8359	0.3315
9	1.9552	13.2887	0.4038	2.1210	10.8861	0.3870
10	2.4731	4.4859	0.3340	2.2824	5.3248	0.3582

Setting of data generation: (VOC=20, DN=1000, NT=100, $1.1 < \sigma < 0.7$.)
Setting of `sdf_nls`: (MaxIte=300, TolX=10^{-14}, TolFun=10^{-14}.)

Effect of the stopping criteria of `sdf_nls`

Nextly, Hypothesis 5 was examined thorough modifying the default stopping rules of `sdf_nls` from (TolX, TolFun, MaxIte) = $(10^{-14}, 10^{-14}, 300)$ to $(10^{-24}, 10^{-24}, 5000)$. The results appear to improve much. For example, the order of median values decrease to 10^{-2} except for the cases TOP = 6 and 7 cases. This order is much smaller than the order 10^{0} for medians of E_{rel}s in the previous setting (Table 5.1) if we ignore TOP = 4 case. Thus, this result seems to support our Hypothesis 5.

Table 5.4: Result from the tensor decomposition approach with GEVD initialization for Hypothesis 5

	MEAN			MEDIAN		
TOP	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME
3	0.3098	1.1625	63.5344	0.0095	1.1217	88.2149
4	0.2627	1.2542	83.4419	0.0114	1.2267	94.5270
5	0.5343	1.3839	99.3627	0.0199	1.3659	99.0165
6	0.7667	1.3851	73.0373	1.2275	1.3477	88.8552
7	0.8098	1.7510	66.3099	1.3222	1.5679	75.4209
8	0.5183	2.0270	73.5369	0.0434	1.8155	75.8972
9	0.3290	1.8188	66.2585	0.0439	1.8268	76.9883
10	0.2961	1.8426	56.0777	0.0481	1.6681	70.7411

Setting of data generation: (VOC=20, DN=1000, NT=100, $1.1 < \sigma < 0.7$.)
Setting of `sdf_nls`: (MaxIte=5000, TolX=10^{-24}, TolFun=10^{-24}.)

To additionally study an effect of the stopping criteria of `sdf_nls`, the convergence plots for the *Table* 5.4 are shown in Figures 5.1 and 5.2. As seen in the left-side figures in Figures 5.1 and 5.2, fluctuations in refval=TolFun are observed roughly below 300 iterations excepting for top right case in Figure 5.2. The range of these fluctuations are so large as to once exceed around 10^{-14}, which is the stopping criteria of TolFun for the default setting, and to again return into the area larger than 10^{-14}. This implies that the setting of TolFun being at around 10^{-14} could result in finishing iterations at early stage because the TolFun criterion is met, even though the corresponding optimal solution is not stable nor possible global minima. For example, the Figure 5.2 is a typical example, where the iteration would have immediately ended up and probably yielded a bad solution if TolFun=10^{-14} had been selected. This is because a larger decrease in an objective function occurred later once such decrease had reached 10^{-14}. Hence, the result shown in Table 5.1 is worse than the one shown in Table 5.4, which again agrees with our Hypothesis 5.

Top left: TOP=10, VOC=20, $1.1 > \sigma > 0.7$,
Top right: TOP=10, VOC=20, $0.05 > \sigma > 0.01$,
Bottom left: TOP=10, VOC=40 $1.1 > \sigma > 0.7$,
Bottom right: TOP=10, VOC=40, $0.05 > \sigma > 0.01$

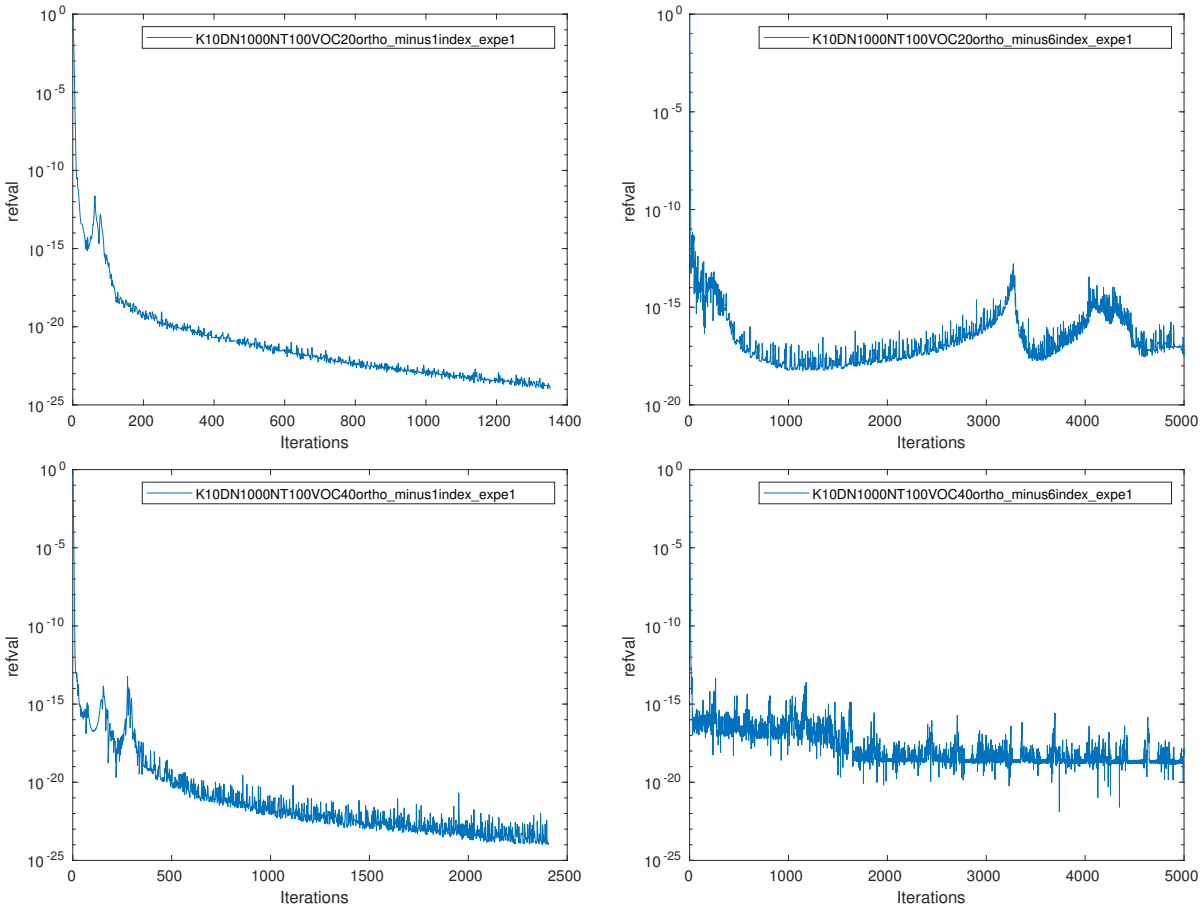

Figure 5.1: Convergence plot

It is also observed that refval does not monotonically decrease. This would be related to our objective function; more specifically, the objective function is non-convex and thus it is expected that the function has many local optima and saddle points [56]. Hence, although the decrease of objective function becomes gradually small as a solution becomes close to a local optima or saddle point per iteration, such decrease can increase once our algorithm escapes from the region around that local optima or saddle point and starts searching another solution. Considering non-convexity property of the objective function described above, it would be recommendable for our tensor method to be executed multiple times and see if solutions are stable.

