Non-parametric Bayesian Methods for Structured Topic Models

Lan Du

October, 2012

A thesis submitted for the degree of Doctor of Philosophy
of the Australian National University

To my wife, Jiejuan Wang, and daughter, Yufei Du

Declaration

The work presented in this thesis is my original research work, except where due acknowledgement has been made in the text of the thesis to all other material used. This work has not been submitted previously for a degree at any university.

Lan Du
October 2012

Acknowledgements

First and foremost, I would like to thank my supervisor, Dr. Wray Buntine, for chairing my supervisor committee and for his supervision, advice and guidance as well as giving me extraordinary experiences throughout my Ph.D. studies. His knowledge and passion in science exceptionally inspire and enrich my growth as a Ph.D. student and a scientific researcher. A large portion of my knowledge of doing scientific research, especially the way of doing rigorous studies, that I have today is learnt from his day to day supervision. I am also grateful to Wray for his support and understanding of family issues that I encountered during my Ph.D. studies at ANU in Canberra. It has been my honour to be his Ph.D. student. I would like to attribute the level of my Ph.D. degree to his encouragement and efforts, and without him this thesis would not have been completed or written.

I would also like to thank my co-supervisor, Dr. Huidong Jin, for his advice and supervision. His discussions and comments have greatly contributed to this thesis. He provided me constant support and encouragement in various ways. His insight, inspiration and feedback have been invaluable. In particular, the scientific writing skill that I have today is largely due to him. Moreover, I appreciate him for using his precious times to listen to and help me resolve my life problems.

Many thanks go to my advisors, Dr. Peter Christen and Dr. Lei Wang for their valuable advice in science discussions. I am grateful in every possible ways to every member in my supervisor committee and hope to keep up our collaboration in the future.

During my Ph.D. studies at ANU, I am quite fortunate to have several overseas research travels that enable me to interact with great researchers all over the world. I would express my gratitude to Prof. Yi Zhang from Sichuan University in China, Prof. Mao Ye from University of Electronic Science and Technology of China and Dr. Yangqiu Song from IBM China research laboratory for hosting my visits. I had a wonderful time at their laboratories.

It is a pleasure to thank all the members of the Statistical Machine Learning group for good advice and collaboration. They made my time at ANU easier and enjoyable. Special thanks go to administration staffs at the ANU Research School of Computer Science for handling all the administrative disorders, and those at NICTA for organising me conference travels. I gratefully acknowledge NICTA for funding me through the NICTA Ph.D. scholarship and the supplementary scholarship, which make my Ph.D. work possible.

Many thanks go in particular to Dr. Guolin Hua, for his thoughtfulness and financial support. Without him, I would not have come to Australia, and I would not have gained my achievement.

I owe my deepest gratitude to my parents, Hanqing Du and Xiaoling Che, who sincerely raised me with their unselfish caring and love. They deserve very special mention for their inseparable support and encouragement. Without them, I cannot pursue my ideals in Australia. I owe them an awful lot.

Words fail me to express my appreciation to my wife, Jiejuan Wang, for her dedication, love and confidence in me. In 2010, she was pregnant, but I could not stay with her in China because I had to carry on my Ph.D. studies in Canberra. I left her alone with my parents, which was inconceivable. However, my wife has never complained about me, so I can successfully complete this thesis. I feel also deeply indebted to my little gorgeous daughter, Yufei Du. The total time I have stayed with her is only about five months, but now she is two years old. Their sacrifices make the completion of this thesis possible and easier. Thank you.

Finally, I would like to thank everybody who has helped me throughout the Ph.D. studies, as well as expressing my apologies that I could not mention all of them one by one.

Abstract

The proliferation of large electronic document archives requires new techniques for automatically analysing large collections, which has posed several new and interesting research challenges. Topic modelling, as a promising statistical technique, has gained significant momentum in recent years in information retrieval, sentiment analysis, images processing, *etc.* Besides existing topic models, the field of topic modelling still needs to be further explored using more powerful tools. One potentially useful area is to directly consider the document structure ranging from semantically high-level segments (*e.g.*, chapters, sections, or paragraphs) to low-level segments (*e.g.*, sentences or words) in topic modelling.

This thesis introduces a family of structured topic models for statistically modelling text documents together with their intrinsic document structures. These models take advantage of non-parametric Bayesian techniques (*e.g.*, the two-parameter Poisson-Dirichlet process (PDP)) and Markov chain Monte Carlo methods. Two preliminary contributions of this thesis are

1. The Compound Poisson-Dirichlet process (CPDP): it is an extension of the PDP that can be applied to multiple input distributions.

2. Two Gibbs sampling algorithms for the PDP in a finite state space: these two samplers are based on the Chinese restaurant process that provides an elegant analogy of incremental sampling for the PDP. The first, a two-stage Gibbs sampler, arises from a table multiplicity representation for the PDP. The second is built on top of a table indicator representation. In a simply controlled environment of multinomial sampling, the two new samplers have fast convergence speed.

These support the major contribution of this thesis, which is a set of structured topic models:

Segmented Topic Model (STM) which models a simple document structure with a four-level hierarchy by mapping the document layout to a hierarchi-

cal subject structure. It performs significantly better than latent Dirichlet allocation and other segmented models at predicting unseen words.

Sequential Latent Dirichlet Allocation (SeqLDA) which is motivated by topical correlations among adjacent segments (*i.e.*, the sequential document structure). This new model uses the PDP and a simple first-order Markov chain to link a set of LDAs together. It provides a novel approach for exploring the topic evolution within each individual document.

Adaptive Topic Model (AdaTM) which embeds the CPDP in a simple directed acyclic graph to jointly model both hierarchical and sequential document structures. This new model demonstrates in terms of per-word predictive accuracy and topic distribution profile analysis that it is beneficial to consider both forms of structure in topic modelling.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In recent years, with the fast development of the web and the advent of various digitisation techniques (*e.g.*, optical character recognition and speech recognition), documents continue to be stored on the Internet in the form of webpages, blogs, twitters, news papers, e-books, scientific articles, *etc.* The proliferation of large electronic document archives requires new techniques for automatically organising, searching, indexing, and browsing large collections, which has posed several new and interesting challenges to researchers in both the machine learning and the data mining communities. In particular, there is an increasing need of automatic methods to semantically visualise and analyse these electronic documents. This thesis presents new probabilistic generative methods based on non-parametric Bayesian techniques (*e.g.*, the Dirichlet processes and the two-parameter Poisson-Dirichlet processes) for effectively modelling text documents by considering their intrinsic document structure.

Documents not only contain meaningful text, but also exhibit a natural structure, which is part of the motivation of the development of SGML, the precursor to HTML. For example, a book has chapters which themselves contain sections; a section is further composed of paragraphs; a blog or a twitter page contains a sequence of comments and links to related blogs/twitters; a scientific article contains appendices and references to related work. Clearly, a complete representation of a document structure ranges from the high-level components (*e.g.*, chapters or sections) to the low-level components (*e.g.*, sentences or words). These components (referred to as segments thereafter) provide rich contextual information for their subcomponents. The layout of the components is always represented in various forms jointly with the document logical structure, *i.e.*, the latent subject structure. Altogether, the segments form a document structure, which will be

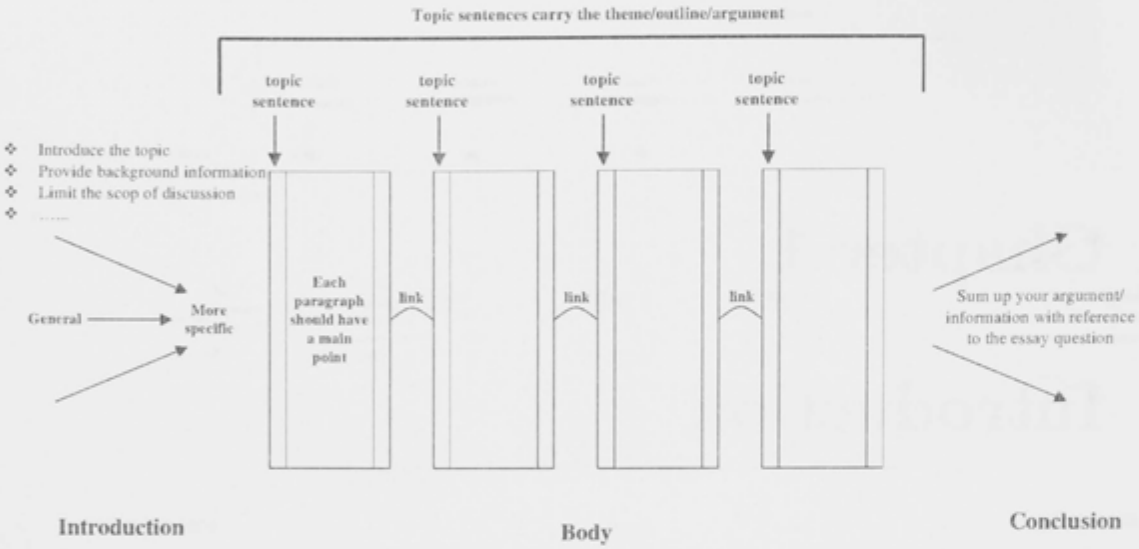

Topic sentences carry the theme/outline/argument
topic sentence
topic sentence
topic sentence
topic sentence
Introduce the topic
Provide background information
Limit the scop of discussion
......
General
More specific
Each paragraph should have a main point
link
link
link
Sum up your argument/ information with reference to the essay question
Introduction
Body
Conclusion

Figure 1.1: An example document structure: an essay structure.

considered in this thesis. It can be very beneficial to directly consider the document structure in statistical document modelling. The structural information can be useful for indexing and retrieving the information contained in the document, for instance, for structured information retrieval and digital libraries.

A well organised document structure can convey two kinds of information. First, the layout of segments (*e.g.*, the chapter sequence in a story book or the paragraph sequence in an essay) in a document gives many clues about the subject structure of the document, which implies some semantic relationships among those segments. These clues can also help readers to navigate documents according to the subject structures. Second, the text content itself can give rich information about the relationships and semantics of text. Analysing the text content and exploring the document structure can provide us information about, for example, how subjects are organised in a document and how they change over the structure. Consequently, modelling documents along with their structures is an interesting and potentially important problem in exploratory and predictive text analytics.

To further explain the document structure, I take as an example an essay structure shown in Figure 1.1. An easily accessible and understandable structure is very important for an essay. Generally, an essay should have a subject which indicates what the essay talks about; then paragraphs, basic structural units in an essay, are organised around the subject. Furthermore, each paragraph should have one or more subtopics, that are somehow linked together to make up the essay subject. It means the subtopics are not isolated, but they can be more specific than the essay subject, and generally be variants of it. The layout and progression of them can give us a meaningful essay structure. Indeed, the above consideration originates from how people normally organise ideas in their writing.

As a consequence, a different challenge in automatic text analysis is the problem of understanding the document structure. The focus of this thesis is to statistically model the text content of documents together with their underlying document structures by taking advantage of both topic modelling (Chapter 4) and non-parametric Bayesian methods (Chapters 2 and 3). In recent years, topic models and non-parametric Bayesian methods become increasingly prominent in machine learning. The former forms a family of models in which documents can be generated with simple probabilistic generative processes. The latter provides a valuable suit of flexible modelling techniques, in which the prior and posterior distributions are general stochastic processes whose support is the space of all distributions.

1.1 Thesis Contribution

The objective of this thesis is to address research challenges for structured text analysis in the context of hierarchical non-parametric Bayesian modelling. This leads to the development of a family of structured topic models. Most existing topic models directly model documents by tokens with the "*bag-of-words*" assumption. They usually neglect the document structure. However, incorporating the document structure in topic modelling, we can derive a richer posterior topical structure that can further facilitate understanding and exploring each individual document.

As discussed in the previous section, a document is usually composed of a certain number of segments. The definition of segments can vary according to different types of documents. They can be chapters in a book, sections in a scientific article, and paragraphs in an essay. Although segments can be defined differently, they are organised logically to form an entire document. The logical organisation is achieved through linkages between the document subject and the segment subtopics. In this thesis, the first set of contributions are models and algorithms I present for modelling the following document structures:

Hierarchical document structure In writing, people usually try to organise segments around the document subject according to subtopics discussed in the segments. The segment subtopics can be more specific than the subject, which means each segment could have its specificity on topics. In general, they can be taken as variants of the document subject. The organisation of segments in a document according to relations between the document subject and the segment subtopics gives us an hierarchical representation of the document structure. One contribution of this thesis is a new Segmented Topic Model (STM, Chapter 5), which directly models the hierarchical document structure by mapping it to a subject hierarchy that is specific for each individual document. Modelling the hierarchical structure, STM has higher fidelity over existing techniques in terms of per-word predictive accuracy.

Sequential document structure The segment sequence in a document, or the layout of segments, also conveys a sequential document structure. The subtopics of segments are not only linked to the document subject, but also linked sequentially to their adjacent ones, because people often try to make the flow of information among segments logical and smooth. Therefore, segments are not actually exchangeable in a sequential context. Another contribution of this thesis is a Sequential Latent Dirichlet Allocation model (SeqLDA, Chapter 6), a novel variant of Latent Dirichlet Allocation (LDA) [Blei et al., 2003], which makes use of a simple first-order Markov chain to model the sequential structure exhibited by each document. It can effectively discover and visualise patterns of topic evolution in each individual document.

Mixture of hierarchical and sequential document structures It is known that a document can simultaneously exhibit both a hierarchical structure and a sequential structure. The mixture of the two structures gives us a full document structure. Now, topic shifts from one part of the document to another can be allowed, like those in a novel. The contribution on topic modelling is therefore the integration of STM and SeqLDA. I call it an Adaptive Topic Model (AdaTM, Chapter 7), in which a simple Directed Acyclic Graph (DAG) is used to model both the hierarchical and the sequential document structures. It can further explore how each segment adapts topics from either the preceding segment subtopic or the document subject, or even both.

Moreover, to handle the above document structures, I use a non-parametric Bayesian method, called the two-parameter Poisson-Dirichlet process (PDP), to model probabilistic dependencies between document subject and its segment subtopics, and those among subtopics. With respect to the PDP, the second set of contributions (Chapter 3) of this thesis includes

A Collapsed Multiplicity Gibbs Sampler (CMGS) The Chinese restaurant process provides an elegant analogy of incremental sampling for the PDP. In a Chinese restaurant metaphor for the PDP, customers arrive sequentially, each of which chooses a dish by choosing a table. In Gibbs sampling dynamically recording the number of customers sitting at each table could be problematic. I introduce a two-stage Gibbs sampling algorithm based on the table *multiplicity* representation for the PDP [Teh, 2006a; Buntine and Hutter, 2010]. It has been successfully used in STM and SeqLDA.

A Blocked Table Indicator Gibbs Sampler (BTIGS) This is joint work[1] with Changyou Chen and Wray Buntine [Chen et al., 2011]. In the Chinese restaurant metaphor, if a customer does not choose to sit at an occupied table to share a dish with other customers, a new table will be created for this customer. Thus, in the new sampling algorithm, we introduce an auxiliary latent variable, called *table indicator*, to record those customers who have chosen an unoccupied table. I have adapted it for doing posterior inference over a DAG, see AdaTM (Chapter 7).

Notice that algorithms used for doing posterior inference for STM, SeqLDA and AdaTM are good enough to test those models based on experimental results in Chapter 3. It will be worth exploring the above algorithms along with other techniques (*e.g.*, variational inference [Jordan et al., 1999; Blei and Jordan, 2005]) to find more efficient methods.

The researches of this thesis have led to a set of published results as follows:

1. Lan Du, Wray Buntine, and Huidong Jin. A segmented topic model based on the two-parameter Poisson-Dirichlet process. *Machine Learning* 2010. [Du et al., 2010b]

2. Lan Du, Wray Buntine, and Huidong Jin. Sequential latent Dirichlet allocation: Discover underlying topic structures within a document. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, 2010 [Du et al., 2010a];

[1] Each author has equal contribution to this work [Chen et al., 2011].

3. Lan Du, Wray Buntine, Huidong Jin, and Changyou Chen. Sequential latent Dirichlet allocation. *Knowledge and Information Systems*, 2012 [Du et al., 2012b];

4. Lan Du, Wray Buntine, Huidong Jin. Modelling Sequential Text with an Adaptive Topic Model. *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012 [Du et al., 2012a];

5. Wray Buntine, Lan Du, and Petteri Nurmi. Bayesian networks on Dirichlet distributed vectors. In *Proceedings of the Fifth European Workshop on Probabilistic Graphical Models* (PGM-2010), 2010 [Buntine et al., 2010];

6. Changyou Chen, Lan Du, and Wray Buntine. Sampling for the Poisson-Dirichlet process. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Database*, 2011 [Chen et al., 2011].

1.2 Thesis Overview

Figure 1.2 illustrates the dependencies of sections in Chapters 2 to 3 and the subsequent chapters. The rest of the thesis is organised as follows.

Chapter 2: In this chapter, I cover the fundamentals of Dirichlet related nonparametric Bayesian methods, which provide necessary background knowledge for the development of models and algorithms in subsequent chapters. These include the Dirichlet distribution, the Dirichlet process, the Poisson-Dirichlet process, and the compound Poisson-Dirichlet process. In the first section, I review basic definitions and properties of the Dirichlet distribution. Subsequently, I discuss the three processes in detail from three main aspects, *i.e.*, definition, two different ways of construction (*i.e.*, the stick-breaking construction and the Chinese restaurant process representation), and hierarchical models. I put emphasis on the Chinese restaurant representation, because it forms the basis of several Gibbs sampling algorithms that are developed in Chapter 3 and used in Chapters 5 to 7.

Chapter 3: In this chapter, I introduce two new Gibbs sampling algorithms for doing posterior inference for the PDP. One is based on the *multiplicity* representation [Buntine and Hutter, 2010], the other is based on the *table*

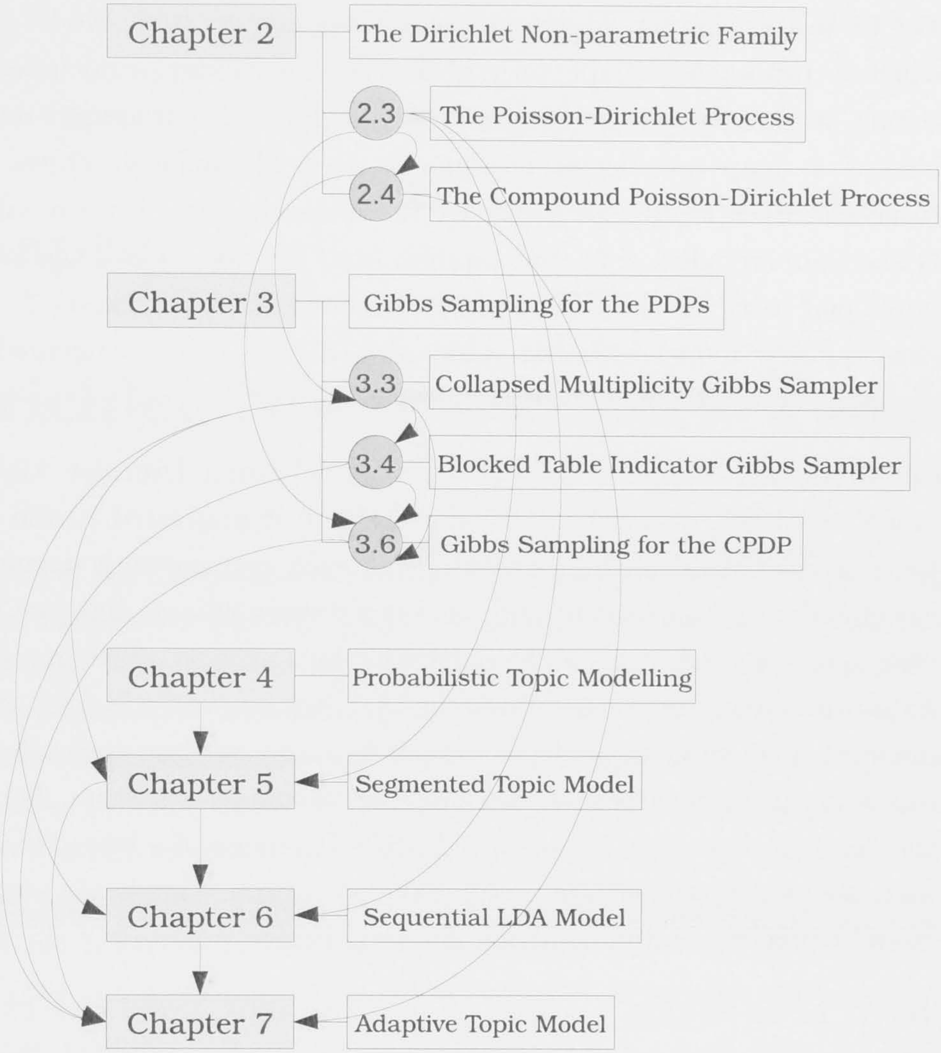

Chapter 2
The Dirichlet Non-parametric Family
2.3
The Poisson-Dirichlet Process
2.4
The Compound Poisson-Dirichlet Process
Chapter 3
Gibbs Sampling for the PDPs
3.3
Collapsed Multiplicity Gibbs Sampler
3.4
Blocked Table Indicator Gibbs Sampler
3.6
Gibbs Sampling for the CPDP
Chapter 4
Probabilistic Topic Modelling
Chapter 5
Segmented Topic Model
Chapter 6
Sequential LDA Model
Chapter 7
Adaptive Topic Model

Figure 1.2: Dependency diagram of chapters and sections

indicator representation [Chen et al., 2011]. I compare the two samplers with Teh's sampling for seating arrangement sampler [Teh, 2006a] in a simple controlled environment of multinomial sampling. The experimental results show that the two new methods converge much faster than Teh's sampler in a simply controlled environment. Thereafter, I also develop Gibbs sampling for the compound Poisson-Dirichlet process by presenting the joint posterior distributions.

Chapter 4: In this chapter, I review probabilistic topic models, especially LDA. I also discuss applications of topic models in various domains, *e.g.*, information retrieval, text analysis and computer vision. Finally, I cover some typical extensions of LDA.

Chapter 5: In this chapter, I introduce a new Segmented Topic Model (STM), which incorporates a simple form of document structure, a document consisting of multiple but exchangeable segments (*e.g.*, paragraphs and sentences). It maps the layout of segments to a hierarchical subject structure. The PDP is used to construct the hierarchy. An effective collapsed Gibbs sampling algorithm that samples from the posterior of the model is developed based on the CMGS algorithm introduced in Chapter 3. I compare the new model with the standard LDA and other segmented topic models on several document collections.

Chapter 6: In this chapter, I present a Sequential latent Dirichlet Allocation model (SeqLDA), a novel extension of LDA. It is motivated by the underlying sequential document structure, *i.e.*, each segment in a document is correlated to its antecedent and subsequent segments via linkages among their topics. Indeed, it maps the sequential document structure to a sequential subject structure, then embeds the PDP in a first-order Markov chain to model the sequential topic dependencies. In such a way, we can explore how topics within a document evolve over the document structure. For doing the posterior inference, I adapt the CMGS algorithm in a hierarchical context. Besides experiments on perplexity comparison, I apply the sequential model to topic evolution analysis of several books.

Chapter 7: In this chapter, I propose an Adaptive Topic Model (AdaTM) that integrates the two models introduced in Chapters 5 and 6. It considers both hierarchical and sequential document structures via a simple DAG structure. I extend the block table indicator Gibbs sampler introduced in Chapter 3 to do the posterior inference over the DAG. Experimental results indicate that AdaTM outperforms STM, SeqLDA and LDA in terms of perplexity, and is able to uncover clear sequential structures in books, such as Herman Melville's "Moby Dick".

Chapter 8 In this chapter, I summarise the key contributions of this thesis and discuss possibilities for future research.

Chapter 2

Dirichlet Non-parametric Family

Hierarchical Bayesian reasoning is fundamental and used throughout the general machine intelligence domain (*e.g.*, text analysis and image processing) to model distributions over observed data. It provides a valuable suite of flexible modelling approaches for high dimensional structured data analysis. Recently, non-parametric methods have become increasingly prominent in the machine learning community. In non-parametric Bayesian methods, the prior and posterior distributions are general *stochastic processes* [Hjort et al., 2010] whose support is a space of distributions. These stochastic processes allow Bayesian inference to be carried out in general infinite dimensional spaces, which can overcome the problem of over-/under-fitting of data encountered by parametric Bayesian methods.

In this chapter, I will focus on the foundation of one of the most important families of non-parametric Bayesian methods, the Dirichlet non-parametric family, which includes:

- the Dirichlet distribution (DD): a conjugate prior for parameters of the multinomial distribution (Section 2.1),

- the Dirichlet process (DP): a probability distribution over distributions (Section 2.2), it extends the DD to other domains,

- the two-parameter Poisson-Dirichlet process (PDP): a two-parameter generalisation of the DP (Section 2.3),

- the compound Poisson-Dirichlet process (CPDP): an extension of the PDP that can be applied to multiple input distributions (Section 2.4).

2.1 Dirichlet Distribution

The Dirichlet distribution [Ferguson, 1973; Antoniak, 1974; Sethuraman, 1994] forms the first step toward understanding the DP/PDP models. It has been widely used in areas such as topic modelling and probabilistic language models [Mackay and Peto, 1995; Steyvers and Griffiths, 2007; Frigyik et al., 2010], where the Dirichlet distribution has been proven to be particularly useful in modelling word distributions. This section will describe the Dirichlet distribution and some of its properties.

The Dirichlet distribution, a multi-parameter generalisation of the Beta distribution, defines a probability distribution on a space of all finite probability vectors, *i.e.*, the sampling result from a Dirichlet distribution is a distribution on some discrete probability space. Formally, the Dirichlet distribution of order k is defined over a $(k-1)$-dimensional probability simplex denoted by $\Delta_k = \{(\theta_1, \theta_2, \ldots, \theta_k) : \sum_{i=1}^{k} \theta_i = 1, \theta_i \geq 0\}$.

Definition 2.1. (*Dirichlet distribution*). Let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_k)$ and $\alpha_i > 0$, for $i = 1, \ldots, k$. A random vector $\boldsymbol{\theta} \in \Delta_k$ is said to be Dirichlet distributed if its probability density function with respect to *Lebesgue measure* is given by

$$p\big((\theta_1, \theta_2, \ldots, \theta_k) \,|\, \boldsymbol{\alpha}\big) = \frac{1}{Beta_k(\boldsymbol{\alpha})} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}, \tag{2.1}$$

and it is denoted as $\boldsymbol{\theta} \sim \mathrm{Dir}(\boldsymbol{\alpha})$. $Beta_k(\boldsymbol{\alpha})$ is a k-dimension Beta function that normalises the Dirichlet, defined as

$$Beta_k(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^{k} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)} \, .$$

The sum $\alpha_0 = \sum_{i=1}^{k} \alpha_i$ serves as a precision parameter that measures the sharpness of $\boldsymbol{\theta}$. It measures how different we expect typical samples $\boldsymbol{\theta}$ to be from the mean $\left(\frac{\alpha_1}{\alpha_0}, \frac{\alpha_2}{\alpha_0}, \ldots, \frac{\alpha_k}{\alpha_0}\right)$. When all the components of $\boldsymbol{\alpha}$ are equal to 1, the Dirichlet distribution reduces to a uniform distribution over the simplex. While they are all greater than 1, the density is concentrated on somewhere in the interior of the simplex. Otherwise, if they are less than one, the density has sharp peaks almost at vertices of the simplex. The support of the Dirichlet distribution is the set of all normalised k-dimensional vectors whose components will be in an interval $(0, 1]$. It means the support does not include vertices or edges.

The Dirichlet distribution is reduced to the Beta distribution when $k = 2$. I now describe some interesting properties of the Dirichlet distribution. More detailed discussions of the Dirichlet distribution can be found in, for example, [Ferguson, 1973; Antoniak, 1974; Sethuraman, 1994; Bernardo and Smith, 1994].

2.1.1 Properties of the Dirichlet

In the general case, the mean vector, covariance, marginal distribution and mode are given as follows.

Property 2.1. *(Mean, Variance, Covariance, marginal, mode). If $\boldsymbol{\theta} \sim Dir(\boldsymbol{\alpha})$ and the precision $\alpha_0 = \sum_{i=1}^{k} \alpha_i$*

$$\begin{aligned}
\mathbb{E}[(\theta_1, \theta_2, \cdots, \theta_k)] &= \left(\frac{\alpha_1}{\alpha_0}, \frac{\alpha_2}{\alpha_0}, \ldots, \frac{\alpha_k}{\alpha_0}\right) \\
\mathbb{V}[\theta_i] &= \frac{\alpha_i(\alpha_0 - \alpha_i)}{\alpha_0^2(1 + \alpha_0)} \\
\mathbb{C}ov[\theta_i, \theta_j] &= \frac{-\alpha_i \alpha_j}{\alpha_0^2(1 + \alpha_0)} \\
(\theta_i, 1 - \theta_i) &\sim Dir(\alpha_i, \alpha_0 - \alpha_i) \\
Mode(\theta_1, \theta_2, \ldots, \theta_k) &= \left(\frac{\alpha_1 - 1}{\alpha_0 - k}, \frac{\alpha_2 - 1}{\alpha_0 - k}, \ldots, \frac{\alpha_k - 1}{\alpha_0 - k}\right)
\end{aligned}$$

Property 2.2. *(Conjugacy) Dirichlet Distribution $\boldsymbol{\theta} \sim Dir(\boldsymbol{\alpha})$ is a conjugate prior of the multinomial $\boldsymbol{n} \,|\, \boldsymbol{\theta} \sim Multi(\boldsymbol{\theta})$.*

Proof. Let a discrete random vector $\boldsymbol{n} = (n_1, n_2, \ldots, n_k)$ with $\sum_{i=1}^{k} n_i = N$, which is multinomial distributed in a k-dimensional space with parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_k)$; and $\boldsymbol{\theta}$ be Dirichlet distributed with parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$. Then, using Bayes rule, the posterior distribution is

$$\begin{aligned}
p(\boldsymbol{\theta} \,|\, \boldsymbol{n}) &\propto p(\boldsymbol{n} \,|\, \boldsymbol{\theta}) p(\boldsymbol{\theta}) \\
&\propto \left(\frac{N!}{n_1!\, n_2! \ldots n_k!} \prod_{i=1}^{k} \theta_i^{n_i}\right) \left(\frac{1}{Beta_k(\boldsymbol{\alpha})} \prod_{i=1}^{k} \theta_i^{\alpha_i - 1}\right) \\
&\propto \prod_{i=1}^{k} \theta_i^{n_i + \alpha_i - 1}
\end{aligned}$$

Hence, $\boldsymbol{\theta} \,|\, \boldsymbol{n} \sim \mathrm{Dir}(\boldsymbol{\alpha} + \boldsymbol{n})$. □

This Dirichlet-Multinomial conjugate property is the key ingredient to compute the conditional posterior distribution in Dirichlet-Multinomial mixture models. It assists in the implementation of efficient Markov Chain Monte Carlo

(MCMC) algorithms. For example, the collapsed Gibbs sampling algorithms in topic models (*e.g.*, Latent Dirichlet Allocation [Griffiths and Steyvers, 2004]) make use of *Dirichlet-Multinomial Conjugacy* to compute the conditional posterior distribution with some latent variables marginalised out, which makes the Gibbs sampling collapsed, instead of sampling the whole latent space. Thereby, a simple approximate inference algorithm can be obtained. The importance of *Dirichlet-Multinomial Conjugacy* will be further observed in the development of structured topic models in Chapters 5, 6 and 7.

In addition to conjugacy, the Dirichlet distribution has a useful fractal-like property, named *aggregation*, that if parts of the sample space are aggregated together, the new partition of the space is still Dirichlet distributed [Ferguson, 1973].

Property 2.3. *(Aggregation) In general, if $I_{1:m}$ is a partition of $\{1, 2, \ldots, k\}$, and $(x_1, x_2, \ldots, x_k) \sim Dir(\alpha_1, \alpha_2, \ldots, \alpha_k)$, then*

$$\left(\sum_{i\in I_1} x_i, \ldots, \sum_{i\in I_m} x_i\right) \sim Dir\left(\sum_{i\in I_1} \alpha_i, \ldots, \sum_{i\in I_m} \alpha_i\right) \tag{2.2}$$

2.1.2 Sampling from the Dirichlet Distribution

The Dirichlet distribution can be constructed in three different ways via the Gamma distribution, the stick-breaking construction, and the Polya Urn scheme respectively. They provide concrete representations of how to generate samples from a Dirichlet distribution.

Dirichlet Distribution through the Gamma Distribution

Ferguson [1973] defined the Dirichlet distribution in a slightly more general way by transforming Gamma-distributed random variables. Generating a Dirichlet distribution from these Gamma random variables has following steps: let $z_1, z_2, \ldots, z_k$ be Gamma-distributed random variables,

1. For $i = 1, 2, \ldots, k$, draw $z_i \sim \mathcal{G}(\alpha_i, 1)$, where $\alpha_i > 0$.
2. For $i = 1, 2, \ldots, k$, $\theta_i = \frac{z_i}{\sum_{i=1}^{k} z_i}$.
3. Then, $(\theta_1, \theta_2, \ldots, \theta_k) \sim \mathrm{Dir}(\alpha_1, \alpha_2, \ldots, \alpha_k)$.

It has been proven that the distribution generated with the steps above is always singular with respect to *Lebesgue measure* in k-dimensional space since

$\sum_{i=1}^{k}\theta_i = 1$ [Ferguson, 1973]. Following this construction, the proof of Property 2.3 is straightforward by using the additive property of the Gamma distribution: if $z_i \sim \mathcal{G}(\alpha_i, 1)$ and $z_j \sim \mathcal{G}(\alpha_j, 1)$, where $i \neq j$, and if z_i and z_j are independent, then $z_i + z_j \sim \mathcal{G}(\alpha_i + \alpha_j, 1)$.

Dirichlet Distribution through the Stick-breaking Construction

The stick-breaking construction is a process of iteratively breaking off pieces of a stick of length one. Random variables drawn from a Dirichlet distribution can be simulated by lengths of the pieces broken off from the stick in a random way, such that the lengths follow a Dirichlet distribution [Sethuraman, 1994; Ishwaran and James, 2001]. This uses the marginalisation property of the Dirichlet distribution, see Property 2.1.

Let $V_1, V_2, \ldots, V_k$ be intermediate random variables drawn from a Beta distribution, *i.e.*, $V_i \sim \text{Beta}\left(\alpha_i, \sum_{j=i+1}^{k}\alpha_j\right)$. A Dirichlet distribution can be constructed via the stick-breaking construction with following steps:

1. Draw $V_1 \sim \text{Beta}\left(\alpha_1, \sum_{j=2}^{k}\alpha_j\right)$, set $\theta_1 = V_1$. The remaining piece has length $1 - V_1$.

2. For $2 \leq i \leq k-1$, draw $V_i \sim \text{Beta}\left(\alpha_i, \sum_{j=i+1}^{k}\alpha_j\right)$, and set $\theta_i = V_i\prod_{j=1}^{i-1}(1 - V_j)$.

3. The length of remaining piece $\prod_{j=1}^{k-1}(1 - V_j)$ is θ_k.

Finally, the derived vector of random variables $(\theta_1, \theta_2, \ldots, \theta_k)$ is Dirichlet distributed with parameters $(\alpha_1, \alpha_2, \ldots, \alpha_k)$.

Dirichlet Distribution through the Urn Scheme

The Dirichlet distribution can be constructed from the Urn model [Johnson and Kotz, 1977]. Blackwell and Macqueen [1973] have shown that the distribution of colors in an urn after n draws converges as $n \to \infty$ to a Dirichlet distribution in a finite space (*i.e.*, the number of colors is finite). It is known as the Polya Urn scheme.

To generate a Dirichlet distribution from the Polya Urn scheme with parameters $(\alpha_1, \alpha_1 \ldots, \alpha_k)$, we start with an urn with $\alpha_0 = \sum_i^k \alpha_i$ balls of which α_i balls[1] are of color i, $1 \leq i \leq k$. At each step, we draw a ball uniformly at random from

[1] In general, α_i is not necessarily an integer, so we might have a rational number of balls of each color in the urn initially.

the urn, and then place it back to the urn along with another ball of the same color. After $n \to \infty$ steps, the proportions of balls of each color converge to a limiting discrete distribution, which is shown to be a Dirichlet distribution.

Mathematically, let X_i be a color random variable, a Dirichlet distribution can be constructed via the Polya Urn scheme as:

1. For the first draw, a ball with color i is drawn with probability
$$p(X_1 = i) = \frac{\alpha_i}{\sum_{i'=1}^{k} \alpha_{i'}}.$$
2. Draw the $(n+1)^{th}$ draw, a ball with color i is drawn with probability
$$p(X_{n+1} = i \,|\, X_1, \ldots, X_n) = \frac{\alpha_i + \sum_{j=1}^{n} 1_{X_j = i}}{\sum_{i'=1}^{k} \alpha_{i'} + n}.$$

2.2 Dirichlet Process

This section provides a brief overview of the Dirichlet process (DP) mainly based on the work of Ferguson [1973]; Antoniak [1974]; Teh [2010]; Teh et al. [2006]. A high-level tutorial can be found in [Jordan, 2005; Teh, 2007]. Along with the basic definition, constructions of a Dirichlet process from the stick-breaking process and the *Chinese restaurant process* (CRP) [Aldous, 1985] will be presented. In addition, I will discuss some hierarchical extensions of the Dirichlet process.

In probability theory, a DP is a stochastic process that can be taken as a probability distribution over distributions. It is, as a non-parametric Bayesian method, most useful in models in which each mixture component is a discrete random variable of unknown cardinality. A canonical example of such a model is the infinite mixture model (*i.e.*, the DP mixture model), where the discrete random variables may indicate clusters.

Let $(\mathcal{X}, \mathcal{B})$ be a measurable space, for a random probability distribution G to be distributed according to a DP, its marginal distributions have to be Dirichlet distributions. Ferguson [1973] gave a formal definition of the DP as follows.

Definition 2.2. (*Dirichlet Process*). Let H be a random measure on $(\mathcal{X}, \mathcal{B})$ and α be positive real number. We say a random probability measure G on $(\mathcal{X}, \mathcal{B})$ is a Dirichlet process with a *base measure* H and a *concentration parameter* α, *i.e.* $G \sim \mathrm{DP}(\alpha, H)$, if for any finite measurable partition $(B_1, B_2, \ldots, B_k)$ of $\mathcal{X}$, the random vector $(G(B_1), G(B_2), \ldots, G(B_k))$ is Dirichlet distributed with parameter $(\alpha H(B_1), \alpha H(B_2), \ldots, \alpha H(B_k))$:
$$(G(B_1), G(B_2), \ldots, G(B_k)) \sim \mathrm{Dir}(\alpha H(B_1), \alpha H(B_2), \ldots, \alpha H(B_k)) \,.$$

The support of G is the same as H. The existence of the DP is guaranteed by either the *Kolmogorov's consistency theorem* or the *de Finetti's theorem*. One important property of the DP is that distributions drawn from a DP are discrete with probability one [Ferguson, 1973]. That means the previously drawn values have strictly positive probability of being redrawn again, which can be proven in the two construction methods of the DP in the next section.

Corollary 2.1. *According to Definition 2.2, if H is a probability vector over a finite space, then the following holds*

$$DP(\alpha,\ Discrete(H)) = Dir(\alpha H)\ .$$

Thus, the DP is an extension of a Dirichlet distribution.

We can draw a sequence of independently and identically distributed (*i.i.d.*) random variables from G. Theoretically, the sequence can be infinite. Then after marginalising out G, these random variables follow a Blackwell-Macqueen distribution [Blackwell and Macqueen, 1973], also known as the CRP. I will show that a DP can be constructed via the CRP in next section.

Property 2.4. *(Mean, Variance, and Covariance) If $G \sim DP(\alpha, H)$, for any measurable set $B \in \mathcal{B}$,*

$$\begin{aligned}
\mathbb{E}(G(B)) &= H(B) \\
\mathbb{V}(G(B)) &= \frac{H(B)(1-H(B))}{\alpha+1} \\
\mathbb{C}ov(G(B), G(B')) &= -\frac{H(B)H(B')}{\alpha+1} \quad s.t.\ B' \cap B = \emptyset
\end{aligned}$$

The base measure H and the concentration parameter α play important roles in the construction of a DP. Specifically, the base measure is the mean of the DP, and the concentration parameter α, also known as a precision parameter [Rodríguez et al., 2008], controls the variance between G and H. Large α means the DP concentrates more mass around the mean. When the base measure is non-atomic (or continuous), $H(X) = 0$ for all $X \sim H$, thus samples from H are almost surely distinct, *e.g.*, a probability distribution such as Gaussian. With respect to discrete applications that are common in computer science and intelligent systems, the non-atomicity of the base measure does not always hold. Thus, when the base measure is atomic, $H(X) > 0$ for all samples $X \sim H$.

The posterior distribution of the DP is still a DP with updated concentration parameter and base measure over partitions of $\mathcal{X}$. Let $x_1, x_2, \ldots, x_n$ be a sequence of *i.i.d.* draws from G, and x_i take values on $\mathcal{X}$. With *Dirichlet-Multinomial conjugacy* (Property 2.2) and some algebra, we can yield the posterior of the DP as

$$G \,|\, \boldsymbol{x}_{1:n} \sim \mathrm{DP}\left(\alpha + n, \frac{\alpha}{\alpha+n}H(\cdot) + \frac{n}{\alpha+n}\frac{\sum_{i=1}^{n}\delta_{x_i}(\cdot)}{n}\right) , \tag{2.3}$$

where $\delta_{x_i}(\cdot)$ is the point mass located at x_i. The updated concentration parameter is $\alpha + n$, and the base measure is changed to $\frac{\alpha H + \sum_{i=1}^{n}\delta_{x_i}(\cdot)}{\alpha+n}$. The predictive distribution $x_{n+1} \,|\, \boldsymbol{x}_{1:n}$ is the updated base measure of the posterior of DP. I will show the derivation of predictive probability in the CRP interpretation for the DP in Section 2.2.1.

2.2.1 Construction of the Dirichlet Process

There are two well known ways of drawing samples from a Dirichlet process. One is the stick-breaking construction for the DP, where two sequences of *i.i.d.* random variables need to be generated. It can be simulated by randomly breaking a unit stick into pieces with different weights. The other is the CRP interpretation according to the Polya Urn scheme [Blackwell and Macqueen, 1973], which gives us a straightforward way of generating posterior samples from a random distribution given observations.

Dirichlet Process via the Stick-breaking Construction

The stick-breaking construction [Sethuraman, 1994] is a concrete representation of draws from G, where $G \sim \mathrm{DP}(\alpha, H)$. It is a weighted sum of the point masses at atoms. The process of stick-breaking also provides a straightforward proof of the existence of DPs [Teh, 2010].

Theorem 2.2. *(The stick-breaking construction for the DP) Let $(V_k)_{k=1}^{\infty}$ and $(X_k^*)_{k=1}^{\infty}$ be independent sequences of i.i.d. random variables, the stick-breaking construction of the DP has the following form:*

$$\begin{aligned} V_k \,|\, \alpha, H &\sim\ Beta(1, \alpha) & X_k^* \,|\, \alpha, H &\sim\ H \\ p_k &= V_k \prod_{j=1}^{k-1}(1 - V_j) & G &= \sum_{k=1}^{\infty} p_k \delta_{X_k^*}(\cdot) \ , \end{aligned}$$

where $\delta_{X_k^*}(\cdot)$ is a discrete probability measure that concentrates at X_k^*, and $\sum_{k=1}^{\infty} p_k = 1$ with probability one.

Metaphorically, the process of generating a sequence of p_k can be understood as iteratively breaking off pieces with random lengths from a stick, subject to the length of the initial stick is one. Similar to the stick-breaking construction for the Dirichlet distribution in Section 2.1.2, the stick-breaking for the DP goes as follows:

1. Take a stick of length one and randomly break it into two parts with proportions V_1 and $1 - V_1$. The first broken stick has length $p_1 = V_1$.

2. Then take the remaining part, of length $1 - V_1$ and apply the same process to randomly break into proportions V_2 and $1 - V_2$. This second broken stick is the first part, of length $p_2 = (1 - V_1)V_2$.