Top left: TOP=3, VOC=20, $1.1 > \sigma > 0.7$,
Top right: TOP=3, VOC=20, $0.05 > \sigma > 0.01$,
Bottom left: TOP=3, VOC=40, $1.1 > \sigma > 0.7$,
Bottom right: TOP=3, VOC=40, $0.05 > \sigma > 0.01$

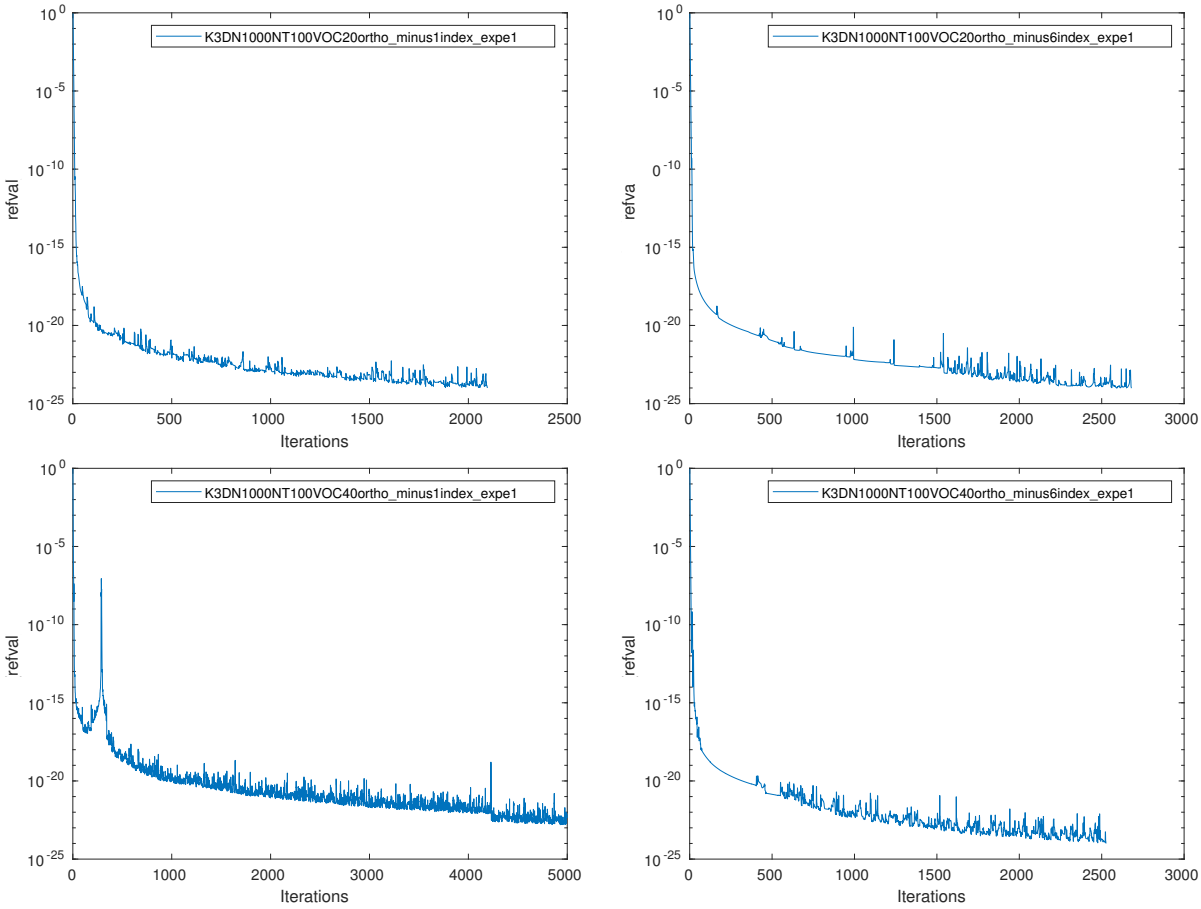

Figure 5.2: Convergence plots.

Since Hypothesis 5 seems to be verified, we further investigate our main Hypothesis 2. First and foremost, although it would be difficult to say that the results from our method perform better than the one from collapsed Gibbs sampling approach, it may be claimed that our tensor method could work as comparable as collapsed Gibbs

sampling method with a few exceptions (TOP=6, 7.) However, this statement would hold only when our method is repeatedly used to recover the parameter $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ for the same dataset and provide 'frequent' solution $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}$. This is because the means of E_{rel} for our method are still worse than E_{rel} of collapsed Gibbs sampling, while the medians for our method are mostly similar to E_{rel} of the other method. Thus, we conclude that Hypothesis 2 is partially verified in the sense that half of the applications of our tensor based algorithm to a set of synthetic datasets generated from the same model would lead to finding $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}$ as correctly as an application of collapsed Gibbs sampling does except for permutation effect.

The discussion above leads to the second important observation that our tensor method is less stable than collapsed Gibbs sampling. Indeed, by observing Table 5.1, it is also worthwhile noting that medians of E_{rel} are smaller than corresponding means such that there is a roughly at least the order 10^{-1} differences between them except for TOP=6, 7 cases. This would mean the instability of our tensor method, which is not observed for the collapsed Gibbs sampling method even when ITE is reduced to 200. (Table 5.2.) The question is how such another CP decomposition can be possible given the fact that our synthetic data are generated with a specific constrain such that corresponding $\tilde{M}_3^{\alpha}$ is orthogonal and thus only one unique CP decomposition exist.

Thirdly, it is found that the geometric condition numbers $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$ for our method are slightly larger than those for collapsed Gibbs sampling excepting for the case (ITE=200, BI=100, TOP=8). In particular, as TOP increases, both means and medians of $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$ for our method appear to increase slightly more than $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$ for collapsed Gibbs sampling do. Since $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$ gives information on sensitivity to perturbation (Section 3.6), our method could be less robust to the contamination of an input data such as noise than the collapsed Gibbs sampling, especially when the number of topic is large.

Finally, it is noticed that the computational times TIME for our algorithm is shorter than the ones for collapsed Gibbs sampling, which suggests that our approach would be more efficient with respect to computational time. For example, the longest mean of TIME of our algorithm, 99.3627 (TOP=5), is less than 25% of the shortest TIME of collapsed Gibbs sampling approach 322.8017 (ITE=200, BI=100, TOP=4). It is important to keep in mind, however, that a solution obtained from our efficient tensor approach is not empirically guaranteed to always be correct as discussed before. Furthermore, an interesting observation regarding TIME over different TOP for our tensor method is that the median of TIME somewhat appears to slightly decrease as TOP increases.

5.6.2 Effect of the dictionary size

Method

In the Section 5.6.1, we partially verify Hypothesis 2. The natural question would be if our tensor method recovers $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ as well as collapsed Gibbs sampling when the synthetic data are generated from a model with a different set of parameters.

Especially, it would be beneficial to modify number of words in the dictionary VOC since it would be unlikely for documents to be composed from only 20 distinctive words in practice; the larger the VOC is, the more likely the synthetic data would represent an example of real document data. To investigate such an effect of VOC, we made a new hypothesis as follows.

Hypothesis 6. *When synthetic data are generated from a LDA model whose* $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ *is a orthogonal matrix and* $TOP = K \geq 2$*, our symmetric CP decomposition based approach should properly recover* $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ *except for limitation of permutation regardless the value of parameter VOC of the LDA model. Hence, it is anticipated that the difference between* $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ *and* $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}$ *taking permutation effect into account is as small as the one obtained from the result of collapsed Gibbs sampling.*

To study Hypothesis 6, we re-generated various synthetic data with VOC ranging from 30 to 70, TOP=5, DN=1000, NT=100, $1.1 < \sigma < 0.7$. Note that some much larger VOC were tried but it turned out that it took too long to calculate $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$ and thus VOC=70 was taken as maximum. After that, our tensor-based method and collapsed Gibbs sampling method were employed to respectively recover and estimate $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$. Considering the result and discussion of Section 5.6.1, the setting (TolX, TolFun, MaxIte)=$(10^{-24}, 10^{-24}, 5000)$ with GEVD initialization was utilized for `sdf_nls` in our method.

Results

Table 5.5: Result from the tensor decomposition approach with GEVD initialization for Hypothesis 6

	MEAN			MEDIAN		
VOC	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME
20	0.5343	1.3839	99.3627	0.0199	1.3659	99.0165
30	1.2681	1.6606	47.3901	1.2580	1.2767	62.2610
40	1.0952	1.2920	74.1439	1.3109	1.2613	87.5978
50	0.5558	1.2794	100.2817	0.0320	1.2534	103.9764
60	0.7879	1.2832	204.5827	1.2383	1.2837	234.8701
70	0.5780	1.3173	253.1754	0.0264	1.2669	243.9350

Setting of data generation: (TOP=5, DN=1000, NT=100, $1.1 < \sigma < 0.7$.)
Setting of `sdf_nls`: (MaxIte=5000, TolX=10^{-24}, TolFun=10^{-24}.)

The results from our method and collapsed Gibbs sampling are displayed in Tables 5.5 and 5.6. Regarding E_{rel}, the values of the collapsed Gibbs sampling approach are much better than the means of those for our tensor based approach. On the other hand, the medians of E_{rel} for our method look moderately close to E_{rel} of the

collapsed Gibbs sampling with ITE=200 and BI=100 in the cases of VOC = 20, 50, and 70. This again indicates the empirical unreliability of our tensor method as already observed in Table 5.4. Since only 3 cases among 6 different VOC setting end in seemingly succeeding in recovering $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$, it would not be enough to verify our Hypothesis 6.

Table 5.6: Result from the collapsed Gibbs sampling approach for Hypothesis 6

	ITE=300, BI=150			ITE=200, BI=100		
VOC	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME
20	0.0158	1.2922	779.3240	0.0174	1.2857	338.8986
30	0.0120	1.0428	728.7444	0.0122	1.0497	435.8451
40	0.0124	1.0803	464.1408	0.0129	1.0737	373.8288
50	0.0174	1.1553	471.3231	0.0175	1.1541	271.4311
60	0.0127	1.0470	457.9043	0.0135	1.0514	400.3102
70	0.0109	1.0712	459.2095	0.0118	1.0625	403.2660
80	0.0112	1.0491	451.5863			
90	0.0250	1.1405	446.0719			
100	0.0102	1.0489	457.8623			

Setting of data generation: (TOP=5, DN=1000, NT=100, $1.1 < \sigma < 0.7$.)
Setting of `sdf_nls`: (MaxIte=5000, TolX=10^{-24}, TolFun=10^{-24}.)

In terms of $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$, the figures for collapsed Gibbs sampling procedure are overall much closer to 1 than the means and medians of those for our tensor method, implying that our method would be more sensitive to a small perturbation effect in an input than collapsed Gibbs sampling. In contrast to, the observed systematic shift in $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$ of the collapsed Gibbs sampling procedure for increasing TOP (Table 5.4), no systematic trends in $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$ seem to exist as VOC rises.

The means and medians of TIME for our tensor based method are again much shorter than the collapsed Gibbs sampling's TIME even with the setting ITE=200. In particularly, it requires at least twice as long computational time as our tensor method. However, the mean and median TIME jump from around 100 to 230 between VOC=50 and VOC=60 setting. Such jumps in TIME might repeatedly happen as VOC is further increased, possibly resulting in a long TIME.

5.6.3 Effect of orthogonality of topic-words matrix

Method

In the previous section, the validity of Hypothesis 6 which is an extention of Hypothesis 2 is investigated. The interesting question is how important the orthogonality constraint on $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ is for a correct parameter recovery. As mentioned in the end of Section 4.2, based on the discovery of Chiantini, Ottaviani, and Vannieuwenhoven [14],

the symmetric CP decomposition of in our tensor based method (Equation (4.14)) is unique when Equation (4.16) is satisfied, which suggests that our method would correctly recover $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ from the data generated from a model where no orthogonal constrain is imposed on $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$. To quantitatively study the above supposition, we made another new hypothesis focusing on that the orthogonality of $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ has a closely related to its smallest singular value σ:

Hypothesis 7. *For synthetic data generated from a LDA model, our symmetric CP decomposition based method would correctly recover* $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ *regardless of the magnitude of its* σ *value, given that the prespecified* K, D *of the LDA model satisfy Equation* (4.16).

To verify Hypothesis 7, synthetic data were generated from LDA models whose σ were respectively limited in different range and whose (TOP, VOC, DN, NT) are $(5, 20, 1000, 100)$, ensuring that Equation (4.16) is satisfied: $5 < \frac{1540}{20} = 77$. For these datasets, our method and collapsed Gibbs sampling approach were applied.

Results

Table 5.7: Result from the tensor decomposition approach with GEVD initialization for Hypothesis 7

	MEAN			MEDIAN		
sigma	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME
$1.1 < \sigma < 0.7$	0.5343	1.3839	99.3627	0.0199	1.3659	99.0165
$0.7 < \sigma < 0.5$	0.2155	2.0524	82.6348	0.0280	1.9839	96.7259
$0.5 < \sigma < 0.1$	0.3646	8.7570	83.8168	0.2627	2.5137	93.8714
$0.1 < \sigma < 0.05$	0.1143	60.2601	94.5441	0.1350	18.0480	94.4915
$0.05 < \sigma < 0.01$	0.0835	171.2094	94.8390	0.0577	2.6812	94.9495

Setting of data generation: (TOP=5, VOC=20, DN=1000, NT=100.)
Setting of `sdf_nls`: (MaxIte=5000, TolX=10^{-24}, TolFun=10^{-24}.)