3. Again, we take the remaining part, of length $(1 - V_1)(1 - V_2)$, and apply the same process repeatedly. So the length of k^{th} stick becomes $p_k = V_k \prod_{j=1}^{k-1}(1 - V_j)$ for $k > 2$.

Ishwaran and James [2001] presented a truncated stick-breaking construction, in which the number of sticks is set to some truncation level K that can be determined by the moments of the random breaking weights, and the $K+1$, $K+2, \ldots$ sticks are discarded. The length of last stick is set to $p_K = 1 - \sum_{j=1}^{K-1} p_j$. In most real world scenarios, the DP mixture models are in an effectively finite state space, such as those adapted for natural language processing, language modelling, and computer vision.

Dirichlet Process via the Chinese Restaurant Process

The *Chinese restaurant process* (CRP), also known as the Blackwell-Macqueen Urn scheme, asymptotically produces a partition of integers [Blackwell and Macqueen, 1973]. It is shown that samples from a Dirichlet process are discrete and exhibit a clustering property [Teh et al., 2006].

The CRP is an elegant analogy of incremental sampling for the DP. It refers to draws from G, instead of referring to G directly, which means it is easy to describe the distribution by specifying how to draw samples from it. Let $\{X_1^*, X_2^*, \ldots, X_K^*\}$ be a set of distinct values drawn from the base measure H (Note H is non-atomic, and K can be infinitely large, *i.e.*, $K \to \infty$). Those distinct values are taken on by random samples $x_1, x_2, \ldots, x_n$ that are *i.i.d.* given G, and $n_k^* = \sum_{i=1}^{n} 1_{x_i = X_k^*}$. With G marginalised out and given the first n observations, the posterior distribution (or the predictive distribution) of the $(n+1)^{th}$ random variable has the following

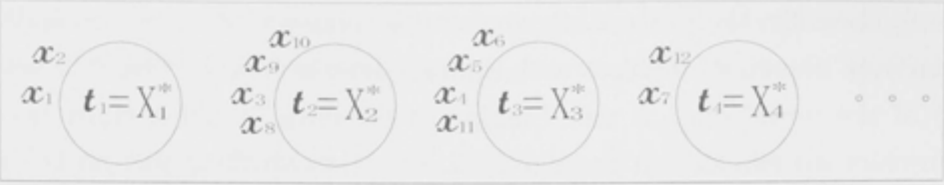


Figure 2.1: A CRP representation for a DP with a continuous base distribution that makes the dish served at each table to be distinct. Circles are tables, each k^{th} table has a label t_k. x_ns are customers, and X_k^*'s are dishes. $t_k = X_k^*$ indicates the k^{th} table serves X_k^*.

form

$$x_{n+1} \,|\, x_1, x_2, \ldots, x_n, \alpha, H \sim \sum_{k=1}^{K} \frac{n_k^*}{n+\alpha} \delta_{X_k^*(\cdot)} + \frac{\alpha}{n+\alpha} H(\cdot) \;. \tag{2.4}$$

We can interpret the posterior distribution of the DP in terms of a Chinese restaurant metaphor, as shown in Figure 2.1. Consider a Chinese restaurant with an infinite number of tables, each of which has infinite seating capacity. Each table t_k serves a dish, *i.e.*, a distinct value X_k^*. A sequence of customers, labeled by $x_1, x_2, \ldots, x_n$, arrive in the restaurant. The first customer sits at the first table; the $(n+1)^{th}$ customer can choose either an occupied table or opening a new table with following probabilities,

$$\begin{aligned} p(k^{th} \text{ occupied table}) &\propto \frac{n_k^*}{n+\alpha} \\ p(\text{next new table}) &\propto \frac{\alpha}{n+\alpha} \;. \end{aligned}$$

In the sense of CRP, the concentration parameter α controls how often a newly arrived customer opens a new table. The larger α is, the more tables will be activated, which further corresponds to the smaller the variance between G and H. Another important property of the DP is the reinforcement effect: the more customers sit at the k^{th} table, the more likely the k^{th} table will be chosen by subsequent customers.

2.2.2 The Hierarchical Dirichlet Process

As a widely used non-parametric Bayesian method for discrete random distributions, the DP has been extended in different ways to deal with dependencies that exists in various data, such as grouped data, streamed data and time-stamped data. For example, MacEachern [1999] introduced the dependent DP (DDP) to handle dependencies in a collection of distributions, which is a quite general framework; and Lin et al. [2010] gave a new Poisson processes based construction for the DDP. One special case of the DDP general framework is the hierarchical DP (HDP) [Teh et al., 2006]. In the HDP, multiple group specific distributions are drawn from a common DP whose base distribution is in turn drawn from another DP. Some other extensions include the nested DP [Rodríguez et al., 2008] and spatial DP [Duan et al., 2007]. In this section, I give a brief overview of the more widely used HDP model.

Motivated by sharing atoms across different data groups, Teh et al. [2006] introduced the HDP, in which the base measure G_0 of a Dirichlet process for each data group is drawn from another Dirichlet process with base measure H. In such a way, G_0 is forced to be discrete, and distinct values drawn from the top level base measure H are shared with different weights among draws from the low level Dirichlet process. The support of draws from the HDP is the same as that of H. The precise definition of the HDP is given in [Teh et al., 2006], and a further description can be found in [Teh and Jordan, 2010].

Definition 2.3. (*Hierarchical Dirichlet Process* [Teh et al., 2006]) Let γ and α be concentration parameters, H is a baseline measure on a measurable space $(\mathcal{X}, \mathcal{B})$, G_0 is the intermediate base measure, the HDP is defined as

$$\begin{aligned} G_0 \mid \gamma,\ H &\sim \mathrm{DP}(\gamma,\ H) \\ G_i \mid \alpha, G_0 &\sim \mathrm{DP}(\alpha,\ G_0),\ \text{for } i \in \{1, \ldots, I\}, \end{aligned}$$

where $\{1,\ 2,\ \ldots,\ I\}$ is an index set which indexes a collection of Dirichlet processes, $\{G_1,\ G_2,\ \ldots,\ G_I\}$. Each G_i corresponds to a data group and is defined on $(\mathcal{X}, \mathcal{B})$.

Obviously, the HDP is also a probability distribution over a set of random distributions over a measurable space $(\mathcal{X}, \mathcal{B})$, and shares similar properties to the DP. It links a number of probability distributions by letting them share the same base measure. Teh et al. [2006] presented the stick-breaking construction and a Chinese restaurant representation for the HDP, analogous to the DP. Here, I describe the latter, named Chinese restaurant franchise (CRF) by Teh et al. [2006], which not only elaborates the combinatorial structure of the HDP in a way of incremental sampling, but also provides the ground of the subsequent discussion of the Poisson-Dirichlet process and various Gibbs sampling algorithms.

The CRF is an analog of the Chinese restaurant process for the HDP with all the G_i and their base measure G_0 marginalised out, *i.e.*, the marginalisation of the HDP. Actually, it extends the CRP representation for the DP to handling multiple Chinese restaurants that are conditionally independent to each other.

Metaphorically, suppose there is a global menu with dishes (*i.e.*, distinct values drawn from H), and at each restaurant, each table is associated with a dish from the menu. A customer chooses a dish by choosing a table. More specifically, arriving in a restaurant that corresponds to G_i, a customer can either choose to sit at an occupied table to share the dish with other customers or to open a new table. If the customer sits at a new table, a dish must be ordered from the global menu. In the CRF, choosing a dish from the global menu is equivalent to sending the new table as a proxy customer [Mochihashi and Sumita, 2008] to the corresponding parent restaurant (G_0), and then repeating the Chinese restaurant process analogy of the DP to choose the dish from H. In this sense, the HDP is a hierarchical CRP. The precise nature of sharing atoms across data groups induced by the HDP is mimicked by sharing dishes among multiple restaurants.

Mathematically, let $\boldsymbol{x}_i = (x_{i,1}, x_{i,2}, \dots, x_{i,J_i})$ be a sequence of J_i customers in a restaurant i, $x_{i,j}$ is one entry in $\boldsymbol{x}_i$ that are random variables distributed according to G_i. Dishes in the global menu are denoted by $\{X_1^*, X_2^*, \dots, X_K^*\}$ that are *i.i.d.* draws from H. Furthermore, let $\boldsymbol{y}_i = (y_{i,1}, y_{i,2}, \dots, y_{i,T_i})$ be the dishes served at tables in the i^{th} restaurant and be distributed according to the intermediate base probability measure G_0, where T_i is the number of currently occupied tables in restaurant i, and $y_{i,t} \in \{X_1^*, X_2^*, \dots, X_K^*\}$. Clearly, all the customers $\boldsymbol{x}_i$ take the values on $\{X_1^*, X_2^*, \dots, X_K^*\}$ via the intermediate random variables $\boldsymbol{y}_i$. Let $n_{i,t}^*$ be the number of customers sitting at table t in restaurant i, and n'_k be the total number of tables in all the restaurants serving dish X_k^*, $n'_k = \sum_{i=1}^{I} \sum_{t=1}^{T_i} 1_{y_{i,t}=X_k^*}$. In this setup, restaurants correspond to DPs associating with data groups, and customers are factors.

Now, the marginal probabilities of the HDP are computed by integrating out the random distributions G_i and G_0 recursively with the CRP. First, integrating out G_i yields the conditional probability of x_{i,J_i+1} given by the posterior (2.4),

$$x_{i,J_i+1} \,|\, \boldsymbol{x}_i,\ \alpha,\ G_0 \sim \sum_{t=1}^{T_i} \frac{n_{i,t}^*}{\alpha + J_i} \delta_{y_{i,t}}(\cdot) + \frac{\alpha}{\alpha + J_i} G_0(\cdot)\ . \tag{2.5}$$

The probability of drawing a random variable from the above mixture is according to mixture proportions on the right-hand side of the formula. Since all $\boldsymbol{y}_{1:I}$ are *i.i.d.* according to G_0, which is again Dirichlet process distributed, we can readily marginalise out G_0 by using the same posterior (2.4). Thus, the conditional probability for y_{i,T_i+1} is

$$y_{i,T_i+1} \,|\, \boldsymbol{y}_{1:I},\ \gamma,\ H \sim \sum_{k=1}^{K} \frac{n'_k}{\gamma + \sum_{k=1}^{K} n'_k} \delta_{X_k^*}(\cdot) + \frac{\gamma}{\gamma + \sum_{k=1}^{K} n'_k} H(\cdot) \,. \tag{2.6}$$

The posterior structure of the HDP can easily be obtained in regard to the posterior of the DP, Equation (2.3). Thus, given $\boldsymbol{x}_{1:I}$ and $\boldsymbol{y}_{1:I}$, the posterior distributions are stipulated respectively by

$$G_0 \,|\, \boldsymbol{y}_{1:I}, \gamma, H \sim \mathrm{DP}\left(\gamma + \sum_{k=1}^{K} n'_k, \frac{\gamma H(\cdot) + \sum_{k=1}^{K} n'_k \delta_{X_k^*}(\cdot)}{\gamma + \sum_{k=1}^{K} n'_k}\right) ,$$

$$G_i \,|\, \boldsymbol{x}_{1:I}, \alpha, G_0 \sim \mathrm{DP}\left(\alpha + \sum_{t=1}^{T_i} n^*_{i,t}, \frac{\alpha G_0(\cdot) + \sum_{k=1}^{K} \sum_{t: y_{i,t} = X_k^*} n^*_{i,t} \delta_{X_k^*(\cdot)}}{\alpha + \sum_{t=1}^{T_i} n^*_{i,t}}\right) .$$

Clearly, the recursive construction of the HDP can be generalised to arbitrary hierarchical structures by recursively putting DPs together [Teh and Jordan, 2010]. However, these hierarchies should be tree structures, since the DP only allows one base measure. In Section 2.4, I will show how to extend the DP to handling multiple base measures.

2.2.3 Variants of the HDP

The HDP allows the sharing of atoms among multiple groups of data. The underlying assumption is that these data groups are exchangeable. The exchangeability is possessed by the *i.i.d.* draws (*i.e.*, $\boldsymbol{G}_{1:I}$) from the same base measure. However, there are many applications for which the exchangeability assumption is not suitable and needs to be removed to further incorporate other dependencies for complex data structures, such as the temporal structure (*e.g.*, time stamped documents and music). These have motivated various extensions of the HDP that have been studied in the Bayesian non-parametric literature.

A way to extend the HDP is to introduce dependence among realisations of independent HDPs, such as the dynamic HDP (DHDP) proposed by Ren et al. [2008], with more details in [Ren et al., 2010]. To consider the statistical dependency among the time-evolving data, the DHDP uses a Hidden Markov Model (HMM) to incorporate time-evolving parameters, such as time stamps, to further chain a set of HDPs in a linear way, *i.e.*, $G_j = (1 - w_{j-1})G_{j-1} + w_{j-1}H_{j-1}$. In the DHDP setting, a set of innovation distributions, $\{H_1, H_2, \cdots, H_{J-1}\}$ and G_1 are draws from the same HDP, as shown in Figure 2.2. We can see that the probability distribution G_j at time stamp j is indeed a weighted sum of the probability

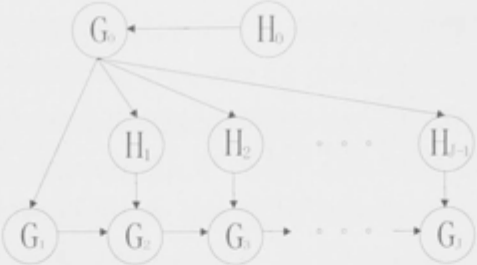


Figure 2.2: The dynamic HDP model

distribution G_{j-1} and the innovation distribution H_{j-1} that are generated at the previous time stamp $j-1$. In this way, G_j can be modified from G_{j-1} by adding an new innovation distribution H_{j-1}. The probability of innovation is controlled by the weight w_{j-1}. Furthermore, since G_1 and all the Hs are drawn from the same HDP, atoms are shared across sequential datasets that are no longer exchangeable. Therefore, the evolution is done by changing the mixture weights associating with atoms sequentially along the time line. A simplified version of the DHDP [Pruteanu-Malinici et al., 2010] has been applied to topic modelling to study the change of topic mixture weights over time. In this simplified version, probability distributions drawn from a HDP are approximated by those drawn from a Dirichlet distribution, see Corollary 2.1.

Unlike the DHDP, Zhang et al. [2010] have extended the HDP to a five-level hierarchy to explore the cluster evolution patterns over time and cross corpora. This model is called an evolutionary HDP (EvoHDP). It uses a coupled Markov chain to link multiple HDPs through their intermediate base distributions, instead of linearly combining a set of probability distributions drawn from a HDP. Specifically, let G_i^j and G_i^{j-1} indicate probability distributions associated with a corpus i at time j and $j-1$ respectively, G_0^j be a global base distribution at time j for all the corpora. G_i^j is generated as $G_i^j \sim \mathrm{DP}\left(\alpha_i^j, w_i^j G_i^{j-1} + (1-w_i^j)G_0^j\right)$, where w_i^j is the mixture weight for the corpus i at time j. Similarly, let G be an overall base distribution drawn from a DP with base distribution H. The global base distribution G_0^j is generated as $G_0^j \sim \mathrm{DP}\left(\alpha_j', w_j' G_0^{j-1} + (1-w_j')G\right)$, where w_j' is the mixture weight for G_0^j. Clearly, the base distribution for drawing either G_i^j or G_0^j is a weighted sum of two distributions. The authors use two chains on G_i^j's and G_0^j's to model patterns of cluster evolution within each individual corpus and across corpora respectively. I will show that their way of constructing the two chains can be taken as a special variant of the compound Poisson-Dirichlet process introduced in Section 2.4. Both DHDP and EvoHDP are two complex models, that are different from models developed in Chapters 5 to 7 in terms of data being modelled and modelling objectives. They may not be suitable for modelling document structures.

These and some other variants of the HDP have the same characteristic that the shared atoms are fixed across data groups, which are either temporally streamed or have other kinds of dependencies, and only the mixture weights are changed in a way according to different data structures. Can atoms themselves change along the data structure without violating the underlying dependencies? Ahmed and Xing [2010] introduced a new dynamic HDP where not only the mixture weights change dynamically, but also the atoms can either retain, die out or emerge over time. This dynamic HDP adapts the recurrent CRP, proposed in [Ahmed and Xing, 2008], which uses a time-decaying kernel to control the life span of atoms over time. Atoms at the current time period are dependent on those at the previous Δ time periods, which enhances the statistical similarity between adjacent time slices.

Replacing random atoms in the DP with random probability distributions, Rodŕiguez et al. [2008] developed the nested Dirichlet process (NDP) to deal with multilevel clustering problems in a nested setting. Under the Chinese restaurant metaphor, the NDP clusters customers within each restaurant and also clusters restaurants at the same time. While clustering the customers within each restaurant, the NDP can borrow the statistical information obtained from the clustering in other restaurants. A distribution drawn from a NDP can be written as $G_j \sim Q$, and $Q \equiv DP(\alpha, DP(\beta, H))$. The stick-breaking construction for the NDP [Rodŕiguez et al., 2008] is

$$\begin{array}{rlrl}
V'_{l,k} \mid \beta & \sim Beta(1, \beta) & X^*_{l,k} \mid H & \sim H(\cdot) \\
\rho^*_{l,k} & = V'_{l,k} \prod_{s=1}^{l-1} (1 - V'_{s,k}) & G^*_k & \equiv \sum_{l=1}^{\infty} \rho^*_{lk} \delta_{X^*_{l,k}}(\cdot) \\
V_k \mid \alpha & \sim Beta(1, \alpha) & \pi^*_k & = V_k \prod_{s=1}^{k-1} (1 - V_s)
\end{array}$$

$$G_j \sim Q \equiv \sum_{k=1}^{\infty} \pi^*_k \delta_{G^*_k}(\cdot) \ .$$

According to this construction for the NDP and that for the HDP [see Teh et al., 2006, Section 4.1], the difference between dependencies induced by the HDP and the DNP are straightforward, even though both of them allow hierarchical data structures. Changing atoms to random probability distributions may provide more flexibility than the HDP to cluster observations together with co-clustering the distributions.

2.2.4 Dirichlet Process Mixture Models

The DP related processes cannot be used to model data directly because the probability distributions drawn from a DP are discrete. Instead, they are more naturally used as a prior on top of hierarchical models, which yields the Dirichlet mixture model, DPM [Antoniak, 1974].

Let w_i be an observation with a distribution $F(\theta_i)$ given factor θ_i that is *i.i.d.* drawn from a random probability measure G. Given θ_i, the observations are conditionally independent to each other. If G is Dirichlet process distributed, we can then derive the DPM as

$$\begin{aligned} w_i &\sim F(\theta_i) && \text{for } i = 1, 2, \dots, n \\ \theta_i &\sim G && \text{for } i = 1, 2, \dots, n \\ G &\sim \mathrm{DP}(\alpha, H) \, . \end{aligned}$$

With respect to *Dirichlet-multinomial conjugacy*, $F(\cdot)$ is set to be a multinomial distribution in many language and image related applications, for instance, the probabilistic topic models, the n-gram model, image processing, *etc*. Similarly, the HDP mixture models can be represented as

$$\begin{aligned} w_{i,j} &\sim F(\theta_{i,j}) && \text{for } i = 1, 2, \dots, I;\ j = 1, 2, \dots, J_i \\ \theta_{i,j} &\sim G_i && \text{for } i = 1, 2, \dots, I;\ j = 1, 2, \dots, J_i \\ G_i &\sim \mathrm{DP}(\alpha, G_0) && \text{for } i = 1, 2, \dots, I \\ G_0 &\sim \mathrm{DP}(\gamma, H) \, . \end{aligned}$$

2.3 Poisson-Dirichlet Process

The *two-parameter Poisson-Dirichlet process* (PDP), also known as the Pitman-Yor process (PYP) [Ishwaran and James, 2001], is a two-parameter generalisation of the *Dirichlet process*. Similar to the DP, it is a probability distribution over distributions over a measurable space $(\mathcal{X}, \mathcal{B})$, and parameterised with a *discount parameter* $0 \leq a < 1$, a *concentration parameter* $b > -a$, and a random base measure H over $\mathcal{X}$, *i.e.*, $\mathrm{PDP}(a, b, H)$. We can write $G \sim \mathrm{PDP}(a, b, H)$, if a probability distribution G is PDP distributed.

As in the case of a DP, the most important application of the PDP is as a non-parametric prior for parameters of mixture models. For example, the PDP and its hierarchical extensions provide useful machinery for improving the standard topic model [Blei et al., 2003; Buntine and Jakulin, 2006; Sato and Nakagawa, 2010], the n-gram model [Teh, 2006a,b] and models of grammar [Johnson et al., 2007; Wallach et al., 2008]. By simply replacing the DP with the PDP, the PDP mixture model can be derived as follows:

$$\begin{aligned} w_i &\sim F(\theta_i) && \text{for } i = 1, 2, \ldots, n \\ \theta_i &\sim G && \text{for } i = 1, 2, \ldots, n \\ G &\sim \mathrm{PDP}(a,\ b,\ H)\ , \end{aligned}$$

where w_i indicates observations, θ_i indicates factors *i.i.d.* drawn the PDP, and $F(\theta_i)$ denotes the factor specific distribution of the observations. If the factor is given, the observations are conditionally independent. Similar to the DP, the PDP is a device for introducing infinite mixture models and for hierarchical Bayesian modelling of discrete probability distributions.

In this section I will give a brief introduction to the PDP, and discuss analogs of the three perspectives presented in Section (2.2) for the DP, *i.e.*, the stick-breaking construction, the CRP representation, and the hierarchical PDP. For more in depth discussion, please refer to Pitman and Yor [1997]; Ishwaran and James [2001]; Buntine and Hutter [2010]. A high-level tutorial from a machine learning perspective can be found in Jordan [2005] and Rodŕiguez [2011].

2.3.1 Poisson-Dirichlet Distribution

Similar to the DP, the basic form of the PDP has as input a random base measure H on a measurable space $(\mathcal{X}, \mathcal{B})$, and yields a discrete distribution on a finite or countably infinite subset of $\mathcal{X}$,

$$\sum_{k=1}^{\infty} p_k \delta_{X_k^*}(\cdot)\ , \tag{2.7}$$

where $\boldsymbol{p} = (p_1, p_2, \ldots)$ is a probability vector so $0 \leq p_k \leq 1$ and $\sum_{k=1}^{\infty} p_k = 1$. Also, $\delta_{X_k^*}(\cdot)$ is a probability mass concentrated at X_k^*. The values $X_k^* \in \mathcal{X}$ are *i.i.d.* according to H, which is referred to as the *base measure*. As discussed in Section 2.2, the *base measure* can be either continuous or discrete. The probability vector $\boldsymbol{p}$ follows a two parameter Poisson-Dirichlet distribution [Pitman and Yor, 1997] given in Definition 2.4.

Definition 2.4. (*Poisson-Dirichlet distribution*) For $0 \leq a < 1$ and $b > -a$, suppose that a probability distribution $P_{a,b}$ governs independent random variables V_k such that V_k has a Beta distribution. Let

$$\begin{aligned} V_k \mid a, b &\sim \text{Beta}(1-a, b+ka) \\ p_k &= V_k \prod_{j=1}^{k-1} (1 - V_j) \qquad \text{for } k = 1, 2, \ldots, \infty, \end{aligned}$$

yielding $\boldsymbol{p} = (p_1, p_2, \ldots)$. Define the *Poisson-Dirichlet distribution* with parameters a, b, abbreviated PDD(a, b) to be the $P_{a,b}$ distribution of $\boldsymbol{p}$.

Note Definition 2.4 assumes a particular ordering of the entries in $\boldsymbol{p}$, but when used in Equation (2.7) any order is lost so this does not matter.

2.3.2 PDP via the Stick-breaking Construction

The stick-breaking construction for the PDP can directly be derived by extending the Poisson-Dirichlet distribution with Equation (2.7).

Theorem 2.3. *(The stick-breaking construction for the PDP) Let V_k be a Beta distributed random variable and p_k be the stick-breaking weight, a probability distribution G drawn from a PDP can be derived by the following constrution*

$$\begin{aligned} V_k \mid a, b &\sim Beta(1-a, b+ka) \\ X_k^* \mid H &\sim H \\ p_k &= V_k \prod_{j=1}^{k-1} (1 - V_j) \\ G &= \sum_{k}^{\infty} p_k \delta_{X_k^*}(\cdot) \ , \ k = 1, 2, \ldots, \infty \end{aligned}$$

It is easy to observe that the PDP stick-breaking construction reduces to the DP stick-breaking construction, see Theorem 2.2, when the discount parameter a is equal to 0. The simulation of iteratively breaking off pieces with random lengths from a stick can be found in Section 2.2.1. More discussion about the PDP stick-breaking construction can also be found in [Ishwaran and James, 2001; Teh and Jordan, 2010; Buntine and Hutter, 2010].

2.3.3 PDP via the Chinese Restaurant Process

Another important approach to construct a PDP is to use the CRP metaphor, a particular interpretation for a marginalised version of the PDP. The CRP also gives an analogy of incremental sampling from the posterior of the PDP. Suppose a sequence of data have been sampled from a random distribution $G \sim \text{PDP}(a, b, H)$. Let the sampled data be $x_1, x_2, ..., x_N$, then what is the conditional distribution of x_{N+1} after marginalising out G? While the base distribution is non-atomic or continuous (the probability of repeated draws is effectively zero), the conditional distribution is [Ishwaran and James, 2001]

$$p(x_{N+1} \,|\, x_1, x_2, ..., x_N, a, b, H) \;=\; \sum_{k=1}^{K} \frac{n_k^* - a}{N+b} \delta_{X_k^*}(\cdot) + \frac{Ka+b}{N+b} H(\cdot) \,, \qquad (2.8)$$

where K is the distinct number of values in $x_1, x_2, \ldots, x_N$ ordered as $X_1^*, X_2^*, ..., X_K^*$ (*i.e.*, draws from H) with respective counts $n_1^*, n_2^*, \ldots, n_K^*$, and theoretically K can be infinitely large. These are modelled with the notion of *tables* in a Chinese restaurant in the CRP terminology, the k-th table has n_k^* *customers seated* and they are having *dish* X_k^*.

When the base distribution is discrete, and all probabilities are finite, this conditional probability must be modified since draws from H can be repeated. That is, when sampling from H, the probability of $X_k^* = X_l^*$ (for $k \neq l$) is positive. It has been observed the dish being served but cannot tell if it comes from the same table or not. For example, there are three tables in Figure 2.3 serving the same dish X_1^*. However, just given X_1^*, we cannot tell which table, t_1, t_4 or t_5, it comes from. This observation also applies to the CRP representation for the DP.

With the discrete base distribution and a finite sample, we can have a latent variable t_k^* that is the number of tables serving the dish X_k^* and the total count n_k^* of customers having dish X_k^* across t_k^* tables is spread with latent counts $m_{k,1}, ..., m_{k,t_k^*}$ where $n_k^* = \sum_{j=1}^{t_k^*} m_{k,j}$. For example, there are thirteen customers in the restaurant in Figure 2.3 that are sitting at five tables. The customers counts are $n_1^* = 5$, $n_2^* = 4$ and $n_3^* = 4$, and the table counts are $t_1^* = 3$, $t_2^* = 1$, $t_3^* = 1$. The customer counts are spread to tables. In the case of just observing the total counts but not the partition across tables, we can derive the following conditional distribution

$$\begin{aligned} p(x_{N+1} \,|\, \boldsymbol{x}, \boldsymbol{m}, \boldsymbol{t}^*, a, b, H) &= \sum_{k=1}^{K} \sum_{j=1}^{t_k^*} \frac{m_{k,j} - a}{N+b} \delta_{X_k^*}(\cdot) + \frac{Ta+b}{N+b} H(\cdot) && (2.9) \\ &= \sum_{k=1}^{K} \frac{n_k^* - a t_k^*}{N+b} \delta_{X_k^*}(\cdot) + \frac{Ta+b}{N+b} H(\cdot) \,, && (2.10) \end{aligned}$$

where $\boldsymbol{t}^* = (t_1^*, t_2^*, \ldots, t_K^*)$, and $T = \sum_{k=1}^{K} t_k^*$. Sampling from the above equations makes explicit whether a new table is created or which existing table is used

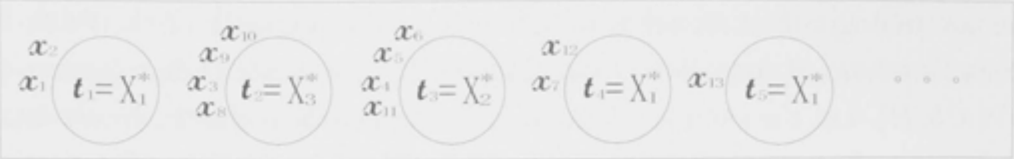


Figure 2.3: A CRP representation for a PDP with a discrete base distribution. There are thirteen customers, five tables with three dishes being served. A dish now can be served by multiple tables, which is different to Figure 2.1.

for the new sample. Equations (2.9) and (2.10) will be used to derive two Gibbs samplers that will be discussed respectively in Sections 3.2 and 3.3.

2.3.4 Hierarchical Dirichlet-Poisson Process

As discussed in Section 2.3.1, the PDP is a probability function on distributions: it takes as input a random base distribution and yields as output a discrete probability distribution, which has a finite or countable set of possible values on the same domain. When the base distribution is itself discrete, the PDP yields a new discrete distribution that is somewhat similar; the greater b is, the smaller variance of the two distributions will be (Generally, the variance is of order $\frac{1-a}{1+b}$ [Buntine and Hutter, 2010].).

The output probability of a PDP can be recursively used as a base distribution for another PDP to create a hierarchy of distributions. This hierarchy is the so-called hierarchical PDP (HPDP), or the hierarchical Pitman-Yor process [Teh, 2006b; Teh and Jordan, 2010]. It is a generalisation of the HDP. Analogous to the HDP, the HPDP is defined as

$$\begin{aligned} G_0 \mid a_0, b_0, H &\sim \mathrm{PDP}(a_0,\ b_0,\ H) \\ G_i \mid a_i, b_i, G_0 &\sim \mathrm{PDP}(a_i,\ b_i,\ G_0),\ \text{for } i \in \{1, \ldots, I\}\ . \end{aligned}$$

With a simple modification of the conditional probabilities given by Equations (2.5) and (2.6), one has the following conditional probabilities for the HPDP

$$\begin{aligned} y_{i,T_i+1} \mid \boldsymbol{y}_{1:I},\ a_0,\ b_0,\ H &\sim \sum_{k=1}^{K} \frac{n'_k - a_0}{b_0 + \sum_{k=1}^{K} n'_k} \delta_{X_k^*}(\cdot) + \frac{b_0 + a_0 K}{b_0 + \sum_{k=1}^{K} n'_k} H(\cdot) \\ x_{i,J_i+1} \mid \boldsymbol{x}_i,\ a_i,\ b_i,\ G_0 &\sim \sum_{t=1}^{T_i} \frac{n^*_{i,t} - a_i}{b_i + J_i} \delta_{y_{i,t}}(\cdot) + \frac{b_i + a_i T_i}{b_i + J_i} G_0(\cdot) \end{aligned}$$

where b_i and b_0 are concentration parameters, and a_i and a_0 are discount parameters. The other notations are the same as in Equations (2.5) and (2.6). Similar to the HDP, the HPDP can be adapted to an infinite limit of finite mixture models as a non-parametric prior.

2.3.5 PDP v.s. DP

The PDP is a generalisation of the DP. Both are probability distributions over distributions over a measurable space. The PDP has similar properties to the DP, *e.g.*, mean, variance and covariance (see the proof of lemma 35 in [Buntine and Hutter, 2010]).

Property 2.5. *(Mean, Variance, and Covariance) If $G \sim PDP(a, b, H)$, for any measurable set B,*

$$\begin{aligned} \mathbb{E}(G(B)) &= H(B) \\ \mathbb{V}(G(B)) &= \frac{1-a}{1+b} H(B)(1 - H(B)) \\ \mathbb{C}ov(G(B), G(B')) &= -\frac{1-a}{1+b} H(B)H(B') \quad s.t.\ B' \cap B = \emptyset\ . \end{aligned}$$

The stick-breaking construction and the Chinese restaurant process have natural generalisations for the PDP and the DP. With respect to applications, both of them are used as non-parametric priors for parameters of mixture models. Nevertheless, the PDP and the DP are different to a certain extent, since the introduction of the *discount parameter* a in the PDP.

The PDP can reduce to the DP, if the *discount parameter* is set to 0. With only the *concentration parameter* b, the DP has some properties such as slower convergence of the sum $\sum_{k=1}^{\infty} p_k$ to one, since the number of unique values taken on by draws from G grows slowly at order $O(b \log N)$, where N is the total number of draws. Actually, with referring to the posterior distribution (2.4) of the DP in the CRP representation, we can have the expected number (K) of unique values computed as follows.

$$\mathbb{E}[K \mid N] = \sum_{n=1}^{N} \frac{b}{b+n-1} \in O(b \log N).$$

If $0 < a < 1$, the PDP behaves according to a "power-law" [Pitman, 2002; Teh, 2006b; Goldwater et al., 2006; Teh and Jordan, 2010], which is in contrast to the logarithmic growth for the Dirichlet Process. The "power-law" behaviour can be observed from either the stick-breaking construction or the CRP representation for the PDP. As discussed by Teh and Jordan [2010], the stick-breaking

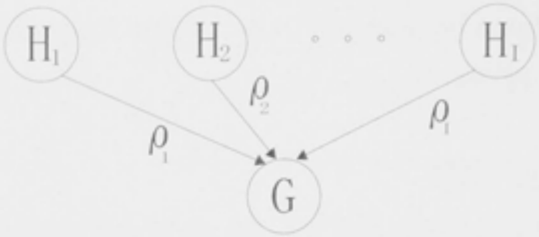


Figure 2.4: The graphical representation of the CPDP. ρ_i is the weight on the directed edge between H_i and G.

construction in Theorem 2.3 shows that the expectation of p_k is of order $O(k^{-1/a})$ if $0 < a < 1$, which indicates the partition size decays according to a "power-law". For the DP, the expectation of p_k (see Theorem 2.2) is of order $O\left((\frac{b}{1+b})^k\right)$, which decreases exponentially in k.

A similar phenomenon can also be observed from the CRP representation of the PDP, in which the proportion of tables with N customers scales as $O(N^{-(a+1)})$, and the total number of tables scales as $O(N^a)$. Involving the discount parameter causes the tail of a distribution drawn from the PDP to be much longer than that drawn from the DP, since there will be a large number of tables with small number of customers, which corresponds to a "power-law" that exists in natural language [Goldwater et al., 2006]. The "power-law" behaviour of the PDP makes it more suitable than the DP for many applications, especially for natural language processing.

2.4 Compound Poisson-Dirichlet Process

As discussed in Section 2.3, the traditional PDP only has one base measure. The expectation of probability distributions drawn from a PDP is the base measure. The variance between those random distributions and the base measure is controlled jointly by the discount and the concentration parameters. However, in modelling problems, such as statistical language model domain adaption [Wood and Teh, 2009] and topic evolutionary analysis [Zhang et al., 2010], it is required to share knowledge (*i.e.*, statistical information) across different domains or among data that impose, for instance, a sequential time dependence. Therefore, it is of great interest to develop a new integrated non-parametric Bayesian method that can adapt or borrow knowledge from different domains to handle more complex data relations.

Although we can linearly combine a set of PDPs to deal with knowledge adap-

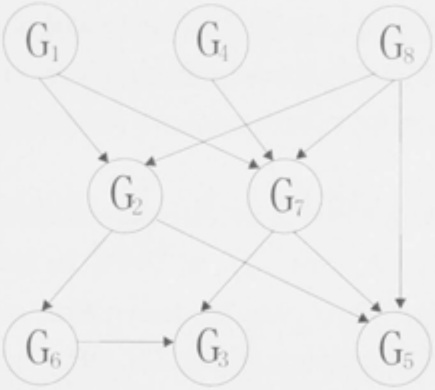


Figure 2.5: A directed acyclic graph (DAG). Each node can be associated with a random probability distribution drawn from a CPDP.

tations, like those in [Ren et al., 2008; Pruteanu-Malinici et al., 2010], another approach is to directly extend the PDP without loosing the generality by replacing the single base measure with an admixture of multiple base measures defined on the same measurable space, as shown in Figure 2.4. I call this method the compound Poisson-Dirichlet process (CPDP). The mixture weights associating with the base measures are normally summed to one to make them probabilistic.

Definition 2.5. (*the compound Poisson-Dirichlet process*) Let $0 \leq a < 1$ be the discount parameter, $b > -a$ be the concentration parameter, $\{H_1, H_2, \ldots, H_I\}$ be a set of base measures over a measurable space $(\mathcal{X}, \mathcal{B})$, each of which is indexed by i, and ρ_i be the mixture weight corresponds to H_i, *s.t.* $\sum_{i=1}^{I} \rho_i = 1$, as shown in Figure 2.4, the compound Poisson-Dirichlet process is

$$G \sim CPDP\left(a, b, \sum_{i=1}^{I} \rho_i H_i\right) .$$

Clearly, Definition 2.5 shows that the CPDP can be understood as a multiple-base-measure generalisation of the PDP. It can easily be adapted to an arbitrary directed acyclic graph (DAG), as shown in Figure 2.5, where each node in the DAG is associated with a random probability distribution drawn from a CPDP. The CPDP takes as a base measure the admixture of random distributions associating with its parent nodes. This forms a network of CPDPs, called the graphical PDP, or the graphical Pitman-Yor process [Wood and Teh, 2009].

Definition 2.6. (*The graphical Poisson-Dirichlet process*) Let $\mathcal{G}$ denote a DAG that composes of nodes indexed by integers $1, \cdots, J$. Each node j in $\mathcal{G}$ is associated with a random probability distribution G_j drawn from a CPDP. The directed edges are indicated by (i, j) for $i \in \mathrm{Pa}(j)$, where $\mathrm{Pa}(j)$ is the set of parent nodes of j. Let $\rho_{i,j}$ be the mixture weight on each edge (i, j), *s.t.* $\sum_{i \in \mathrm{Pa}(j)} \rho_{i,j} = 1$. The graphical PDP is

$$G_j \sim \mathrm{CPDP}\left(a_j, b_j, \sum_{i \in \mathrm{Pa}(j)} \rho_{i,j} G_i\right). \tag{2.11}$$

For example, in Figure 2.5, G_2 can be drawn from a CPDP with a mixture of G_1 and G_8, and G_7 with a mixture of G_1, G_4 and G_8. Note, when $|\mathrm{Pa}(j)| = 1$, then the single hyper-parameter $\rho_{i,j}$ is equal to one, so the sum degenerates to G_i, thus the CPDP defined on one node reduces to the PDP. The mixture weights $\boldsymbol{\rho}_j$ can be modelled as $\boldsymbol{\rho}_j \sim \mathrm{Dir}_{|\mathrm{Pa}(j)|}(\boldsymbol{\varrho})$, where $\boldsymbol{\varrho}$ is a Dirichlet parameter, whose dimensionality is $|\mathrm{Pa}(j)|$. Obviously, the CPDP inherits most of the properties of the PDP and can be embedded not only in hierarchical structures but also in large networks with arbitrary structures, such as a DAG. Consequently, one can no longer commit to a single base measure, so there is more flexibility of modelling complex data, for example, the adaptor grammar [Johnson et al., 2007] in the context of probabilistic context-free grammars.

2.4.1 CPDP via the Stick-breaking Construction

Developing a stick-breaking construction for the CPDP provides a concrete representation of draws from the CPDP, and it provides insight into the sharing of atoms drawn from multiple base measures with probabilities proportional to the mixture weights.

The generation of stick-breaking weights $(p_k)_{k=1}^{\infty}$ is the same as that in the PDP stick-breaking construction, see Theorem 2.3. The problem is now how to generate random variables $(X_k^*)_{k=1}^{\infty}$ from base measure(s). In the PDP stick-breaking construction, all X_k^*'s are drawn from a single base measure. However, there are multiple base measures that are linearly combined as one measure in the CPDP. Therefore, to generate a X_k^*, we need to decide exactly from which base measure this X_k^* is drawn. Thus the core difference between the CPDP and the PDP is the different way of drawing $(X_k^*)_{k=1}^{\infty}$.

According to Definition (2.5), the base measure for drawing a probability distribution G from a CPDP is the mixture of I base measures $H_1, H_2, \cdots, H_I$. Since the sum of the mixture weights is equal to one, $\sum_{i=1}^{I} \rho_i = 1$, we can treat $\boldsymbol{\rho} = (\rho_1, \rho_2, \cdots, \rho_I)$ as a probability vector, a parameter of a multinomial distribution, and then samples drawn from this multinomial distribution can be used to decide which base measure a X_k^* is drawn from. Therefore, the probability of drawing a X_k^* from a specific base measure H_i could be proportional to the mixture weight ρ_i, then the stick-breaking construction which we derive now is straightforward.

Let ϕ_k be a random variable distributed according to a multinomial distribution with parameter $\boldsymbol{\rho}$, and attached to X_k^*, then $(X_k^*)_{k=1}^{\infty}$ are generated as:

$$\begin{aligned} \phi_k \,|\, \boldsymbol{\rho} &\sim \text{Discrete}(\boldsymbol{\rho}) \\ X_k^* \,|\, \phi_k, H_1, H_2, \cdots, H_I &\sim H_{\phi_k} \end{aligned}$$

If $|\text{Pa}(j)| = 1$, as discussed before, the above procedure reduces to directly sampling X_k^* from a single base distribution. Moreover, the support of each G is contained within the support of the mixture of all its base measures.

If the discount parameter is set to zero, the CPDP reduces to the compound DP, and the graphical PDP reduces to the graphical DP. There exists a stick-breaking construction for the graphical DP, such as the one elaborated in [Zhang et al., 2010]. The admixture of base measures can be done through the admixture of stick-breaking weights. The stick-breaking representation for the Evolutionary HDP proposed by Zhang et al. [2010] deals with two base measures that are random distributions drawn from a DP. It can be generalised to handle multiple base distributions, say J base distributions as follows.

Let G_j, $j \in J$, be a probability distribution drawn from a DP according to Theorem 2.2, and G be a probability distribution drawn from $\text{DP}(\alpha, \sum_{j=1}^{J} \rho_j G_j)$, then the corresponding stick-breaking construction is

$$\begin{aligned} G_j &= \sum_{k=1}^{\infty} p'_{j,k} \delta_{X_k^*} && \text{for } j = 1, \cdots, J \\ \boldsymbol{p} &\sim \text{DP}\left(\alpha, \sum_{j=1}^{J} \rho_j \boldsymbol{p}'_j\right) && \text{s.t. } \sum_{j=1}^{J} \rho_j = 1 \\ G &= \sum_{k=1}^{\infty} p_k \delta_{X_k^*} \,. \end{aligned}$$

Zhang et al. [2010] have given a Gibbs sampling based on this construction. Note we can also make G_j to be drawn from a compound DP.

2.4.2 CPDP via Chinese Restaurant Process

Recall the CRP representations for the DP, the HDP, and the PDP. The CPDP can also be represented by a Chinese restaurant metaphor as follows. Draws

$\boldsymbol{x} = (x_1, x_2, \cdots, x_N)$ from G correspond to customers, dishes served at tables are draws (X_k^*s) from the mixture base measure $\sum_{i=1}^I \rho_i H_i$. Let n_k^* be the number of customers eating X_k^*. If all the base measures are non-atomic, after G being marginalised out, the conditional of x_{N+1} can be derived by slightly modifying Equation (2.8) as

$$\begin{aligned} & p\left(x_{N+1} \,|\, \boldsymbol{x}, a, b, \boldsymbol{\rho}, H_1, H_2, \cdots, H_I\right) \\ = & \sum_{k=1}^{K} \frac{n_k^* - a}{N+b} \delta_{X_k^*}(\cdot) + \frac{a \times K + b}{N+b} \left(\sum_{i=1}^{I} \rho_i H_i(\cdot) \right) \end{aligned} \tag{2.12}$$

where $\delta_{X_k^*}$ is the probability mass at X_k^* and K is the number of dishes served at all tables.