The effect of an orthogonal structure of a topic-word matrix $[\boldsymbol{\mu}_1|\boldsymbol{\mu}_2|\cdots|\boldsymbol{\mu}_K]$ on the parameter recovery (inference) for each method with different settings is summarized in the Tables 5.7 and 5.8. Generally, it can be said that the collapsed Gibbs sampling approach results in lower relative errors with the same order in comparison with our tensor based method in which the order of E_{rel} is unstable and fluctuates for both its mean and median. Especially, the differences in E_{rel} between two methods $\Delta(E_{\text{rel}})$ seem to roughly become large as σ becomes small, meaning that $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ becomes less and less orthogonal. In fact, when the median of E_{rel} of Table 5.7 and E_{rel} of ITE=200 case of Table 5.8 are compared, $(\Delta(E_{\text{rel}})_{1.1<\sigma<0.7}, \Delta(E_{\text{rel}})_{0.7<\sigma<0.5}) = (0.0025, 0.0081)$ with their order 10^{-3}, while $(\Delta(E_{\text{rel}})_{0.5<\sigma<0.1}, \Delta(E_{\text{rel}})_{0.1<\sigma<0.05}, \Delta(E_{\text{rel}})_{0.05<\sigma<0.01}) =$

$(0.2424, 0.0963, 0.0379)$ with their order at least more than 10^{-2}. From the observation above, it would be difficult to claim that σ values are not associated to the quality of $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ recovery. Thus, we conclude that Hypothesis 7 is not validated in this experiment.

Regarding $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$, those for the collapsed Gibbs sampling approach significantly rise up as the true topic word matrix are less orthogonal, which is not the case for medians of condition numbers calculated from the result of our tensor method. Since there is a big difference between medians and means of condition numbers for the tensor based method, implying that $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$ would tend to fluctuate and thus it could be difficult to stably estimate a perturbation effect for our method.

In terms of computational time, the results of our tensor based method are faster than collapsed Gibbs sampling. For example, even a tensor method with maximum 5000 iterations needs only one third of computationaltime for the collapsed Gibbs sampling with 200 iterations.

Table 5.8: Result from the collapsed Gibbs sampling approach for Hypothesis 7

	ITE=300, BI=150			ITE=200, BI=100		
sigma	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	TIME
$1.1 < \sigma < 0.7$	0.0158	1.2922	779.3240	0.0174	1.2857	338.8986
$0.7 < \sigma < 0.5$	0.0195	1.2125	727.7543	0.0199	1.2392	346.8678
$0.5 < \sigma < 0.1$	0.0203	3.3419	959.4164	0.0203	3.7763	328.2348
$0.1 < \sigma < 0.05$	0.1690	587.9084	946.2755	0.0387	513.8209	355.0938
$0.05 < \sigma < 0.01$	0.0169	12160.1061	988.1139	0.0198	13648.2333	324.3537

Setting of data generation: (TOP=5, VOC=20, DN=1000, NT=100.)
Setting of `sdf_nls`: (MaxIte=5000, TolX=10^{-24}, TolFun=10^{-24}.)

5.6.4 Theoretical condition number vs empirical condition number

In Section 5.6.3, it is found that condition numbers tend to be large as σ is small. For instance, the order of the largest $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$ for collapsed Gibbs sampling even reaches 10^4. To further investigate this phenomena, we computed theoretical condition numbers $\kappa_{\{\boldsymbol{\mu}\}_{k=1}^{k=K}}$ using true $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ and compared them with $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$. The results are shown in Tables 5.9 and 5.10.

Table 5.9: Mean of theoretical and empirical condition numbers

	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	$\kappa_{\{\boldsymbol{\mu}\}_{k=1}^{k=K}}$	BE_{rel}	$BE_{rel} \times \kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$
σ					
$1.1 < \sigma < 0.7$	0.5343	1.3839	1.1124	0.1332	0.1843
$0.7 < \sigma < 0.5$	0.2155	2.0524	1.4631	0.0576	0.1182
$0.5 < \sigma < 0.1$	0.3646	8.7570	31.0065	0.0721	0.6314
$0.1 < \sigma < 0.05$	0.1143	60.2601	886.8186	0.0206	1.2414
$0.05 < \sigma < 0.01$	0.0835	171.2094	48499.9857	0.0219	3.7495

Setting of data generation: (TOP=5, VOC=20, DN=1000, NT=100.)
Setting of `sdf_nls`: (MaxIte=5000, TolX=10^{-24}, TolFun= 10^{-24}.)

As seen in Table 5.9, a larger difference between means of $\kappa_{\{\boldsymbol{\mu}\}_{k=1}^{k=K}}$ and $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$ is observed as $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ is less and less orthogonal. For example, the difference in the order of magnitude between the means of $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}} = 171.2094$ and $\kappa_{\{\boldsymbol{\mu}\}_{k=1}^{k=K}} = 48499.9857$ is 10^2 with $0.05 < \sigma < 0.01$ setting, while it is 10^0 with $1.1 < \sigma < 0.7$ setting. This implies that when a set of vectors of true parameter $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ is not orthogonal, finding $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ is far more difficult than finding $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}$ on average in the sense that even a tiny perturbation in an input tensor can be significantly amplified to result in a completely different solution. It is important, however, to note that $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}} = 171.2094$ is already so large that finding this solution is already very complex.

Compared to the means of $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$ and $\kappa_{\{\boldsymbol{\mu}\}_{k=1}^{k=K}}$, the medians of them shown in Table 5.10 are much smaller. In fact, for synthetic data generated with $0.05 < \sigma < 0.01$ setting, the medians of $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}} = 2.6812$ and $\kappa_{\{\boldsymbol{\mu}\}_{k=1}^{k=K}} = 22994.5520$ are much more smaller than corresponding means. This suggests that the complexity of problem itself fluctuates in a large extent. Furthermore, the median of $\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$ is approximately 10^4 times as large as $\kappa_{\{\boldsymbol{\mu}\}_{k=1}^{k=K}}$. This implies that recovering true parameter is far more difficult than discovering an approximate solution in many trials.

Table 5.10: Median of theoretical and empirical condition numbers

	E_{rel}	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	$\kappa_{\{\boldsymbol{\mu}\}_{k=1}^{k=K}}$	BE_{rel}	$BE_{rel} \times \kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$
σ					
$1.1 < \sigma < 0.7$	0.0199	1.3659	1.0827	0.0475	0.0649
$0.7 < \sigma < 0.5$	0.0280	1.9839	1.5083	0.0360	0.0714
$0.5 < \sigma < 0.1$	0.2627	2.5137	2.8431	0.0283	0.0711
$0.1 < \sigma < 0.05$	0.1350	18.0480	962.4312	0.0196	0.3537
$0.05 < \sigma < 0.01$	0.0577	2.6812	22994.5520	0.0190	0.0509

Setting of data generation: (TOP=5, VOC=20, DN=1000, NT=100.)
Setting of `sdf_nls`: (MaxIte=5000, TolX=10^{-24}, TolFun= 10^{-24}.)

5.6.5 Relationship between condition numbers and relative errors

Table 5.11: Relationship between relative errors and condition numbers.