Similar to the PDP, when all the base measures are discrete, and all probabilities are finite, Equation (2.12) must be modified since draws from the mixture of base measures can be repeated. With the same notations used in Equation (2.9) and (2.10), we have

$$\begin{aligned} & p\left(x_{N+1} \,|\, \boldsymbol{x}, \boldsymbol{m}, \boldsymbol{t}^*, a, b, \boldsymbol{\rho}, H_1, H_2, \cdots, H_I\right) \\ = & \sum_{k=1}^{K} \sum_{j=1}^{t_k^*} \frac{m_{k,j} - a}{N+b} \delta_{X_k^*}(\cdot) + \frac{a \times T + b}{N+b} \left(\sum_{i=1}^{I} \rho_i H_i(\cdot) \right) \quad (2.13) \\ = & \sum_{k=1}^{K} \frac{n_k^* - a * t_k^*}{N+b} \delta_{X_k^*}(\cdot) + \frac{a \times T + b}{N+b} \left(\sum_{i=1}^{I} \rho_i H_i(\cdot) \right) \quad (2.14) \end{aligned}$$

In the Chinese restaurant metaphor, customers choose a dish by sitting at a table. If a customer chooses to sit at an unoccupied table, a new dish should be sampled from the base measure. In the CPDP, the number of base measures could be more than one, so we need to decide from which base measure a dish is sampled. Since the base measure is an admixture, and the sum of the mixture weights is equal to one, it can be achieved by first choosing a base measure H_i with probability proportional to ρ_i, then sampling a dish from H_i. This procedure is also known as multi-floor Chinese restaurant franchise [Wood and Teh, 2009].

Specifically, in the CRP metaphor for the CPDP, a restaurant corresponding to G has I menus, each of which is generated from H_i. Tables are clustered and allocated to different floors according to dishes served by them. The number of floors is equal to the number of menus. If a table serves a dish that is drawn from a menu i, then this table will be allocated to the i^{th} floor. Arriving at a multi-floor restaurant, a customer can choose to sit either at an occupied table in a floor or at an unoccupied table. If the customer chooses to sit at an unoccupied table, a dish

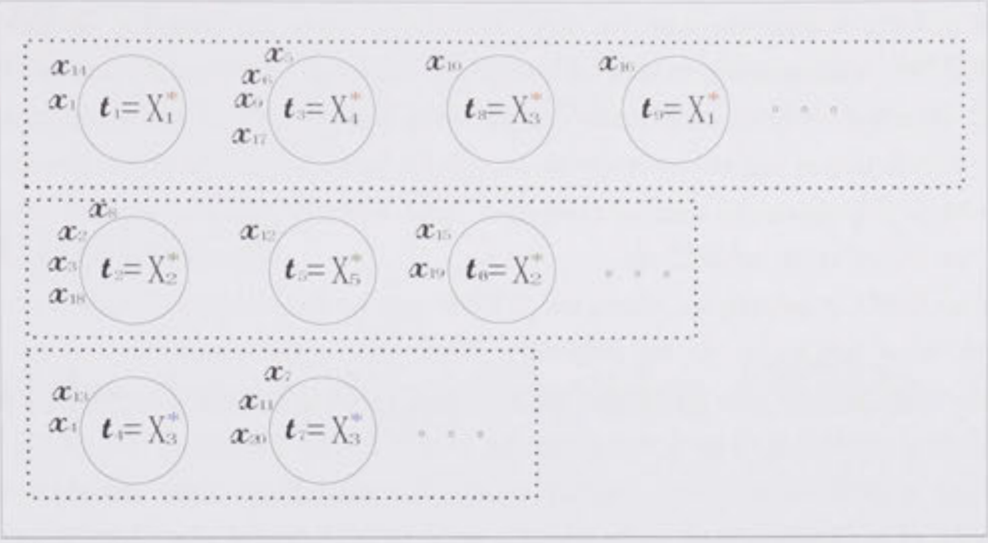


Figure 2.6: A multi-floor CRP representation for a CPDP with three base measures. The outer rectangle indicates a restaurant, the inter rectangles with dotted lines are floors, circles are tables, x_n's are customers, and X_k^*'s are dishes.

is then ordered from any one of the I menus. The probability of ordering a dish from menu i is proportional to ρ_i. If the ordered dish is from menu i, the newly created table will then be allocated in the i^{th} floor. Figure 2.6 shows a multi-floor Chinese restaurant metaphor for a CPDP with three base measures. It has three floors that correspond to the three base measures, nine occupied tables, and 20 customers. In this CRP representation, each table could be associated with a latent variable, named *menu indicator* (shown as stars with different colors in Figure 2.6), that indicates from which menu the dish on the table is ordered. All the menu indicators can be taken as *i.i.d.* draws from a multinomial distribution with parameter $\boldsymbol{\rho}$, and they cluster tables into I number of floors. We can consider putting a prior on $\boldsymbol{\rho}$, such as a Dirichlet distribution. I will show how to introduce a Dirichlet distribution as a prior on $\boldsymbol{\rho}$ in Section 3.6.

2.4.3 CPDP v.s. Other Related Models

The CPDP is different in several perspectives from the related models, such as the dynamic HDP [Ren et al., 2008] and the Pachinko allocation model (PAM) [Li and McCallum, 2006].

The dynamic HDP (DHDP), see Section 2.2.3, shares the statistical information (*e.g.*, atoms) across sequential data (*e.g.*, music) by linearly combining two probability distributions. In contrast, the CPDP combines several base distributions to form a mixture base for the PDP, before drawing samples. In this way, the CPDP could propagate shared knowledge through, for example, an arbitrary DAG structure where each node has at least one parent. Combining multiple base distributions together means sharing knowledge across different domains, or statistical dependencies among streamed data. Clearly, the former is a mixture, and the latter is an admixture.

The DAG structure to which the CPDP can be inserted is flexible such that it can be a hierarchy, or an arbitrary DAG with cross-connected edges. It is worth pointing out the Pachinko allocation model (PAM) proposed by Li and McCallum [2006] and its extensions [Li et al., 2007; Mimno et al., 2007]. The PAM is a DAG-structured mixture model for modelling topic correlations. It consists of a DAG, where each interior node is a Dirichlet distribution over its child nodes. The Dirichlet distribution has the same dimension as the number of child nodes. It can be seen that directed edges in its DAG indicate that parents are Dirichlet distributions over the corresponding linked children. Indeed, the DAG in PAM partitions the space to different parts (*i.e.*, the whole topic space to subtopic spaces). However, in the DAG structure with the CPDP's, each node is associated with a random probability distribution drawn from a CPDP. The directed edges show how one node can be generated from the admixture of the linked parents via a stochastic process.

2.5 Summary

In this chapter, I have discussed the Dirichlet distribution and the Dirichlet related non-parametric Bayesian methods that include the Dirichlet process, the two-parameter Poisson-Dirichlet process and their hierarchical extensions. I also discussed a new class of non-parametric methods, named the compound Poisson-Dirichlet processes that can handle multiple input distributions. The corresponding stick-breaking construction and Chinese restaurant process representation were presented. All of these provide a foundation for models and algorithms presented in Chapters 3 to 7.

Chapter 3

Gibbs Sampling for the PDPs

In this chapter, I will discuss computational aspects of doing inference for non-parametric Bayesian models based on the Poisson-Dirichlet process that is an important non-parametric method in statistical machine learning. There are various mathematical representations available for PDPs, which can be combined in different ways to build a range of inference algorithms, *e.g.*, Neal [2000]; Ishwaran and James [2001]; Blei and Jordan [2005]; Ren et al. [2008]; Zhang et al. [2010]; Ren et al. [2010]. Here I will focus on Gibbs sampling algorithms for sampling from posterior distributions of the PDPs, based on the Chinese restaurant process (CRP) representation, particularly in a finite state space. In subsequent sections, I will discuss three Gibbs sampling algorithms for the PDP. They are respectively

- Teh's sampling for seating arrangement sampler (SSA) [Teh, 2006a] (Section 3.2);
- Collapsed multiplicity Gibbs sampler (CMGS, Section 3.3);
- Blocked table indicator Gibbs sampler (BTIGS, Section 3.4).

After comparing these three samplers in Section 3.5, I will present two Gibbs sampling techniques for the CPDPs based on the CMGS and the BTIGS in Section 3.6.

3.1 Joint Marginalized Likelihood

In this section I discuss the joint marginalised likelihood over a specific seating arrangement of customers in a restaurant. It will help in understanding the sampling algorithms that will be discussed in Sections 3.2, 3.3 and 3.4.

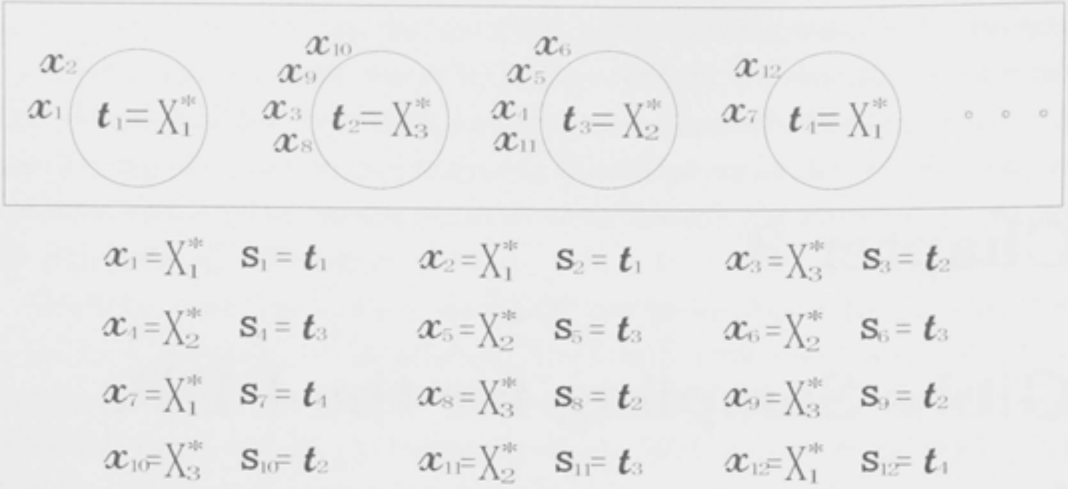

x_2
x_1
$t_1 = X_1^*$
x_{10}
x_9
x_3
x_8
$t_2 = X_3^*$
x_6
x_5
x_4
x_{11}
$t_3 = X_2^*$
x_{12}
x_7
$t_4 = X_1^*$
$\circ \circ \circ$
$x_1 = X_1^*$ $s_1 = t_1$ $x_2 = X_1^*$ $s_2 = t_1$ $x_3 = X_3^*$ $s_3 = t_2$
$x_4 = X_2^*$ $s_4 = t_3$ $x_5 = X_2^*$ $s_5 = t_3$ $x_6 = X_2^*$ $s_6 = t_3$
$x_7 = X_1^*$ $s_7 = t_4$ $x_8 = X_3^*$ $s_8 = t_2$ $x_9 = X_3^*$ $s_9 = t_2$
$x_{10} = X_3^*$ $s_{10} = t_2$ $x_{11} = X_2^*$ $s_{11} = t_3$ $x_{12} = X_1^*$ $s_{12} = t_4$

Figure 3.1: A specific seating arrangement in a Chinese restaurant. The *customer-dish* assignment $x_n = X_k^*$ indicates a customer x_n eats the dish X_k^*, the *customer-table* assignment $s_n = t_i$ indicates x_n sits at table t_i, and the *table-dish* assignment $t_i = X_k^*$ indicates table t_i serves X_k^*.

In the Chinese restaurant metaphor, after all customers have been seated, each restaurant has a seating arrangement of those customers. Figure 3.1 shows a specific seating arrangement in a restaurant that has twelve customers, each of which sits at a table t_i. There are four occupied tables, each of which serves a dish X_k^*. We can see that a seating arrangement includes the total number of customers, the total number of occupied tables, the *customer-table* assignments (*i.e.*, the table identities of the customers ($s_n = t_i$)), the *customer-dish* assignments ($x_n = X_k^*$), and the *table-dish* assignments ($t_i = X_k^*$). Actually, given the *customer-table* assignments, the *table-dish* assignments can be reconstructed from the *customer-dish* assignments, and vice versa. For example, in Figure 3.1, $x_1 = X_1^*$ and $s_1 = t_1$, so $t_1 = X_1^*$. Therefore, we only need to keep either the *customer-table* and *customer-dish* assignments or the *customer-table* and *table-dish* assignments.

Now the joint marginal likelihood over a particular seating arrangement can be computed as follows. Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ be a sequence of customers; K the total number of dishes in a restaurant where each dish is denoted by X_k^*; t_k^* the number of tables serving X_k^*; n_k^* the number of customers eating X_k^*; m_{k,t_i} the number of customers sitting at table t_i serving X_k^*; T the total number of occupied tables; and N the total number of customers. The *customer-table* assignments are indicated by $\boldsymbol{s} = (s_1, s_2, \ldots, s_N)$. Each entry s_n of $\boldsymbol{s}$ corresponds to the table assignment of customer x_n, and takes values on $\{t_1, t_2, \ldots, t_T\}$. The seating arrangement can now be interpreted mathematically as

$$\begin{aligned}
x_n &\in \{X_1^*, X_2^*, \ldots, X_K^*\} && \text{for } n \in \{1, 2, \ldots, N\} \\
s_n &\in \{t_1, t_2, \ldots, t_T\} && \text{for } n \in \{1, 2, \ldots, N\} \\
m_{k,t_i} &= \sum_{n=1}^{N} 1_{x_n = X_k^*} 1_{s_n = t_i} & n_k^* &= \sum_{n=1}^{N} 1_{x_n = X_k^*} \\
t_k^* &= \sum_{i=1}^{T} 1_{t_i = X_k^*} = \sum_{i=1}^{T} 1_{m_{k,t_i} > 0} & T &= \sum_{k=1}^{K} t_k^* \\
N &= \sum_{i=1}^{T} m_{k,t_i} = \sum_{k=1}^{K} n_k^* \, .
\end{aligned}$$

The joint marginal likelihood over a specific seating arrangement (*i.e.*, $\boldsymbol{x}$ and $\boldsymbol{s}$) can be derived by multiplying up the conditional probabilities, given by Equation (2.9) in Chapter 2, for all assignments of customers to tables. It has the following form

$$p(\boldsymbol{x}, \boldsymbol{s} \,|\, a, b, H) \;=\; \frac{(b|a)_T}{(b)_N} \prod_{k=1}^{K} H(X_k^*)^{t_k^*} \prod_{j=1}^{t_k^*} (1-a)_{m_{k,\zeta_j} - 1} \, , \tag{3.1}$$

where ζ_j takes values on $\{t_1, t_2, \ldots, t_T\}$, H is a probability distribution over dishes, $(x)_N$ is given by $(x|1)_N$, and $(x|y)_N$ denotes the Pochhammer symbol with increment y, it is defined as

$$(x|y)_N \;=\; x(x+y)\ldots(x+(N-1)y) = \begin{cases} x^N & \text{if } y = 0 \\ y^N \times \frac{\Gamma(x/y+N)}{\Gamma(x/y)} & \text{if } y > 0 \, , \end{cases} \tag{3.2}$$

where $\Gamma(\cdot)$ denotes the standard Gamma function.

This joint marginal likelihood function will be used as the basis for the derivation of Gibbs sampling algorithms discussed in Sections 3.2, 3.3 and 3.4. In particular, I will show how it can be used to compute equations in Section 3.3. Notations used in this section will be reused in subsequent sections.

3.2 Teh's Sampling for Seating Arrangement

The *sampling for seating arrangement* algorithm [Teh, 2006a], denoted by SSA, returns samples from the posterior distribution over the seating arrangement. It

Algorithm 1 Sampling for seating arrangements algorithm

```
for each customer $x_n$ sitting at a table that serves a dish $X^*_k$ do
    Sample to find the table from which $x_n$ will be removed, i.e., suppose for
    each dish, $X^*_k$, there are a total of $t^*_k$ tables serving $X^*_k$, each of which
    is indexed by $\zeta_j$ ($j \in 1, 2, \ldots, t^*_k$) and has $m_{k,\zeta_j}$ customers. Then, the
    probability of removing $x_n$ from the $j$-th of these tables is $m_{k,\zeta_j}/n^*_k$.
    Decrement the corresponding customer count $m_{k,\zeta_j}$ by one. If the count
    goes down to zero, the table becomes unoccupied, then decrease the table
    count $t^*_k$ by one.
    Reinsert the customer back to the restaurant using the standard CRP
    sampling probabilities computed by Equation (2.9) which are propor-
    tional to
        a.) $m_{k,\zeta_j} - a$: seat $x_n$ at ${\zeta_j}^{th}$ occupied table in the restaurant. The
    dish served on this table is then assigned to $x_n$. Update related counts.
        b.) $b + aT$: seat $x_n$ at a new table. A dish needs to be sampled from
    $H$. Let it be denoted by $X^*_{k'}$. Then, assign the dish to the new table and
    the customer $x_n$. Increase $t^*_{k'}$ by one, and initialise $m_{k',\zeta_{t^*_{k'}}}$ to one.
end for
```

only keeps track of the number of customers sitting around each table (*i.e.*, m_{k,t_i}), rather than explicitly recording all *customer-table* assignments[1].

From the joint marginalised likelihood over a seating arrangement, Equation (3.1), we can easily observe that given the dishes all customers eat, the table assignments have no effect on the joint marginalised likelihood of the data. Therefore, at every restaurant, the SSA algorithm only keeps track of the number of tables t^*_k, and all the customer counts m_{k,ζ_1}, m_{k,ζ_2}, $\ldots$, $m_{k,\zeta_{t^*_k}}$ at tables for each X^*_k, where ζ_j is defined in Equation (3.1). The assignments of individual customers to individual tables, denoted by $\boldsymbol{s}$, are not recorded but rather reconstructed randomly during sampling. This can be done because the assignments $\boldsymbol{s}$ do not appear explicitly in either Equation (2.9) or Equation (3.1), rather appear indirectly via the customer count at each table. Thus they can be uniformly sampled as long as the counting constraints are maintained, *i.e.*, $n^*_k \geq t^*_k$. The SSA algorithm performs cycles as shown in Algorithm 1. We should keep in mind that

[1] Although m_{k,ζ_j} in Equation (3.1) can be reconstructed from the *customer-table* assignments, and recording these assignments could lead to a better mixing of a Markov chain, it may still require a large storage space in cases, such as a topic model that needs to be trained on a large number of documents.

SSA requires a dynamic storage for customer counts at all tables. Being placed in a hierarchical context where deletion and creation of tables lead us to recursively carry out the removing and reinserting operations up through all the nodes in the hierarchy.

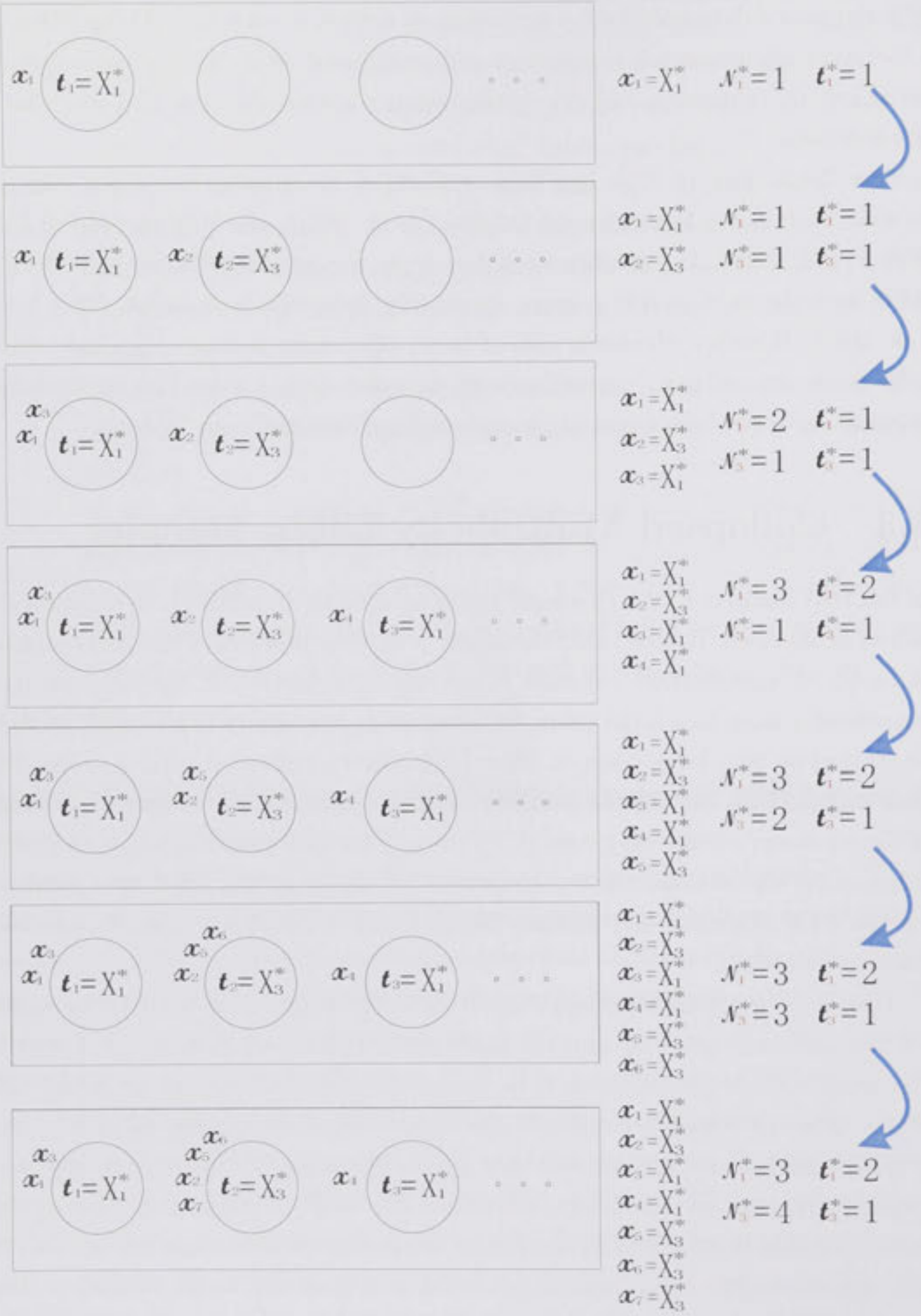
The basic idea of SSA has been embedded in samplers for more complex models, such as the hierarchical LDA [Blei et al., 2010], the HDP variant of LDA [Teh et al., 2006], the doubly nested n-gram model [Mochihashi and Sumita, 2008] and the side-by-side n-gram models for language adaptation [Wood and Teh, 2009]. However, the basic idea of these algorithms remains the same, which is to move the customer currently being sampled up to the end of the customer sequence so that the sequential formula of Equation (2.9) can be used.

3.3 Collapsed Multiplicity Gibbs Sampler

In the SSA algorithm, the customer count at each table needs to be dynamically stored in memory. It could still encounter a storage problem if the total number of tables at a restaurant becomes large, which is possible if the concentration parameter b is set to a large value, for example in a language model that needs to be trained on very large corpora. Here I introduce a collapsed version of the SSA that marginalises out all the possible seating arrangements so that the storage of the customer counts $m_{k,\zeta_1}, m_{k,\zeta_2}, \ldots, m_{k,\zeta_{t_k^*}}$ for each dish X_k^* is not needed. It is based on the *multiplicity* representation of tables in the CRP interpretation for the PDP [Buntine and Hutter, 2010]. I call it *Collapsed Multiplicity Gibbs Sampler* (CMGS).

In the *multiplicity* representation, two observations that need to be stored are the customer count n_k^* and the table count t_k^* for each dish X_k^*, as shown in Figure 3.2. This representation is no longer sequential, since the actual identity of the table at which a customer sits cannot be reconstructed from n_k^*'s and t_k^*'s, and neither can one tell whether a dish being served comes from the same table or not (see Section 2.3.3). Therefore, the CRP based sequential sampling methods using Equation (2.9) cannot be used. Nevertheless, in order to describe this representation, the terminologies, such as restaurant, table, customer, dish, will still be used.

Definition 3.1. (*Multiplicity*) In the CRP representation for a PDP, assume the base distribution H is discrete, which means the probability of a same dish being served by multiple tables is positive with probability one. The number of tables



x_1 $t_1=X_1^*$
$x_1=X_1^*$ $n_1^*=1$ $t_1^*=1$
x_1 $t_1=X_1^*$ x_2 $t_2=X_3^*$
$x_1=X_1^*$ $n_1^*=1$ $t_1^*=1$
$x_2=X_3^*$ $n_3^*=1$ $t_3^*=1$
x_3 x_1 $t_1=X_1^*$ x_2 $t_2=X_3^*$
$x_1=X_1^*$ $x_2=X_3^*$ $x_3=X_1^*$
$n_1^*=2$ $t_1^*=1$
$n_3^*=1$ $t_3^*=1$
x_3 x_1 $t_1=X_1^*$ x_2 $t_2=X_3^*$ x_4 $t_3=X_1^*$
$x_1=X_1^*$ $x_2=X_3^*$ $x_3=X_1^*$ $x_4=X_1^*$
$n_1^*=3$ $t_1^*=2$
$n_3^*=1$ $t_3^*=1$
x_3 x_1 $t_1=X_1^*$ x_5 x_2 $t_2=X_3^*$ x_4 $t_3=X_1^*$
$x_1=X_1^*$ $x_2=X_3^*$ $x_3=X_1^*$ $x_4=X_1^*$ $x_5=X_3^*$
$n_1^*=3$ $t_1^*=2$
$n_3^*=2$ $t_3^*=1$
x_3 x_1 $t_1=X_1^*$ x_6 x_5 x_2 $t_2=X_3^*$ x_4 $t_3=X_1^*$
$x_1=X_1^*$ $x_2=X_3^*$ $x_3=X_1^*$ $x_4=X_1^*$ $x_5=X_3^*$ $x_6=X_3^*$
$n_1^*=3$ $t_1^*=2$
$n_3^*=3$ $t_3^*=1$
x_3 x_1 $t_1=X_1^*$ x_6 x_5 x_2 x_7 $t_2=X_3^*$ x_4 $t_3=X_1^*$
$x_1=X_1^*$ $x_2=X_3^*$ $x_3=X_1^*$ $x_4=X_1^*$ $x_5=X_3^*$ $x_6=X_3^*$ $x_7=X_3^*$
$n_1^*=3$ $t_1^*=2$
$n_3^*=4$ $t_3^*=1$

Figure 3.2: A *multiplicity* representation for the PDP. The empty circles are unoccupied tables, the others are occupied tables. There are seven customers who arrive in a restaurant sequentially. The statistics kept are n_k^*'s and t_k^*'s. The arrival of each customer will increase either n_k^* or both, which depends on the way in which the customers choose a table to sit at.

t_k^* serving the same dish X_k^* is defined as the *multiplicity* of the tables. In general, the multiplicity is the frequency of a distinct value drawn from the base measure appearing in the data.

Notice that given the seating arrangement, the conditional probability or the predictive probability, Equation (2.10), only depends on the number of customers eating dish X_k^*, n_k^*, and the number of tables serving X_k^*, *i.e.*, the *multiplicity* t_k^*. With all the customer counts $\boldsymbol{m}_{1,1:t_1^*}$, $\boldsymbol{m}_{2,1:t_2^*}$, ..., $\boldsymbol{m}_{K,1:t_K^*}$ being eliminated and just the total counts being kept, the joint posterior distribution of customers[2] $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ and multiplicities $\boldsymbol{t}^* = (t_1^*, t_2^*, \ldots, t_K^*)$ is derived by marginalising out all the possible seating arrangements[3] with Equation (3.1) [Teh, 2006a; Buntine and Hutter, 2010; Du et al., 2010b]

$$p(\boldsymbol{x}, \boldsymbol{t}^* \mid a, b, H) \;=\; \frac{(b|a)_T}{(b)_N} \prod_{k=1}^{K} H(X_k^*)^{t_k^*} S_{t_k^*, a}^{n_k^*} \,, \tag{3.3}$$

where $S_{M,a}^N$ is the generalised Stirling number [Hsu and Shiue, 1998] given by the linear recursion [Buntine and Hutter, 2010; Teh, 2006a]

$$\begin{aligned} S_{0,a}^N &= \delta_{0,N} && \\ S_{M,a}^N &= 0 && \text{for } M > N \\ S_{M,a}^{N+1} &= S_{M-1,a}^N + (N - Ma) S_{M,a}^N && \text{for } M \leq N \,. \end{aligned} \tag{3.4}$$

As a consequence, it follows that $S_{N,a}^N = 1$ and $S_{1,a}^N = \frac{\Gamma(N-a)}{\Gamma(1-a)}$. The major hurdle for using the joint distribution (3.3) is to compute the Stirling numbers. To avoid the intensive computation of order $O(NM)$, we can tabulate or cache the Stirling numbers for the required discount parameter a. In addition, these numbers rapidly become very large so computation needs to be done in a log space using a logarithmic addition to prevent overflow. Therefore, Equation (3.4) is computed in log space as

$$\log S_{M,a}^{N+1} = \log S_{M,a}^N + \log\left(\exp\left(\log S_{M-1,a}^N - \log S_{M,a}^N\right) + (N - M\,a)\right) \,.$$

The log() and exp() functions make the evaluation fairly slow. When keeping a fixed, we can overcome this problem by placing a maximum value on M, say

[2] Note customers $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$ are explicitly represented by the customer counts $(n_1^*, n_2^*, \ldots, n_K^*)$ in Equation (3.3).

[3] The last product in Equation (3.1) is changed to a Stirling number in Equation (3.3) by marginalising out all the specific seating arrangements [see Teh, 2006a, Equation (26)].

Algorithm 2 Collapsed multiplicity Gibbs sampling algorithm

```
1. for each customer $x_n$ eating a dish $X_k^*$ do
2.     Remove $x_n$ from the restaurant by decreasing $n_k^*$, subject to $n_k^* \geq t_k^*$ and $n_k^* = 0$ iff $t_k^* = 0$. If currently, $n_k^* = t_k^*$, then $t_k^*$ is decreased by one.
3.     Conditioned on the current $t_k^*$, sample for $x_n$ a new dish $X_{k'}^*$ according to Equation (3.3), and add it back to the restaurant by increasing $n_{k'}^*$. if $t_{k'}^* = 0$, then increase $t_{k'}^*$ by one.
4.     Conditioned on the new dish $X_{k'}^*$, sample $t_{k'}^*$ according to Equation (3.3), where $t_{k'}^*$ should be in the interval $[1, n_{k'}^*]$.
5. end for
```

100 or 1000 to limit the cache[4]. Furthermore, the computation with the two functions could suffer from fixed precision truncation error, especially for large M. In practice, the type of Stirling numbers is normally set to *Double* in the computation. In order to save space without seriously losing precision, we could instead save the final table in *Float*.

Equation (3.3) gives the joint posterior distribution of two random variables, x and t, thus a simple two-stage Gibbs sampling algorithm [Robert and Casella, 2005] can be adapted to sample each variable interchangeably conditioned on each other. Before presenting the two-stage Gibbs sampler in detail, I first discuss some constraints on the two counts n_k^* and t_k^*. Intuitively, the number of occupied tables in a restaurant should be less than or equal to the total number of customers currently being seated; and the table count is equal to zero if and only if the customer count is zero. These constraints apply to n_k^* and t_k^*. Specifically, the number of customers eating a dish X_k^* should be greater than or equal to t_k^*, and if $t_k^* = 0$, n_k^* must be zero, *i.e.*,

$$\begin{cases} t_k^* = 0 & \text{if and only if } n_k^* = 0 \\ n_k^* \geq t_k^* . \end{cases} \tag{3.5}$$

When removing and adding a customer, we must always bear in mind these constraints.

Now I present the two-stage Gibbs sampler in Algorithm 2. We can see that the CRP based Gibbs sampling algorithms, like SSA, are no longer applicable,

[4] The value of M depends on different applications. For instance, in the experiments of the segmented topic model in Chapter 5, I set M to the maximum number of words in segments. In the sequential LDA model, see Chapter 6, I set M to the double of this maximum number.

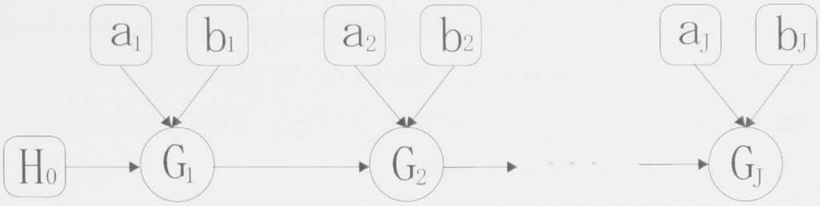


Figure 3.3: An example of multi-level hierarchical PDP

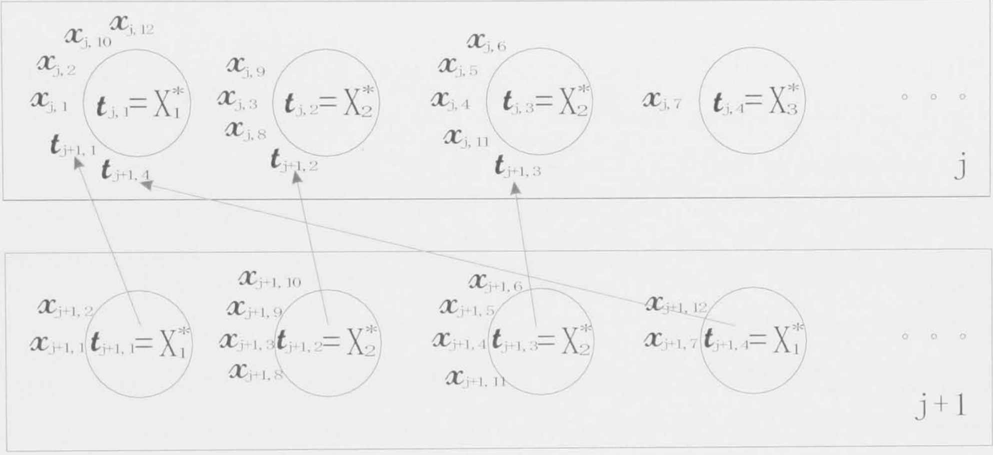


Figure 3.4: An hierarchical CRP representation for a multi-level hierarchical PDP. There are two types of customers in each restaurant, $x_{j,n}$'s who arrive by themselves and $t_{j+1,m}$'s who are sent by the corresponding child restaurant.

since the underlying sequence of customer-table assignments are lost due to Equation (3.3). I will empirically compare this CMGS with SSA in Section 3.5.

The joint posterior distribution given by Equation (3.3) can be adapted hierarchically to sample from the posterior of the PDP embedded in a multi-level hierarchy, see for instance Figure 3.3. The base distribution at level j is now recursively drawn from a PDP at level $j-1$. Then recursively applying Equation (3.3), we can derive the following joint posterior distribution

$$\begin{aligned} & p(\boldsymbol{x}_{1:J}, \boldsymbol{t}^*_{1:J} \,|\, \boldsymbol{a}_{1:J}, \boldsymbol{b}_{1:J}, H_0) \\ &= \prod_{k}^{K} H_0(X_k^*)^{t^*_{1,k}} \prod_{j=1}^{J} \frac{(b_j|a_j)_{T_j}}{(b_j)_{N_j+T_{j+1}}} \prod_{k=1}^{K} S^{n_{j,k}+t^*_{j+1,k}}_{t^*_{j,k},a_j} \,, \end{aligned} \tag{3.6}$$

where H_0 is the base distribution for the highest level PDP, $n_{j,k}$ is the number of customers that arrive by themselves and eat X_k^*, $N_j = \sum_{k=1}^{K} n_{j,k}$. The recursion is done according to Figure 3.4, a hierarchical CRP representation for a multi-level hierarchical PDP. In the figure, rectangles represent the Chinese restaurants that are indexed by j, circles are tables ($t_{j,m}$'s) and customers are $x_{j,n}$'s. $t_{j,m} = X_k^*$

indicates a dish X_k^* is served at table $t_{j,m}$. Red arrows indicate tables in the child restaurant are sent as proxy customers to its parent restaurant, so the total number of customers in restaurant j is $N_j + T_{j+1}$. This shows how the last two products in Equation (3.6) are derived.

3.4 Blocked Table Indicator Gibbs Sampler

In addition to SSA and CMGS algorithms, another promising sampling algorithm is a block Gibbs sampling algorithm based on an auxiliary latent variable, *table indicator*, which is introduced by Chen, Du and Buntine in [Chen et al., 2011]. I call the new algorithm the *Blocked Table Indicator Gibbs Sampler* (BTIGS). It is based on the *table indicator* representation built on top of the Chinese restaurant metaphor.

Definition 3.2. (*Table indicator*) The table indicator u associated with each customer x is an auxiliary latent variable that indicates whether x takes the responsibility of opening (or contributing) a new table or not. If x opens a new table, *i.e.*, the customer choosing to sit at an unoccupied table, $u = 1$, otherwise, $u = 0$.

This representation could compensate for the information loss that might be caused by CMGS. As discussed in Section 3.3, *customer-table* assignments cannot be reconstructed from the two observations (n_k^* and t_k^*) kept by CMGS. Losing the assignment information may result in bad mixing of the Markov chain in the sampling stage. However, recording all *customer-table* assignments requires large storage space. In order to reduce the information loss and the large space requirement, the *table indictor* representation records table contributions of customers, rather than the *customer-table* assignments.

The basic idea of the *table indicator* representation is that a table indicator variable is introduced to dynamically record the table contribution of each customer. If a customer x_n takes the responsibility of opening a new table, let $u_n = 1$; otherwise $u_n = 0$, which indicates x_n has chosen to share a dish with other customers. Figure 3.5 shows how the table indicator representation works with seven customers in a restaurant. There are only three customers that have opened a new table. They are x_1, x_2 and x_4 respectively. Their indicators are set to 1. In this sense, the *table indicator* keeps track of table contribution of each customer, rather than the table identity. The table *multiplicity* t_k^* defined in

CMGS can be constructed from the indicators as

$$t_k^* = \sum_{n=1}^{N} u_n 1_{x_n = X_k^*} \ . \tag{3.7}$$

For example, in Figure 3.5, for dish X_1^*, the table count $t_1^* = u_1 + u_3 + u_4 = 2$; for dish X_3^*, $t_3^* = u_2 + u_5 + u_6 + u_7 = 1$. This construction implies that the statistics need to be kept for the *table indicator* representation can be the same as those kept in the *multiplicity* representation. Note given n_k^*'s and t_k^*'s, we can compute the probability that a customer opened a table.

Conditioned on customer x_n eating dish X_k^* and given n_k and t_k, the probability of this customer contributing a table is proportional to $\frac{t_k^*}{n_k}$. For instance, the probability of customer x_3 contributing a table is $\frac{2}{3}$ in Figure 3.5. Indeed, there is a uniformity in the table indicator assignment. Therefore, *table indicators* can be randomly assigned in order to recover the table contributions, and thus the explicitly recording table indicators for all the customers is unnecessary.

The posterior distribution of the PDP from the *table indicator* representation can now be derived as follows. Let $\boldsymbol{t}^* = (t_1^*, t_2^*, \dots, t_K^*)$ be a vector of the table multiplicities, $\boldsymbol{u} = (u_1, u_2, \dots, u_N)$ be a vector of latent table indicators, and $\boldsymbol{x} = (x_1, x_2, \dots, x_N)$ be a sequence of customers that are explicitly represented by per dish customer counts n_k^*s. It is easy to see from Equation (3.7) together with Figure 3.5 that a specific *table indicator* assignment corresponds to a unique *multiplicity* representation, but a *multiplicity* representation gives $\prod_k \frac{n_k^*!}{t_k^*!(n_k^*-t_k^*)!}$ possible *table indicator* assignments. This choose term says any t_k^* of the n_k^* customers are equally likely to contribute a table. As a consequence, Equation (3.3) can be computed in terms of the joint posterior distribution of $\boldsymbol{x}$ and $\boldsymbol{u}$ as

$$p(\boldsymbol{x}, \boldsymbol{t}^* \,|\, a, b, H) = \left(\prod_k \frac{n_k^*!}{t_k^*!(n_k^* - t_k^*)!} \right) p(\boldsymbol{x}, \boldsymbol{u} \,|\, a, b, H) \ . \tag{3.8}$$

This formula lets us convert the *multiplicity* representation $(\boldsymbol{x}, \boldsymbol{t}^*)$ to the table indicator representation $(\boldsymbol{x}, \boldsymbol{u})$. Consequently, modifying the joint posterior, Equation (3.3), along with Equation (3.8), we can write down the joint posterior distribution of $\boldsymbol{x}$ and $\boldsymbol{u}$ as

$$p(\boldsymbol{x}, \boldsymbol{u} \,|\, a, b, H) \ = \ \frac{(b|a)_T}{(b)_N} \prod_{k=1}^{K} \left(H(X_k^*)^{t_k^*} S_{t_k^*, a}^{n_k^*} \frac{t_k^*!(n_k^* - t_k^*)!}{n_k^*!} \right) . \tag{3.9}$$

It can be observed that this joint posterior distribution is exchangeable in the pairs $(x_n,\ u_n)$, since the posterior and related statistics used are all sums over

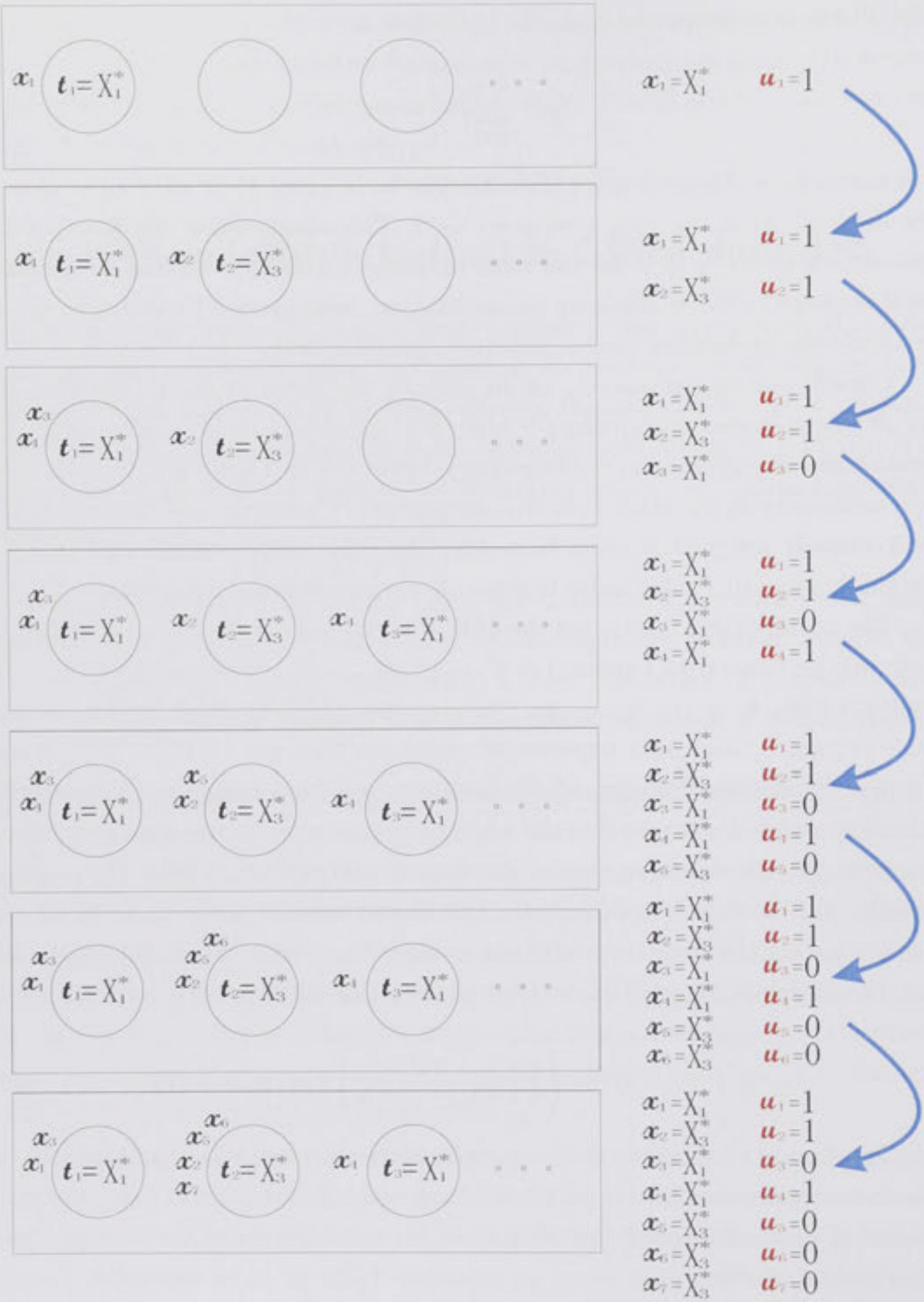


Figure 3.5: A *table indicator* representation of the PDP. The empty circles are unoccupied tables, the others are occupied tables. There are seven customers who arrive in a restaurant sequentially. Actually, only three of them have the responsibility of opening a new table.