	MEAN				MEDIAN			
	$\mathrm{E}_{\mathrm{rel}}$	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	$\mathrm{BE}_{\mathrm{rel}}$	$\mathrm{BE}_{\mathrm{rel}} \times \kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	$\mathrm{E}_{\mathrm{rel}}$	$\kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$	$\mathrm{BE}_{\mathrm{rel}}$	$\mathrm{BE}_{\mathrm{rel}} \times \kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$
TOP								
3	0.3098	1.1625	0.1133	0.1317	0.0095	1.1217	0.0157	0.0176
4	0.2627	1.2542	0.0868	0.1089	0.0114	1.2267	0.0339	0.0416
5	0.5343	1.3839	0.1332	0.1844	0.0199	1.3659	0.0475	0.0649
6	0.7667	1.3851	0.1900	0.2631	1.2275	1.3477	0.2572	0.3466
7	0.8098	1.7510	0.1849	0.3237	1.3222	1.5679	0.2322	0.3641
8	0.5183	2.0270	0.1261	0.2555	0.0434	1.8155	0.0830	0.1507
9	0.3290	1.8188	0.1222	0.2222	0.0439	1.8268	0.0942	0.1721
10	0.2961	1.8426	0.1382	0.2547	0.0481	1.6681	0.1012	0.1688
VOC								
20	0.5343	1.3839	0.1332	0.1844	0.0199	1.3659	0.0475	0.0649
30	1.2681	1.6606	0.2903	0.4820	1.2580	1.2767	0.2932	0.3743
40	1.0952	1.2920	0.3084	0.3985	1.3109	1.2613	0.4057	0.5117
50	0.5558	1.2794	0.1556	0.1990	0.0320	1.2534	0.0488	0.0612
60	0.7879	1.2832	0.2215	0.2842	1.2383	1.2837	0.2423	0.3111
70	0.5780	1.3173	0.1821	0.2399	0.0264	1.2669	0.0488	0.0618
σ								
$1.1 < \sigma < 0.7$	0.5343	1.3839	0.1332	0.1843	0.0199	1.3659	0.0475	0.0649
$0.7 < \sigma < 0.5$	0.2155	2.0524	0.0576	0.1182	0.0280	1.9839	0.0360	0.0714
$0.5 < \sigma < 0.1$	0.3646	8.7570	0.0721	0.6314	0.2627	2.5137	0.0283	0.0711
$0.1 < \sigma < 0.05$	0.1143	60.2601	0.0206	1.2414	0.1350	18.0480	0.0196	0.3537
$0.05 < \sigma < 0.01$	0.0835	171.2094	0.0219	3.7495	0.0577	2.6812	0.0190	0.0509

Finally, we investigated Hypothesis 3 based on the results we obtained so far. From Equation (3.30), it is expected that the true relative error $\mathrm{E}_{\mathrm{rel}}$ is bounded by the value of condition number multiplied by relative backward error. As shown in Table 5.11, the means of $\mathrm{BE}_{\mathrm{rel}} \times \kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$ for different TOP and VOC are all smaller than corresponding $\mathrm{E}_{\mathrm{rel}}$, which does not match Equation (3.30). On the other hand, the medians of $\mathrm{BE}_{\mathrm{rel}} \times \kappa_{\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}}$ are larger than the medians of $\mathrm{E}_{\mathrm{rel}}$ for all TOP, VOC, and σ, excepting for the cases $0.5 < \sigma < 0.1$ and $0.05 < \sigma < 0.01$. This seems to match Equation (3.30) well. This leads us to a conclusion that the condition number of the tensor corresponding to $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}$ could be a relatively good indicator for the relative error roughly for half trials among all but not all the cases. Hence, Hypothesis 3 is too strong statement to be verified based on these experimental results.

5.7 Real dataset: NIPS proceedings papers

Until here, the results of our study on our tensor based method with synthetic data are shown. In this section, we show results of our experiment on the performance

of our method for real dataset, which was done to observe how our tensor based method behaves with real document dataset.

As real dataset, the NIPS proceeding data `bagofwords_nips.mat, words_nips.mat` in MATLAB `topictoolbox` [54] were used. The data are composed from 2301375 words over 1740 documents and have a document index between 1 and 1740 and vocabulary index between 1 and 13649, which is the dictionary size. While the Gibbs sampling approach is directly applicable to the original dataset, our method is not. This is because our algorithm is not scalable. For example, our algorithm requires an input #total words $\times$ VOC matrix and it would need 234.0GB to be stored if the original data were used. Therefore, among 13649 distinct words, only from the most frequently appearing word up to the 100th most frequently appearing word were extracted and considered.

5.7.1 Nips proceeding datasets: Gibbs sampling vs tensor approach

Table 5.12 and Table 5.13 show the experimental results for the subset of NIPS dataset. For each topic under the different dictionary size (VOC) setting, the words with higher probability of occurrence based on the estimated $[\boldsymbol{\mu}_1 \boldsymbol{\mu}_2]$ are listed. As can be seen in those tables, the result obtained from the Gibbs sampling approach looks roughly similar to the one gained from our tensor decomposition approach with the setting maximum iteration (MaxIte) 5000, the lower bound of the step size (TolX) and the lower bound of the decrease of an objective function (TolFun) 10^{-30}, if we ignore a strict order of words. By observing those words for each topic for results from both approaches, we could come up with topic labels "neural network" and "machine learning".

However, it is true that the order of words is not a difference between results from both methods. For example, there are a few words that only appear in the result from either method such as "error" of the result from tensor approach or "data" and "figure" of the result from Gibbs sampling approach. Furthermore, the output from Gibbs sampling with VOC = 90 for one topic looks different from others: "analysis processing information models distribution feature single". This might be due to unconvergence of Markov chain. Except for this seemingly special case, our assigned topic labels "neural network" and "machine learning" would be still reasonable. Thus, for this NIPS dataset, our tensor method could be regarded to function as well as Gibbs sampling method under the assumption that the number of topics is 2.

Regarding the setting of stopping rule for `sdf_nls`, it can be said that TolFun and TolX need to be carefully chosen such that they are sufficiently small. In fact, when the setting of Maxite=300 and TolX=TolFun=10^{-12} for the tensor method was used, its results looks more different from ones of Gibbs sampling method.

Table 5.12: Words with higher probabilities for each topic: the Gibbs sampling vs tensor decomposition approach

VOC	Gibbs sampling: ITE=100	Tensor decomposition: MaxIte=300, TolX=TolFun=10^{-12} GEVD initialization
10	{'network neural input'} {'learning model data' }	{'learning' {'network neural'}
20	{'network input neural networks training output units'} {'model learning data algorithm function time state',}	{'learning state function time error',} {'network neural networks training input output units'}
30	{'model learning data algorithm time function state',} {'network input neural networks training output units'}	{'network input time learning neural system output'} {'data models results based neural model algorithm'}
40	{'network input neural networks training output units'} {'model learning data function time algorithm figure' }	{'learning network training networks error set units',} {'network neural information networks results number input'}
50	{'network input neural networks output training units'} {'model learning data function algorithm set state',}	{'data state linear space vector number model',} {'network output algorithm units networks weight problem'}
60	{'network input networks training neural output units'} {'model learning data time function figure algorithm' }	{'data layer neurons performance input vector neural',} {'distribution single network pattern rate large figure'}
70	{'network input neural networks output time units' } {'learning model data function algorithm set state'}	{'based time weight model simple learning space',} {'problem single class hidden approach results network'}
80	{'network input neural networks output time figure',} {'learning data model function algorithm set training'}	{'learning network time units model control neural',} {'performance work neural error number analysis data'}
90	{'network learning training set data networks function'} **{'analysis processing information models distribution feature single'}**	{'model time figure state system neural input' {'paper network hidden point gaussian task model',}
100	{'network input neural networks output figure time' {'learning data function set algorithm model training'}	{'level single unit process feature distribution points'} {'time trained cells features weight distribution net',}

NIPS proceedings papers (bag of words) in MATLAB *topictoolbox* is used. Since a tensor decomposition based method can not apply with the original vocabulary size in a dictionary 13649, most frequently occurring 20 ~ 100 distinct words are used as vocabulary size. Setting: The numebr of Topics = 2, the total number of observed words = 144549(#VOC = 10) ~ 523333(#VOC = 100),the number of documents = 1740

Table 5.13: Words with higher probabilities for each topic: Tensor decomposition approach

VOC	MaxIte=300, TolX=TolFun=10^{-12}(random initialization)	MaxIte=5000, TolX=TolFun=10^{-30}(GEVD initialization)	Final iterations
10	{'learning function'} {'model data' }	{'network neural' } {'learning function'}	27
20	{'network neural networks training input output units'} {'model data learning set' }	{'learning state function time algorithm' } {'network neural networks training input output units'}	30
30	{'training figure performance output units network models'} {'network learning networks neural input units output' }	{'learning state function time algorithm model error' } {'network neural networks training input output units'}	97
40	{'network neural networks input training output units'} {'learning state function time algorithm error model' }	{'network neural networks training input output units'} {'learning state function time algorithm model error' }	71
50	{'figure learning state weights linear space system' } {'error units training results systems point network'}	{'learning state function time algorithm model error' } {'network networks neural input training units output'}	202
60	{'function input point learning large models problem' } {'network learning networks training input unit hidden'}	{'learning state function time model algorithm error' } {'network neural networks input training units output'}	163
70	{'learning large networks input time figure algorithms'} {'model network control noise image training neural' }	{'network neural networks input training units output'} {'learning state function time model algorithm error' }	103
80	{'hidden task unit time shows output data' } {'test feature performance inputs small signal network'}	{'learning state function time model algorithm error' } {'network neural networks input training units output'}	105
90	{'learning weights process network neurons state weight'} {'case signal figure layer functions level distribution'}	{'learning state function time model algorithm error' } {'network neural networks input training units output'}	75
100	{'noise representation recognition examples classification state similar'} {'distribution learning figure order size work form' }	{'learning state function time model algorithm error' } {'network neural networks input training units output' }	89

5.8 Conclusions

In this section, the experimental results of our method on synthetic and real dataset are introduced. Through the investigation of multiple hypotheses, we empirically studied in which situation or setting our method based on symmetric CP decomposition works. Based on our experimental results, we can use our new tensor method when the following conditions are all satisfied.

- The long iterations are used; in other words, a stricter stopping criteria are employed.
- The true topic matrix $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ is approximately orthogonal.
- A corpus does not contain either a large total number of words or a large number of distinct words over the corpus.
- The number of topics K of the LDA model is small; for example at most 10 based on our experiment.
- The number of words in a dictionary D is relatively small; for example at most 70 based on our experiment.
- Multiple datasets modeled by the same LDA model are available and median can be considered.

Moreover, we also found that significantly shorter computational time to attain $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}$ is achievable for our tensor method compared with collapsed Gibbs sampling. This would be a strong reason why this new method of ours is so attractive once an empirical validation of this method is established.

However, we could not empirically verify that our method works for slightly more relaxed conditions considered in Hypothesis 7, which suggests that our new method should not be used to recover parameter of a LDA in general at this moment. More specifically, there are some difficulties in using this new method based on our experimental results.

Firstly, limitation of the input data. Our method cannot be applied to the data of a corpus consisting of numerous total number of words or the large number of distinct words, which is actually common in practice. In addition, our method was empirically shown to work almost only when true $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ is approximately orthogonal, which would be pretty rare in practice.

Secondly, a lack of empirical evidence that our method surely recovers $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ as correctly as collapsed Gibbs sampling does. Although there is a theoretical guarantee that our method works as discussed in Chapter 4, we cannot obtain empirical evidence sufficient enough to support this through our experiments. In fact, as quite a large difference between means and medians of E_{rel} shown over the experimental results suggests, it is found that $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}$ recovered from our method is very unstable in the sense that $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}$ is sometimes close to $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$ and sometimes not. Thus, even when we have the data to which our method is applicable, the obtained result can be different from true $\{\boldsymbol{\mu}\}_{k=1}^{k=K}$.

Despite such an absence of empirical support for validation of our new method discussed above, we still applied it to the real dataset by modifying the data. As a result, it was observed that our method appeared to succeed in extracting the topic labels similar to those obtained with collapsed Gibbs sampling, although $\{\hat{\boldsymbol{\mu}}\}_{k=1}^{k=K}$ recovered with our method does not correctly match to the one estimated with collapsed Gibbs sampling. Thus, our method might perform enough to find hidden topics if TOP is small.

However, it is important to note that this result can be wrong because our method is not be able to directly apply to the real data in most cases at this moment. As we have discussed in Section 5.7, the data need to be modified so that the size of input matrix is smallened and our method can be applied to them. Because of such a modification, the result would be no longer be ensured to be similar to the one obtained with application of collapsed Gibbs sampling on the original dataset.

Chapter 6

Conclusion

In this thesis, we have presented a parameter recovery method for LDA based on symmetric CP decomposition to recover a set of vectors of topic-words distribution $\{\boldsymbol{\mu}\}_{k=1}^{K}$ and $\boldsymbol{\alpha}$ of Dirichlet prior for $\{\boldsymbol{\mu}\}_{k=1}^{K}$ from a collection of text data. Our method is fundamentally based on two proven theories: one is that the slightly modified third order moment computed from word-triplets under the LDA model is asymptotically symmetric CP decomposable, and the other is that generic low-rank symmetric CP decomposition is unique. These strong theoretical supports are advantages of our method.

Furthermore, we have empirically investigated if our proposed method successfully recovers the true parameter $\{\boldsymbol{\mu}\}_{k=1}^{K}$ using synthetic data. As a result, we have experimentally determined that our method does not stably recover correct $\{\boldsymbol{\mu}\}_{k=1}^{K}$ even when $\{\boldsymbol{\mu}\}_{k=1}^{K}$ is orthogonal. Moreover, we have substantiated that recovered parameters with our method tend to become more and more incorrect as $\{\boldsymbol{\mu}\}_{k=1}^{K}$ is less and less orthogonal. In addition, we have seen that condition numbers of the tensors corresponding to the recovered $\{\boldsymbol{\mu}\}_{k=1}^{K}$ are not stable indicators of true relative errors. What is more, we have empirically verified that the nonlinear least square method adopted in our method requires the stricter stopping rule criteria and larger iterations to find a plausibly correct solution. This is probably associated to the non-convexity of the objective function, which would be also related to unstableness of results of our method. These results suggest that the application of our method in practice is not recommendable at this moment.