Algorithm 3 Blocked table indicator Gibbs sampling algorithm

```
for each customer $x_n$ eating a dish $X_k^*$ do
    Sample $u_n$ for $x_n$ according to Equations (3.10) to remove $x_n$.
    Jointly sample $x_n$ and $u_n$ based on the joint posterior, Equation (3.9)
    Update both $n_k^*$ and $t_k^*$ based on the sampled values of $x_n$ and $u_n$ respectively.
end for
```

data [see Chen et al., 2011, Corollary 1]. Thus, different sampling orders and table assignment $\boldsymbol{s}$ can yield the same indicator assignment $\boldsymbol{u}$.

To sample from the posterior, I introduce an adapted version (*i.e.*, BTIGS) of the block Gibbs sampling algorithm proposed in [Chen et al., 2011]. It is shown in Algorithm 3. BTIGS is different to both CMGS and its variants proposed by Buntine et al. [2010]; Du et al. [2010b,a, 2012b], all of which adopt the two-stage Gibbs sampling algorithm to interchangeably and iteratively sampling x_n and t_k^*. Instead, BTIGS allows jointly sample x_n and u_n by cancelation of terms in Equation (3.9).

As I mentioned before, u_n is randomly assigned in the sampling procedure, rather than dynamically stored. While removing a customer from a restaurant, we need first to sample the value of u_n with following probabilities

$$p(u_n = 1 \,|\, x_n = X_k^*) = \frac{t_k^*}{n_k^*} \qquad p(u_n = 0 \,|\, x_n = X_k^*) = \frac{n_k^* - t_k^*}{n_k^*} \,. \tag{3.10}$$

It is interesting that the constraints put on the t_k^* and n_k^* discussed in CMGS (see Section 3.3) are implicitly guaranteed by the two probabilities. For example, if $n_k^* = t_k^*$ and $t_k^* > 0$, removing a customer $x_n = X_k^*$ must cause t_k^* to be decreased by one. In this case, Equation (3.10) always has $p(u_n = 1 \,|\, x_n = X_k^*) = 1$ so that removing a table is guaranteed. The only case to which a careful attention should be paid is that a table cannot be removed for x_n if $t_k^* = 1$ and $n_k^* > t_k^*$. Therefore, it should be assured that $p(u_n = 1 \,|\, x_n = k) = 0$ and $p(u_n = 0 \,|\, x_n = k) = 1$ in the implementation of BTIGS.

3.5 Empirical Comparison of the Three Samplers

All the three Gibbs sampling algorithms, *i.e.*, SSA, CMGS, and BTIGS, can be embedded into an hierarchical context. However, it is difficult in experiments to

Table 3.1: Experiment parameter settings for comparing SSA, CMGS and BTIGS

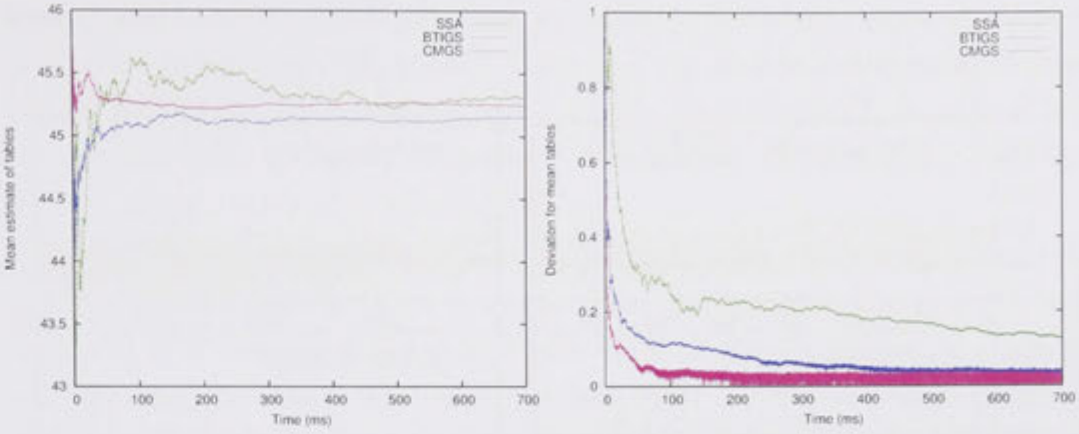
Setting No.	K	a	b
Setting No.1	50	0	10
Setting No.2	50	0	100
Setting No.3	50	0.5	100
Setting No.4	100	0	10
Setting No.5	100	0	100
Setting No.6	100	0.5	100

isolate the side effects in complex hierarchies, such as those caused by different implementations, different hierarchical modelling methods and different hyperparameter settings or estimations. Therefore, in order to reduce the side effects as much as we can, the three samplers are investigated in this section in a simply controlled environment of multinomial sampling, since the use of plain PDP or DP (*i.e.*, the PDP/DP without a hierarchical structure) on a discrete domain corresponds to multinomial sampling. In this way, we can do a precise qualitative comparison of these three samplers, which can further help explain their comparative performance in a more complex hierarchical context [Chen et al., 2011].

The goal of the following experiments is to compare the relative convergence speed of the three samplers in order to quantify the improvement of CMGS and BTIGS, compared with SSA. High precision of convergence is not a main concern here, since high precision would typically not be achieved in our hierarchical modelling context in which the DP and the PDP are normally used. Moreover, within the simply controlled environment, we can repeat dozens of Gibbs runs within a short time due to the fast computation. Therefore, the use of convergence diagnostics [Cowles and Carlin, 1996] or related theory is not required in this simple case.

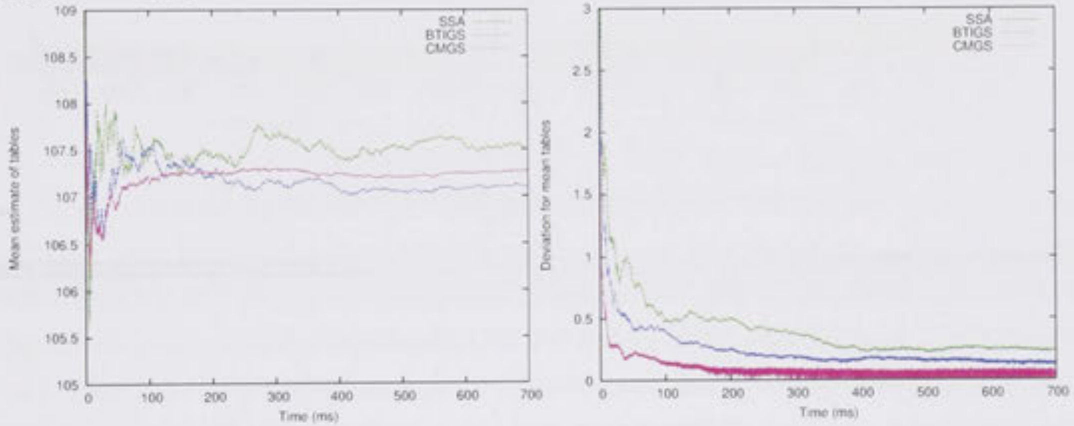
In all the experiments, the discount parameter a and the concentration parameter b are fixed, the base distribution is uniform on a fixed dimension K, denoted by $\boldsymbol{U}_K$. Table 3.1 shows six different parameter settings being used. For each of these parameter settings, 20 independent Gibbs runs are made. For each run, N samples are drawn from a single discrete probabilistic distribution $\boldsymbol{\mu}_K$ that is randomly sampled from the PDP as follows.

$$\begin{aligned} \boldsymbol{\mu}_K &\sim \mathrm{PDP}(a, b, \boldsymbol{U}_K) \\ \boldsymbol{n}_k &\sim \mathrm{multinomial}(\boldsymbol{\mu}_K, N) \,, \end{aligned}$$



(a) (b)

Figure 3.6: The plots of mean estimates of T for one of the 20 Gibbs runs (a) and the standard deviation of the 20 mean estimates (b) with $a = 0$, $b = 10$, $K = 50$ and $N = 500$.



(a) (b)

Figure 3.7: The plots of mean estimates of T for one of the 20 Gibbs runs (a) and the standard deviation of the 20 mean estimates (b) with $a = 0.5$, $b = 10$, $K = 50$ and $N = 500$.

where N is set to $10K$ in all the experiments, and the sum of entries in the counting vector $\boldsymbol{n}_K$ is equal to N.

The basic quantity estimated during each Gibbs run is the total number of tables T. For the six parameter settings, a rough determination is done for convergence time required in milliseconds. Let C indicate the sampler's convergence time. A burn-in for each individual Gibbs sampling run is done for $\frac{C}{10}$ milliseconds. Different convergence times are used for the different parameter settings. For $K = 50$, $C = 1000$ms, and for $K = 100$, $C = 10000$ms. Then the mean es-

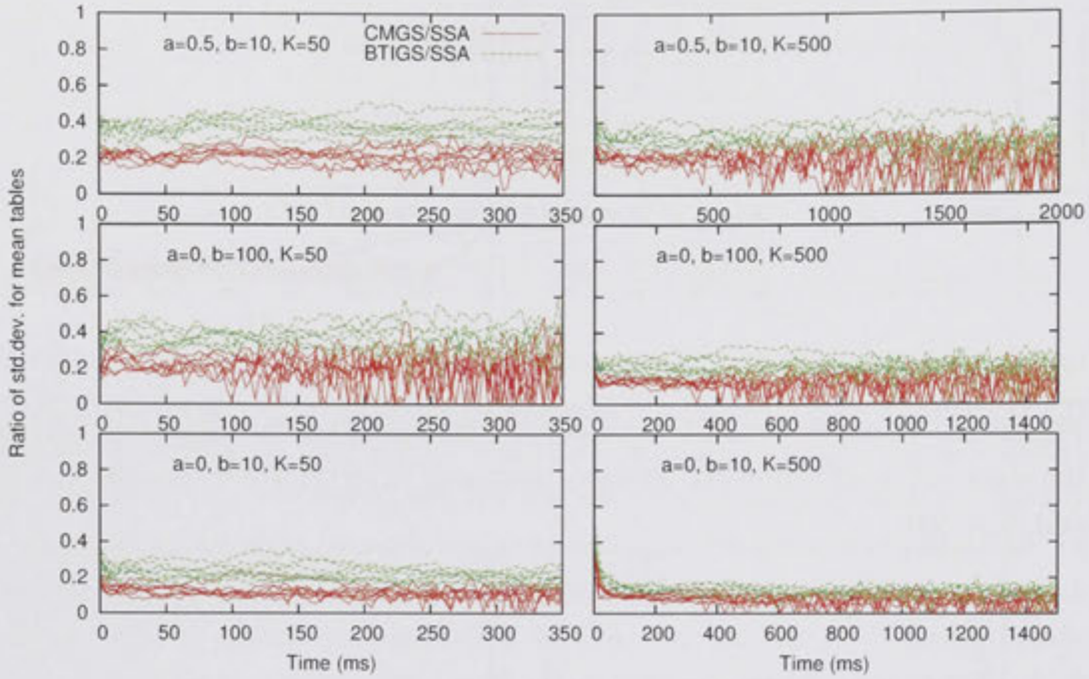


Figure 3.8: The relative standard deviations of T

timates of T from all major Gibbs cycles from burn-in up to the current time are recorded. In addition, since there are 20 independent Gibbs runs, a sample standard deviation of the 20 means is also recorded at corresponding cycles. The time series of means for individual runs and the sample standard deviations allow one to assess empirically how fast the Gibbs samplers are converging.

Figures 3.6 and 3.7 show examples of the time series of mean estimates of the total table counts and the time series of standard deviations of the 20 means with two different parameter settings. When $a = 0$, the PDP is indeed the DP. As we can see from these figures, regardless of the PDP or the DP, the mean estimates and standard deviations for BTIGS and CMGS become relatively stable more quickly than those for SSA, especially for BTIGS. Besides, both BTIGS and CMGS have smaller standard deviations than SSA, which indicates a faster convergence.

In order to further assess the relative performance of the three algorithms, the relative values of standard deviations are computed in ratios as

$$\frac{s.d._{CMGS}}{s.d._{CMGS} + s.d._{SSA}} \qquad \frac{s.d._{BTIGS}}{s.d._{BTIGS} + s.d._{SSA}},$$

which should be close to 0.5 if the two standard deviations are about equal. If the ratio score is less than 0.5, say 0.25, it means the standard deviation for either CMGS or BTIGS is about three times smaller than that for SSA, thus CMGS and BTIGS converge faster. Otherwise, the standard deviation for the SSA is smaller, which means SSA converges faster.

Figure 3.8 shows six plots for the six different parameter settings. Each plot overlays 9 time series of the ratio scores of the standard deviations on the mean estimates of T computed across 20 Gibbs runs for the 9 different data samples $\boldsymbol{n}_K$. We can see that in the context where both the concentration parameter and the discount parameter are fixed correctly, both CMGS and BTIGS are significantly faster than SSA, and CMGS has perhaps half the standard deviation of BTIGS. Moreover, the improvement of CMGS seems more pronounced with the higher dimension.

3.6 Gibbs Sampling for the CPDP

In Section 3.5, I showed the superiority of CMGS and BTIGS over SSA in a controlled environment of multinomial sampling. In this section, I will generalise CMGS and BTIGS to do posterior inference for the CPDP in a discrete space, where all probabilities are finite and discrete. For easy understanding, I will describe the two techniques in context of a DAG structure, *i.e.* Equation (2.11). Note it is also worth pointing out that a SSA based sampling algorithm can be found in [Wood and Teh, 2009], and Zhang et al. [2010] introduced a Gibbs sampling algorithm based on the stick-breaking construction discussed in Section 2.4.1.

The challenge of doing Gibbs sampling over the posterior of CPDP is to handle to multiple base measures. It is more complex than the PDP. In the Chinese restaurant metaphor for the CPDP embedded in a DAG structure (*i.e.*, the graphical PDP in Definition 2.6), all the restaurants (*i.e.*, nodes in the DAG) are linked to multiple parent restaurants. In each restaurant, dishes served in different floors are drawn from different parent restaurants. The number of floors is equal to the number of parents. Then, how can we decide from which parent a particular dish is drawn? or how can we decide to which parent the newly opened table is sent as a proxy customer?

Figure 3.9 shows an example of the CRP representation of the CPDP within a DAG structure. There are three restaurants, labeled with 1, 2 and 8, that correspond to three nodes in Figure 2.5, *i.e.*, G_1, G_2 and G_8 respectively. G_1 is

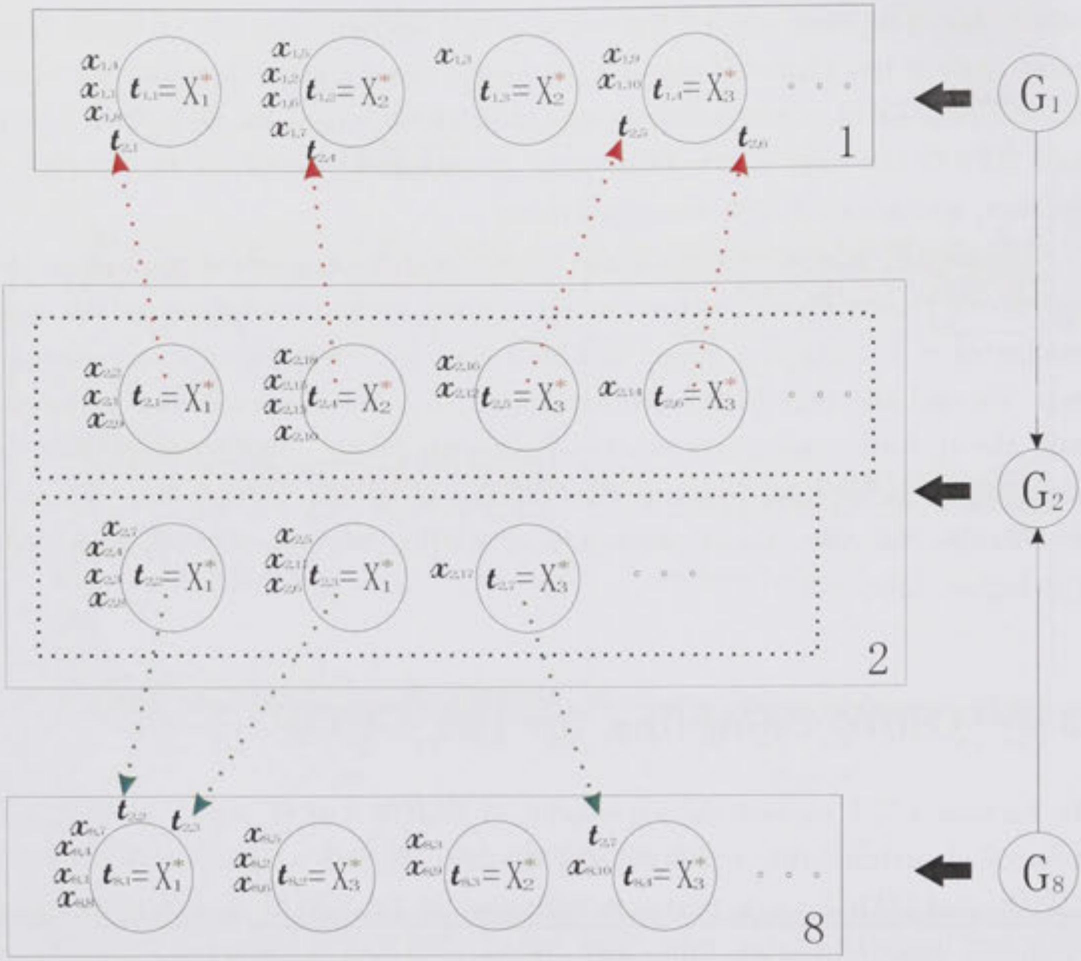


Figure 3.9: A CRP representation for a CPDP embedded in a simple DAG structure taken out from Figure 2.5. The red stars indicate that the dishes are drawn from G_1, and the green ones indicate that the dishes are drawn from G_8.

drawn from a CPDP with the admixture of G_2 and G_8 as base distributions. To demonstrate clearly the representation, I assume that all probability distributions associated with root nodes (nodes with no parents) in Figure 2.5 are drawn from the same CPDP with a single discrete base distribution H_0. Thus, atoms (*i.e.*, dishes in the global menu) drawn from H_0 can be shared among all the nodes (*i.e.*, restaurants). Therefore, dishes, denoted by X_k^* in Figure 3.9, with the same subscripts but different colored stars are the same dish drawn from the global menu. Different colors are just used to indicate these dishes are ordered through different parent restaurants and served by tables located in different floors in the restaurant. For example, X_1^* and X_1^* are the same dish, but ordered through restaurant 1 and restaurant 8 respectively. As indicated by dotted arrows with different colors, tables in different floors of restaurant 2 are sent as proxy customers to either restaurant 1 or restaurant 8. For example, $t_{2,2}$ is sent to restaurant 8, because the dish it serves is drawn from G_8.

Now, we are ready to modify CMGS and BTIGS algorithms to make them applicable to the CPDP. First, I adapt the *multiplicity* representation of the PDP to the CPDP. Refer to Equation (2.11), G_j is a random probability distribution associated with node j, $\mathrm{Pa}(j)$ is a set of parent nodes of j, each of which is indicated by G_i, (i,j) is the directed edge from node i to node j, and $\rho_{i,j}$ is the mixture weight on (i,j), *s.t.* $\sum_{i\in \mathrm{Pa}(j)} \rho_{i,j} = 1$.

Now, let $n^*_{j,k}$ be the count of customers eating X^*_k at node j, which includes the customers arriving by themselves and those sent by the child nodes of j ($\mathrm{Cd}(j)$), see Figure 3.9, and $t^*_{j,k}$ be the *table multiplicity*. Equation (3.3) can be modified to yield the joint posterior of all customer counts $\boldsymbol{n}_j$ and table multiplicities $\boldsymbol{t}^*_j$ for a CPDP at node j as

$$p(\boldsymbol{x}_j, \boldsymbol{t}^*_j \mid a_j, b_j, G_1, G_2, \cdots, G_{|\mathrm{Pa}(j)|}) = \frac{(b_j|a_j)_{T_j}}{(b_j)_{N_j}} \prod_{k=1}^{K} S^{n^*_{j,k}}_{t^*_{j,k}, a_j} \left(\sum_{i\in \mathrm{Pa}(j)} \rho_{i,j} G_i(X^*_k) \right)^{t^*_{j,k}} , \tag{3.11}$$

where $T_j = \sum_{k=1}^{K} t^*_{j,k}$ and $N_j = \sum_{k=1}^{K} n^*_{j,k}$. To expand the sum with multinomial identity, $t^*_{j,k}$ can be decomposed into parts, $s^*_{i,j,k}$'s, each of which indicates the number of tables serving X^*_k in the i^{th} floor of restaurant j, *s.t.* $s^*_{i,j,k} \geq 0$. The $s^*_{i,j,k}$ tables are sent to parent i as proxy customers. Thus, we have

$$\begin{aligned} t^*_{j,k} &= \sum_{i\in \mathrm{Pa}(j)} s^*_{i,j,k} & n^*_{j,k} &= n_{j,k} + \sum_{c\in \mathrm{Cd}(j)} s^*_{j,c,k} \\ n^*_{j,k} &\geq t^*_{j,k} & t^*_{j,k} &= 0 \text{ iff } n^*_{j,k} = 0 \ , \end{aligned}$$

where $n_{j,k}$ is the number of customers arriving by themselves. For example, there are 3 tables serving dish X^*_3 in restaurant 2 in Figure 3.9, *i.e.*, $t^*_3 = 3$. Two of them ($t_{2,5}$ and $t_{2,6}$) are sent to restaurant 1 and the left one ($t_{2,7}$) is sent to restaurant 8. Thus, $n^*_{1,3} = 2 + 2 = 4$, and $n^*_{8,3} = 4 + 1 = 5$.

As a consequence, we can decide with this decomposition how many tables serve a dish ordered from a specific parent. The problem of involving multiple base measures can now be solved. The final joint posterior distribution of $\boldsymbol{x}_j$ and

$\boldsymbol{s}^*_j$ (*i.e.*, $(s^*_{1,j,k}, s^*_{2,j,k}, \ldots, s^*_{|\mathrm{Pa}(j)|,j,k})$) is derived as

$$p\left(\boldsymbol{x}_j,\, \boldsymbol{s}^*_j \,|\, a_j,\, b_j,\, G_1, G_2, \cdots, G_{|\mathrm{Pa}(j)|}\right) = \\ \frac{(b_j|a_j)_{T_j}}{(b_j)_{N_j}} \prod_{k=1}^{K} S^{n^*_{j,k}}_{t^*_{j,k},a_j} C^{t^*_{j,k}}_{\boldsymbol{s}^*_{j,k}} \prod_{i\in\mathrm{Pa}(j)} \rho_{i,j}^{s^*_{i,j,k}} G_i(X^*_k)^{s^*_{i,j,k}} \tag{3.12}$$

where $C^{t^*_{j,k}}_{\boldsymbol{s}^*_{j,k}}$ is a multinomial coefficient which is computed as

$$C^{t^*_{j,k}}_{\boldsymbol{s}^*_{j,k}} = \frac{t^*_{j,k}!}{\prod_{i\in\mathrm{Pa}(j)} s^*_{i,j,k}!} \,.$$

Indeed, Equation (3.12) can be treated as a generalisation of Equation (3.3). Replace the equations in Algorithm 2, we can derive the Gibbs sampling algorithm for the CPDP.

Now, I generalise BTIGS for the CPDP. Unlike the *table indicator* representation for the PDP (see Section 3.4), given a multiplicity representation, there are $\prod_{k=1}^{K} C^{n^*_{j,k}}_{\boldsymbol{s}^*_{j,k},(n^*_{i,k}-t^*_{j,k})}$ choices of the table indicator configurations, where

$$C^{n^*_{j,k}}_{\boldsymbol{s}^*_{j,k},(n^*_{j,k}-t^*_{j,k})} = \frac{n^*_{j,k}!}{\left(\prod_{i\in\mathrm{Pa}(j)} s^*_{i,j,k}!\right)(n^*_{j,k}-t^*_{j,k})!} \,.$$

Therefore, the joinst posterior for the CPDP at node j based on the *multiplicity* representation can be reconstructed from that based on the *table indicator* representation as (similar to Equation (3.8))

$$\begin{aligned} & p\left(\boldsymbol{x}_j,\, \boldsymbol{s}^*_j \,|\, a_j,\, b_j,\, G_1, G_2, \cdots, G_{|\mathrm{Pa}(j)|}\right) \\ & = \left(\prod_{k=1}^{K} C^{n^*_{j,k}}_{\boldsymbol{s}^*_{j,k},(n^*_{j,k}-t^*_{j,k})}\right) p\left(\boldsymbol{x}_j,\, \boldsymbol{u}_j \,|\, a_j,\, b_j,\, G_1, G_2, \cdots, G_{|\mathrm{Pa}(j)|}\right), \end{aligned} \tag{3.13}$$

then, modifying Equation (3.12) with reference to Equation (3.13) gives the joint posterior distribution of $\boldsymbol{x}_j$ and table indicators $\boldsymbol{u}_j$ as follows

$$\begin{aligned} & p\left(\boldsymbol{x}_j,\, \boldsymbol{u}_j \,|\, a_j,\, b_j,\, G_1, G_2, \cdots, G_{|\mathrm{Pa}(j)|}\right) \\ & = \frac{(b_j|a_j)_{T_j}}{(b_j)_{N_j}} \prod_{k=1}^{K} S^{n^*_{j,k}}_{t^*_{j,k},a_j} \frac{C^{t^*_{j,k}}_{\boldsymbol{s}^*_{j,k}}}{C^{n^*_{j,k}}_{\boldsymbol{s}^*_{j,k},(n^*_{j,k}-t^*_{j,k})}} \prod_{i\in\mathrm{Pa}(j)} \rho_{i,j}^{s^*_{i,j,k}} G_i(X^*_k)^{s^*_{i,j,k}} \\ & = \frac{(b_j|a_j)_{T_j}}{(b_j)_{N_j}} \prod_{k=1}^{K} S^{n^*_{j,k}}_{t^*_{j,k},a_j} \frac{1}{C^{n^*_{j,k}}_{t^*_{i,k}}} \prod_{i\in\mathrm{Pa}(j)} \rho_{i,j}^{s^*_{i,j,k}} G_i(X^*_k)^{s^*_{i,j,k}} \end{aligned} \tag{3.14}$$

Here the definition of *table indicator* for the CPDP is slightly different from that for the PDP (*i.e.*, Definition 3.2). For the CPDP embedded in a DAG, there can be multiple parents for each node. Tables contributed by customers at one node can be sent to different parents. Thus, the values that the *table indicator* can take on should be the indices of all the parent nodes if a table is created. Otherwise, the *table indicator* is zero. For example, in Figure 2.5, if a customer contributes a new table at node G_5 and the new table is sent to node G_7 as a proxy customer, then the table indicator for this customer is $u = 7$. Note although I just elaborated how table indicator works for the CPDP, dynamically recording all the table indicators is not required in practice. Like BTIGS, we can randomly assign table indictors by sampling.

While adapting Algorithm 3 for doing sampling for the CPDP, we should pay attention to Step 2, sampling to remove a customer ($x_{j,n}$) eating dish X_k^* from node j (*i.e.*, restaurant j in the CRP representation), since the decomposition of $t_{j,k}^*$ makes the sampling more complex than in Algorithm 3. If $x_{j,n}$ has contributed a table, which floor is the table located in? Based on the recorded counts, $n_{j,k}^*$'s, $s_{i,j,k}^*$'s and $t_{j,k}^*$'s, we cannot tell the exact floor (recall that each floor corresponds to a parent) because there could be multiple floors serving X_k^*. For example, X_1^* and X_3^* are served in both floors in restaurant 2 in Figure 3.9. Consequently, it is necessary to consider all possibilities by computing the probabilities of allocating a table contributed by $x_{j,n}$ to any floors serving X_k^*. That is, given $x_n = X_k^*$ and all the counts, we have

$$p(u_{j,n} = i \,|\, x_{j,n} = X_k^*) = \frac{s_{i,j,k}^*}{n_{j,k}^*} \qquad \text{for } i \in \mathrm{Pa}(j) \tag{3.15}$$

$$p(u_{j,n} = 0 \,|\, x_{j,n} = X_k^*) = \frac{n_{j,k}^* - t_{j,k}^*}{n_{j,k}^*} \tag{3.16}$$

Finally, sampling for the CPDP at each node, we can adapt Algorithm 3 by replacing Equation (3.9) with Equation (3.14), and Equations (3.10) with Equations (3.15) and (3.16). However, to do sampling in the whole DAG, one needs to modify Algorithm 3 with recursions. I will give a concrete example in Chapter 7 by embedding the CPDP in a document structure.

To deal with the mixture weights, $\boldsymbol{\rho}_j$, we can predefine the weights with respect to how *important* each parent node i is to the node j, or we can even make $\boldsymbol{\rho}_j$ uniformly distributed. The approach adopted here is to put either an informative or a non-informative prior on $\boldsymbol{\rho}_j$. In regard to *Dirichlet-Multinomial* conjugacy and applications to discrete domains, such as language processing, a $|\mathrm{Pa}(j)|$-dimension Dirichlet distribution is used. *i.e.*, $\boldsymbol{\rho}_j \sim \mathrm{Dir}_{|\mathrm{Pa}(j)|}(\boldsymbol{\varrho})$. With $\boldsymbol{\rho}_j$ marginalised out, Equations (3.12) and (3.14) can be further changed to respectively

$$\begin{aligned} & p\left(\boldsymbol{x}_j, \boldsymbol{t}_j^* \,|\, a_j, b_j, G_1, G_2, \cdots, G_{|\mathrm{Pa}(j)|}\right) \\ = \; & \frac{\Gamma\left(\sum_{i\in\mathrm{Pa}(j)} \varrho_{i,j}\right)}{\prod_{i\in\mathrm{Pa}(j)} \Gamma(\varrho_{i,j})} \frac{\prod_{i\in\mathrm{Pa}(j)} \Gamma\left(\varrho_{i,j} + \sum_{k=1}^{K} s^*_{i,j,k}\right)}{\Gamma\left(\sum_{i\in\mathrm{Pa}(j)} \varrho_{i,j} + \sum_{k=1}^{K} t^*_{j,k}\right)} \\ & \frac{(b_j|a_j)_{T_j}}{(b_j)_{N_j}} \left(\prod_{k=1}^{K} S^{n^*_{j,k}}_{t^*_{j,k},a_j} C^{t^*_{j,k}}_{\boldsymbol{s}^*_{j,k}} \prod_{i\in\mathrm{Pa}(j)} G_i(X^*_k)^{s^*_{i,j,k}}\right) \end{aligned} \tag{3.17}$$

$$\begin{aligned} & p\left(\boldsymbol{x}_j, \boldsymbol{u}_j \,|\, a_j, b_j, G_1, G_2, \cdots, G_{|\mathrm{Pa}(j)|}\right) \\ = \; & \frac{\Gamma\left(\sum_{i\in\mathrm{Pa}(j)} \varrho_{i,j}\right)}{\prod_{i\in\mathrm{Pa}(j)} \Gamma(\varrho_{i,j})} \frac{\prod_{i\in\mathrm{Pa}(j)} \Gamma\left(\varrho_{i,j} + \sum_{k=1}^{K} s^*_{i,j,k}\right)}{\Gamma\left(\sum_{i\in\mathrm{Pa}(j)} \varrho_{i,j} + \sum_{k=1}^{K} t^*_{j,k}\right)} \\ & \frac{(b_j|a_j)_{T_j}}{(b_j)_{N_j}} \left(\prod_{k=1}^{K} S^{n^*_{j,k}}_{t^*_{j,k},a_j} \left(C^{n^*_{j,k}}_{t^*_{j,k}}\right)^{-1} \prod_{i\in\mathrm{Pa}(j)} G_i(X^*_k)^{s^*_{i,j,k}}\right) \end{aligned} \tag{3.18}$$

The differences between the Gibbs sampling scheme proposed in [Wood and Teh, 2009] and those discussed above reside in the differences between the SSA and the CMGS/BTIGS. Wood and Teh's scheme is based on a modified version of the SSA (see Equation 3.1) according to the multi-floor Chinese restaurant franchise representation, which is a direct extension of the Chinese restaurant franchise representation in [Teh et al., 2006]. In their method, we have to do bookkeeping of *menu indicator* variables (*i.e.*, floor variables in [Wood and Teh, 2009]) as discussed in Section 2.4, when unseating and reseating customers. The purpose of the bookkeeping is to keep track of the parent restaurants to which each table should be sent as a proxy customer in the DAG structure. Therefore, in the sampling procedure, if unseating (reseating) a customer causes removing (adding) a table in a restaurant, it is essential to recursively sample to remove (add) a proxy customer (and a table if necessary) to the corresponding parent restaurant, see the predictive distribution shown in Equation 2.13.

However, the schemes based on the CMGS/BTIGS do not require to maintain the *menu indicator* variables, since both the CGMS and BTIGS have integrated out all the possible seating arrangements, refer to Sections 3.3 and 3.4. All need to be kept are table counts, *i.e.*, $s^*_{i,j,k}$'s in each restaurant. In order to compute the table counts, we must recursively check whether a table needs to be removed (added) from a restaurant and its corresponding parent restaurants, if a customer is unseated (reseated). Specifically, if a table is added or removed from a restaurant, we need to consider all the possible linked paths from this restaurant towards the root, where a proxy customer (and a table if necessary) can be recursively added or removed. Because *menu indicator* variables are not dynamically recorded, one needs to sample over all the possible paths to add or remove a table. See for example the inference scheme for the AdaTM in Chapter 7, which used Equation 3.18 based on the BTIGS.

3.7 Summary

In this Chapter, I have reviewed Teh's sampling for seating arrangement (SSA) sampler, and introduced the collapsed multiplicity Gibbs sampler (CMGS) and the blocked table indicator Gibbs sampler (BTIGS). The results of experiments run in a simply controlled environment of multinomial sampling have preliminarily shown that the CMGS and BTIGS converges much faster than the SSA does. It would be very interesting to further compare the three samplers in different contexts, for instance, to compare the three samplers in topic models or the word segmentation models by [Goldwater et al., 2009].

The techniques for doing posterior inference with networks of PDPs or CPDPs can be readily developed from these likelihoods, *i.e.*, Equations (3.6), (3.9), (3.12) and (3.14). In Chapters 5 , 6 and 7, I will show these can be used to do inference for structured topic models.

Chapter 4

Probabilistic Topic Modelling

Topic modelling is an increasingly useful class of techniques for analysing not only large unstructured documents but also data that posit "*bag-of-words*" assumption, such as genomic data [Flaherty et al., 2005] and discrete image data [Wang and Grimson, 2008]. As a promising unsupervised learning approach with wide application areas, it has gained significant momentum recently in machine learning, data mining and natural language processing communities. In this chapter, I discuss briefly the fundamentals (*e.g.*, basic idea and posterior inference) of topic models, especially Latent Dirichlet Allocation (LDA) by Blei et al. [2003] that acts as a benchmark model in the topic modelling community, since these are the important prerequisites for understanding the structured topic models that will be developed in Chapters 5, 6 and 7.

This chapter is organised as follows. The basic idea of the probabilistic topic models is discussed in Section 4.1. Section 4.2 gives an fairly detailed introduction to LDA, then the Gibbs sampling algorithm for LDA is presented in Section 4.3. Finally, I will discuss applications of topic models in Section 4.4.

4.1 Probabilistic Topic Models

Probabilistic topic models [Deerwester et al., 1990; Hofmann, 1999, 2001; Blei et al., 2003; Girolami and Kabán, 2003; Buntine and Jakulin, 2006; Steyvers and Griffiths, 2007; Blei and Lafferty, 2009; Heinrich, 2008] are a discrete analogue to principal component analysis (PCA) and independent component analysis (ICA) that model *topic* at the word level within a document [Buntine, 2009]. They have many variants such as Non-negative Matrix Factorisation (NMF) [Lee and Seung, 1999], Probabilistic Latent Semantic Indexing (PLSI) [Hofmann, 1999] and LDA [Blei et al., 2003], and have applications in fields such as genetics [Pritchard et al., 2000; Flaherty et al., 2005], text and the web [Wei and Croft, 2006; Bíró et al., 2008], image analysis [Li and Perona, 2005; Wang and Grimson, 2008; He and Zemel, 2008; Cao and Fei-Fei, 2007; Wang et al., 2009a], social networks [McCallum et al., 2007; Mei et al., 2008] and recommender systems [Pennacchiotti and Gurumurthy, 2011]. A unifying treatment of these models and their relationship to PCA and ICA is given by Buntine and Jakulin [2006]. The first Bayesian treatment was due to Pritchard et al. [2000] and the broadest model is the Gamma-Poisson model of Canny [2004].

Specifically, probabilistic topic models are a family of generative models for learning the latent semantic structure of a corpus by using of a hierarchical Bayesian analysis of the text content. Their fundamental idea is that each document is a convex mixture of latent topics, each of which is a probability distribution over words in a vocabulary. Clearly, a topic model is a factor model that specifies a simple probabilistic process by which documents can be generated. It reduces the complex process of generating a document to a small number of probabilistic steps by assuming exchangeability.

To generate a new document, a distribution over topics (*i.e.*, a *topic distribution*) is first drawn from a probability distribution over a measurable space. Then, each word in the document is drawn from a *word distribution* associated with a topic that is drawn from the generated *topic distribution*. The semantic properties of words and documents can be expressed in terms of probabilistic topics. Let $\boldsymbol{\mu}$ be document specific *topic distribution*, $\boldsymbol{\phi}_{1:K}$ be topic specific *word distributions*, z_i be a topic associated with word w_i, where $z_i \in \{1, \dots, K\}$, a topic model can be interpreted in term of a mixture model as

$$\begin{aligned} w_i \mid \boldsymbol{\Phi},\ z_i &\sim F(\phi_{z_i}) && \text{for } i = 1, 2, \dots, n \\ z_i \mid \boldsymbol{\mu} &\sim \boldsymbol{\mu} && \text{for } i = 1, 2, \dots, n, \end{aligned}$$

where $F(\cdot)$ is set in general to a multinomial distribution, and a Dirichlet distribution (see Chapter 2) is put as a prior on $\boldsymbol{\mu}$.

Applying standard Bayesian inference techniques, we can invert the generative process to infer a set of optimal latent topics that maximises the likelihood (or the posterior probability) of a collection of documents. Compared with the purely spatial representation (*e.g.*, Vector Space Model [Salton and McGill, 1986]), the superiority of representing the content of words and documents in means of probabilistic topics is that each topic can be individually interpretable as a probability distribution over words, it picks out a coherent cluster of correlated terms

[Steyvers and Griffiths, 2007]. We should also note that each word can appear in multiple clusters, just with different probabilistic weights, which indicates topic models could be able to capture polysemy [Steyvers and Griffiths, 2007]. The generative process is purely based on the "*bag-of-words*" assumption where only word occurrence information (*i.e.*, frequencies) is taken into consideration. This well corresponds to the assumption of *exchangeability* in Bayesian statistics. However, word-order is ignored even though it might contain important contextual cues to the original content.

As a probabilistic generative process, variants and extensions of topic models can be used to postulate complex latent semantic structures responsible for a collection of documents, making it possible to use Bayesian inference to recover those structures. The goal of fitting those topic models is to find the best set of latent topics that can well explain the observed data (*e.g.*, documents). In topic modelling literature, there are two ways in general to do approximate posterior inference, one is variational inference [Jordan et al., 1999], the other is Gibbs sampling [Neal, 2000; Robert and Casella, 2005]. The latter is discussed in Section 4.3.

4.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) [Blei et al., 2003], a full Bayesian extension of PLSI, is a three-level hierarchical Bayesian model for collections of discrete data, *e.g.*, documents. It is also known as multinomial PCA [Buntine, 2002].

Compared with PLSI, LDA puts a Dirichlet prior on *topic distributions*, which overcomes the difficulty, faced by PLSI, in the generalisability of modelling the unseen documents. Girolami and Kabán [2003] showed that PLSI is a maximum a posterior estimate of LDA with a uniform Dirichlet prior. Choosing the Dirichlet prior simplifies the problem of posterior inference due to the *Dirichlet-Multinomial* conjugacy, see Property 2.2 in Chapter 2. Moreover, if the Kullback-Leibler measure is used, instead of least square, then NMF behaves like a maximum likelihood version of LDA.

As a fundamental model for topic modelling, LDA is usually used as a benchmark model in the empirical comparison with its various extensions. Figure 4.1 illustrates its graphical representation using plate notation (see [Buntine, 1994] for an introduction). In this notation, shaded and unshaded nodes indicate observed and unobserved (*i.e.*, latent or hidden) variables respectively; arrows in-

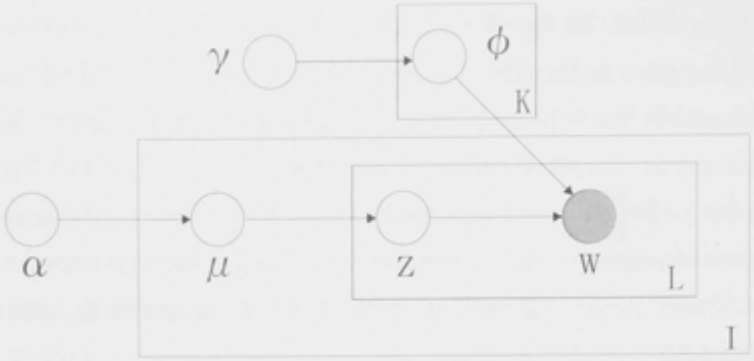


Figure 4.1: Latent Dirichlet allocation

dicate conditional dependencies among variables; and plates indicate repeated sampling.

For document analysis, LDA is a hidden variable model of documents. The observed data are words $\boldsymbol{w}$ of each document, the two hidden variables are $\boldsymbol{\mu}$ (*the topic distribution*) and z (the *word-topic assignment*), and the model parameters are the Dirichlet prior $\boldsymbol{\alpha}$ and $\boldsymbol{\phi}_{1:K}$ (*word distributions*). To generate documents, LDA assumes the following generative process:

1. For each topic k where $k \in \{1, \dots, K\}$
    (a) Draw word distribution $\boldsymbol{\phi}_k \sim \mathrm{Dir}_V(\boldsymbol{\gamma})$
2. For each document $i \in \{1, \dots, I\}$
    (a) Draw topic distribution $\boldsymbol{\mu}_i \mid \boldsymbol{\alpha} \ \sim \ \mathrm{Dir}_K(\boldsymbol{\alpha})$
    (b) For each word $w_{i,l}$ in i, where $i \in \{1, \dots, L_i\}$
        i. Draw a topic $z_{i,l} \mid \boldsymbol{\mu}_i \sim \mathrm{Discrete}(\boldsymbol{\mu}_i)$
        ii. Draw a word $w_{i,l} \mid z_{i,l}, \ \boldsymbol{\phi}_{1:K} \ \sim \ \mathrm{Discrete}(\boldsymbol{\phi}_{z_{i,l}})$.