On the other hand, we have verified that our method needs much less computational time than collapsed Gibbs sampling as one of the advantages. This observation would support motivation for further research on methods of parameter recovery based on moments; namely tensors and their decompositions.

Considering LDA is regarded as a simple topic model and a variety of extended LDA models exist, a future work could include exploring if our method is applicable to some of those models.

Chapter 7

Appendix

All the MATLAB and Python codes used in experiments are available online: https://github.com/Nozomi-Takemura/Master_thesis

Bibliography

[1] Absil, P.-A., Mahony, R., and Sepulchre, R. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, USA, 2007.

[2] Allman, E. S., Matias, C., and Rhodes, J. A. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics 37*, 6A (2009), 30993132.

[3] Anandkumar, A., Foster, D. P., Hsu, D., Kakade, S. M., and Liu, Y.-K. A spectral algorithm for latent dirichlet allocation. *Algorithmica 72*, 1 (Mar 2014), 193214.

[4] Anandkumar, A., Foster, D. P., Hsu, D. J., Kakade, S. M., and kai Liu, Y. A spectral algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 917–925.

[5] Anandkumar, A., Ge, R., Hsu, D., Kakade, S. M., and Telgarsky, M. Tensor decompositions for learning latent variable models. *J. Mach. Learn. Res. 15*, 1 (Jan. 2014), 2773–2832.

[6] Blei, D. Video lecture: Topic models. http://videolectures.net/mlss09uk_blei_tm/, 2009. "Accessed: 2018-05-23".

[7] Blei, D. M. Introduction to probabilistic topic models. In *In Communications of the ACM* (2011).

[8] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res. 3* (Mar. 2003), 993–1022.

[9] Breiding, P. *Numerical and Statistical Aspects of Tensor Decompositions*. PhD thesis, Technical University of Berlin, 2017. Accessed: 2018-05-17.

[10] Breiding, P., and Vannieuwenhoven, N. The condition number of join decompositions. *ArXiv e-prints* (Nov. 2016).

[11] Buntine, W. L. Operations for learning with graphical models. *CoRR abs/cs/9412102* (1994).

[12] 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 (sep 1970), 283–319.

[13] Celeux, G., Hurn, M., and Robert, C. P. Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association 95*, 451 (2000), 957970.

[14] Chiantini, L., Ottaviani, G., and Vannieuwenhoven, N. On generic identifiability of symmetric tensors of subgeneric rank. *Transactions of the American Mathematical Society 369*, 6 (Aug 2016), 40214042.

[15] Clark, S. Lecture slides: Topic modelling and latent dirichlet allocation. https://www.cl.cam.ac.uk/teaching/1213/L101/clark_lectures/lect7.pdf, 2013. Accessed: 2018-05-28.

[16] Comon, P. Tensors : A brief introduction. *IEEE Signal Processing Magazine 31*, 3 (2014), 4453.

[17] Comon, P., Golub, G., Lim, L.-H., and Mourrain, B. Symmetric tensors and symmetric tensor rank. *SIAM J. Matrix Anal. Appl. 30*, 3 (Sept. 2008), 1254–1279.

[18] Domanov, I. *Study of Canonical Polyadic Decomposition of Higher-Order Tensors*. PhD thesis, KU LEUVEN, 2013.

[19] Domanov, I., and Lathauwer, L. D. Canonical polyadic decomposition of third-order tensors: Reduction to generalized eigenvalue decomposition. *SIAM Journal on Matrix Analysis and Applications 35*, 2 (2014), 636660.

[20] For, C., and Harshman, R. A. Foundations of the parafac procedure: Models and conditions for an "explanatory " multimodal factor analysis, 1970.

[21] Geman, S., and Geman, D. Stochastic relaxation, gibbs distributions and the bayesian restoration of images∗. $Journal of Applied Statistics 20, 5 - 6(jan 1993), 25 - -62$.

[22] Griffiths, T. L., and Steyvers, M. Finding scientific topics. *Proceedings of the National Academy of Sciences 101*, Supplement 1 (Oct 2004), 52285235.

[23] Gu, L. Lecture slides: Dirichlet distribution, dirichlet process and dirichlet process mixture. https://www.cs.cmu.edu/~epxing/Class/10701-08s/recitation/dirichlet.pdf. Accessed: 2018-05-28.

[24] Hateley, J. C. Introduction to the tensor product. http://web.math.ucsb.edu/~jhateley/project/tensor.pdf. "Accessed: 2018-05-28".

[25] Heinrich, G. Parameter estimation for text analysis. *Technical report* (2009).

[26] Hillar, C. J., and Lim, L.-H. Most tensor problems are np-hard. *J. ACM 60*, 6 (Nov. 2013), 45:1–45:39.

[27] Hitchcock, F. L. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics 6*, 1-4 (1927), 164189.

[28] Ho, B. T. Lecture slides: Graphical models and topic modeling. http://www.jaist.ac.jp/~bao/VIASM-SML/Lecture/L5-Graphical%20Model%20and%20TopicModeling.pdf. Accessed: 2018-04-27.

[29] Hong, L. Hong, liangjie | head of data science at etsy: Generate synthetic data for lda. http://www.hongliangjie.com/2010/09/30/generate-synthetic-data-for-lda/, Sep 2010. Accessed: 2018-05-20.

[30] Hotelling, H. Relations between two sets of variates. *Biometrika 28*, 3/4 (1936), 321.

[31] Ikko. Latent dirichlet allocation (mcmc with model selection) - file exchange - matlab central. https://nl.mathworks.com/matlabcentral/fileexchange/56919-latent-dirichlet-allocation--mcmc-with-model-selection-, May 2016. Accessed: 2018-05-13.

[32] Jelodar, H., Wang, Y., Yuan, C., and Feng, X. Latent Dirichlet Allocation (LDA) and Topic modeling: models, applications, a survey. *ArXiv e-prints* (Nov. 2017).

[33] Kolda, T. G., and Bader, B. W. Tensor decompositions and applications. *SIAM Rev. 51*, 3 (Aug. 2009), 455–500.

[34] Kruskal, J. B. Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications 18*, 2 (1977), 95138.

[35] Kruskal, J. B. Multiway data analysis. North-Holland Publishing Co., Amsterdam, The Netherlands, The Netherlands, 1989, ch. Rank, Decomposition, and Uniqueness for 3-way and N-way Arrays, pp. 7–18.

[36] Lahat, D., Adalı, T., and Jutten, C. Multimodal Data Fusion: An Overview of Methods, Challenges and Prospects. *Proceedings of the IEEE 103*, 9 (Aug. 2015), 1449–1477.

[37] Lathauwer, L. D. A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization. *SIAM Journal on Matrix Analysis and Applications 28*, 3 (2006), 642666.

[38] Link, W. A., and Eaton, M. J. On thinning of chains in mcmc. *Methods in Ecology and Evolution 3*, 1 (2011), 112115.