Here, the hyper-parameter $\boldsymbol{\gamma}$ is a Dirichlet prior on *word distributions* (*i.e.*, a Dirichlet smoothing on the multinomial parameter $\boldsymbol{\phi}_k$ [Blei et al., 2003]), and $\mathrm{Dir}_K(\cdot)$ indicates a K-dimensional Dirichlet distribution. The model parameters can be estimated from data. The hidden variables can be inferred for each document by simply inverting the generative process. These hidden variables are useful for ad-hoc document analysis, for example, information retrieval [Wei and Croft, 2006] and document summarisation [Arora and Ravindran, 2008a,b]. With this process, LDA models documents on a low-dimensional topic space[1], which

[1] Note the number of topics associated with a document collection is usually far smaller than the vocabulary size, since documents in a collection tend to be heterogeneous.

provides not only an explicit semantic representation of a document, but also a hidden topic decomposition of the document collection [Blei and Lafferty, 2009].

Given the Dirichlet priors $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$, and the observed documents $\boldsymbol{w}_{1:I}$, the joint distribution of both the observed and the hidden variables can be read directly from Figure 4.1 using distributions given in the above generative process as:

$$\begin{aligned} & p(\boldsymbol{\mu}_{1:I},\ \boldsymbol{z}_{1:I},\ \boldsymbol{w}_{1:I} \mid \boldsymbol{\alpha},\ \boldsymbol{\gamma}) \\ = & \prod_{k=1}^{K} p(\boldsymbol{\phi}_k \mid \boldsymbol{\gamma}) \prod_{i=1}^{I} p(\boldsymbol{\mu}_i|\boldsymbol{\alpha}) \prod_{l=1}^{L_i} p(z_{i,l}|\boldsymbol{\mu}_i) p(w_{i,l}|\boldsymbol{\phi}_{z_{i,l}}) \ . \end{aligned} \tag{4.1}$$

The variables $\boldsymbol{\phi}_{1:K}$ are corpus level variables, which are assumed to be sampled once for the corpus; document level variables $\boldsymbol{\mu}_i$'s are sampled once for each document; and variables $z_{i,l}$'s are word level variables that are sampled once per word in each document.

Given the observed documents $\boldsymbol{w}_{1:I}$, the task of Bayesian inference is to compute the posterior distribution over the model parameters $\boldsymbol{\phi}_{1:K}$ and the hidden variables, $\boldsymbol{\mu}_{1:I}$, and $\boldsymbol{z}_{1:I,1:L}$. The posterior is

$$\begin{aligned} & p(\boldsymbol{\mu}_{1:I}, \boldsymbol{z}_{1:I},\ \boldsymbol{\phi}_{1:K} \mid \boldsymbol{w}_{1:I},\ \boldsymbol{\alpha},\ \boldsymbol{\gamma}) \\ = & \frac{p(\boldsymbol{\mu}_{1:I},\ \boldsymbol{z}_{1:I},\ \boldsymbol{w}_{1:I} \mid \boldsymbol{\alpha},\ \boldsymbol{\gamma})}{\int_{\boldsymbol{\mu}} \int_{\boldsymbol{\phi}} \sum_{\boldsymbol{z}} p(\boldsymbol{\mu}_{1:I},\ \boldsymbol{z}_{1:I},\ \boldsymbol{w}_{1:I} \mid \boldsymbol{\alpha},\ \boldsymbol{\gamma})} \ . \end{aligned}$$

Although LDA is a relatively simple model, a direct computation of this posterior is infeasible due to the summation over topics in the integral in the denominator. Training LDA on a large collection with millions of documents can be challenging and efficient exact algorithms have not been found [Buntine, 2009]. Therefore, one has to appeal to approximate inference algorithms and the following methods are used, *i.e.*, the mean field variational inference [Blei et al., 2003], the collapsed variational inference [Teh et al., 2007], the expectation propagation [Minka and Lafferty, 2002], and Gibbs sampling [Griffiths and Steyvers, 2004]. Buntine and Jakulin [2006] have given a fairly detailed discussion on some of those methods. They also mentioned some other methods, such as the direct Gibbs sampling by Pritchard et al. [2000] and Rao-Blackwellised Gibbs sampling by Casella and Robert [1996].

Furthermore, Wallach et al. [2009] have studied several classes of structured priors for LDA, *i.e.*, asymmetric or symmetric Dirichlet priors on $\boldsymbol{\mu}$ and $\boldsymbol{\phi}$. They have shown that LDA with an asymmetric prior on $\boldsymbol{\mu}$ significantly outperforms that with a symmetric prior. However, there is no benefit while putting an asymmetric prior on $\boldsymbol{\phi}$. Sato and Nakagawa [2010] have further put a PDP prior (see

Section 2.3) on $\boldsymbol{\phi}$ to introduce the power-law phenomenon of a word distribution in topic models. They have shown a better performance than the standard LDA.

Out of all the proposed approximate inference algorithms, each of which has advantages and disadvantages, a thorough comparison of these algorithms is not a goal of this thesis. Hereafter I will focus on the collapsed Gibbs sampling algorithm introduced in [Griffiths and Steyvers, 2004], details can be found in [Steyvers and Griffiths, 2007]. The collapsed Gibbs sampler is found to be good as others. It is also general enough to be a good base for extensions of LDA.

4.3 Approximate Inference via Gibbs Sampling

Since we can easily write down the full conditional distribution $p(z_{i,l}|\boldsymbol{z}^{-z_{i,l}}, \boldsymbol{w})$ by marginalising out the document-topic distributions, *i.e.*, $\boldsymbol{\mu}_{1:I}$, from the joint distribution, Equation (4.1), it is straightforward to use Gibbs sampling [Geman and Geman, 1990], a special case of the Metropolis-Hastings algorithm in the Markov chain Monte Carlo (MCMC) family. The collapsed Gibbs sampling algorithm for LDA marginalises out $\boldsymbol{\mu}_{1:I}$ and $\boldsymbol{\phi}_{1:K}$, instead of explicitly estimating them. The strategy of marginalising out some hidden variables is usually referred to as "collapsing" [Neal, 2000], which is the same as Rao-Blackwellised Gibbs sampling [Casella and Robert, 1996]. The collapsed algorithm samples in a collapsed space, rather than sampling parameters and hidden variables simultaneously [Teh et al., 2007]. So, Griffiths and Steyvers' algorithm is also known as a collapsed Gibbs sampler.

The principle of Gibbs sampling is to simulate the high-dimensional probability distribution by conditionally sampling a lower-dimensional subset of variables via a Markov chain, given the values of all the others fixed. The sampling proceeds until the chain becomes stable (*i.e.*, after the so-called "*burn-in*" period, the chain will burn-in to a stable local optimum). Theoretically, the probability distribution drawn from the chain after the "*burn-in*" period will asymptotically approach the true posterior distribution. In regard to LDA, the collapsed Gibbs sampler considers all word tokens in a document collection, and iterates over each token to estimate the probability of assigning the current token to each topic, conditioned on topic assignments of all the other tokens.

To derive the full conditional distributions, we need first to compute the joint distribution given in Equation (4.1) by using the Dirichlet integral. Let $\boldsymbol{n}_k = (n_{k,1}, n_{k,2}, \ldots, n_{k,V})$ where $n_{k,v}$ is the number of times word v is assigned a topic k; and $\boldsymbol{m}_i = (m_{i,1}, m_{k,2}, \ldots, m_{i,K})$ where $m_{i,k}$ is the number of word tokes in document i to which topic k is assigned. Thus, given all the documents, Equation (4.1) is further computed as

$$p(\boldsymbol{z}_{1:I},\ \boldsymbol{w}_{1:I} \mid \boldsymbol{\alpha},\ \boldsymbol{\gamma}) = \prod_{k=1}^{K} \frac{\text{Beta}_V(\boldsymbol{\gamma} + \boldsymbol{n}_k)}{\text{Beta}_V(\boldsymbol{\gamma})} \prod_{i=1}^{I} \frac{\text{Beta}_K(\boldsymbol{\alpha} + \boldsymbol{m}_i)}{\text{Beta}_K(\boldsymbol{\alpha})} \tag{4.2}$$

With a simple cancelation of Equation (4.2), the full conditional distribution can be derived as

$$p(z_{i,l} = k | \boldsymbol{z}_{1:I}^{-z_{i,l}}, \boldsymbol{w}_{1:I}, \boldsymbol{\alpha}, \boldsymbol{\gamma}) \propto \frac{n_{k,w_{i,l}} + \gamma_{w_{i,l}}}{\sum_{v=1}^{V} (n_{k,v} + \gamma_v)} \frac{m_{i,k} + \alpha_k}{\sum_{k'=1}^{K} (m_{i,k'} + \alpha_{k'})}\ , \tag{4.3}$$

After a sufficient number of Gibbs iterations, which means the sampler has burned-in, the Markov chain is ready to sample. Given the posterior sample statistics, the latent variable $\boldsymbol{\mu}$ and the model parameter $\boldsymbol{\phi}$ can be estimated using the expectation of the Dirichlet distribution (see Section 2.1) as:

$$\phi_{k,v} = \frac{n_{k,v} + \gamma_v}{\sum_{v'=1}^{V} (n_{k,v'} + \gamma'_v)} \qquad \mu_{i,k} = \frac{n_{i,k} + \alpha_k}{\sum_{k'=1}^{K} (n_{i,k'} + \alpha_{k'})}$$

Due to the extensive computations required by the topic sampling for each word token, particularly while the number of topics and the corpus size are large (which is usually the case in real applications), Porteous et al. [2008] presented a fast collapsed Gibbs sampling algorithm, an efficient variant of Griffiths and Steyvers' sampler. This fast version significantly reduces the sampling operations based on the notion of skewed sampling distribution, which means the probability mass is always put on a small fraction of K topics. With the same motivation, Xiao and Stibor [2010] gave another version of fast sampling method that puts a multinomial distribution on the number of times each word type is sampled in a document.

4.4 Applications and Extensions

Since the first introduction of topic models, particularly PLSI and LDA, they have been broadly applied for machine learning and data mining, particularly in information retrieval, text analysis and computer vision. For instance,

Information retrieval Azzopardi et al. [2004]; Buntine et al. [2004]; Wei and Croft [2006]; Chemudugunta et al. [2007]; Tang et al. [2011] have adapted topic models to information retrieval. Some of them have shown that topic model based information retrieval methods can outperform alternative methods that are based on, for example, Latent Semantic Indexing [Deerwester et al., 1990], the mixture of uni-grams model [McCallum, 1999]. Bíró et al. [2008] employed a modified LDA, so-called multi-corpus LDA, to handle the problem of web spam filtering, which demonstrates a relative improvement over a strong content and link feature baseline.

Text analysis Multi-document summarisation is an interesting application of topic models in the text analysis domain. It is an automatic procedure that aims at the identification of the essence of a set of related documents. The LDA mode can be used to decompose a document collection into different topics, and then find the sentences that adequately represent these topics. Arora and Ravindran [2008a,b] described an approach that uses LDA to capture the underlying topics of a set of documents, and further uses Singular Value Decomposition (SVD) to find the most orthogonal representation (topic vector) of sentences. Those sentences can be chosen to be put together as a summarisation. Purver et al. [2006]; Misra et al. [2009]; Blei and Moreno [2001] have further used topic models in semantic text segmentation. The segment boundaries are determined based on the topic change. Some other applications of topic models in text analysis are such as sentiment analysis [Mei et al., 2007; Titov and McDonald, 2008a,b; Lin and He, 2009; Brody and Elhadad, 2010], sparse text classification (*e.g.*, twitter [Ritter et al., 2010], web segments [Phan et al., 2008] and microblogs [Ramage et al., 2010]), entity resolution [Bhattacharya and Getoor, 2006], and word sense disambiguation [Boyd-Graber et al., 2007].

Computer vision Topics models have also been adapted for computer vision. For example, LDA has been used to discover objects from images [Cao and Fei-Fei, 2007], and to classify images into different categories [Li and Perona, 2005], and to assort human actions [Niebles et al., 2008]. In particular, Wang and Grimson [2008] proposed a Spatial Latent Dirichlet Allocation (SLDA) model which encodes spatial structures among visual words (or image patches). It partitions the visual words that are close in space into the same documents. In discovering objects from a collection of images, SLDA outperforms LDA.

Furthermore, standard topic models, especially LDA, have been extended in several ways to relax assumptions (*i.e.*, "*bag-of-words*" and the fixed number of topics) or to incorporate beyond the "*bag-of-words*" information.

In order to model how topics evolve over time in large sequentially organised documents, [Blei and Lafferty, 2006b] introduced a dynamic topic model (DTM) which removes the document exchangeability assumption (*i.e.*, the joint posterior distribution is invariant to permutations of the ordering of documents [Blei, 2011]) made by LDA. The DTM puts a random walk model on the natural parameters of multinomial distributions (denoted by $\boldsymbol{\beta}_{1:K}$) to model sequential data dependencies. In the DTM, corpus is divided by time slice. Documents within each time slice are modelled with a K-component LDA. Topics associated with time slice t evolve from those associated with time slice $t-1$ [Blei and Lafferty, 2009]. The DTM chains the natural parameters for each topic k ($\boldsymbol{\beta}_{t,k}$) at different time slices in a random walk model that evolves with Gaussian noise as

$$\boldsymbol{\beta}_{t,k} \,|\, \boldsymbol{\beta}_{t-1,k} \sim \mathcal{N}(\boldsymbol{\beta}_{t-1,k}, \sigma^2 I).$$

It is clear that the natural parameters at time slice $t-1$ are the expectation for the distribution of the natural parameters at time slice t, and the correlation of samples from the above distribution is controlled through adjusting the distribution variance. Then, $\boldsymbol{\beta}_{t,k}$ is mapped to the multinomial mean parameters $\boldsymbol{\phi}_{t,k}$ by

$$\phi_{t,k,w} = \frac{e^{\beta_{t,k,w}}}{\sum_{w'} e^{\beta_{t,k,w'}}}.$$

However, the nonconjugacy of the Gaussian and the multinomial makes exact posterior inference intractable. The authors adapted Kalman Filtering to do an approximated inference. Here, we should note that the DTM allows topics themselves to change over time.

In the DTM, data are required to be divided into discretised time slices. Wang et al. [2008a] argued that "*the choice of discretisation affects the memory requirements and computational complexity of posterior inference*". They further generalised the DTM to handle the continuous time space using a Brownian motion model. In the continuous DTM, the natural parameters of the multinomial distributions evolve as

$$\boldsymbol{\beta}_{t,k} \,|\, \boldsymbol{\beta}_{t-1,k} \sim \mathcal{N}(\boldsymbol{\beta}_{t-1,k}, v\Delta_t I),$$

where Δ_t is the elapsed time between time points $t-1$ and t. Thus, we can see that the difference between the continuous DTM and the DTM resides in the way of handling time. Other models following the DTM are used quite often in data mining to analyse streamed data to identify topic trends, *e.g.*, the on-line

LDA model [AlSumait et al., 2008], topics over time model [Wang and McCallum, 2006], the inheritance topic model [He et al., 2009], the Markov topic model [Wang et al., 2009b] and the dynamic mixture model [Wei et al., 2007].

The "*bag-of-words*" assumption made by LDA assumes the order of the words in each document does not matter, but this assumption is sometimes unrealistic and can cause to mistakenly neglect the important contextual information conveyed by word-orders. Griffiths et al. [2005] and Wallach [2006] presented two extensions of LDA to model words unexchangeably. Griffiths et al. [2005] used a combined model to capture the syntactic (word-orders) and semantic (topics) by alternating between a standard HMM and LDA. While Wallach [2006] inserted a Dirichlet bigram language model [Mackay and Peto, 1995] into LDA to generate topics conditioned on the context. A similar independence assumption is also made on topics, which says the learned topics are unrelated to each other. However, considering topic correlations may give us a rich posterior topic structure. These models include the correlated topic model (CTM) [Blei and Lafferty, 2006a], the Pachinko allocation model [Li and McCallum, 2006] and the hierarchical LDA (hLDA) [Blei et al., 2010] and so on.

In the CTM, topic proportions are modelled by a logistic normal distribution that allows for covariance structure among topics, instead of a Dirichlet distribution. Now, the topic proportion $\boldsymbol{\mu}$ in Figure 4.1 is generated by mapping a multivariate random variable from $\mathbb{R}^K$ to the K-simplex as follows:

$$\boldsymbol{\mu}' \mid \eta, \Sigma \sim \mathcal{N}(\eta, \Sigma) \qquad\qquad \boldsymbol{\mu}_k = \frac{e^{\mu'_k}}{\sum_{k'} e^{\mu'_{k'}}},$$

where $\{\eta, \Sigma\}$ is a K-dimensional mean and covariance matrix, in which each entry specifies the correlation between a pair of topics. Clearly, the CTM uses the covariance of the Gaussian to model the correlations between topics. Thus topics are allowed to be correlated to each other. One should also note that the number of parameters in the covariance matrix grows as $O(K^2)$. The PAM captures the topic correlations with a directed acyclic graph. It extends to the concept of topic to be a distribution not only over words, but also over interior topics, see Section 2.4.3. The hLDA is built on top of nested Chinese restaurant process (nCRP) which is defined as "*a stochastic process that assigns probability distributions to ensembles of infinitely deep, infinitely branching trees*" [Blei et al., 2010]. In the hLDA, topics are organised in a tree hierarchy on which nCRP is used as a prior. To generate a document, the hLDA first draws a topic path from the tree, then samples topics from the path. In this way, the hLDA can cluster documents according to the topic tree with multiple levels of abstraction.

How to incorporate meta-information (besides time) into topic modelling is another line of research that is interesting in, for example, computer vision and text mining, where the collected data usually come with meta-information, *e.g.*, class labels, review rating, authors, and citations. The well-known models for this kind of research include

The supervised LDA model [Blei and McAuliffe, 2007] that puts a logistic regression on the word-topic assignments to generate observed features, such as class labels;

The Dirichlet-multinomial regression model [Mimno and McCallum, 2008] that can in principal incorporate arbitrary features;

The correlated labelling model [Wang et al., 2008b] that builds directly the class label into the generative process;

The author-topic model [Rosen-Zvi et al., 2004; Steyvers et al., 2004] in which the word-topic assignments are generated according to the topics distributions associated with different authors;

The linked-LDA model [Nallapati et al., 2008] that jointly models the text and citations.

4.5 Summary

In conclusion, topic models have broad applications across different disciplines, generally from machine learning to data mining. Although these models are slightly different in the sense of assumptions, they share the same fundamental idea: mixtures of topics and probability distributions over words. It is worth pointing out that most of them have to deal with the "*bag-of-words*" assumption, and no one has paid attention to the subject structure of each individual document that is buried in the high levels of document structures. However, Embedding the document structures directly in topic models could yield a rich posterior topic structure for each document, which can further help in ad-hoc document analysis.

Chapter 5

Segmented Topic Model

The structure of documents into headings, sections, and thematically coherent parts, implies something about shared topics, and also plays an important role in document browsing and retrieval. In this chapter I take the simplest form of structure, a document consisting of multiple segments, as the basis for a new form of topic model, named Segmented Topic Model (STM), which leverages the structure of a document, instead of learning it. To make the model computationally feasible, and to allow the form of collapsed Gibbs sampling that has worked well to date with topic models, the marginalised PDP posterior (see Section 3.3) is used to handle the hierarchical modelling. I compare it with the standard topic models (*e.g.*, LDA reviewed in Chapter 4) and existing segmented models. The new model significantly outperforms standard topic models on either whole document or segment, and the existing segmented models, based on the held-out perplexity measure.

This chapter is organised as follows. In Section 5.1 I give an introduction to the motivation of STM. In section 5.2, I discuss related works in the literature of topic modelling. Then, I describe STM in detail and the posterior inference based on the PDP in Sections 5.3 and 5.4 respectively. In Section 5.5, I compare STM with LDA and the existing segmented models. The experimental results on several document collections are reported in Section 5.6.

5.1 Introduction

In recent years, documents continue to be digitised and stored in the form of web pages, blogs, twitters, books, scientific articles and so on. A majority of these documents come naturally with structure. They are structured into semantically coherent parts to ease understanding and readability of texts. A complete representation of the document structure ranges from the semantically high-level components (*e.g.*, chapters and sections) to the low-level components (*i.e.*, sentences and words). For instance, a book has chapters which itself contains sections, a section is further composed of paragraphs; a blog/twitter page contains a sequence of comments and links to related blogs/twitters; a scientific article contains appendices and references to related work.

In text analysis, some forms of structure are modelled with links in a document, and many different approaches follow from the key initial paper by Cohn and Hofmann [2001]. Some forms of structure are readily modelled simply by typing tokens, separating out the words, the links, maybe the names, into different multinomials in the topic model, easily done with existing theory [Buntine and Jakulin, 2006, Section 5.2]. Other forms of structure work with the topic space themselves [Blei et al., 2003; Mimno et al., 2007]. However, a different challenge in text analysis is the problem of understanding the document structure. Here I look at the original layout of each document as the guide to structure by following the ideas of Shafiei and Milios [2006], who developed a hierarchical model of the segments in a document.

Given a collection of documents, each of which consists of a set of segments (*e.g.*, sections, paragraphs, or sentences), each segment contains a group of words, it is interesting to explore the latent subject structure of each document by taking into account segments and their layout. I believe segments in a document not only have meaningful content but also provide preliminarily structural information, which can aid in the analysis of the original text. This idea actually originates from the way in which people normally compose documents (*e.g.*, essays, theses or books). When starting to write a document, people always bear in mind that they need first come up with some main ideas that they want to talk about; then decide a structure to organise these ideas logically and smoothly through, for example, chapters in a book, or sections in an article; and the ideas assigned to different segments could vary around the main ideas.

Can we statistically model documents in this manner? I adopt the probabilistic generative models called topic models to test this hypothesis. The basic idea is that each document is a random mixture over several latent topics, each of which is a distribution over words. Topic models specify a simple probabilistic process by which words can be generated, see Chapter 4. Here, we can consider LDA, as a way of modelling "ideas" with topics. However, LDA cannot simultaneously learn main ideas and sub-ideas under the same latent topic settings.

Extending LDA to involve segments of a document, Shafiei and Milios [2006] presented a Latent Dirichlet Co-Clustering (LDCC) model. It assumes there are two kinds of topics, *document-topics* (*i.e.*, distributions over segments) and *word-topics* (*i.e.*, distributions over words). In LDCC, documents are random mixtures of *document-topics*, and segments are random mixtures of *word-topics*. To generate a *word-topic* distribution for a segment, one needs to first draw for each segment a *document-topic* from the *document-topic* distribution. Clearly, LDCC does not share topics between documents and their segments. It also assumes that each segment is associated with only one *document-topic*. I will argue that these assumptions can be removed by using distributions over topics (*i.e.*, topic proportions), which require more powerful statistical tools.

In subsequent sections, I develop a simple structured topic model using the PDP in a finite discrete space, which is discussed in Section 2.3. This has the advantage of allowing a collapsed Gibbs sampler (see Section 3.3) to be developed for a hierarchical structure model. The proposed new topic model takes into account the beyond "*bag-of-words*" information, *i.e.*, a simple document structure, to enhance the understanding of the original text content.

5.2 Related work

Generative probabilistic topic models, see Chapter 4, are designed to identify topical representations of the textural data, which can reveal word usage patterns within or across documents. They have been widely applied to different kinds of documents, such as articles [Griffiths and Steyvers, 2004; Blei et al., 2003], emails [Mccallum et al., 2004], web blogs [Ramage et al., 2010], web spams [Bíró et al., 2008], customer profiles [Xing and Girolami, 2007], *etc*. They share a common assumption, "*bag-of-words*" that is the most widely used representation of text documents [Sebastiani, 2002].

Recently, some researchers have given attention to the study of how to explore the beyond "*bag-of-words*" information in topic modelling, such as the word order and topic structure. Griffiths et al. [2005] presented a composite model that makes use of the short-range syntactic dependencies among the words within the limit of a sentence. This model consists of two parts, a hidden Markov model (HMM) and a topic model. The former handles the syntactic word dependencies, the latter deals with the word semantics. Wallach [2006] gave another topic model that extends LDA by incorporating a notion of word orders via the combination of the n-gram statistics and latent topics.

The other models, which discover the structure of latent topics, include the correlated topic model (CTM) [Blei and Lafferty, 2006a], the Pachinko allocation model (PAM) [Mimno et al., 2007], the hierarchical Dirichlet process (HDP) [Teh et al., 2006], the hierarchical LDA (HLDA) [Blei et al., 2010], *etc.* Since the Dirichlet distribution is usually used in topic modelling as a prior to generate document topic proportions, a latent assumption is that topics are nearly independent. However, it is common to have correlations among topics in textual data. The CTM tries to capture the pairwise topic correlations by replacing the Dirichlet distribution with a logistic normal distribution. The PAM extends the concept of topics to include distributions not only over words but also over topics. The HDP (see Section 2.2.2) is built on top of pre-clustered data, *i.e.*, data groups, that have a pre-defined hierarchical structure. The HLDA organises topics into a tree with different levels of abstraction. The nested Chinese restaurant process defines a prior on the tree. For more discussion, see Section 4.4.

All these models attempt to capture the intra-topic correlation (*i.e.*, the hierarchical structure of topics themselves) that is quite different from the document structure this chapter deals with. The benefit of modelling document structure is

Table 5.1: List of notations for STM

Notation.	Description.
K	number of topics
I	number of documents
J_i	number of segments in document i
$L_{i,j}$	number of words in document i, segment j
W	number of words in dictionary
$\boldsymbol{\alpha}$	base distribution for document topic probabilities
$\boldsymbol{\mu}_i$	document topic probabilities for document i, base distribution for segment topic probabilities
$\boldsymbol{\nu}_{i,j}$	segment topic probabilities for document i and segment j
$\boldsymbol{\Phi}$	word probability vectors as a $K \times W$ matrix
$\boldsymbol{\phi}_k$	word probability vector for topic k, entries in Φ
$\boldsymbol{\gamma}$	W-dimensional vector for the Dirichlet prior for each $\boldsymbol{\phi}_k$
$w_{i,j,l}$	word in document i, segment j, at position l
$z_{i,j,l}$	topic for word in document i, segment j, at position l

that it can help understand the hierarchical subject structure of each individual document. Some previous research considers document internal structure with topic modelling. A considerable body of this line of research is in the field of topic segmentation, *i.e.*, division of a text into topically coherent segments. For example, the aspect HMM model [Blei and Moreno, 2001] assumes that each segment is generated from a unique topic assignment, and those latent topics have Markovian relations. Similar models include the hidden topic Markov model [Gruber et al., 2007], the structural topic model [Wang et al., 2011]. Instead of assuming each segment is assigned one topic, Purver et al. [2006] proposed a topic segmentation model in which each segment is associated with a topic distribution drawn from a Dirichlet distribution, like the multinomial mean shift model [Mochihashi and Matsumoto, 2006]. Indeed, STM shares a similar assumption as the model by Purver et al. [2006]. However, those models were designed to learn the topical structure of documents, while STM tries to leverage the structure in topic modelling.

5.3 STM Generative Process

The segmented topic model (STM) is a four-level probabilistic generative topic model with two levels of topics proportions, a level of topics and a level of words.

Before specifying STM, I list all notations and terminologies being used. Notation is depicted in Table 5.1. The following terms and dimensions are defined:

- A *word* is the basic unit of the text data, indexed by $\{1, \ldots, W\}$ in a vocabulary.
- A *segment* is a sequence of L words. It can be a section, paragraph, or even sentence. In this chapter, I assume segments are paragraphs or sentences.
- A *document* is an assemblage of J segments, as shown in the left of Figure 5.1, where d indicates a document, s_js are segments, and w_ls are words. Notice that J is known a priori.
- A *corpus* is a collection of I documents.

The basic idea of STM is to assume that each document i has a certain mixture of latent topics, denoted by probability vector $\boldsymbol{\mu}_i$, and is composed of meaningful segments; each of those segments also has a mixture over the same space of latent topics as those for the document, and these are denoted by probability vector $\boldsymbol{\nu}_{i,j}$

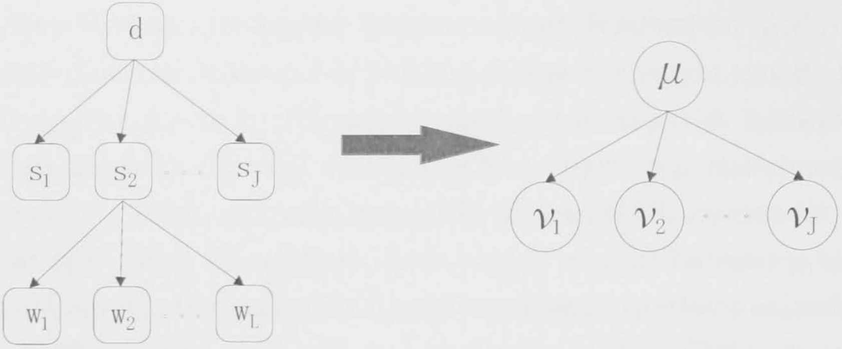


Figure 5.1: Graphical representation of mapping a document layout to a document subject structure in the STM. The left is the layout, and the right is the subject structure.

for segment j of document i. Both the main ideas of a document and sub-ideas of its segments are modelled here by the topic distributions. Sub-ideas are taken as variants of the main ideas, and thus sub-ideas can be linked to the main ideas, given correlations between a document and its segments, as shown in Figure 5.1.

How do the segment proportions $\boldsymbol{\nu}_{i,j}$ vary around the document proportions $\boldsymbol{\mu}_i$? The use of the PDP as $\boldsymbol{\nu}_{i,j} \sim \mathrm{PDP}(a, b, \boldsymbol{\mu}_i)$ distribution is a key innovation here. One would be happy to use, instead, a distribution such as $\boldsymbol{\nu}_{i,j} \sim \mathit{Dirichlet}(b\boldsymbol{\mu}_i)$ where b plays the role of "equivalent sample size". However, such a distribution makes the prior not conjugate to the likelihood so general MCMC sampling is required and parameter vectors such as $\boldsymbol{\mu}_i$ can no longer be integrated out to yield an efficient collapsed Gibbs sampler. I therefore employ the following lemma adapted from [Buntine and Hutter, 2010]:

Lemma 5.1. *The following approximations on distributions hold*

$$\begin{aligned} PDP(0, b, Discrete(\boldsymbol{\theta})) &= Dir(b\boldsymbol{\theta})\ , \\ PDP(a, 0, Discrete(\boldsymbol{\theta})) &\approx Dir(a\boldsymbol{\theta}) \qquad (as\ a \to 0), \end{aligned}$$

The first approximation is justified because the means and the first two central moments (orders 2 and 3) of the LHS and RHS distributions are equal. The second approximation is justified because the mean and first two central moments (orders 2 and 3) agree with error $O(a^2)$.

The PDP is a prior conjugate to the multinomial likelihoods, so allows collapsed Gibbs samplers of the kind used for LDA. Thus, conditioned on the model parameters $\boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\Phi}$ and the PDP parameters a, b, STM assumes the following generative process for each document i:

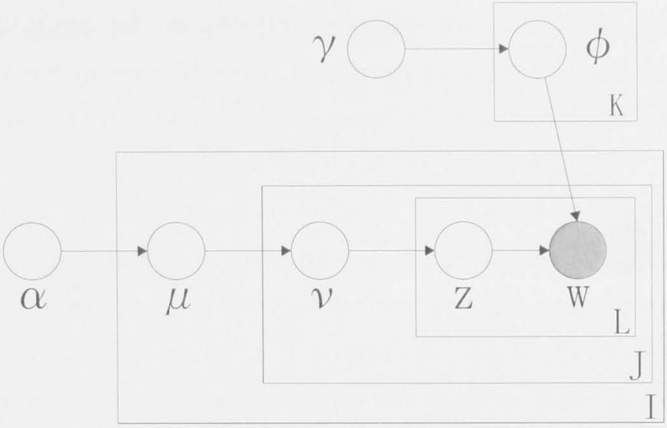


Figure 5.2: Segmented topic model. The inner rectangle indicates repeated sampling on words, the middle one indicates segments, the outer indicates documents.

1. Draw $\boldsymbol{\mu}_i \sim \mathrm{Dir}_K(\boldsymbol{\alpha})$
2. For each segments $j \in \{1, \ldots, J_i\}$
   (a) draw $\boldsymbol{\nu}_{i,j} \sim \mathrm{PDP}(a, b, \boldsymbol{\mu}_i)$
   (b) For each $w_{i,j,l}$, where $l \in \{1, \ldots, L_{i,j}\}$
      i. Select a topic $z_{i,j,l} \sim \mathrm{Discrete}_K(\boldsymbol{\nu}_{i,j})$
      ii. Generate a word $w_{i,j,l} \sim \mathrm{Discrete}_W(\boldsymbol{\phi}_{z_{i,j,l}})$.

I have assumed the number of topics (*i.e.*, the dimensionality of the Dirichlet distribution) is known and fixed, and the word probabilities are parameterised by a $K \times W$ matrix $\boldsymbol{\Phi}$. The graphical representation of STM is shown in Figure 5.2. The complete-data likelihood of each document i (*i.e.*, the joint distribution of all observed and latent variables) can be read directly from the graph using the distributions given in the above generative process.

5.4 Approximate Inference by CMGS

Having described the motivation behind STM, I now elaborate on the procedures for the posterior inference and parameters estimation. In order to use this model, the key inference problem that needs to be solved is to compute the posterior distribution of latent variables (*i.e.*, $\boldsymbol{\mu}$, $\boldsymbol{\nu}$ and $\boldsymbol{z}$) given the model parameters $\boldsymbol{\alpha}$, $\boldsymbol{\Phi}$, a, b and observations $\boldsymbol{w}$, *i.e.*,

$$p(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{z} \mid \boldsymbol{w}, \boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b) = \frac{p(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{z}, \boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b)}{\int_\mu \int_\nu \sum_z p(\boldsymbol{\mu}, \boldsymbol{\nu}, \boldsymbol{z}, \boldsymbol{w} \mid \boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b)} .$$

Unfortunately, this posterior distribution cannot be computed directly, due to the intractable computation of marginal probabilities in the denominator. We must appeal to an approximated inference, where some of the parameters (e.g. $\boldsymbol{\mu}$, $\boldsymbol{\nu}$ and $\boldsymbol{\Phi}$) can be integrated out rather than explicitly estimated. Two standard approximation methods have been applied to topic models: variational inference [Blei et al., 2003] and collapsed Gibbs sampling [Griffiths and Steyvers, 2004]. I use the latter in order to take the advantage of the collapsed Gibbs sampler for the PDP, *i.e.*, CMGS discussed in Section 3.3. Table 5.2 lists all statistics, which are needed for the development of the Gibbs algorithm. The table count $t^*_{i,j,k}$ (*i.e.*, the table *multiplicity*) and its derivatives are introduced in Section 3.3.

5.4.1 Model Likelihood

To build a collapsed Gibbs sampling algorithm, we need first to derive the joint distribution over observations $\boldsymbol{w}$, topic assignments $\boldsymbol{z}$ and the multiplicities $\boldsymbol{t}^*$, and then use this joint distribution to compute the full conditional distributions, *i.e.*,

- $p(z_{i,j,l} \,|\, \boldsymbol{z}^{-z_{i,j,l}}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^*_{1:I,1:J}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, a, b)$, and

Table 5.2: List of statistics for STM

Statistic.	Description.
$M_{i,k,w}$	topic by word total sum in document i, the number of words with dictionary index w and topic k, *i.e.*, $M_{i,k,w} = \sum_{j=1}^{J_i} \sum_{l=1}^{L_{i,j}} 1_{z_{i,j,l}=k} 1_{w_{i,j,l}=w}$.
$M_{k,w}$	$M_{i,k,w}$ totalled over documents i, *i.e.*, $\sum_i M_{i,k,w}$
$\boldsymbol{M}_k$	vector of W values $M_{k,w}$
$n^*_{i,j,k}$	topic total in document i and paragraph j for topic k. $n^*_{i,j,k} = \sum_{l=1}^{L_{i,j}} 1_{z_{i,j,l}=k}$
$N_{i,j}$	topic total sum in document i and segment j, *i.e.*, $\sum_{k=1}^{K} n^*_{i,j,k}$.
$\boldsymbol{n}^*_{i,j}$	topic total vector, *i.e.*, $(n^*_{i,j,1}, n^*_{i,j,2}, \ldots, n^*_{i,j,K})$.
$t^*_{i,j,k}$	table count in the CRP for document i and segment j, for topic k. This is the number of tables active for the k-th value.
$T_{i,j}$	total table count for document i and segment j, *i.e.*, $\sum_{k=1}^{K} t^*_{i,j,k}$.
$\boldsymbol{t}^*_{i,j}$	table count vector, *i.e.*, $(t^*_{i,j,1}, t^*_{i,j,2}, \ldots, t^*_{i,j,K})$.

- $p(t^*_{i,j,k} \mid \boldsymbol{z}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^{*-t^*_{i,j,k}}_{1:I,1:J}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, a, b)$.

The Dirichlet priors put on $\boldsymbol{\mu}_i$ and the PDP priors on $\boldsymbol{\nu}_{i,j}$ are conjugate to the multinomial distributions, and the PDP is also conjugate to the Dirichlet distribution. The conjugacy makes the marginalisation much easier. Thus, the joint conditional distribution of $\boldsymbol{z}_i$, $\boldsymbol{t}^*_{i,1:J_i}$, $\boldsymbol{w}_i$ can easily be computed by integrating out $\boldsymbol{\mu}_i$, $\boldsymbol{\nu}_{i,1:J_i}$ and $\boldsymbol{\Phi}$ respectively as follows.

First, integrating out the segment topic distribution $\boldsymbol{\nu}_{i,j}$ by using the joint posterior distribution of observations and multiplicities for the PDP, see Equation 3.3, we have

$$
\begin{aligned}
& p(\boldsymbol{\mu}_i, \boldsymbol{z}_{i,1:J_i}, \boldsymbol{w}_{i,1:J_i}, \boldsymbol{t}^*_{i,1:J_i} \mid \boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b) \\
&= P(\boldsymbol{\mu}_i \mid \boldsymbol{\alpha}) \int \prod_{j=1}^{J_i} \underbrace{p(\boldsymbol{\nu}_{i,j} \mid \boldsymbol{\mu}_i, a, b)}_{\boldsymbol{\nu}_{i,j} \sim \mathrm{PDP}(a,b,\boldsymbol{\mu}_i)} \prod_{l=1}^{L_{i,j}} p(z_{i,j,l} \mid \boldsymbol{\nu}_{i,j}) p(w_{i,j,l} \mid \boldsymbol{\phi}_{z_{i,j,l}}) d\boldsymbol{\nu}_{i,j} \\
&= \left(\frac{1}{\mathrm{Beta}_K(\boldsymbol{\alpha})} \prod_{k=1}^{K} \mu_{i,k}^{\alpha_k - 1} \right) \prod_{j=1}^{J_i} \underbrace{\left(\frac{(b|a)_{T_{i,j}}}{(b)_{N_{i,j}}} \prod_{k=1}^{K} S^{n^*_{i,j,k}}_{t^*_{i,j,k},a} \, \mu_{i,k}^{t^*_{i,j,k}} \right)}_{\text{see Equation 3.3}} \prod_{k=1}^{K} \prod_{w=1}^{W} \phi_{k,w}^{M_{i,k,w}} \\
&= \left(\frac{1}{\mathrm{Beta}_K(\boldsymbol{\alpha})} \prod_{k=1}^{K} \mu_{i,k}^{\alpha_k + \left(\sum_{j=1}^{J_i} t^*_{i,j,k}\right) - 1} \right) \prod_{j=1}^{J_i} \left(\frac{(b|a)_{T_{i,j}}}{(b)_{N_{i,j}}} \prod_{k=1}^{K} S^{n^*_{i,j,k}}_{t^*_{i,j,k},a} \right) \prod_{k=1}^{K} \prod_{w=1}^{W} \phi_{k,w}^{M_{i,k,w}} ,
\end{aligned}
$$

where $\mathrm{Beta}_K(\boldsymbol{\alpha})$ is K dimensional Beta function that normalises the Dirichlet (see Definition 2.1), and the last two products are derived by

$$
p(\boldsymbol{w}_{i,1:J_i} \mid \boldsymbol{z}_{i,1:J_i}, \boldsymbol{\Phi}) = \prod_{k=1}^{K} \prod_{w=1}^{W} \phi_{k,w}^{M_{i,k,w}} . \tag{5.1}
$$

Then, integrating out all the document topic distributions $\boldsymbol{\mu}_i$ and the topic-word matrix $\boldsymbol{\Phi}$ with Dirichlet integral, as is usually done for collapsed Gibbs sampling in topic models, gives

$$
\begin{aligned}
& p(\boldsymbol{z}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^*_{1:I,1:J} \mid \boldsymbol{\alpha}, \boldsymbol{\gamma}, a, b) \\
&= \int \left(\prod_{i=1}^{I} \int p(\boldsymbol{\mu}_i, \boldsymbol{z}_i, \boldsymbol{w}_i, \boldsymbol{t}^*_{i,1:J_i} \mid \boldsymbol{\alpha}, \boldsymbol{\Phi}, a, b) d\boldsymbol{\mu}_i \right) d\boldsymbol{\Phi} \\
&= \prod_{i=1}^{I} \left(\frac{\mathrm{Beta}_K\left(\boldsymbol{\alpha} + \sum_{j=1}^{J_i} \boldsymbol{t}^*_{i,j}\right)}{\mathrm{Beta}_K\left(\boldsymbol{\alpha}\right)} \prod_{j=1}^{J_i} \left(\frac{(b|a)_{T_{i,j}}}{(b)_{N_{i,j}}} \prod_{k=1}^{K} S^{n^*_{i,j,k}}_{t^*_{i,j,k},a} \right) \right) \\
& \prod_{k=1}^{K} \frac{\mathrm{Beta}_W\left(\boldsymbol{\gamma} + \boldsymbol{M}_k\right)}{\mathrm{Beta}_W\left(\boldsymbol{\gamma}\right)} .
\end{aligned} \tag{5.2}
$$

5.4.2 Collapsed Gibbs Sampling Algorithm

Collapsed Gibbs sampling is a special form of MCMC simulation, which should proceed until the Markov chain has "converged", although in practice it is run for a fixed number of cycles. While the proposed algorithm does not directly estimate $\boldsymbol{\mu}$, $\boldsymbol{\nu}$ and $\boldsymbol{\Phi}$, I will show how they can be approximated using the posterior sample statistics of $\boldsymbol{z}$ and $\boldsymbol{t}^*$. I adapt the CMGS algorithm proposed in Section 3.3 to divide the sampling procedure to two stages. First, given all the table counts $\boldsymbol{t}^*_{1:I,1:J_i}$, the latent topic assignment $z_{i,j,l}$ for each word is sampled. Second, given all the topic assignments of words $\boldsymbol{z}_{1:I,1:J}$, the table count $t^*_{i,j,k}$ is sampled for each topic under each segment.

Now, the full conditional distribution for $z_{i,j,l}$ can be obtained by focusing on a $z_{i,j,l}$, and looking at the proportionalities in Equation (5.2). For this, $t^*_{i,j,k}$ is mostly constant, as is $N_{i,j}$. Also, we have to take care of constraints on $t^*_{i,j,k}$, *i.e.*, $t^*_{i,j,k} \leq n^*_{i,j,k}$ (see constraints (3.5)). Note that $t^*_{i,j,k}$ can be forced to decrease when $n^*_{i,j,k}$ decreases by removing the current $z_{i,j,l}$. Therefore, to compute the final conditional distribution we have to distinguish among three cases:

1. Removing $z_{i,j,l} = k$ forces $n^*_{i,j,k} = t^*_{i,j,k} = 0$.
2. Before removing $z_{i,j,l} = k$, $n^*_{i,j,k} = t^*_{i,j,k} > 0$, so $t^*_{i,j,k}$ should decrease by one, *i.e.*, $t^*_{i,j,k} = t^*_{i,j,k} - 1$.
3. Adding $z_{i,j,l} = k$ forces both $n^*_{i,j,k}$ and $t^*_{i,j,k}$ to change from zero to one.

Taking into account all cases, we can obtain the final full conditional distribution

$$\begin{aligned} & p(z_{i,j,l} = k \mid \boldsymbol{z}^{-z_{i,j,l}}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^*_{1:I,1:J}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, a, b) \\ \propto \quad & \left(\frac{\alpha_k + \sum_{j=1}^{J_i} t^*_{i,j,k}}{\sum_{k=1}^{K} \left(\alpha_k + \sum_{j=1}^{J_i} t^*_{i,j,k} \right)} (b + aT_{i,j}) \right)^{1_{n^*_{i,j,k}=0}} \\ & \left(\frac{S^{n^*_{i,j,k}+1}_{t^*_{i,j,k},a}}{S^{n^*_{i,j,k}}_{t^*_{i,j,k},a}} \right)^{1_{n^*_{i,j,k}>0}} \quad \frac{\gamma_{w_{i,j,l}} + M_{k,w_{i,j,l}}}{\sum_{w=1}^{W} (\gamma_w + M_{k,w})} \end{aligned} \tag{5.3}$$

Given the current state of topic assignment of each word, the conditional distribution for table count $t^*_{i,j,k}$ can be obtained by cancelation of terms in Equation (5.2), yielding

$$\begin{aligned} & p(t^*_{i,j,k} \mid \boldsymbol{z}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^{*-t^*_{i,j,k}}_{1:I,1:J}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, a, b) \\ & \quad \propto \frac{\Gamma\left(\alpha_k + \sum_{j=1}^{J_i} t^*_{i,j,k}\right)}{\Gamma\left(\sum_{k=1}^{K}\left(\alpha_k + \sum_{j=1}^{J_i} t^*_{i,j,k}\right)\right)} (b|a)_{T_{i,j}} S^{n^*_{i,j,k}}_{t^*_{i,j,k},a} , \end{aligned} \tag{5.4}$$

which stochastically samples the multiplicity $t^*_{i,j,k}$. We should note that the value of $t^*_{i,j,k}$ should be in a specific interval to obey the constraints on $n^*_{i,j,k}$ and $t^*_{i,j,k}$. The interval is $[1, n^*_{i,j,k}]$, if $n^*_{i,j,k} \geq 2$. There is no sampling $t^*_{i,j,k}$ required if $n^*_{i,j,k} < 2$. Algorithm 4 gives the collapsed Gibbs sampler for STM that is derived from Algorithm 2.

From the statistics obtained after the *burn-in* of the Markov chain, we can easily estimate the document topic distribution $\boldsymbol{\mu}$, the segment topic distribution $\boldsymbol{\nu}$, and topic-word distributions $\boldsymbol{\Phi}$. They can be approximated from the following posterior expected values via sampling:

$$\widehat{\mu}_{i,k} = \mathbb{E}_{\boldsymbol{z}_{i,1:J_i}, \boldsymbol{t}^*_{i,1:J_i} \mid \boldsymbol{w}_{i,1:J_i}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, a, b} \left[\frac{\alpha_k + \sum_{j=1}^{J_i} t^*_{i,j,k}}{\sum_{k=1}^{K}\left(\alpha_k + \sum_{j=1}^{J_i} t^*_{i,j,k}\right)} \right] \tag{5.5}$$

$$\widehat{\nu}_{i,j,k} = \mathbb{E}_{\boldsymbol{z}_{i,1:J_i}, \boldsymbol{t}^*_{i,1:J_i} \mid \boldsymbol{w}_{i,1:J_i}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, a, b} \left[\frac{n^*_{i,j,k} - a \times t^*_{i,j,k}}{b + N_{i,j}} + \mu_{i,k} \frac{T_{i,j} \times a + b}{b + N_{i,j}} \right] \tag{5.6}$$

$$\widehat{\phi}_{k,w} = \mathbb{E}_{\boldsymbol{z}_{1:I,1:J}, \boldsymbol{t}^*_{1:I,1:J} \mid \boldsymbol{w}_{1:I,1:J}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, a, b} \left[\frac{\gamma_w + M_{k,w}}{\sum_{w'=1}^{W}(\gamma_{w'} + M_{k,w'})} \right] . \tag{5.7}$$

5.4.3 Sampling the Concentration Parameter

Initial experiments showed the concentration parameter b of the PDP can strongly affect perplexity results and seemed difficult to set by optimisation. I therefore developed a simple sampling method using auxiliary variables as follows. Each segment j of document i has an auxiliary probability $q_{i,j} \sim \text{Beta}(b, N_{i,j})$. From this, using an improper prior for b of the form $1/b$, the posterior for b is given by

$$\begin{aligned} & b \mid \boldsymbol{q}_{1:I,1:J}, \boldsymbol{z}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^*_{1:I,1:J}, \boldsymbol{\alpha}, \boldsymbol{\gamma}, a \\ & \quad \sim \text{Gamma}\left(\sum_{i=1}^{I} \sum_{j=1}^{J_i} T_{i,j}, \sum_{i=1}^{I} \sum_{j=1}^{J_i} \log 1/q_{i,j} \right) . \end{aligned} \tag{5.8}$$

Sampling using these auxiliary variables operates every major Gibbs cycle as follows:

Algorithm 4 Collapsed Gibbs sampling algorithm for STM

```
Require: $a$, $b$, $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$, $K$, $Corpus$, $MaxIteration$
Ensure: topic assignments for all words and all table counts
Topic assignment initialisation: randomly initialise topic assignments for all the words.
Table count initialisation: randomly initialise all $t^*_{i,j,k}$ $s.t.$ $0 \leq t^*_{i,j,k} \leq n^*_{i,j,k}$. If $n^*_{i,j,k} > 0$, $t^*_{i,j,k}$ must be greater than 0.
Compute all statistics listed in Table 5.2
for $iter \leftarrow 1$ to $MaxIteration$ do
    foreach document $i$ in corpus do
        foreach segment $j$ in $i$ do
            foreach word $w_{i,j,l}$ in $j$ do
                Exclude $w_{i,j,l}$, and update all the related statistics with current topic $z_{i,j,l} = k'$ removed. The constraints on $n^*_{i,j,k'}$ and $t^*_{i,j,k'}$ must be satisfied.
                Sample new topic $k$ for $w_{i,j,l}$ using Equation (5.3).
                Update all the statistics related to the new topic.
                Remove the value of the current table count $t^*_{i,j,k}$ from the statistics.
                Sample new table count $t^*_{i,j,k}$ for the new topic $k$ using Equation (5.4).
                Update the statistics with the new table count.
            end for
        end for
    end for
end for
```

1. Sample $q_{i,j} \sim \mathrm{Beta}(b, N_{i,j})$ for each document i and segment j and compute $\sum_{i=1}^{I} \sum_{j=1}^{J_i} \log 1/q_{i,j}$.

2. Sample b according to the condition distribution (5.8).

5.5 Comparison with other Topic Models

In this section I compare STM, in terms of text modelling, with two topic models[1], Latent Dirichlet Allocation (LDA) [Blei et al., 2003] and Latent Dirichlet Co-Clustering (LDCC) [Shafiei and Milios, 2006].

5.5.1 Latent Dirichlet Allocation

LDA is a three-level probabilistic generative model, the idea of which is that documents are random mixtures over latent topics, where each topic is a distribution over words, see Chapter 4 for detailed discussion. Compared with LDA, instead of sampling a topic $z_{i,j,l}$ directly from the document topic distribution $\boldsymbol{\mu}_i$, STM adds another layer between $z_{i,j,l}$ and $\boldsymbol{\mu}_i$, which is the segment topic distribution $\boldsymbol{\nu}_{i,j}$. Adding this distribution implies a higher fidelity of STM over LDA on modelling the correlation between the document topics and its segment topics (*i.e.*, the subject structure inside a document). LDA could also model the correlation by having two runs through documents and their segments separately. Nevertheless, the consistency of underlying topics between two separate runs cannot be guaranteed, since different runs will come up with different latent topics (due to unsupervised learning). Therefore, LDA cannot simultaneously model document topic distributions and segment topic distributions under the same latent topic space, as does STM.

It is interesting that STM can reduce to LDA, if the concentration parameter b of the PDP is set to an extremely large value, such as a value far larger than the number of observations. The proof is quite straight forward. In STM, $\boldsymbol{\nu}_{i,j}$ is drawn from a PDP with base measure $\boldsymbol{\mu}_i$, which itself is drawn from a Dirichlet distribution. Therefore, the base measure is discrete. See Property 2.5, the mean and variance of $\boldsymbol{\nu}_{i,j}$ are

$$\mathbb{E}[\boldsymbol{\nu}_{i,j}] = \boldsymbol{\mu}_i \ ; \qquad \mathbb{V}[\boldsymbol{\nu}_{i,j}] = \frac{1-a}{1+b}\left(\mathrm{diagonal}(\boldsymbol{\mu}_i) - \boldsymbol{\mu}_i\boldsymbol{\mu}_i^{\dagger}\right). \tag{5.9}$$

[1] I have changed some notations from the original papers to make them consistent with those used in STM.

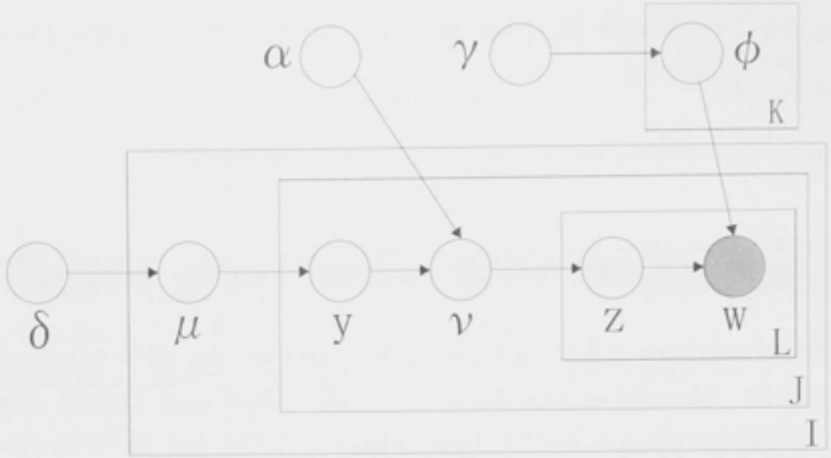


Figure 5.3: The latent Dirichlet co-clustering model

We can see that if $b \to \infty$, the variance approaches zero, so $\boldsymbol{\nu}_{i,j}$ is almost equal to $\boldsymbol{\mu}_i$. Now drawing topics from $\boldsymbol{\nu}_{i,j}$ can be equivalent to drawing topics directly from $\boldsymbol{\mu}_i$, which makes STM become LDA. It can also be proven by observing the conditional distribution given by Equation (2.10). If $b \to \infty$, the probability of customers choosing an occupied table approaches zero. These mean each customer will choose a new table to sit at and each table will just have one customer. Thus, there are $T_{i,j}$ approaching $N_{i,j}$ and $n^*_{i,j,k}$ approaching $t^*_{i,j,k}$ for all k's. Therefore, the ratios of Pochhammer symbols and the values of the Stirling numbers in Equation (5.2) become one. Taking out the two products over Pochhammer symbols and Stirling numbers from Equation (5.2), we can see that the marginal distribution of STM is the same as that of LDA (see Equation (4.2)).

5.5.2 Latent Dirichlet Co-clustering

LDCC is a four-level probabilistic model, as STM. It tries to extend LDA by assuming documents are random mixtures over *document-topics*, each of those topics is characterised by a distribution over segments; and segments are random mixtures over *word-topics*, each *word-topic* is a distribution over words. The two different kinds of topics are connected by hyper-parameters $\boldsymbol{\alpha}$, under the assumption that each *document-topic* is a mixture of *word-topics*. It is a kind of nested LDA, as shown in Figure 5.3. LDCC also assumes that each segment is associated with only one *document-topic* (y in Figure 5.3), which is quite a strong assumption in my view.

In contrast, STM allows documents and segments to share same latent topics, rather than assuming two different kinds, as I believe a document and its segments should be generated from the same topic space. Moreover, STM relaxes the assumption on segments by assuming each segment still has a topic distribution drawn from its document topic distribution. Thus, each segment can also exhibit multiple topics, which includes the case that it has only one topic, if the distribution highly concentrates on one topic. In this sense, STM does not make the strong assumptions, as LDCC does.

5.6 Experimental Results

I implemented the three models in C, and ran them on a desktop with Intel(R) Core(TM) Quad CPU (2.4GHz), although the codes are not multi-threaded. The training time, for instance, on the NIPS dataset with 100 topics and 1000 Gibbs iterations is approximately 5 hours for LDA, 33 hours for LDCC and 20 hours for STM. In subsequent sections, I report the following sets of experimental results:

The in depth study of characteristics of STM I first discuss the experimental results on two patent datasets (G06-1000 and G06-990) to analyse topic variability among segments. The goal of this set of experiments is to study how the concentration parameter b and the discount parameter a can influence topic proportions.

Perplexity comparisons I then compare STM with LDA and LDCC in terms of per-word predictive accuracy on unseen documents. Besides the aforementioned two patent datasets, the three models are further applied to another two patent datasets (A-1000 and F-1000), the NIPS datasets[2], and an extract from the Reuters RCV1 corpus [Lewis et al., 2004]. The perplexity comparisons on held-out testing documents evidently demonstrate the advantage of STM over the other two models.

5.6.1 Data Sets and Evaluation Criteria

The two patent datasets, G06-1000 and G06-990, are randomly selected from 5000 U.S. patents[3] granted between Jan. and Mar. 2009 under the class "*computing; calculating; counting*" with international patent classification (IPC) code G06. Patents in G06-1000 are split into paragraphs according to the original

[2] It is available at `http://nips.djvuzone.org/txt.html`

[3] All patents are from Cambia, `http://www.cambia.org/daisy/cambia/home.html`

structure. Patents in G06-990[4] are split into sentences with a Perl package (Lingua::En:Sentence). All stop-words, extremely common words (*e.g.*, top 40 for G06-1000), and less common words (*i.e.*, words appear in less than 5 documents) have been removed. This leads to a vocabulary size of 10385 unique words in G06-1000 and 11518 in G06-990. The G06-1000 dataset contains 1,000 patents, 60,564 paragraphs, and 2,513,087 words. The G06-990 dataset contains 990 patents, 249,102 sentences, and 2,832,364 words. Paragraphs or sentences are treated as segments, and 80% of each dataset are hold out for training and 20% for testing.

In order to evaluate the generalisation capability of these models to unseen data, perplexity is computed, which is a standard measure for estimating the performance of probabilistic language models. The perplexity of a collection $\mathcal{D}_{test}$ of I test documents that is defined as:

$$perplexity(\mathcal{D}_{test}) = \exp\left\{ - \frac{\sum_{i=1}^{I} \ln p(\boldsymbol{w}_i)}{\sum_{i=1}^{I} N_i} \right\} \tag{5.10}$$

where $\boldsymbol{w}_i$ indicates all words in document i, and N_i indicates the total number of words in i. A lower perplexity over unseen documents means better generalisation capability. In the following experiments, it is computed based on the held-out method introduced by Rosen-Zvi et al. [2004]. In order to calculate the likelihood of each unseen word in STM, we need to compute the document topic probability vector $\boldsymbol{\mu}$, the segment topic probability vector $\boldsymbol{\nu}$, and word probability matrix $\boldsymbol{\Phi}$. Here, I estimate them using a Gibbs sampler and Equations (5.5), (5.6) and (5.7) for each sample of assignments $\boldsymbol{z}, \boldsymbol{t}$.

5.6.2 Topic Variability Analysis among Segments

I first investigate the variability between topic proportions (*i.e.*, distributions) of documents and those of their segments. As I discussed in Section 5.3, it is modelled by the PDP with two parameters, a and b. Here I present studies on how a and b act on the diversity among document topic proportions (*i.e.*, $\boldsymbol{\mu}_i$) and their segment topic proportions (*i.e.*, $\boldsymbol{\nu}_{i,j}$).

The standard deviation is used to measure the variation of $\boldsymbol{\nu}_{i,j}$, and entropy to show the expected number of topics in either documents or segments. The prior mean and variance of $\boldsymbol{\nu}_{i,j}$ have been given in Equations (5.9). For all figures in this section, STM_P and STM_S indicate STM running on paragraphs (G06-1000) and sentences (G06-990) respectively; STM_P_mu and STM_S_mu indicate

[4] I randomly selected 1000 patents, but 10 were deleted after pre-processing, because they were too small.

entropies computed based on $\boldsymbol{\mu}$, and STM_P_nu and STM_S_nu denote those computed based on $\boldsymbol{\nu}$.

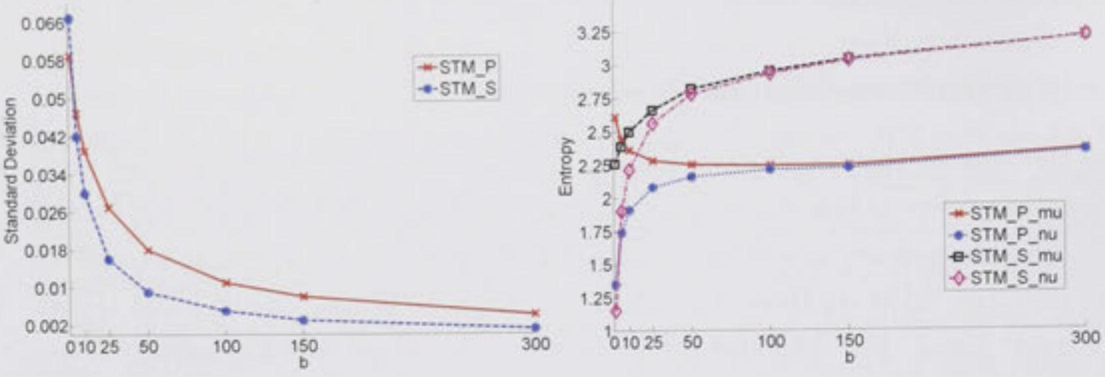
Study of the Concentration Parameter b

The purpose of this set of experiments is to investigate how b influences topic proportions after isolating the effect of a. In the experiments, I fix $a = 0.2$ for the G06-1000 dataset and $a = 0$ for the G06-990 dataset, change b from 0.1 to 300.0, and then run STM on those two datasets with $k = 50$ and $\alpha = 0.5$. As shown in Figure 5.4(a), the standard deviation decreases while b is increasing. When b is small, the variance of topic proportions in segments is large. Hereby, the topic proportion $\boldsymbol{\nu}_{i,j}$ of a segment could be quite different from the topic proportion $\boldsymbol{\mu}_i$ of the corresponding document, as indicated in Figure 5.4(b) by the different expected number of topics. In contrast, when b gets quite large, the variance of segment topic proportions becomes small. Figure 5.4(b) shows the expected number of topics in each segment gets close to the number of topics in the corresponding document. In this case, there could be no difference between a document topic proportion and its segment topic proportions, and segments loose their specificity on topics. We can observe that the perplexity turns out to be larger when b is quite small or quite large in Figure 5.6(a). Consequently, we can conclude that the topic deviation between a document and its segments should be neither too small nor too big, which somehow complies with the way in which people structure ideas in writing.

Study of the Discount Parameter a

To study how a influences topic proportions, I ran another set of experiments on the two patent datasets by fixing b to 10 and changing a from 0.0 to 0.9. According to Equations (5.9), the variance of segment topic distribution gets small while a is getting large, given b fixed.

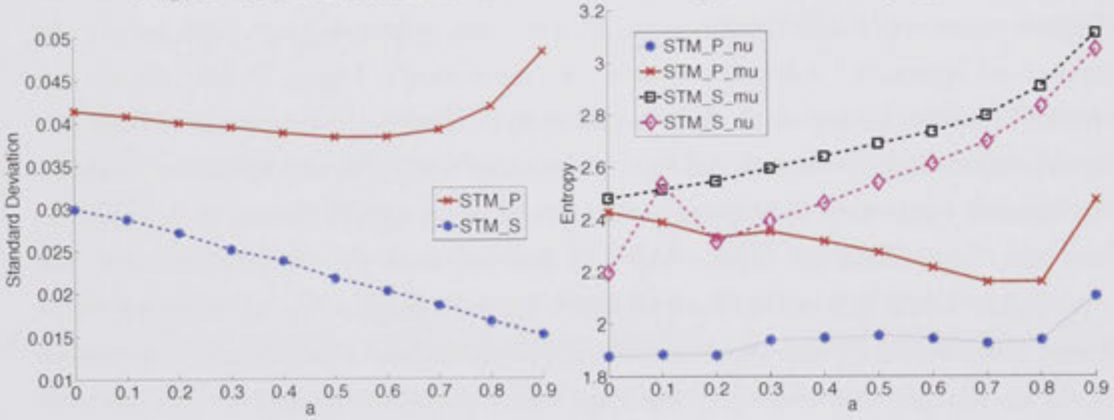
I plotted the standard deviation in Figure 5.5(a), the entropy in Figure 5.5(b), and perplexity in Figure 5.6(b). For the G06-990 dataset, while a is increasing, the standard deviation decreases, and the expected number of topics in each segment gets close to the expected number of topics in the document. However, the perplexity increases significantly when a changes from 0.6 to 0.9, which is also observed in the G06-1000 dataset. It is interesting that both the standard deviation and the entropy drop first and then increase for the G06-1000 dataset. Figure 5.6(b) shows there is no big difference while a is between 0 and 0.5. We may



(a) Standard deviation *w.r.t.* b

(b) Entropy *w.r.t.* b

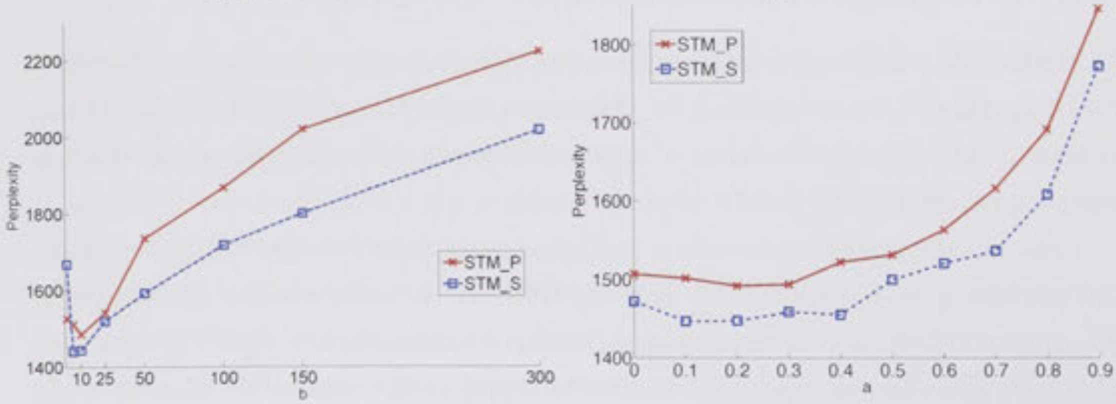
Figure 5.4: Standard deviation and entropy with changing a fixed



(a) Standard deviation *w.r.t.* a

(b) Entropy *w.r.t.* a

Figure 5.5: Standard deviation and entropy with b fixed



(a) Perplexity w.r.t. b

(b) Perplexity w.r.t. a

Figure 5.6: Perplexity with either a or b fixed

conclude that the influence of a, especially $a < 0.6$, on topic proportions is not significant when b is set to 10 on the two patent datasets.

Topic Proportion Examples

To further show topic variability among document topic proportion and its segment topic proportions, I plot as an example those topic proportions of a patent from G06-1000 in Figure 5.7. They are extracted from an experiment with following settings: $K = 50$, $a = 0.2$, $b = 10$ and $\alpha = 0.5$. This patent has 10 paragraphs, and talks about web authentication security systems for finance, as indicated by four topics with the highest ratios in document topic distribution mu in Figure 5.7. They are T-12, T-16, T-31 and T-44 in Table 5.3 (Note topic numbers following "T-" correspond to topic indices in Figure 5.7.). As indicated by the blue bars, segment topic proportions are variants of the document topic proportion with different ratios for the four main topics. For example, the first paragraph (see nu_1) covers all the four topics and topic T-15, it is indeed an

Table 5.3: 11 topic examples learnt by STM from the G06-1000 dataset

T-12	T-16	T-20	T-21	T-31	T-32
web	systems	path	component	key	files
page	performance	tree	management	security	volume
browser	large	nodes	engine	authentication	copy
site	required	price	electronic	hash	site
internet	multiple	paths	applications	keys	update
pages	problem	decision	modules	encryption	backup
content	high	failure	external	chip	directory
report	single	period	desktop	encrypted	local
users	cost	graph	install	protected	delta
website	typically	model	installation	secure	updates
T-37	**T-42**	**T-44**	**T-46**	**T-47**	
value	window	card	skilled	state	
threshold	selected	transaction	understood	event	
segment	displayed	account	patent	error	
maximum	screen	customer	specific	status	
size	view	payment	intended	current	
amount	button	terminal	limited	action	
rang	selection	cards	modifications	recovery	
determined	box	ic	incorporated	events	
index	select	identification	disclosed	determines	
equal	text	merchant	detail	routine	

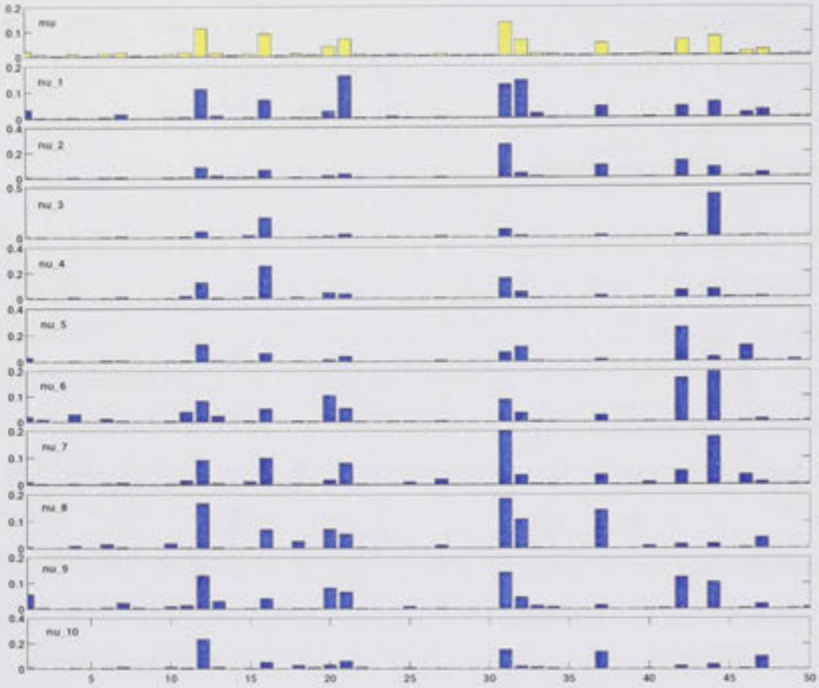


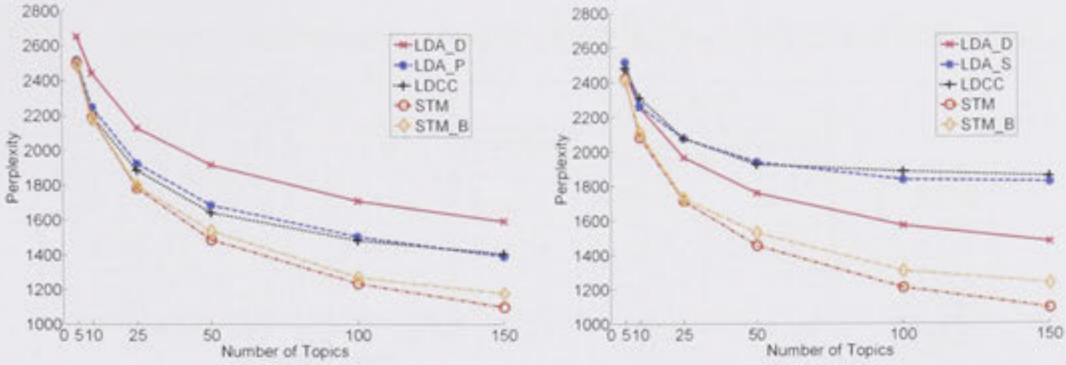
Figure 5.7: Plots of topic distributions for a patent from G06-1000. Document topic distribution mu is on the top, and the others are 10 segment topic distributions, labeled with nu_j, $j \in \{1, 2, \ldots, 10\}$. The label of X-axis is topic which is indexed from 1 to 50, the Y-axis is topic proportion.

introduction paragraph; the fifth paragraph (see nu_5) focuses on the interface design, see topic T-42; and the seventh and the eighth paragraphs (see nu_7 and nu_8) discuss more about technical issues of an authentication system. It can be seen that STM can capture the variability among topic proportions.

5.6.3 Perplexity Comparison

I follow the standard way in topic modelling to evaluate the per-word predicative perplexity of STM, LDA and LDCC. In the training procedure, each Gibbs sampler is initialised randomly and runs for 500 burn-in iterations. Then a total number of 5 samples are drawn at a lag of 100 iterations. These samples are averaged to obtain the final trained model, as in [Li et al., 2007].

I set hyper-parameters fairly in order to make a scientific comparison, as they are important to these models. Symmetric Dirichlet priors (*i.e.*, $\boldsymbol{\alpha}$ for LDA and STM, $\boldsymbol{\delta}$ for LDCC) were simply used in the following experiments, although we can estimate them from data using, for instance, the Moment-Matching algorithm



(a) Perplexity comparison on the G06-1000 (b) Perplexity comparison on the G06-990

Figure 5.8: Perplexity comparisons on the G06-1000 and G06-990 datasets

proposed by Minka [2000]. With γ fixed to $200/W$, I ran different settings of $\boldsymbol{\alpha}$ and $\boldsymbol{\delta}$ (from 0.01 to 0.9) for different number of topics (*i.e.*, 5, 10, 25, 50, 100, and 150), and empirically chose the optimal parameters for LDA and LDCC. It has been observed, for example, LDA trained on $\alpha = 0.1$ was always better on both G06-1000 and G06-990 datasets than on other settings, but LDCC varied quite a bit (*e.g.*, $\delta = 0.9$ for 25 *word-topics*, $\delta = 0.01$ for 100 *word-topics*). The number of *document-topics* in LDCC was fixed to 20 for all experiments and $\boldsymbol{\alpha}$ was estimated using the moment-match algorithm, as in [Shafiei and Milios, 2006]. I used $\alpha = 0.5$ in STM for all the numbers of topics without tuning, and set $a = 0.2$ and $b = 10$ for both the G06-1000 dataset and the G06-990 dataset. When optimising b, I set $a = 0$. Note that optimising the parameter settings for the two competitors (LDA and LDCC) enables us to draw sound conclusions on the performance of STM.

Figure 5.8(a) presents experimental results for these models on the G06-1000 dataset. LDA has been run on document level (LDA_D) and paragraph level (LDA_P) separately. It is interesting to see that LDA_P is better than LDA_D. LDCC exhibits better performance than LDA_D, but it is only comparable with LDA_P. The paired t-test, shown in Table 5.4, gives p-value= 0.05 to the slight improvement. In contrast, STM (with or without sampling b using the scheme presented in Section 5.4.3, indicated by STM and STM_B respectively) consistently performs better than all the other models. The advantage is especially obvious for large numbers of topics. Table 5.5 shows the optimised b values. The superiority of STM over LDA and LDCC is statistically significant according to the paired t-test with p-values shown in the third and fourth columns of Table 5.4.

Table 5.4: P-values for paired t-test on two patent datasets

	G06-1000			G06-990		
	LDCC	STM	STM_B	LDCC	STM	STM_B
LDA_D	7.0e-5	1.3e-3	5.4e-4	2.9e-2	4.8e-3	2.2e-3
LDA_P/S	5.0e-2	1.5e-2	8.0e-3	3.9e-1	9.1e-3	6.3e-3
LDCC		3.9e-2	2.8e-2		1.1e-2	7.7e-3

Table 5.5: Optimised b values, when $a = 0$

	K=5	K=10	K=25	K=50	K=100	K=150
G06-1000	1.53	1.89	2.46	2.92	3.42	3.54
G06-990	1.21	1.36	1.90	2.15	2.36	2.44

Similar comparison on the G06-990 dataset is shown in Figure 5.8(b). I ran LDA (indicated by LDA_S), LDCC and STM on the sentence level. The perplexity of LDCC becomes slightly larger than LDA_S when the number of topics is greater than 50. It is comparable to LDA_S, as LDCC *v.s.* LDA_P in Figure 5.8(a). Interestingly, the performance of either LDA or LDCC on the sentence level turns out to be much worse than LDA on the document level. However, the paired t-test results in the last two columns of Table 5.4 show that STM is still significantly better than both LDA and LDCC. STM could certainly retain its good generalisation capability even on sparse text on the segment level.

Evidently, the results illustrated in both Figure 5.8(a) and Figure 5.8(b) demonstrate that STM can work remarkably well on both the paragraph level and the sentence level.

5.6.4 Further Experiments

In order to further exhibit the advantage of STM, I also ran it on another two patent datasets (A-1000 and F-1000), the NIPS dataset and an extract of the Reuters dataset using $a = 0$ and sampling the concentration parameter b according to the scheme in Section 5.4.3. Table 5.6 shows the optimised b values. The Dirichlet prior $\boldsymbol{\alpha}$ for LDA is optimised by using the method[5] proposed by [Minka, 2000].

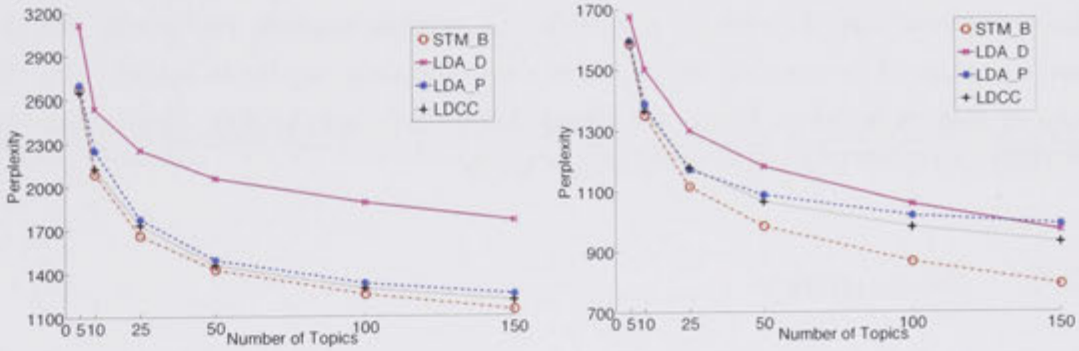
[5] The code is modified from the Minka's Matlab code that is downloaded from `http://research.microsoft.com/en-us/um/people/minka/software/fastfit/`

Table 5.6: Optimised b values on the A-1000, the F-1000, the NIPS and the Reuters datasets with $a = 0$.

	K=5	K=10	K=25	K=50	K=100	K=150
A-1000	0.94	1.08	1.36	1.56	1.71	1.77
F-1000	1.64	2.10	2.75	3.31	3.99	4.49
NIPS	1.46	1.97	2.7	3.4	4.04	4.33
Reuters	2.98	3.54	3.17	2.26	1.50	1.20

Table 5.7: Dataset statistics

	A-1000	F-1000	NIPS	Reuters
Number of documents	1,000	1,000	1,629	2,640
Number of segments	78,653	55,149	174,747	38,182
Number of words	3,108,479	2,127,878	1,773,365	405,531
Vocabulary size	18,988	9,760	13,327	13,884

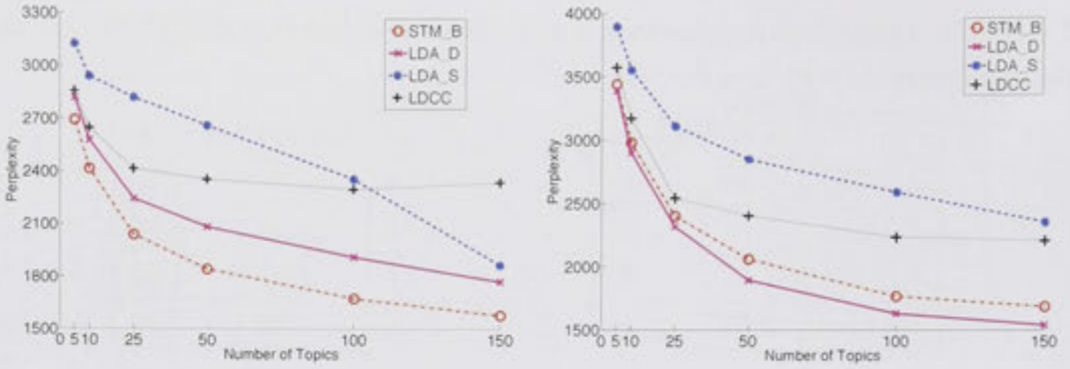


(a) Perplexity comparison on the A-1000 dataset

(b) Perplexity comparison on the F-1000 dataset

Figure 5.9: Perplexity comparisons on the A-1000 and the F-1000 patent datasets

The two patent datasets, A-1000 and F-1000, are randomly selected from the U.S. patents granted in 2010 with IPC code A ("*human necessities*") and F ("*mechanical engineering; lighting; heating; weapons; blasting*") respectively. All the patents in the two datasets are split into paragraphs, as done for G06-1000. The NIPS dataset is processed to remove bibliography material (everything after "References") and header material (everything before "Abstract"); the Reuters articles are extracted from 20-25/8/1996, and the articles in categories CCAT, ECAT and MCAT are dropped. All the documents in the NIPS dataset and the



(a) Perplexity comparison on the NIPS dataset (b) Perplexity comparison on the Reuters dataset

Figure 5.10: Perplexity comparisons on the NIPS and the Reuters datasets

Reuters dataset are split into sentences. Table 5.7 shows the statistics of the four datasets. Again 80% were used for training and 20% for testing. Perplexity results appear in Figures 5.9 and 5.10. It is interesting that the performance of LDA running on the document level is slightly better than STM on Reuters articles. I have observed that the average size of Reuter articles is about 150 words, but the average sizes of documents in the other three datasets are much larger than the size of Reuter articles. There are about 3100 words for A-1000, 2100 words for F-1000 and 1100 words for NIPS, respectively.

5.7 Summary

In this chapter, I have presented a segmented topic model (STM) that directly models the document structure with a four-level hierarchy. An effective collapsed Gibbs sampling algorithm based on the CMGS has been developed. The ability of STM to explore correlated segment topics (*i.e.*, the latent subject structure of a document buried in the document layout) has been demonstrated in the experiments by the significant improvement in terms of per-word predictive perplexity compared with the standard topic model (LDA) and previous segmented model (LDCC). I also found that STM is approximately equal to LDA on quite short documents.

The primary benefit of STM is that it allows us to simultaneously model document topic distributions and segment topic distributions in the same latent topic space, without separate runs as LDA or introducing different kinds of topics as LDCC. Although the experiments I have done were just on either the paragraph level or sentence level, STM readily models other segments, like sections and chapters. Moreover, the success of STM has indicated that it is beneficial to consider the document structure directly in topic modelling. Although I think the inference algorithm I proposed is good enough to test STM, it is still worth exploring other inference algorithms, such as variational inference for Dirichlet process mixture models [Blei and Jordan, 2005; Teh et al., 2008].

Chapter 6

Sequential LDA Model

Understanding how topics within a document evolve over the structure of the document is an interesting and potentially important problem in exploratory and predictive text analytics. In this chapter, I address the problem of topic evolution by presenting a novel variant of LDA: Sequential LDA (SeqLDA). This variant directly considers the underlying sequential structure, *i.e.*, a document consists of multiple segments (*e.g.*, chapters or paragraphs), each of which is correlated to its antecedent and subsequent segments. Such progressive sequential dependency is captured by using the HPDP (see Section 2.3.4). I also develop an effective collapsed Gibbs sampling algorithm based on CMGS (see Section 3.3). SeqLDA outperforms the standard LDA in terms of perplexity and yields a nicer sequential topic structure than LDA in topic evolution analysis on several books such as Melville's 'Moby Dick'.

This chapter is organised as follows. I briefly discuss the related work in Section 6.2 after the introduction in Section 6.1. I then elaborate the derivation of SeqLDA, and compare it with some related models in Section 6.3. Section 6.4 discusses the collapsed Gibbs sampling algorithm that samples from the posterior of SeqLDA. In Section 6.6, I present experimental results on patents and several books. Section 6.7 gives a brief discussion and concluding comments.

6.1 Introduction

As I discussed in the previous chapter, many documents in corpora come naturally with structure. They consist of meaningful segments (*e.g.*, chapters, sections, or paragraphs), each of which contains a group of words, *i.e.*, a document-segment-word structure. STM proposed in Chapter 5 focuses on mapping a simple docu-

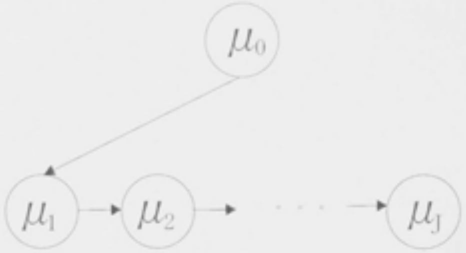


Figure 6.1: A subject structure modelled by SeqLDA. The node $\boldsymbol{\mu}_0$ is the document topic distribution, and the other nodes are the segment topic distributions. The subscripts of $\boldsymbol{\mu}$ follow the order of segments in a document. The arrows indicate dependencies.

ment structure to a hierarchical topic structure, as shown in Figure 5.1. It tries to capture the hierarchical relationship between a document subject and the corresponding segment subtopics. The benefit of incorporating the document structure into topic modelling has been shown by the significantly better performance of STM over LDA and LDCC. However, the underlying assumption of STM is that segments in a document are exchangeable (*i.e.*, given the document topic distribution, the segment topic distributions are conditionally independent.). For the problem of analysing how topics evolve over the document, the exchangeability assumption is not suitable and needs to be removed to further incorporate topic dependencies existing in the sequential document structure, *i.e.*, the segment sequence according to the document layout.

Here I take an essay as an example. It is composed of multiple paragraphs, each of which is associated with a subtopic, as shown in Figure 1.1. All the subtopics are combined in a way together to form the subject of the essay. This document structure conveys two kinds of topic structures that are necessary for writing a cohesive and easily accessible essay. The first kind of topic structure is that subtopics are linked to the essay subject, which gives a topic hierarchy, as shown in the right of Figure 5.1. Thus, paragraphs are organised according to the topic hierarchy, and implicitly assumed to be exchangeable. In this chapter, I am interested in the second kind of topic structure that subtopics are linked sequentially according to the paragraph sequence (*i.e.* the original layout of paragraphs). These linkages are indicated by arcs labeled with "link" in Figure 1.1. Now paragraphs are no longer exchangeable, and I believe the paragraph sequence can provide some useful contextual information that can help to understand the original text content. We can further use the contextual information to analyse how topics change within a document. Figure 6.1 shows a graphical representation of the sequential topic structure according to the segment sequence in a document.

As with STM, I adapt topic models for explicitly modelling the sequential topic structure. In the context of topic modelling, both the subject of a document and the subtopics of its segments can be modelled by distributions over the same set of latent topics, each of which is a distribution over words. The sequential topic structure is modelled through the probabilistic dependencies among the topic distributions, as indicated by arrows in Figure 6.1. However, most of the existing topic models are not aware of the underlying document subject structure. They only consider one level, *i.e.*, document-words, and usually neglect the contextual information buried in the higher levels of document structure.

In SeqLDA, the progressive topic dependency is captured using a multi-level extension of the HPDP, see Section 2.3.4. Thus, a segment topic distribution can be recursively drawn from a PDP with a base distribution that is the topic distribution of its preceding segment. Using the PDP chain of topic distributions allows us to explore how topics are evolving among, for example, paragraphs in an essay, or chapters in a novel; and to detect the rising and falling of a topic in prominence. The topic evolution can be estimated by exploring how topic proportions change in segments. Tackling topic modelling together with the subject structure of a document provides a solution for going beyond the "*bag-of-words*" assumption that is widely used in text analytics (*e.g.*, natural language processing and information retrieval).

6.2 Related Work

To capture topic evolution in temporal data, integrating time stamps into topic models has been around for a while. Existing work focuses mainly on learning topic evolution patterns from a time-varying corpus, instead of exploring how topics progress within each individual document by following the latent topic structure. These works explore how topics change, rise and fall, by considering time stamps associated with document collections. In general, they can be put into two categories, Markov chain based models and non-Markov chain based models.