[39] Liu, L., Tang, L., Dong, W., Yao, S., and Zhou, W. An overview of topic modeling and its current applications in bioinformatics. *SpringerPlus 5*, 1 (2016).

[40] MacKay, D. J. C. *Information Theory, Inference & Learning Algorithms*. Cambridge University Press, New York, NY, USA, 2002.

[41] McCallum, A. K. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[42] Mimno, D. [topic-models] the computational complexity of lda. https://lists.cs.princeton.edu/pipermail/topic-models/2008-April/000213.html, Apr 2008. Message to the lists.cs.princeton.edu Mailing Lists.

[43] Minka, T., and Lafferty, J. Expectation-propagation for the generative aspect model. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence* (San Francisco, CA, USA, 2002), UAI'02, Morgan Kaufmann Publishers Inc., pp. 352–359.

[44] Nico, V., Otto, D., Laurent, S., Marc, V. B., and Lieven, D. L. Structured data fusion. https://www.esat.kuleuven.be/sista/tensorlab/doc/sdf-basic.html. Accessed: 2018-05-25.

[45] Nico, V., Otto, D., Laurent, S., Marc, V. B., and Lieven, D. L. Tensorlab 3.0. https://www.tensorlab.net, 2016. Accessed: 2018-05-28.

[46] OEDING, L. Tensor decomposition and algebraic geometry. https://simons.berkeley.edu/sites/default/files/docs/1718/slidesoeding.pdf, 2014. Accessed: 2018-05-17.

[47] Pearson, K. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 185* (Jan 1894), 71110.

[48] Rabanser, S., and Shchur, O. Introduction to Tensor Decompositions and their Applications in Machine Learning. 1–13.

[49] Ruffini, M., Casanellas, M., and Gavaldà, R. A New Spectral Method for Latent Variable Models. *ArXiv e-prints* (Dec. 2016).

[50] Sidiropoulos, N. D., Lathauwer, L. D., Fu, X., Huang, K., Papalexakis, E. E., and Faloutsos, C. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing 65*, 13 (jul 2017), 3551–3582.

[51] Silva, V. D., 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 (2008), 10841127.

[52] Song, Z., Woodruff, D., and Zhang, H. Sublinear time orthogonal tensor decomposition. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 793–801.

[53] Sorber, L., Barel, M. V., and Lathauwer, L. D. Structured data fusion. *IEEE Journal of Selected Topics in Signal Processing 9*, 4 (2015), 586600.

[54] Steyvers, M. Matlab topic modeling toolbox 1.4. http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm#Installation_&_Licensing. Accessed: 2018-05-13.

[55] Takane, Y., W. Young, F., and De Leeuw, J. Nonmetric individual differences multidimensional scaling: An alternating least squares method with optimal scaling features. 7–67.

[56] Tomioka, R., Hayashi, K., and Kashima, H. Estimation of low-rank tensors via convex optimization. *ArXiv e-prints* (Oct. 2010).

[57] Trefethen, L. N., and Bau, D. *Numerical Linear Algebra*. SIAM, 1997.

[58] Vannieuwenhoven, N. Tensor decompositions for machine learning applications. https://lirias.kuleuven.be/bitstream/123456789/596392/1/DTAI_Seminar_20171009.pdf. Accessed: 2017-11-28.

[59] Vannieuwenhoven, N. Presentation slides: The geometry of the tensor rank decomposition. https://www.researchgate.net/publication/324529816_The_geometry_of_the_tensor_rank_decomposition, 2016. Accessed: 2018-05-17.

[60] Walsh, B. Lecture notes for eeb 581: Markov chain monte carlo and gibbs sampling. http://nitro.biosci.arizona.edu/courses/EEB519A-2007/pdfs/Gibbs.pdf, April 2004. Accessed: 2018-04-22.

[61] Yildirim, I. Bayesian inference: Gibbs sampling. http://www.mit.edu/~ilkery/papers/GibbsSampling.pdf, August 2012. Accessed: 2018-04-22.

[62] Zhang, C., Gutiérrez, E. D., Asplund, A., and Pescatore, L. Tensor Decomposition for Topic Models: An Overview and Implementation.

[63] Zhang, X., and Sisson, S. A. Blocking Collapsed Gibbs Sampler for Latent Dirichlet Allocation Models. *ArXiv e-prints* (Aug. 2016).

[64] Zou, J., Hsu, D., Parkes, D., and Adams, R. P. Contrastive learning using spectral methods. In *Advances in Neural Information Processing Systems 26* (12/2013 2013).

Master's thesis filing card

Student: Nozomi Takemura

Title: Tensor decompositions for latent dirichlet allocation in topic modeling

UDC: 621.3

Abstract:

Latent Dirichlet allocation (LDA) is one of the most popular topic models, which is used to learn unobserved topics from a collection of documents by assuming a specific data generation process specified with the dependency between latent and observed variables. One of the challenges of LDA is the inference of latent variables, which has been tackled with a variety of approaches but all of them exhibit some disadvantages.

Recently, however, tensor decomposition-based methods stemming from the method of moments have been proposed to recover a subset of LDA parameters, which are supported by the solidly theoretical framework and hugely researched computational methods advanced in the field of linear algebra or numerical analysis.

In this thesis, as one such moment-based parameter recovery methods for LDA, we propose symmetric Canonical Polyadic Decomposition. The main aim of this thesis is to empirically investigate if this method works since there is no previous research with empirical results. Furthermore, one of the biggest motivations for proposing this method is that it theoretically recovers the set of unique parameter vectors of topic-word distribution with probability 1, while most of other previously introduced methods, such as ones based on Markov chain Monte Carlo, do not have such strong theoretical guarantees for parameter estimation.

Although the basic concept of our proposed method has been presented earlier in literature, most of the proposed methods are dependent on the orthogonalization of an input tensor. This leads to the second motivation for proposing our method that does not involve such orthogonalization process. Specifically, the inclusion of such an orthogonalization step theoretically poses a much stricter limit on the maximum number topics that can be considered in an LDA model than our proposed method.

To further study if our proposed method actually works, we conduct experiments using synthetic data generated from LDA models with various parameter settings, subsequently comparing generated parameter with the recovered parameter. Moreover, parameters of the same LDA models are inferred from the same synthetic data using popularly used collapsed Gibbs sampling. By doing this, we compare the results from our method to the ones from collapsed Gibbs sampling. Finally, we apply our method to pieces of real datasets and extract hidden topics, which are again compared with the ones obtained from collapsed Gibbs sampling.

The main contribution of this thesis is that we experimentally determine that our proposed method does not correctly recover the parameter of topic-word distributions of LDA despite the existence of a theoretical guarantee.

Thesis submitted for the degree of Master of Science in Statistics, option General Statistical Methodology

Thesis supervisors: Prof. dr. ir. Karl Meerbergen
Prof. dr. ir. Johan Suykens
Dr. Nick Vannieuwenhoven

Assessors: Prof. dr. ir. Luc De Raedt
Prof. dr. ir. Jan Aerts

Mentor: Mr. Bruno Coussement