In the Markov chain based models, the dynamic behaviours (*i.e.*, topic evolution in my perspective) are captured by state transitions. The state at time $t+\Delta_t$

is dependent on the state of t. For instance, the dynamic topic model (DTM) [Blei and Lafferty, 2006b], the dynamic mixture model (DMM) [Wei et al., 2007], and the dynamic extensions of HDP [Ren et al., 2008; Ahmed and Xing, 2010].

The DTM captures the topic evolution in document collections that are organised sequentially into several discrete time periods, and then within each period an LDA model is trained on the documents. The Gaussian distributions are used to tie a collection of LDAs by chaining the Dirichlet prior and the model parameters of each topic. Indeed, the parameter at time $t-1$ is the expectation for the distribution of parameter at time t, the idea of which is similar to that used in our SeqLDA. Unfortunately, Gaussian distributions are not conjugate to multinomial distributions, which results in complex approximation in inference.

The DMM assumes that the mixture of latent variables (*i.e.*, topic distribution) for all data streams is dependent on the mixture of the previous time stamp, *i.e.*, the expectation of topic distribution at time t is the topic distribution at time $t-1$, as used in the DTM. Although the structure of the DMM is similar to SeqLDA (*i.e.*, both put first-order Markov assumption on topic distributions), SeqLDA capitalises on the *self-conjugacy*[1] of the PDP to chain a series of LDAs, instead of using Dirichlet distributions. The problem with the Dirichlet distribution is that it is not self-conjugate, which could not facilitate an effective inference algorithm.

Recently, the HDP has been extended to incorporate time dependence to model the time-evolving properties of temporal data, such as the dynamic HDP (DHDP) [Ren et al., 2008]. As shown in Figure 2.2, the DHDP captures the time dependence via a weighted mixture of two distributions drawn from the same HDP, *i.e.*, the distribution G_t at time t is equal to $(1 - w_{t-1})G_{t-1} + w_{t-1}H_{t-1}$, where G_{t-1} is the distribution at time $t-1$, and H_{t-1} is the innovation distribution. It is easy to see that G_t is modified from G_{t-1} by the weighted mixture. See Section 2.2.2 for detailed discussion. Compared to the DHDP, our SeqLDA takes G_{t-1} as the expectation of G_t, which is done by drawing G_t from a PDP with base distribution G_{t-1}. The correlation of samples at adjacent times can be controlled by adjusting the variance of the two distributions. Therefore, the difference between the DHDP and SeqLDA resides in the way of handling the dynamic relationship from G_{t-1} to G_t.

Instead of assuming the Markovian dependence over time, the second class

[1] PDP is conjugate to itself when applied to the discrete data. Equation 3.6 shows that we can recursively integrate out the real valued probability vectors (*i.e.*, G) in the HPDP with an auxiliary variable, *i.e.*, table *multiplicity*.

of models treats time as an observed variable that can be jointly generated with words by latent topics, for example, the topics over time (ToT) model [Wang and McCallum, 2006]. In the ToT, the topic over time is captured by a Beta distribution. Drawing all time stamps from the same Beta distribution might not be appropriate for, such as, stream data [Wei et al., 2007]. Some other approaches are, for instance, He et al. [2009] developed inheritance topic model to understand topic evolution by leveraging the citation information; Kandylas et al. [2008] analysed the evolution of knowledge communities based on the clustering over time method, called Streemer.

Significantly, the difference between these models and SeqLDA is that, instead of modelling topic trends in document collections based on documents' time stamps, SeqLDA models topic progress within each individual document by using the correlations among segments, *i.e.*, the underlying sequential topic structure, according to the original document layout. The Markovian dependencies are put on the topic distributions. In this way, we can directly model the topical dependency between a segment and its successor.

Although one may argue that the models just discussed can also be adapted to the individual document by treating the sequence of segments as time stamps, the computation complexity and space complexity of those models could be significantly increased with the growth of the latent variables and hyper-parameters. In contrast, I use a single integrated model based on the HPDP, in which the real valued parameters can be integrated out because PDP's are *self-conjugate* .

6.3 SeqLDA Generative Process

Now I present the Sequential Latent Dirichlet Allocation model (SeqLDA) which models how topics evolve over segments in individual documents. I assume that there could be some latent sequential topic structure within each individual document, *i.e.*, topics within a document evolve smoothly from one segment to another, especially in various books (*e.g.*, novels). This assumption intuitively originates from the way in which people normally organise ideas in their writing. Before specifying SeqLDA, I list notation and terminology used in this chapter. Notation is given in Table 6.1. The terms and dimensions used in the SeqLDA model are the same as those in STM, see Section 5.3. In this chapter I assume segments are either paragraphs or chapters.

The basic idea of SeqLDA is to assume that each document i is a certain mixture of latent topics, denoted by a topic distribution $\boldsymbol{\mu}_{i,0}$, and is composed of a sequence of meaningful segments; each of these segments also has a mixture over the same set of latent topics as those for the document, and these are indicated by a topic distribution $\boldsymbol{\mu}_{i,j}$ for segment j. Obviously, both the document and its segments share the same topic space. Notice that the index of a segment should comply with its position in the original document layout, which means the first segment is indexed by $j = 1$, the second segment is indexed by $j = 2$, and so on. Both the subject of a document and subtopics of its segments are modelled here by these distributions over topics. Take the book, called "The Prince", as an example. The whole book is treated as a document, each chapter is a segment in the experiments carried out in Section 6.6. The theme of each chapter is simulated by the distribution (*i.e.*, $\boldsymbol{\mu}_{i,j}$) over latent topics. The linkage between theme is modelled by the change among topic distributions.

The development of a sequential structured generative model according to the above idea is based on the HPDP, and it models how the subtopic of a segment is correlated to its previous and following segments. Specifically, the correlation

Table 6.1: List of notations used in SeqLDA

Notation.	Description.
K	number of topics
I	number of documents
J_i	number of segments in document i
$L_{i,j}$	number of words in document i, segment j
W	number of words in dictionary
a_i	the discount parameter of the PDP
b_i	the concentration parameter of the PDP
$\boldsymbol{\alpha}$	K-dimensional vector for the Dirichlet prior for document topic distributions
$\boldsymbol{\mu}_{i,0}$	document topic distribution for document i
$\boldsymbol{\mu}_{i,j}$	segment topic distribution for segment j in document i
$\boldsymbol{\Phi}$	word probability vectors as a $K \times W$ matrix
$\boldsymbol{\phi}_k$	word probability vector for topic k, entries in $\boldsymbol{\Phi}$
$\boldsymbol{\gamma}$	W-dimensional vector for the Dirichlet prior for each $\boldsymbol{\phi}_k$
$w_{i,j,l}$	word in document i, segment j, at position l
$z_{i,j,l}$	topic for word in document i, segment j, at position l

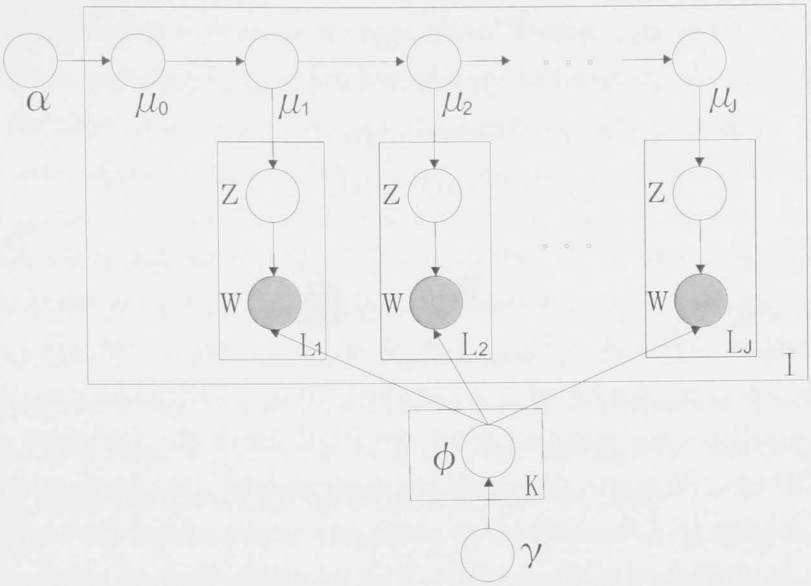


Figure 6.2: SeqLDA

is simulated by the progressive dependency among topic distributions. That is, the j^{th} segment topic distribution $\boldsymbol{\mu}_{i,j}$ is the base distribution of the PDP for drawing the $(j+1)^{th}$ segment topic distribution $\boldsymbol{\mu}_{i,j+1}$; for the first segment, its topic distribution $\boldsymbol{\mu}_{i,1}$ is drawn from the PDP with document topic distribution $\boldsymbol{\mu}_{i,0}$ as the base distribution. The concentration parameter b_i and discount parameter a_i control the variation between the adjacent topic distributions. Figure 6.2 shows the graphical representation of SeqLDA. Shaded and unshaded nodes indicate observed and latent variables respectively. An arrow indicates a conditional dependency between variables, and plates indicate repeated sampling.

In terms of a generative process, SeqLDA can also be viewed as a probabilistic sampling procedure that describes how words in documents can be generated based on the latent topics. It can be depicted as follows: Step 1 samples the word distributions for topics, and Step 2 samples each document by breaking it up into segments:

1. For each topic k in $\{1, \ldots, K\}$,
   (a) Draw $\boldsymbol{\phi}_k \sim \text{Dir}_W(\boldsymbol{\gamma})$
2. For each document i in $\{1, \ldots, I\}$
   (a) Draw $\boldsymbol{\mu}_{i,0} \sim \text{Dir}_K(\boldsymbol{\alpha})$
   (b) For each segment $j \in \{1, \ldots, J_i\}$

i. Draw $\boldsymbol{\mu}_{i,j} \sim \mathrm{PDP}(a_i, b_i, \boldsymbol{\mu}_{i,j-1})$

ii. For each word $w_{i,j,l}$, where $l \in \{1, \ldots, L_{i,j}\}$

   A. draw $z_{i,j,l} \sim \mathrm{Discrete}_K(\boldsymbol{\mu}_{i,j})$

   B. draw $w_{i,j,l} \sim \mathrm{Discrete}_W(\boldsymbol{\phi}_{z_{i,j,l}})$.

Like STM, the number of topics (*i.e.*, the dimensionality of the Dirichlet distribution) is assumed to be known and fixed (*i.e.*, K), and the word probabilities are parameterised by a $K \times W$ matrix $\boldsymbol{\Phi} = (\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_K)$, and will be estimated through the learning process. $\boldsymbol{\mu}_{i,0}$ is sampled from the Dirichlet distribution with prior $\boldsymbol{\alpha}$, and others are sampled from the PDP. Both the Dirichlet distribution and the PDP are conjugate priors for the multinomial distribution, and the PDP is also self-conjugate. Choosing these conjugate priors makes the statistical inference easier, as discussed in Section 6.4. The joint distribution of all observed and latent variables can be constructed directly from Figure 6.2 using the distributions given in the above generative process, as follows:

$$\begin{aligned} & p(\boldsymbol{\mu}_{i,0}, \boldsymbol{\mu}_{i,1:J_i}, \boldsymbol{z}, \boldsymbol{w} | \boldsymbol{\alpha}, \boldsymbol{\Phi}, a_i, b_i) \\ = \; & p(\boldsymbol{\mu}_{i,0}|\boldsymbol{\alpha}) \prod_{j=1}^{J_i} \left(p(\boldsymbol{\mu}_{i,j}|a_i, b_i, \boldsymbol{\mu}_{i,j-1}) \prod_{l=1}^{L_j} p(z_{i,j,l}|\boldsymbol{\mu}_{i,j}) p(w_{i,j,l}|\boldsymbol{\phi}_{z_{i,j,l}}) \right) , \end{aligned} \quad (6.1)$$

where $p(\boldsymbol{\mu}_{i,j}|a_i, b_i, \boldsymbol{\mu}_{i,j-1})$ is given by $\mathrm{PDP}(a_i, b_i, \boldsymbol{\mu}_{i,j-1})$.

From the notion of the proposed model, we can find the obvious distinction between SeqLDA and LDA (shown in Figure 4.1): SeqLDA takes into account the sequential structure of each document, *i.e.*, the segment sequence in a document that LDA ignores. SeqLDA aims to use the information conveyed in the document layout, to capture how topics evolve within a document. Although LDA can also be applied to segments directly, the progressive topical dependency between two adjacent segments could be lost by treating segments independently. LDCC [Shafiei and Milios, 2006], shown in Figure 5.3, has an implicit assumption that segments within each document are exchangeable, which is not always appropriate, so does STM proposed in Chapter 5. Furthermore, assigning just one topic to each segment in LDCC cannot capture the evolution of each topic depicted in the document. Like SeqLDA, STM assumes each segment has a topic distribution, and each segment topic distribution is drawn from document topic distribution via a PDP. As discussed earlier in Section 6.1, STM is developed to explore only the hierarchical relationship between a document subject and its segment subtopics. The exchangeability assumption imposed by STM may make it unsuitable for describing the sequential topic structure.

Thus, if documents indeed have some latent sequential structure, considering this dependency means a higher fidelity of SeqLDA over LDA and LDCC. However, if the correlation among subtopics of some adjacent segments is not obvious, taking the topic distribution of the j^{th} segment as the base distribution of the $(j+1)^{th}$ segment may mis-interpret the document topic structure. In this sense, SeqLDA may be a deficient generative model, but it is still a useful model and remains powerful if the progressive dependency is dynamically changed by optimising concentration and discount parameters (a and b) for each individual segment within each document. In all the reported experiments, I ran one set of experiments with fixed a and b for each corpus, and another set of experiments with a fixed but b optimised for each document i (*i.e.*, b_i).

6.4 Inference Algorithm via CMGS

In this section, I derive the collapsed Gibbs sampling algorithm for doing inference, and parameter estimation in the proposed model. Collapsed Gibbs sampling take advantage of the conjugacy of priors to compute the conditional posteriors. Thus, it always yields relatively simple algorithms for approximate inference in high-dimensional probability distributions. Note that I use conjugate priors in SeqLDA, *i.e.*, Dirichlet prior $\boldsymbol{\alpha}$ on $\boldsymbol{\mu}_0$ and $\boldsymbol{\gamma}$ on $\boldsymbol{\Phi}$, the PDP prior on $\boldsymbol{\mu}_j$; thus $\boldsymbol{\mu}_{0:J}$ and $\boldsymbol{\Phi}$ can be integrated out. Although the proposed sampling algorithm does not directly estimate $\boldsymbol{\mu}_{0:J}$ and $\boldsymbol{\Phi}$, I will show how they can be approximated using the posterior sample statistics.

Table 6.2 lists all the statistics required in the proposed algorithm. The SeqLDA sampling is a collapsed version of what is known as the nested Chinese restaurant process (CRP) used as a component of different topic models [Blei et al., 2010].

6.4.1 Model Likelihood

To derive a collapsed Gibbs sampler for the above model, we need to compute the marginal distribution over the observation $\boldsymbol{w}$, the corresponding topic assignment $\boldsymbol{z}$, and the table multiplicities $\boldsymbol{t}^*$. We do not need to include, *i.e.*, can integrate out, the parameter sets $\boldsymbol{\mu}_{0:J}$ and $\boldsymbol{\Phi}$, since they can be interpreted as statistics of the associations among $\boldsymbol{w}$, $\boldsymbol{z}$ and $\boldsymbol{t}^*$. Hence, we can first recursively apply Equation (3.6) (the joint posterior of HPDP, see Section 3.3) to integrating out the segment topic distributions $\boldsymbol{\mu}_{i,1:J}$ from Equation (6.1) as follows.

$$
\begin{aligned}
&p(\boldsymbol{z}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^*_{1:I,1:J}, \boldsymbol{\mu}_{1:I,0} \,|\, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{\Phi}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I}) \\
&= \prod_{i=1}^{I} p(\boldsymbol{\mu}_{i,0} \,|\, \boldsymbol{\alpha}) \prod_{j=1}^{J_i} \int \underbrace{p(\boldsymbol{\mu}_{i,j} \,|\, a_i, b_i, \boldsymbol{\mu}_{i,j-1})}_{\boldsymbol{\mu}_{i,j} \sim PDP(a_i, b_i, \boldsymbol{\mu}_{i,j-1})} \prod_{l=1}^{L_{i,j}} p(z_{i,j,l} \,|\, \boldsymbol{\mu}_{i,j}) p(w_{i,j,l} \,|\, \boldsymbol{\phi}_{z_{i,j,l}}) \, d\boldsymbol{\mu}_{i,j} \\
&= \prod_{i=1}^{I} \left(\left(\frac{1}{\mathrm{Beta}_K(\boldsymbol{\alpha})} \prod_{k=1}^{K} \mu_{i,0,k}^{\alpha_k - 1} \right) \prod_{j=1}^{J_i} \left(\frac{(b_i | a_i)_{T_{i,j}}}{(b_i)_{N_{i,j} + T_{i,j+1}}} \prod_{k=1}^{K} S^{n_{i,j,k} + t^*_{i,j+1,k}}_{t^*_{i,j,k}, a_i} \right) \prod_{k=1}^{K} \mu_{i,0,k}^{t^*_{i,1,k}} \right) \\
&\quad \prod_{k=1}^{K} \prod_{w=1}^{W} \phi_{k,w}^{M_{i,k,w}} \\
&= \prod_{i=1}^{I} \left(\left(\frac{1}{\mathrm{Beta}_K(\boldsymbol{\alpha})} \prod_{k=1}^{K} \mu_{i,0,k}^{\alpha_k + t^*_{i,1,k} - 1} \right) \prod_{j=1}^{J_i} \left(\frac{(b_i | a_i)_{T_{i,j}}}{(b_i)_{N_{i,j} + T_{i,j+1}}} \prod_{k=1}^{K} S^{n_{i,j,k} + t^*_{i,j+1,k}}_{t^*_{i,j,k}, a_i} \right) \right) \\
&\quad \prod_{k=1}^{K} \prod_{w=1}^{W} \phi_{k,w}^{M_{i,k,w}} \, , \qquad (6.2)
\end{aligned}
$$

Table 6.2: List of statistics used in SeqLDA

Statistic.	Description.
$M_{i,k,w}$	topic by word total sum in document i, the number of words with dictionary index w and topic k, *i.e.*, $M_{i,k,w} = \sum_{j=1}^{J_i} \sum_{l=1}^{L_{i,j}} 1_{z_{i,j,l}=k} 1_{w_{i,j,l}=w}$.
$M_{k,w}$	$M_{i,k,w}$ totalled over documents i, *i.e.*, $\sum_{i=1}^{I} M_{i,k,w}$
$\boldsymbol{M}_k$	vector of W values $M_{k,w}$
$n_{i,j,k}$	topic total in document i and segment j for topic k, *i.e.* $n_{i,j,k} = \sum_{l=1}^{L_{i,j}} 1_{z_{i,j,l}=k}$. It is the total number of customers in the CRP that arrive by themselves, rather than being sent by the child restaurant.
$N_{i,j}$	topic total sum in document i and segment j, *i.e.*, $\sum_{k=1}^{K} n_{i,j,k}$
$t^*_{i,j,k}$	table count in the CRP for document i and segment j, for topic k. This is the number of tables active for the k-th value. Necessarily, $t^*_{i,j,k} \leq n^*_{i,j,k}$ and $t^*_{i,j,k} > 0$ whenever $t^*_{i,j,k} > 0$. In particular, if $n^*_{i,j,k} = 1$ then $t^*_{i,j,k} = 1$.
$T_{i,j}$	total table count in the CRP for document i and segment j, *i.e.* $\sum_{k=1}^{K} t^*_{i,j,k}$.
$\boldsymbol{t}^*_{i,j}$	table count vector, *i.e.*, $(t^*_{i,j,1}, ..., t^*_{i,j,K})$ for segment j.
$u_{i,k}$	the smallest segment index j' in i, where $t^*_{i,j',k} = 0$.

where $t^*_{i,j,k} \leq n_{i,j,k} + t^*_{i,j+1,k}$ and $t^*_{i,j,k} = 0$ *iff* $n_{i,j,k} + t^*_{i,j+1,k} = 0$; $\text{Beta}_K(\boldsymbol{\alpha})$ is a K dimensional beta function that normalises the Dirichlet; $(x)_N$ is given by $(x|1)_N$, and $(x|y)_N$ denotes the Pochhammer symbol (see Section 3.1 for its definition); $S^N_{M,a}$ is the generalised Stirling number (see Section 3.3). Figure 3.4 shows how the segment level topic distributions can be marginalised out in a recursive way to yield Equation 6.2.

Finally, integrate out the document topic distributions $\boldsymbol{\mu}_{i,0}$ and the topic-word matrix $\boldsymbol{\Phi}$, as is usually done for collapsed Gibbs sampling in topic models. The joint distribution of $\boldsymbol{z}_{1:I}$, $\boldsymbol{w}_{1:I}$, $\boldsymbol{t}^*_{1:I,1:J_i}$ is

$$\begin{aligned}
& p(\boldsymbol{z}_{1:I}, \boldsymbol{w}_{1:I}, \boldsymbol{t}^*_{1:I} \,|\, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I}) \\
& = \prod_{i=1}^{I} \left(\frac{\text{Beta}_K(\boldsymbol{\alpha} + \boldsymbol{t}_{i,1})}{\text{Beta}_K(\boldsymbol{\alpha})} \prod_{j=1}^{J_i} \left(\frac{(b_i|a_i)_{T_{i,j}}}{(b_i)_{N_{i,j}+T_{i,j+1}}} \prod_{k=1}^{K} S^{n_{i,j,k}+t^*_{i,j+1,k}}_{t^*_{i,j,k}, a_i} \right) \right) \\
& \quad \prod_{k} \frac{\text{Beta}_W(\boldsymbol{\gamma} + \boldsymbol{M}_k)}{\text{Beta}_W(\boldsymbol{\gamma})} \,.
\end{aligned} \tag{6.3}$$

6.4.2 The Collapsed Gibbs sampler

In each cycle of the Gibbs sampling algorithm, a subset of variables are sampled from their conditional distributions with values of all the other variables given. In SeqLDA, distributions that we need to sample from are the posterior distributions of topics ($\boldsymbol{z}$), and table counts ($\boldsymbol{t}^*$), given a collection of documents. Since the full joint posterior distribution is intractable and difficult to sample from, in each cycle of Gibbs sampling we will sample respectively from two conditional distributions: 1) the conditional distribution of topic assignment ($z_{i,j,l}$) of a single word ($w_{i,j,l}$) given topic assignments for all the other words and all the table counts; 2) the conditional distribution of table count ($t^*_{i,j,k}$) of the current topic given all the other table counts and all the topic assignments. In particular, the sampling strategy adopted here is CMGS discussed in Section 3.3. Notice that sampling table counts from the latter can be taken as a stochastic process of rearranging the seating plan of a Chinese restaurant in the CRP.

In SeqLDA, documents are indexed by i, segments of each document are indexed by j according to their original layout, and words are indexed by l. Thus, with documents indexed by the above method, we can readily yield a Gibbs sampling algorithm for SeqLDA: for each word, the algorithm computes the probability of assigning the current word to topics from the first conditional distribution, while topic assignments of all the other words and table counts are fixed. Then the current word would be assigned to a sampled topic, and this assignment will be stored while the Gibbs sampling cycles through other words. While scanning through the list of words, we should also keep track of table counts for each segment. For each new topic that the current word is assigned to, the Gibbs sampling algorithm estimates the probabilities of changing the corresponding table count to different values by fixing all the topic assignments and all the other table counts. These probabilities are computed from the second conditional distribution. Then, a new value will be sampled and assigned to the current table count. Note that the values of the table count should be subject to some constraints that I will discuss in detail when deriving the two conditional distributions. Consequently, the aforementioned two conditional distributions need to be computed are, respectively,

1. $p(z_{i,j,l} = k \,|\, \boldsymbol{z}_{1:I,1:J}^{-z_{i,j,l}}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^*_{1:I,1:J}, \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I})$,

2. $p(t^*_{i,j,k} \,|\, \boldsymbol{z}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}_{1:I,1:J}^{*-t^*_{i,j,k}}, \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I})$,

where $z_{i,j,l} = k$ indicates the assignment of the l^{th} word in the j^{th} segment of document i to topic k, $\boldsymbol{z}_{1:I,1:J}^{-z_{i,j,l}}$ presents all the topic assignments not including the l^{th} word, and $\boldsymbol{t}_{1:I,1:J}^{*-t^*_{i,j,k}}$ denotes all the table counts except for the current table count $t^*_{i,j,k}$. Before elaborating the derivation of these two distributions, I discuss constraints on the table count ($t^*_{i,j,k}$) and the word count ($n_{i,j,k}$) for each topic. Following the CRP formulation (see Chapter 3), customers are words, dishes are topics and restaurants are segments. All restaurants share a finite number of dishes, *i.e.*, K dishes. From Equation (6.3) and also seen from Equation (3.6) in Section 3.3, tables of the $(j+1)^{th}$ restaurant are customers of the j^{th} restaurant in hierarchical CRPs, as depicted in Figure 3.4. These counts have to comply with the following constraints:

1. $t^*_{i,j,k} = 0$ if and only if $n_{i,j,k} + t^*_{i,j+1,k} = 0$;

2. $t^*_{i,j,k} > 0$ if either $n_{i,j,k} > 0$ or $t^*_{i,j+1,k} > 0$;

3. $n_{i,j,k} + t^*_{i,j+1,k} \geq t^*_{i,j,k} \geq 0$.

For instance, the third constraint says that the total number of occupied tables serving the k^{th} dish must be less than or equal to the total number of customers eating this dish. That is because each occupied table must at least have one customer. Handling the constraints on all the table counts $t^*_{i,j,k}$ is the key challenge in the development of the collapsed Gibbs algorithm.

Considering the procedure of sampling a new topic for a word $w_{i,j,l}$, we need to remove the current topic (referred to as old topic) from the statistics. Assume the value of old topic $z_{i,j,l}$ is k, the number of words assigned to k in the j^{th} segment of document i, $n_{i,j,k}$, should decrease by one; then recursively check the table count $t^*_{i,j',k}$ for $1 \leq j' \leq j$ according to the above constraints, and remove one if needed to satisfy the constraints, this check will proceed until somewhere the constraints hold; and finally assign the smallest j' to $u_{i,k}$ where the first constraint holds. Similarly, the same process should be done when assigning the current word to a new topic. It is easy to prove, by recursion, that no $t^*_{i,j,k}$ goes from zero to non-zero or *vice versa* unless an $n_{i,j,k}$ does, so we only need to consider the case where $n_{i,j,k} + t^*_{i,j+1,k} > 0$. Moreover, the zero $t^*_{i,j,k}$ forms a complete suffix of the list of segments, so $t^*_{i,j,k} = 0$ if and only if $u_{i,k} \leq j \leq J_i$ for some $u_{i,k}$.

Now, beginning with the joint distribution, Equation (6.3), using the chain rule, and taking into account all cases, we can obtain the final full conditional distribution

$$\begin{aligned} & p(z_{i,j,l} = k \mid \boldsymbol{z}^{-z_{i,j,l}}_{1:I,I:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^*_{1:I,1:J}, \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I}) \\ = & \frac{p(\boldsymbol{z}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^*_{1:I,1:J} \mid \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I})}{p(\boldsymbol{z}^{-z_{i,j,l}}_{1:I,1:J}, \boldsymbol{w}_{1:I}, \boldsymbol{t}^*_{1:I,1:J} \mid \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I})} \end{aligned}$$

with three different cases according to the value of $u_{i,k}$ as follows.

When $u_{i,k} = 1$, which means all the table counts $t^*_{i,j',k}$ for $1 \leq j' \leq J_i$ are zero,

$$\begin{aligned} & p(z_{i,j,l} = k \mid \boldsymbol{z}^{-z_{i,j,l}}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^*_{1:I,1:J}, \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I}) \\ = & \frac{(\alpha_k + t^*_{i,1,k})\,(b_i + a_i T_{i,1})}{\sum_{k=1}^{K} (\alpha_k + t^*_{i,1,k})} \prod_{j'=2}^{j} \left(\frac{b_i + a_i T_{i,j'}}{b_i + N_{i,j'-1} + T_{i,j'}} \right) \frac{\gamma_{w_{i,j,l}} + M_{k,w_{i,j,l}}}{\sum_{w=1}^{W} (\gamma_w + M_{k,w})} \,. \end{aligned} \tag{6.4}$$

When $1 < u_{i,k} \leq j$, which means all the table counts $t^*_{i,j',k}$ for $u_{i,k} \leq j' \leq J_i$ are zero, the conditional probability is

$$\begin{aligned} & p(z_{i,j,l} = k \mid \boldsymbol{z}^{-z_{i,j,l}}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^*_{1:I,1:J}, \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I}) \\ = & \prod_{j'=u_{i,k}}^{j} \left(\frac{b_i + a_i T_{i,j'}}{b_i + N_{i,j'-1} + T_{i,j'}} \right) \frac{S^{n_{i,u_{i,k}-1,k}+1}_{t^*_{i,u_{i,k}-1,k}, a_i}}{S^{n_{i,u_{i,k}-1,k}}_{t^*_{i,u_{i,k}-1,k}, a_i}} \frac{\gamma_{w_{i,j,l}} + M_{k,w_{i,j,l}}}{\sum_{w=1}^{W} (\gamma_w + M_{k,w})} \,. \end{aligned} \tag{6.5}$$

When $j < u_{i,k}$, which means the current table count $t^*_{i,j,k} > 0$ (no recursive check), it is simplified to

$$\begin{aligned} & p(z_{i,j,l} = k \mid \boldsymbol{z}^{-z_{i,j,l}}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^*_{1:I,1:J}, \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I}) \\ = & \frac{S^{n_{i,j,k}+1+t^*_{i,j+1,k}}_{t^*_{i,j,k}, a_i}}{S^{n_{i,j,k}+t^*_{i,j+1,k}}_{t^*_{i,j,k}, a_i}} \frac{\gamma_{w_{i,j,l}} + M_{k,w_{i,j,l}}}{\sum_{w=1}^{W} (\gamma_w + M_{k,w})} \,. \end{aligned} \tag{6.6}$$

After sampling the new topic for a word, we need to stochastically sample the table count for this new topic, say k. Although we have summed out the specific seating arrangements (*i.e.*, different tables and specific table assignments) of the customers in the collapsed Gibbs sampler, it is still needed to sample how many tables are serving the k^{th} dish (*i.e.*, topic k in SeqLDA), given the current number of customers (*i.e.*, words) eating the k^{th} dish. If $n_{i,j,k} + t^*_{i,j+1,k} > 1$, the value of $t^*_{i,j,k}$ should be in the following interval:

$$t^*_{i,j,k} \in \left[\max\left(1, t^*_{i,j-1,k} - n_{i,j-1,k}\right), n_{i,j,k} + t^*_{i,j+1,k}\right].$$

Thus, given the current state of topic assignment of each word, the conditional distribution for table count $t^*_{i,j,k}$ can be obtained by similar arguments, as follows.

$$\begin{aligned} & p(t_{i,j,k} \,|\, \boldsymbol{z}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^{*-t^*_{i,j,k}}_{1:I,1:J}, \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I}) \\ & = \frac{p(\boldsymbol{z}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^*_{1:I,1:J} \,|\, \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I})}{p(\boldsymbol{z}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{t}^{*-t^*_{i,j,k}}_{1:I,1:J} \,|\, \boldsymbol{\alpha}, \boldsymbol{a}_{1:I}, \boldsymbol{b}_{1:I})} \\ & \propto \left(\frac{\Gamma\left(\alpha_k + t^*_{i,1,k}\right)}{\Gamma\left(\sum_{k=1}^{K}\left(\alpha_k + t^*_{i,1,k}\right)\right)}\right)^{\delta_{j,1}} \left(\frac{S^{n_{i,j-1,k}+t^*_{i,j,k}}_{t^*_{i,j-1,k},a_i}}{(b_i)_{N_{i,j-1}+T_{i,j}}}\right)^{1-\delta_{j,1}} \quad (b_i|a_i)_{T_{i,j}} S^{n_{i,j,k}+t^*_{i,j+1,k}}_{t^*_{i,j,k},a_i} . \end{aligned} \tag{6.7}$$

The collapsed Gibbs sampling algorithm for SeqLDA is outlined in Algorithm 5. This algorithm is started by randomly assigning words to topics in $[1, \ldots, K]$, and if the total number of customer, $n_{i,j,k} + t^*_{i,j+1,k}$, is greater than zero, the table count $t^*_{i,j,k}$ is initialised to 1. Each Gibbs circle then applies Equations (6.4), (6.5) or (6.6) to every word in the document collection; and applying Equation (6.7) to each table count. Note Steps 18 and 19 will be detailed in Section 6.5. A number of initial samples, *i.e.*, samples before *burn-in* period, have to be discarded. After that, the Gibbs samples should theoretically approximate the target distribution (*i.e.*, the posterior distribution of topics ($\boldsymbol{z}$), and table counts ($\boldsymbol{t}$)). Now, a number of Gibbs samples are drawn at regularly spaced intervals. In experiments discussed in Section 6.6, I averaged these samples to obtain the final sample, as done in [Rosen-Zvi et al., 2004]. This collapsed Gibbs sampling algorithm is easy to implement and requires little memory.

6.4.3 Estimating Topic/Word Distributions

Now, we can easily estimate the topic distribution $\boldsymbol{\mu}$ and topic-word distribution $\boldsymbol{\Phi}$, from statistics obtained after the convergence of the Markov chain. They can

Algorithm 5 Collapsed Gibbs sampling algorithm for SeqLDA

Require: $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$, K, $Corpus$, $MaxIteration$

Ensure: topic assignments for all words and all table counts

```
Topic assignment initialisation: randomly initialise the topic assignment for all words.
Table count initialisation: randomly initialise all $t^*_{i,j,k}$ $s.t.$ $0 \leq t^*_{i,j,k} \leq n_{i,j,k} + t^*_{i,j+1,k}$
Compute statistics listed in Table 6.2
for $iter \leftarrow 1$ to $MaxIteration$ do
  foreach document $i$ do
    foreach segment $j$ in $i$, according to the original layout do
      foreach word $w_{i,j,l}$ in $j$ do
        Exclude $w_{i,j,l}$, and update the statistics with current topic $k' = z_{i,j,k}$ removed
        Recursively check all table counts, $t^*_{i,j',k'}$, where $1 \leq j' \leq j$, to make sure $0 \leq t^*_{i,j',k'} \leq n_{i,j',k'} + t^*_{i,j'+1,k'}$ holds;
        Look for the smallest $1 \leq j' \leq j$, $s.t.$ $t^*_{i,j',k'} = 0$, and assign it to $u_{i,k'}$
        Sample new topic $k$ for $w_{i,j,l}$ using Equations (6.4), (6.5) or (6.6) depending on the value of $u_{i,k}$
        Update the statistics with the new topic, and also update the value of $u_{i,k}$ if needed
        Remove the current table count $t^*_{i,j,k}$ from the statistics
        Sample new table count $t^*_{i,j,k}$ for the new topic $k$ using Equation (6.7)
        Update the statistics with the new table count
      end for
    end for
    Update $\boldsymbol{\alpha}$ by Newton-Raphson method
    Sample $b_i$ with adaptive rejection sampling
  end for
end for
```

be approximated from the following mean posterior expected values (using the mean of a Dirichlet distribution (see Property 2.1) and the mean of the PDP (see Property 2.5)) via sampling. For the document topic distribution $\boldsymbol{\mu}_{i,0}$, we have

$$\widehat{\mu}_{i,0,k} = \mathbb{E}_{\boldsymbol{z}_{i,1:J_i},\boldsymbol{t}^*_{i,1:J_i} \mid \boldsymbol{w}_{i,1:J_i},\boldsymbol{\alpha},\boldsymbol{\gamma},a_i,b_i} \left[\frac{\alpha_k + t^*_{i,0,k}}{\sum_{k=1}^{K} \left(\alpha_k + t^*_{i,0,k}\right)} \right]. \tag{6.8}$$

And the segment topic distribution $\boldsymbol{\mu}_{i,j}$ $(1 \leq j \leq J_i)$ can be estimated as

$$\begin{aligned} &\widehat{\mu}_{i,j,k} = \\ &\quad \mathbb{E}_{\boldsymbol{z}_{i,1:J_i},\boldsymbol{t}^*_{i,1:J_i} \mid \boldsymbol{w}_{i,1:J_i},\boldsymbol{\alpha},\boldsymbol{\gamma},a_i,b_i} \left[\frac{a_i T_{i,j} + b_i}{b_i + N_{i,j} + T_{i,j+1}} \mu_{i,j-1,k} + \frac{(n_{i,j,k} + t^*_{i,j+1,k}) - a_i t^*_{i,j,k}}{b_i + N_{i,j} + T_{i,j+1}} \right]. \end{aligned} \tag{6.9}$$

Then, the topic-word distribution is given by

$$\widehat{\phi}_{k,w} \;=\; \mathbb{E}_{\boldsymbol{z}_{1:I,1:J},\boldsymbol{t}_{1:I,1:J} \mid \boldsymbol{w}_{1:I,1:J},\boldsymbol{\alpha},\boldsymbol{\gamma},\boldsymbol{a},\boldsymbol{b}} \left[\frac{\gamma_w + M_{k,w}}{\sum_{w'=1}^{W} (\gamma_{w'} + M_{k,w'})} \right]. \tag{6.10}$$

6.5 Estimating Hyper-parameters

Since the PDP is quite sensitive to the concentration parameters (*i.e.*, $\boldsymbol{b}_{1:I}$), which has been observed in our initial experiments, also see Section 5.6.2, I thus propose an algorithm to sample b_i for each documents using the Beta/Gamma auxiliary variable trick, as those in [Du et al., 2010b; Teh, 2006a]. The sampling routine is based on the joint distribution Equation (6.3).

First let us consider the case when the discount parameter $a_i = 0$, which is same to what has been discussed in Section 5.4.3. The posterior for b_i is proportional to

$$\prod_{j=1}^{J_i} \frac{b_i^{T_{i,j}} \Gamma(b_i)}{\Gamma(b_i + N_{i,j} + T_{i,j+1})} \; .$$

Now I introduce an auxiliary variable $q_{i,j} \sim Beta(b_i, N_{i,j} + T_{i,j+1})$ for each segment i, j. Then the joint posterior distribution for $q_{i,j}$ and b_i is proportional to

$$b_i^{\sum_{j=1}^{J_i} T_{i,j}} \prod_{j}^{J_i} q_{i,j}^{b_i - 1} (1 - q_{i,j})^{N_{i,j} + T_{i,j+1} - 1} \; . \tag{6.11}$$

Given sampled values of all the auxiliary variables, we can sample b_i according to their conditional distributions,

$$\begin{aligned} q_{i,j} &\sim Beta(b_i, N_{i,j} + T_{i,j+1}) \\ b_i &\sim Gamma\left(\sum_{j=1}^{J_i} T_{i,j}, \sum_{j=1}^{J_i} log(1/q_{i,j}) \right) \; . \end{aligned}$$

For the case when $a_i > 0$, the sampling scheme become a bit more elaborate. Now the posterior for b_i is proportional to

$$a_i^{\sum_j^{J_i} T_{i,j}} \prod_j \frac{\Gamma(b_i/a_i + T_{i,j})}{\Gamma(b_i/a_i)} \frac{\Gamma(b_i)}{\Gamma(b_i + N_{i,j} + T_{i,j+1})} .$$

Introducing the same auxiliary variables, as those for $a = 0$, we can yield a joint posterior distribution proportional to

$$a_i^{\sum_j^{J_i} T_{i,j}} \prod_j^{J_i} \frac{\Gamma(b_i/a_i + T_{i,j})}{\Gamma(b_i/a_i)} q_{i,j}^{b_i-1} (1 - q_{i,j})^{N_{i,j}+T_{i,j+1}-1}. \tag{6.12}$$

It is easy to show that the above distribution is log concave in b_i, so I here adopt an adaptive rejection sampling algorithm [Gilks and Wild, 1992]. Sampling the concentration parameter b allows a different value for each document, even for each segment with only a slight modification of Equations (6.11) and (6.12). In addition, although I did not study the discount parameter a_i in this chapter, it could also be optimised or sampled.

Instead of using symmetric Dirichlet prior $\boldsymbol{\alpha}$, we can use an asymmetric Dirichlet prior whose components have to be estimated. As argued by Wallach et al. [2009], the use of asymmetric prior on $\mu_{i,0}$ could lead to a significant performance improvement. Algorithms for estimating Dirichlet priors proposed in the literature are based on either maximum likelihood or maximum a posteriori, such as the Moment-Matching and the Newton-Raphson iteration. Here, I adopt the Newton-Raphson method following the early work by Minka [2000]. According to Equation (6.3), the gradient of the log-likelihood is

$$\begin{aligned}\frac{\partial f(\boldsymbol{\alpha})}{\partial \alpha_k} &= \sum_{i=1}^{I} \left(\Psi \left(\sum_{k=1}^{K} \alpha_k \right) - \Psi \left(\sum_{k=1}^{K} (\alpha_k + t_{i,1,k}) \right) \right) \\ &\quad + \sum_{i=1}^{I} \left(\Psi \left(\alpha_k + t_{i,1,k} \right) - \Psi \left(\alpha_k \right) \right),\end{aligned}$$

where $\Psi(\cdot)$ is known as the digamma function that is the first derivative of log gamma function, and $f(\boldsymbol{\alpha})$ is the model log likelihood parameterised with $\boldsymbol{\alpha}$,

$$f(\boldsymbol{\alpha}) \propto log \left(\prod_{i=1}^{I} \frac{\mathrm{Beta}_K\left(\boldsymbol{\alpha} + \boldsymbol{t}_{i,1}\right)}{\mathrm{Beta}_K\left(\boldsymbol{\alpha}\right)} \right).$$

Then, the Hessian of the log-likelihood is

$$\begin{aligned}
\frac{\partial f(\boldsymbol{\alpha})}{\partial \alpha_k^2} &= \sum_{i=1}^{I}\left(\Psi'\left(\sum_{k=1}^{K}\alpha_k\right) - \Psi'\left(\sum_{k=1}^{K}(\alpha_k + t_{i,1,k})\right)\right) \\
&\quad + \sum_{i=1}^{I}\left(\Psi'\left(\alpha_k + t_{i,1,k}\right) - \Psi'\left(\alpha_k\right)\right) \\
\frac{\partial f(\boldsymbol{\alpha})}{\partial \alpha_k\, \partial \alpha_{k'}} &= \sum_{i=1}^{I}\left(\Psi'\left(\sum_{k=1}^{K}\alpha_k\right) - \Psi'\left(\sum_{k=1}^{K}(\alpha_k + t_{i,1,k})\right)\right) \qquad \text{where } k \neq k' ,
\end{aligned}$$

and $\Psi'(\cdot)$ is the trigamma function, *i.e.*, the second derivative of gamma function. Now, a Newton iteration can be computed to optimise Dirichlet prior $\boldsymbol{\alpha}$. In the reported experiments, I interchangeably upgrade $\boldsymbol{b}$ and $\boldsymbol{\alpha}$ after each main Gibbs sampling iteration. For example, I optimise $\boldsymbol{\alpha}$ for the first 300 iterations with $\boldsymbol{b}$ fixed; then, optimise $\boldsymbol{b}$ for the next 300 iterations with $\boldsymbol{\alpha}$ fixed, and so on. As we can see, I adopt a more greedy approach to optimise the two parameters simultaneously, which may not give a global optimum.

6.6 Experimental Results

I implemented LDA, LDCC and SeqLDA in C, and ran them on a desktop with Intel(R) Core(TM) Quad CPU (2.4GHz), even though my code is not multi-threaded. The experiment environment is the same as the environment for STM. The previous experimental results, presented in STM, show that, LDCC performs quite similarly to LDA working on the segment level in terms of document modelling accuracy. On the other hand, LDCC is not designed to uncover sequential topic structure either, neither does STM. Thus, I compare SeqLDA directly with LDA working on both the document and the segment levels to facilitate easy comparison.

In this section, I first discuss the perplexity (see Equation 5.10) comparison between SeqLDA and LDA on a patent dataset. The held-out perplexity measure [Rosen-Zvi et al., 2004] is employed to evaluate the generalisation capability to the unseen data. Then, I present topic evolution analysis on two books, available at `http://www.gutenberg.org`. The former will show that SeqLDA is significantly better than LDA with respect to document modelling accuracy as measured by perplexity; and the latter will typically demonstrate the superiority of SeqLDA in topic evolution analysis.

Table 6.3: Dataset statistics

	The Prince	Moby Dick	Pat-1000	
			Training	Testing
No. of documents	1	1	800	200
No. of segments	26	135	49,200	11,360
No. of words	10,588	88,802	2,048,600	464,460
Vocabulary	3,292	16,223	10,385	

6.6.1 Data Sets

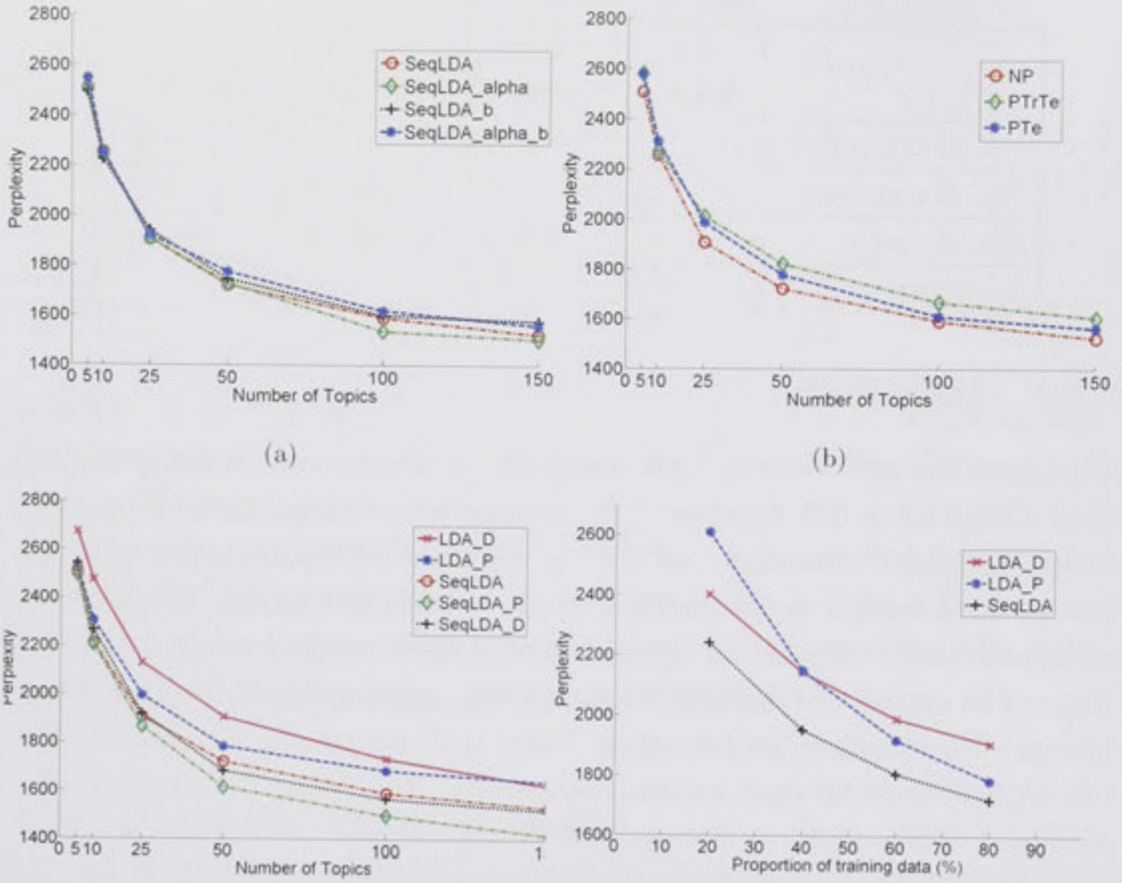
The patent dataset (*i.e.*, Pat-1000) has 1000 patents that are randomly selected from a large set of U.S. patents[2]. They are granted between Jan. and Apr. 2009 under the class "*computing; calculating; counting*". All patents are split into paragraphs according to the original layout in order to preserve the document structure. I have removed all stop-words, extremely common words (*i.e.*, most frequent 50 words), and less common words (*i.e.*, words appear in less than 5 documents). No stemming has been done. I here treat paragraphs as segments. The two books I choose for topic evolution analysis are "The Prince" by Niccolò Machiavelli and "Moby Dick" by Herman Melville, also known as "The Whale". They are split into chapters which are treated as segments, and only stop-words are removed. Table 6.3 shows the statistics of these datasets.

6.6.2 Document modelling

I follow the standard way in document modelling to evaluate the per-word predicative perplexity of SeqLDA and LDA on the Pat-1000 dataset with 20% held out for testing. In order to calculate the likelihood for each unseen word in the SeqLDA model, we need to integrate out the sampled distributions (*i.e.*, $\boldsymbol{\mu}$ and $\boldsymbol{\Phi}$) and sum over all possible topic assignments. Here, I approximate the integrals using a Gibbs sampler with Equations (6.8), (6.9) and (6.10) for each sample of assignments $\boldsymbol{z}$ and $\boldsymbol{t}$. In sampling procedures, I run each Gibbs sampler for 2,000 iterations with 1,500 burn-in iterations. After the burn-in period, a total number of 5 samples are drawn at a lag of 100 iterations. These samples are averaged to yield the final trained model.

I first investigate the performance of SeqLDA with or without the hyper-

[2] All patents are from Cambia, `http://www.cambia.org/daisy/cambia/home.html`



(a)

(b)

(c)

(d)

Figure 6.3: Perplexity comparison on the Pat-1000 dataset.

Table 6.4: P-values for paired t-test for results in Figure 6.3(a)

	Pat-1000		
	SeqLDA_alpha	SeqLDA_b	SeqLDA_alpha_b
SeqLDA	2.2e-1	2.8e-1	1.3e-2

Table 6.5: P-values for paired t-test for results in Figure 6.3(c)

	Pat-1000		
	SeqLDA	SeqLDA_D	SeqLDA_P
LDA_D	7.5e-4	3.3e-4	3.2e-5
LDA_P	3.0e-3	1.9e-2	3.6e-3

parameter estimation proposed in Section 6.5. Four sets of experiments[3] have been done. They are, respectively, SeqLDA with $\alpha = 0.10$ (*i.e.*, symmetric $\boldsymbol{\alpha}$), $b = 10$ and $a = 0.2$ (SeqLDA); with $\boldsymbol{\alpha}$ optimised by Newton-Raphson method, $b = 10$ and $a = 0.2$ (SeqLDA_alpha); with $\alpha = 0.10$, b optimised by sampling method and $a = 0.2$ (SeqLDA_b); and with both $\boldsymbol{\alpha}$ and b optimised and $a = 0.2$ (SeqLDA_alpha_b). Note that for simplicity, b is optimised for each document, even though we can optimise b for each individual segment. Figure 6.3(a) shows the results in terms of perplexity.

According to the p-values of the paired t-test (as shown in Table 6.4), there is no significant difference between the manually optimised SeqLDA and the automatically optimised models at the significant level 5%. It has been observed that the average value of the optimised asymmetric α is close to 0.10. The perplexity of SeqLDA with only alpha optimised becomes lower than others when k is getting larger ($k > 50$). In contrast, SeqLDA with both α and b optimised yields slightly higher perplexity. This might be because the way that I used to carry out the optimisation is approximately greedy, which cannot reach a global optimum for both α and b. We can therefore conclude that the hyper-parameter optimisation algorithms work as well as the manual optimisation. And, I can further claim that these hyper-parameters are not difficult to set up in order to get good results.

Secondly, I ran another set of experiments to verify whether there indeed exists a sequential topical dependency among segments of each document. Instead of retaining the original layout of segments (*i.e.*, the original order of paragraphs in a patent), I have randomly permuted the order of the segments for both the training dataset and the testing dataset. In Figure 6.3(b), "NP" indicates seqLDA trained and tested without permutation, "PTrTe" indicates the model trained and tested with permutation, and "PTe" indicates the model tested with permutation but trained without permutation. Taking $k = 25$ as an example, the perplexity corresponding to the original layout (1905.2) is much lower than that corresponding to the randomly permuted order (2009.8). Thus, the significant difference shows that the sequential topical structure does exist in the patents, and considering this structure can improve the accuracy of text analysis in terms of perplexity.

Thirdly, I compare SeqLDA with LDA. In order to make a fair comparison, I set hyper-parameters fairly, since they are important for the two models. The

[3] I have first done a series of experiments with the value of α ranging from 0.01 to 0.90 to manually choose the optimal one, which is 0.10. And the values of b and a are chosen empirically based on the initial experiments. They are $b = 10$ and $a = 0.20$

Moment-Matching algorithm Minka [2000] is used to optimise $\boldsymbol{\alpha}$ for LDA, and all the parameters for SeqLDA are fixed as: $a = 0.2$, $b = 10$, $\alpha = 0.1$. And γ is set to $200/W$ for both models. Note that I seek to automatically optimise the parameter settings for LDA, which enables one to draw fair conclusions on SeqLDA's performance.

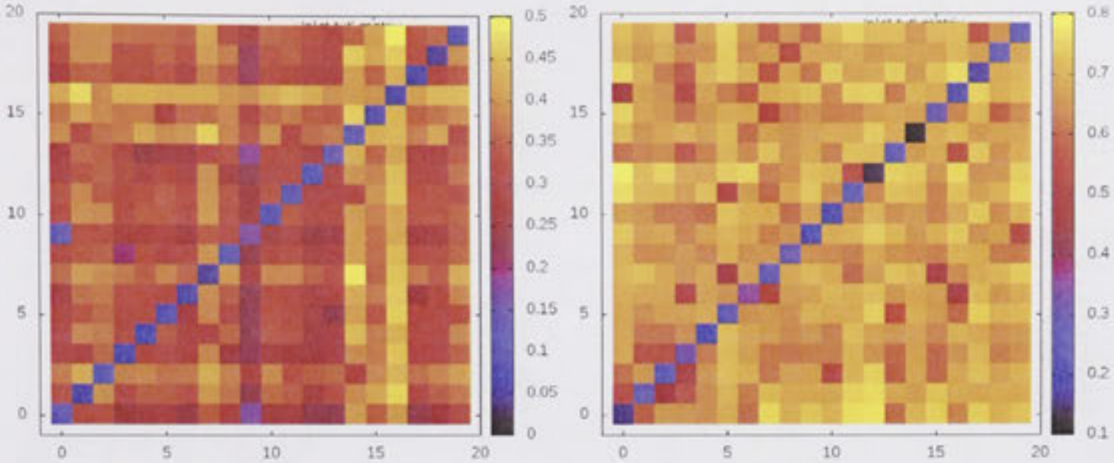
Figure 6.3(c) demonstrates the perplexity comparison for different number of topics. LDA has been tested on document level (LDA_D) and paragraph level (LDA_P) separately. I have also run SeqLDA with or without being boosted by either LDA_D (SeqLDA_D) or LDA_P (SeqLDA_P). The boosting is done by using the topic assignments learnt by LDA to initialise SeqLDA. As shown in the figure, SeqLDA, either with or without boosting, consistently performs better than both LDA_D and LDA_P. The p-values from the paired-t test shown in Table 6.5 are always smaller than 0.05, which has clearly indicated that the advantage of SeqLDA over LDA is statistically significant. Evidently, the topical dependencies information propagated through the sequential document structure, for the patent dataset, indeed exists; and explicitly considering the dependency structure in topic modelling, as SeqLDA does, can be valuable to help understand the original text content.

In my last set of experiments for perplexity comparison, I show the perplexity comparison by changing the proportion of training data. In these experiments, the number of topics for both LDA and SeqLDA are assumed to be fixed and equal to 50. As shown in Figure 6.3(d), SeqLDA (without boosting) always performs better than LDA as the proportion of training data increases. The training time, for example, with 80% patents for training and 2000 Gibbs iterations, is approximately 5 hours for LDA, and 25 hours for SeqLDA, which indicates that SeqLDA is still reasonably manageable in terms of training time.

6.6.3 Topic Distribution Profile over Segments

Besides better modelling perplexity, another key contribution of SeqLDA is the ability to discover underlying sequential topic evolution within a document. With this, one can further perceive how the author organises, for instance, her stories in a book or her ideas in an essay. Here, I test SeqLDA on two books with following parameter settings: $a = 0$, $\alpha = 0.5$, $k = 20$, $b = 25$ for "The Prince", and $b = 50$ for "Moby Dick".

To compare the topics of SeqLDA and LDA, we have to solve the problem of topic alignment, since topics learnt in separate runs have no intrinsic align-



(a) The Prince

(b) Moby Dick

Figure 6.4: Topic alignment by confusion matrix

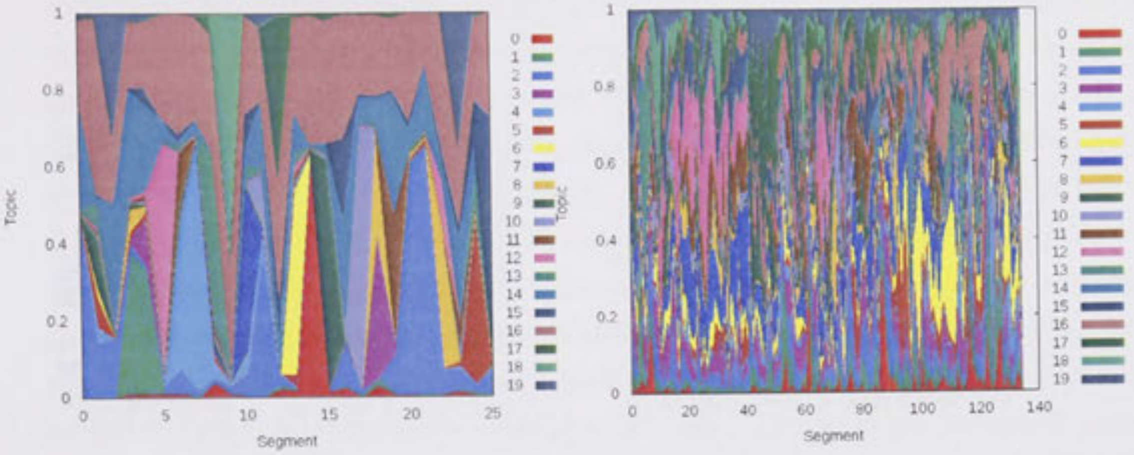
ment. The approach I adopt is to start the SeqLDA's Gibbs sampling with the topic assignments learnt from LDA. Figure 6.4(a) and Figure 6.4(b) show the confusion matrices between the topic distributions generated by SeqLDA and LDA with Hellinger Distance, where SeqLDA topics run along the X-axis. Most topics are well aligned (with blue on the diagonal and yellow off diagonal), especially those for "Moby Dick". For "The Prince", the major confusion is with topic-0 and 9 yielding some blueish off diagonal. Table 6.6 shows some topic examples learnt from "The Prince".

After aligning the topics, I plot the topic distributions (*i.e.*, subtopics) as a function of chapter to show how each topic evolves, as shown in Figure 6.5 and Figure 6.6 respectively. Immediately, we can see that the topic evolving patterns over chapters learnt by SeqLDA are much clearer that those learnt by LDA. For example, compare the subfigures in these two figures, it is a bit hard to find the topic evolution patterns in Figure 6.5(b) learnt by LDA; in contrast, we can find the patterns in Figure 6.6(b), for example, topic-7, which is about men on board ship generally, and topic-12, which is about the speech of old ("thou," "thee," "aye," "lad") co-occur together from chapters 15 to 40 and again around chapters 65-70, which is coherent with the book.

Moreover, Figure 6.7(a) and Figure 6.7(b) depict the Hellinger distances (also as a function of chapters) between the topic distributions of two consecutive chapters to measure how smoothly topics evolve through the books. Obviously, the topic evolution learnt by SeqLDA is much better than that learnt by LDA. SeqLDA always yields smaller Hellinger distances and smaller variance of

Table 6.6: Typical topics learnt from "The Prince". Top 30 words are listed as examples.

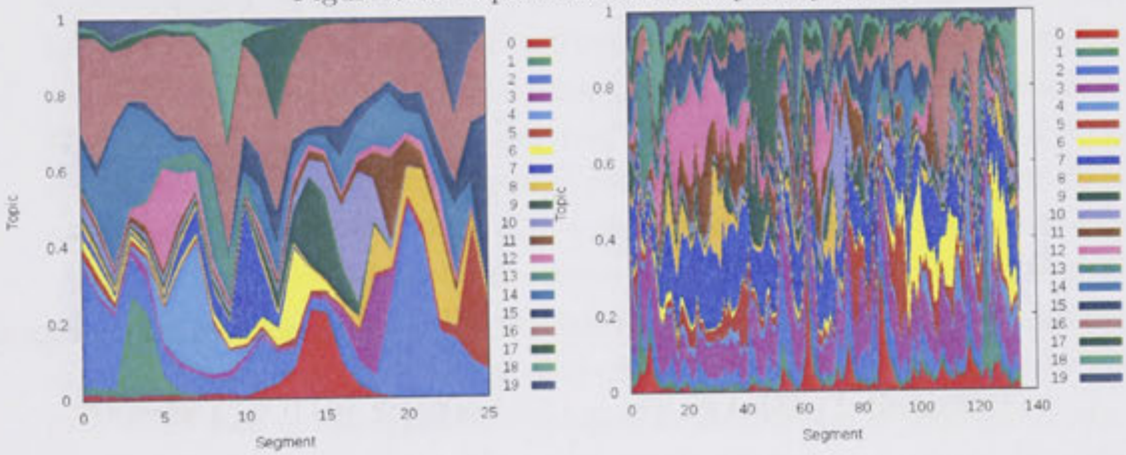
LDA	topic-0	servant servants pandolfo good opinion cares honours recognise honest comprehends venafro trust attention fails praise judgment honouring form thinking correct error clever choosing rank disposed prime useless Sinea faithfull study
	topic-9	truth emperor flatterers opinions counsels wisdom contempt advice listen preserved bold counsel resolutions speaking maximilain patient unite born deceived case affairs short anger prove receive support steadfast guarding discriminating inferred
SeqLDA	topic-0	servant flatterers pandolfo opinions truth good hones question emperor counsels form cares opinion servants wisdom comprehends enable interests honours contempt fails venafro preserved maximilain choosing advantageous listen thinking capable recognise
	topic-9	support cardinals labours fortify walls temporal fortified courageous pontificate spirits resources damage town potentates character barons burnt ecclesiastical principalities defence year firing hot attack pursuit loss showed enemy naturally
	topic-15	people nobles principality favour government times hostile ways oppressed enemies secure give messer friendly rule security courage authority satisfy arises fail rome receive finds adversity civil builds aid expect cities
	topic-16	prince men great good state princes man things make time fear considered subject found long wise army people affaires defend whilst actions life fortune difficulty present mind faithful examples roman



(a) The Prince

(b) Moby Dick

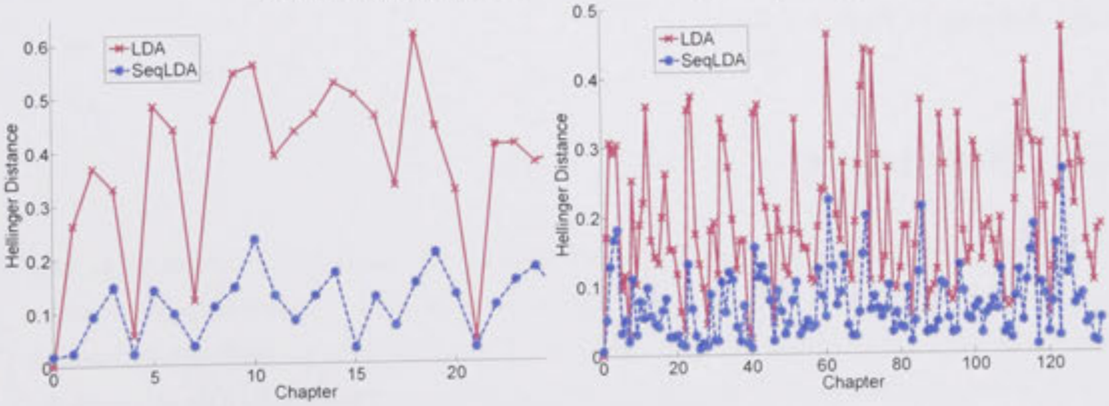
Figure 6.5: Topic evolution analysis by LDA



(a) The Prince

(b) Moby Dick

Figure 6.6: Topic evolution analysis by SeqLDA



(a) The Prince

(b) Moby Dick

Figure 6.7: Topic evolution by Hellinger Distance

distances. The big topic shifts found by LDA are also highlighted by SeqLDA, such as Chapter 7 to 10 in Figure 6.7(a). Evidently, SeqLDA has avoided heavy topic drifting, and makes the topic flow between chapters much smoother than LDA does. An immediate and obvious effect is that this can help readers understand more precisely how a book is organised.

Consider "The Prince" in more detail. The topic that is most unchanged in "The Prince" is topic-16 (having the lightest yellow in off-diagonal in Figure 6.4(a)), also show in Table 6.6. This topic occurs consistently through the chapters in both models and can be seen to really be the core topic of the book. Topic-15 is another topic that has not changed much, and it has its occurrence broadened considerably; for SeqLDA it now occurs throughout the second half of the book starting at chapter 10; the topic is about the nature of governing principalities as opposed to the first 9 chapters which cover how principalities are formed and how princes gain their titles. Now consider the issue of topic-0 and 9. Inspection shows topic-9 learnt by LDA occurring in Chapters 2 and 16 is split into two by SeqLDA: the chapter 16 part joins topic-0 which has its strength in the neighbouring Chapter 15, and the topic-0 part broadens out amongst the three chapters 1-3. These topics are illustrated in Table 6.6 and it can be seen that topic-0 and topic-9 by LDA talk about related themes.

Now consider "Moby Dick" in more detail. In some cases SeqLDA can be seen to refine the topics and make them more coherent. Topic-6, for instance, in SeqLDA is refined to be about the business of processing the captured whale with hoists, oil, blubber and so forth. This occurs starting at chapter 98 of the book. For LDA this topic was also sprinkled about earlier. In other cases, SeqLDA seems to smooth out the flow of otherwise unchanged topics, as seen for topic-0, 1 and 2 at the bottom of Figure 6.6(b).

6.7 Summary

In this chapter, I have proposed a novel generative model, the Sequential Latent Dirichlet Allocation (SeqLDA) model by explicitly considering the document structure in the hierarchical modelling. The sequential topical dependencies buried in the higher level of document structure are captured by the dependencies among the segments' subtopics (or ideas) which are further approximated by topic distributions. Thus, the topic evolution can be estimated by observing how topic distributions change among segments. Unlike other Markov chain based models, SeqLDA, as an integrated model, detects the rise and fall of topics within each individual document by putting the Markov assumption on the topic distributions.

I have also developed for SeqLDA an efficient collapsed Gibbs sampling algorithm based on the CMGS for the HPDP (Equation 3.6). Instead of sampling for the full customer seating arrangement, this algorithm uses the table multiplicities to sum out the exact customer partitions in the restaurants. In this way, the real valued parameter of the PDP can easily be integrated out. Having observed that the PDP is sensitive to the concentration parameters (*i.e.*, b), I introduced an adaptive rejection sampling method to optimise b. Besides the advantage over LDA in terms of improved perplexity, the ability of SeqLDA to discover more coherent sequential topic structure (*i.e.*, how topics evolves among segments within a document) has been demonstrated in the experiments. The experimental results also indicate that the document structure can aid in the statistical text analysis, and structure-aware topic modelling approaches provide a solution going beyond the "bag-of-words" assumption.

There are various ways to extend SeqLDA which I hope to explore in the future. The model could be applied to conduct document summarisation and text segmentation, where sequential structures could play an important role. The two parameters a and b in the PDP can be optimised dynamically for each segment in order to handle sizeable topic drift among segments *i.e.*, where the correlations between two successive segments are not very strong.

based models, SeqLDA, as an integrated model, detects the rise and fall of topics within each individual document by putting the Markov assumption on the topic distributions.

I have also developed for SeqLDA an efficient collapsed Gibbs sampling algorithm based on the CMGS for the HPDP (Equation 3.6). Instead of sampling for the full customer seating arrangement, this algorithm uses the table multiplicities to sum out the exact customer partitions in the restaurants. In this way, the real valued parameter of the PDP can easily be integrated out. Having observed that the PDP is sensitive to the concentration parameters (*i.e.*, b), I introduced an adaptive rejection sampling method to optimise b. Besides the advantage over LDA in terms of improved perplexity, the ability of SeqLDA to discover more coherent sequential topic structure (*i.e.*, how topics evolves among segments within a document) has been demonstrated in the experiments. The experimental results also indicate that the document structure can aid in the statistical text analysis, and structure-aware topic modelling approaches provide a solution going beyond the "bag-of-words" assumption.

There are various ways to extend SeqLDA which I hope to explore in the future. The model could be applied to conduct document summarisation and text segmentation, where sequential structures could play an important role. The two parameters a and b in the PDP can be optimised dynamically for each segment in order to handle sizeable topic drift among segments *i.e.*, where the correlations between two successive segments are not very strong.

Chapter 7

Adaptive Topic Model

In this chapter, I present another structured topic model, called an adaptive topic model (AdaTM), based on the compound Poisson-Dirichlet process (CPDP) discussed in Section 2.4. This new model integrates STM (in Chapter 5) and SeqLDA (in Chapter 6) to incorporate the full document structure, so that two kinds of subject structures (*i.e.*, the latent hierarchical structure and the sequential structure) buried in the high levels of document structures can be modelled simultaneously. It is evaluated on five sets of U.S. patents with different International Patent Classification (IPC) codes and two books. Experimental results show that with topic adaptation, AdaTM can outperform STM, SeqLDA and LDA in terms of per-word predicting likelihood, and it is able to uncover clear topic evolution structure in the books, like SeqLDA.

This chapter is organised as follows. Section 7.1 gives the motivation of the new model. Section 7.2 elaborates the model in detail, then the blocked Gibbs sampling algorithm based on BTIGS is developed in Section 7.3. The experimental results are reported in Section 7.4. Section 7.5 concludes this chapter.

7.1 Introduction

In Chapters 5 and 6, I developed two structured topic models, *i.e.*, STM and SeqLDA, that explore the hierarchical document structure and the sequential document structure respectively. The former maps the hierarchical document structure to a document topic hierarchy by using the PDP (see Figure 5.1); and the latter deals with the underlying sequential topic dependencies (see Figure 6.1) conveyed by the segment sequence (*i.e.*, the order of segments in the document layout) by extending the HPDP with a multi-level hierarchy. Both models have better predictive accuracy than the standard LDA and other segmented topic models, which suggests that document structure can be important in analysing the original text content.

However, documents (*e.g.*, books, scientific articles and patents) usually exhibit both hierarchical and sequential structures. Recall that documents are composed of segments, each of which contains a group of words. The definition of segments can vary according to different types of documents. For example, segments can be chapters in books, sections in articles, or paragraphs in essays. All segments are organised logically to form a document. The logical organisation is done through linkages between the document subject and subtopics associating with segments, and those among the subtopics. The former linkages form the hierarchical topic structure, and the latter ones form the sequential topic structure. All the linkages establish the complete document structure.

The problems of modelling these two kinds of structures separately could be:

1. Modelling the document subject and its corresponding segment subtopics in a hierarchical way has assumed segments in a document are exchangeable. This implies that there are no direct relations among subtopics. However, in writing, people usually try to link a segment to its antecedent and subsequent segments in order to make topics change smoothly from one segment to another. Therefore, the exchangeability assumption is not always appropriate, especially if documents indeed exhibit some latent sequential topic structure. This can be the reason why STM could be inappropriate for doing analysis of topic evolution .

2. In contrast, only modelling the sequential structure may misinterpret the document structure if correlations among subtopics of adjacent segments are not strong. Here I take books, especially novels as examples. In many books, one can have topic shifts from one chapter to another. This was discussed in the analysis of topic distribution profile over chapters of two novels in Section 6.6.3. The topic shifts may interrupt the sequential structure, so it is possible that stories written in different chapters do not exhibit obvious sequential relations, though they altogether make up a complete story. In this case, modelling the sequential structure with the hierarchical structure may yield a better performance.

In Figure 7.1, the graph on the top illustrates an example of a document structure that consists of both kinds of topic structures. The sets ($\{\nu_1, \nu_2\}$, $\{\nu_3, \nu_4, \nu_5\}$

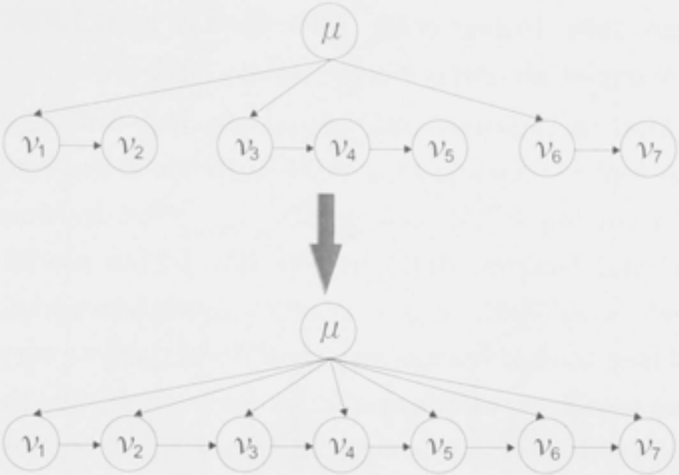


Figure 7.1: An example of a full document structure with seven segments, each of which is denoted by a circle with label ν_i. The top one shows the case when the document structure is known a priori. The button one shows the case when the document structure is not known a priori, thus we need to model the document structure with a fully directed graph. The latter is usually the case in document analysis.

and $\{\nu_6, \nu_7\}$) that contain different number of linked nodes exhibit a hierarchical structure, and nodes in each set exhibit a sequential structure. Topic shifts can be simulated by adapting a new subtopic from the document subject at breaking points of the chain, *i.e.*, at nodes ν_3 and ν_6. If document topic structures are known a priori, we can linearly combines STM and SeqLDA. However, document topic structures are not always known a priori. They need to be learnt from the original text content and the physical layout. As a consequence, we need an integrated model that models the two kinds of structure together, as shown by the fully directly linked graph in Figure 7.1. The integrated model should allow the data themselves to decide whether they exhibit both structures, or just one. Thus, each node can inherit topical features from both parent nodes. Therefore, the subtopic of one segment is now an admixture of its preceding segment subtopic and the document subject.

In this chapter, I am interested in developing a new topic model that can go beyond a strictly sequential model (*e.g.*, SeqLDA) while allowing some hierarchical influence. I employ the hybrid shown at the buttom of Figure 7.1, and associate relative strengths with the arrows. These relative strengths can be used to adaptively allow this hybrid to approximate the one in the top of Figure 7.1. Thus, one needs to depart from the earlier HMM style models, see, *e.g.*, [Blei and Moreno, 2001; Purver et al., 2006; Gruber et al., 2007; Eisenstein and Barzilay, 2008; Wang et al., 2011; Nguyen et al., 2012].

Research in Machine Learning and Natural Language Processing has attempted to model various topical dependencies. Some work considers structure within the sentence level by mixing HMMs and topics on a word by word basis: the aspect HMM [Blei and Moreno, 2001] and the HMM-LDA model [Griffiths et al., 2005] that models both short-range syntactic dependencies and longer semantic dependencies. These models operate at a finer level than we are considering at a segment (like paragraph or section) level. To make a tool like the HMM work at higher levels, one needs to make stronger assumptions, for instance assigning each sentence a single topic and then topic specific word models can be used: the hidden topic Markov model [Gruber et al., 2007] that models the transitional topic structure; a global model based on the generalised Mallows model [Chen et al., 2009], and a HMM based content model [Barzilay and Lee, 2004]. Researchers have also considered time-series of topics: various kinds of dynamic topic models, following early work of [Blei and Lafferty, 2006b], represent a collection as a sequence of sub-collections in epochs. Here, one is modelling the collections over broad epochs, not the structure of a single document that AdaTM considers.

7.2 AdaTM Generative Process

In this section, I develop a new adaptive topic model (AdaTM), a fully structured topic model, by using the CPDP discussed in Section 2.4 to simultaneously model the hierarchical and the sequential topic structures. As for STM and SeqLDA, topic distributions are used to mimic the subjects of documents and subtopics of their segments. The notations and terminologies used in the following sections are the same as those in STM, see Table 5.1. In addition, $\rho_{i,j}$, drawn from a Beta distribution (*i.e.*, a two-dimensional Dirichlet distribution), is the mixture weight associating with the link between document distribution $\boldsymbol{\mu}_i$ and segment topic distribution $\boldsymbol{\nu}_{i,j}$. It is first introduced in the CPDP, see Section 2.4.

In AdaTM, the two topic structures are captured by drawing topic distributions from the CPDPs with two base distributions as follows. The document topic distribution $\boldsymbol{\mu}_i$ and the j^{th} segment topic distribution $\boldsymbol{\nu}_{i,j}$ are the two base distributions of the CPDP for drawing the $(j+1)^{th}$ segment topic distribution $\boldsymbol{\nu}_{i,j+1}$. The topic distribution of the first segment, *i.e.*, $\boldsymbol{\nu}_{i,1}$, is drawn directly from a PDP with the base distribution $\boldsymbol{\mu}_i$. I call this generative process *topic*

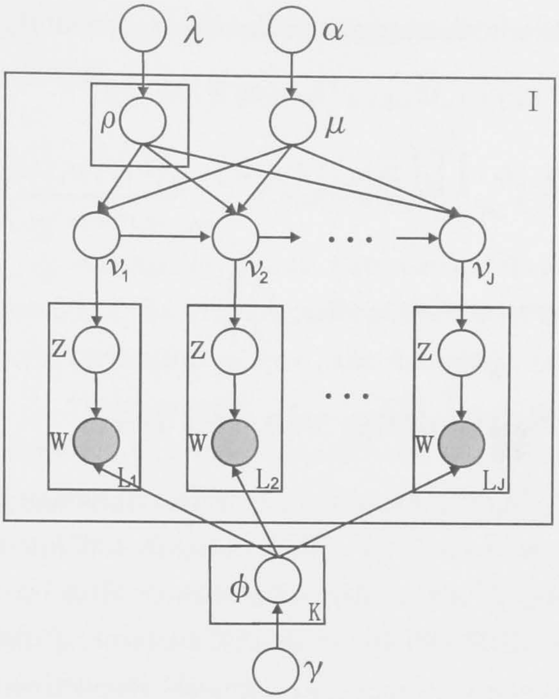


Figure 7.2: Adaptive topic model. $\boldsymbol{\mu}$ is the document topic distribution for the document subject. $\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \ldots, \boldsymbol{\nu}_J$ are the segment topic distributions for the segment subtopics. $\boldsymbol{\rho}$ is a set of the mixture weights associated with segments.

adaptation. Clearly, recursively drawing segment topic distribution with a CPDP forms a simple DAG structure over topic vectors. The graphical representation of AdaTM is shown in Figure 7.2.

The complete probability construction of AdaTM is:

$$\begin{aligned}
\boldsymbol{\phi}_k &\sim \text{Dirichlet}_W(\boldsymbol{\gamma}) && \text{for each } k\\
\boldsymbol{\mu}_i &\sim \text{Dirichlet}_K(\boldsymbol{\alpha}) && \text{for each } i\\
\rho_{i,j} &\sim \text{Beta}(\lambda_S, \lambda_T) && \text{for each } 1 \le j \le J_i\\
\boldsymbol{\nu}_{i,j} &\sim \text{PYP}\left(\rho_{i,j}\boldsymbol{\nu}_{i,j-1} + (1-\rho_{i,j})\boldsymbol{\mu}_i, a, b\right) && \text{for each } 1 \le j \le J_i\\
z_{i,j,l} &\sim \text{Discrete}_K(\boldsymbol{\nu}_{i,j}) && \text{for each } i, j, l\\
w_{i,j,l} &\sim \text{Discrete}_K\left(\boldsymbol{\phi}_{z_{i,j,l}}\right) && \text{for each } i, j, l
\end{aligned}$$

Here, for notational convenience, let $\boldsymbol{\nu}_{i,0} = \boldsymbol{\mu}_i$. Like in STM and SeqLDA, I have assumed the dimensionality of the Dirichlet distribution (*i.e.*, the number of topics) is known and fixed, and word probabilities are parameterised with a $K \times W$ matrix $\boldsymbol{\Phi}$. The complete-data likelihood can be read directly from

Figure 3.11 using distributions given in the above probability model, *i.e.*,

$$
\begin{aligned}
& p(\boldsymbol{\mu}_{1:I}, \boldsymbol{\nu}_{1:I,1:J}, \boldsymbol{z}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J} \,|\, a, b, \boldsymbol{\alpha}, \boldsymbol{\Phi}, \lambda_S, \lambda_T) \\
&= \prod_{i=1}^{I} \Bigg(p(\boldsymbol{\mu}_i \,|\, \boldsymbol{\alpha}) \prod_{j=1}^{J_i} \Bigg(p(\rho_{i,j} \,|\, \lambda_S, \lambda_T) \; \underbrace{p(\boldsymbol{\nu}_{i,j} \,|\, \boldsymbol{\mu}_i, \boldsymbol{\nu}_{i,j-1}, \rho_{i,j}, a, b)}_{\boldsymbol{\nu}_{i,j} \sim \mathrm{CPDP}(\rho_{i,j}\boldsymbol{\nu}_{i,j-1} + (1-\rho_{i,j})\boldsymbol{\mu}_i, a, b)} \\
&\qquad \prod_{l=1}^{L_{i,j}} p(z_{i,j,l} \,|\, \boldsymbol{\nu}_{i,j}) p(w_{i,j,l} \,|\, \boldsymbol{\phi}_{z_{i,j,l}}) \Bigg) \Bigg)
\end{aligned}
\tag{7.1}
$$

7.3 Gibbs Sampling via BTIGS

For the posterior inference, I elaborate a blocked Gibbs sampling algorithm based on the BTIGS (see Section 3.4) to do approximated inference. Table 7.1 lists all the statistics needed in the algorithm. Notice that for easy understanding, terminologies in the CRP will be used, *i.e.*, customers, dishes and restaurants, which correspond to words, topics and segments respectively. The basic theories of the CRP for the PDP and the CPDP are discussed in Chapters 2 and 3. It is worth reminding ourselves that tables in a child restaurant are sent as proxy customers to its parent restaurants, see Figure 3.9.

7.3.1 Model Likelihood

To adapt the blocked table indictor Gibbs sampling algorithm for AdaTM, we first compute the marginal distribution of the observations $\boldsymbol{w}_{1:I,1:J}$ (words), the topic assignments $\boldsymbol{z}_{1:I,1:J}$ and the table indicators $\boldsymbol{u}_{1:I,1:J}$. Specifically, the Dirichlet integral is used to integrate out the document topic distributions $\boldsymbol{\mu}_{1:I}$ and the topic-by-words matrix $\boldsymbol{\Phi}$, and the joint posterior distribution computed in Equation (3.18) is used to recursively marginalise out the segment topic distributions $\boldsymbol{\nu}_{1:I,1:J}$. With these variables marginalised out, we derive the following marginal distribution

$$
\begin{aligned}
& p(\boldsymbol{z}_{1:I,1:J}, \boldsymbol{w}_{1:I,1:J}, \boldsymbol{u}_{1:I,1:J} \,|\, \boldsymbol{\alpha}, \boldsymbol{\gamma}, a, b, \lambda_S, \lambda_T) \\
&= \prod_{i=1}^{I} \left(\frac{\mathrm{Beta}_K\left(\boldsymbol{\alpha} + \sum_{j=1}^{J_i} \boldsymbol{s}_{i,j}\right)}{\mathrm{Beta}_K\left(\boldsymbol{\alpha}\right)} \prod_{j=1}^{J_i} \left(\mathrm{Beta}\left(S_{i,j} + \lambda_S, T_{i,j} + \lambda_T\right) \frac{(b|a)_{T_{i,j}+S_{i,j}}}{(b)_{N_{i,j}+T_{i,j+1}}} \right.\right. \\
&\qquad \left.\left. \prod_{k=1}^{K} \binom{n_{i,j,k} + t^*_{i,j+1,k}}{t^*_{i,j,k} + s^*_{i,j,k}}^{-1} S^{n_{i,j,k}+t^*_{i,j+1,k}}_{t^*_{i,j,k}+s^*_{i,j,k}, a} \right) \right) \prod_{k=1}^{K} \frac{\mathrm{Beta}_W\left(\boldsymbol{\gamma} + \boldsymbol{M}_k\right)}{\mathrm{Beta}_W\left(\boldsymbol{\gamma}\right)} \, ,
\end{aligned}
\tag{7.2}
$$

where $\mathrm{Beta}_K\left(\boldsymbol{\alpha}\right)$ is a K dimensional beta function that normalises the Dirichlet; $(x)_N$ is given by $(x|1)_N$, and $(x|y)_N$ denotes the Pochhammer symbol (see

Equations (3.2)); $S^N_{M,a}$ is a generalised Stirling number (see Section 3.3). Note the following constraints apply:

$$t^*_{i,j,k} + s^*_{i,j,k} \leq n_{i,j,k} + t^*_{i,j+1,k}, \tag{7.3}$$

$$t^*_{i,j,k} + s^*_{i,j,k} = 0 \qquad \text{if and only if } n_{i,j,k} + t^*_{i,j+1,k} = 0 \ . \tag{7.4}$$

For convenience, $t_{i,J_i+1,k} = 0$ and $t_{i,1,k} = 0$. The reason for setting $t_{i,1,k}$ to zero is that the topic distribution of the first segment of each document is always drawn from a PDP with base distribution $\boldsymbol{\mu}_i$ (*i.e.*, the document topic distribution), as shown in Figure 7.2.

As discussed in Section 3.4, table indicators are not required to be recorded, instead, randomly sampled in Gibbs sampling iterations; and all the statistics needed are the same as those in the CMGS. The table indicators can be used to reconstruct the table multiplicities, and vice versa. See Chapter 3 for detailed discussions. Furthermore, the table indicator $u_{i,j,l}$ for word $w_{i,j,l}$ has two components in AdaTM. It is defined specifically as

$$u_{i,j,l} = (u_1, u_2) \text{ s.t. } u_1 \in [-1, 0, 1] \text{ and } u_2 \in [1, \cdots, j],$$

Table 7.1: List of statistics used in AdaTM

Statistic.	Description.
$M_{i,k,w}$	the total number of words in document i with dictionary index w and being assigned to topic k.
$M_{k,w}$	$M_{i,k,w}$ totalled over documents i, *i.e.*, $\sum_i M_{i,k,w}$
$\boldsymbol{M}_k$	vector of W values $M_{k,w}$
$n_{i,j,k}$	topic total in document i and segment j for topic k, *i.e.*, $n_{i,j,k} = \sum_{l=1}^{L_{i,j}} 1_{z_{i,j,l}=k}$. It counts customers arriving by themselves in the CRP representation.
$N_{i,j}$	topic total sum in document i and segment j, *i.e.*, $\sum_{k=1}^{K} n_{i,j,k}$
$t^*_{i,j,k}$	table count in the CPR for document i and paragraph j, for topic k that is inherited back to paragraph $j-1$ and $\boldsymbol{\mu}_{i,j-1}$.
$s^*_{i,j,k}$	table count in the CPR for document i and paragraph j, for topic k that is inherited back to the document and $\boldsymbol{\mu}_i$.
$T_{i,j}$	total table count in the CRP for document i and segment j.
$S_{i,j}$	total table count in the CRP for document i and segment j.
$\boldsymbol{t}^*_{i,j}$	table count vector, *i.e.*, $(t^*_{i,j,1}, ..., t^*_{i,j,K})$ for segment j.
$\boldsymbol{s}^*_{i,j}$	table count vector, *i.e.*, $(s^*_{i,j,1}, ..., s^*_{i,j,K})$ for segment j.

where u_2 indicates the restaurant (*i.e.*, a segment denoted by node ν_j in Figure 3.11) up to which $w_{i,j,l}$ contributes a table. Given u_2, $u_1 = -1$ denotes $w_{i,j,l}$ contributes a table count to $s_{i,u_2,k}$ and $t^*_{i,j',k}$ for $u_2 < j' \leq j$; $u_1 = 0$ denotes $w_{i,j,l}$ does not contribute a table to node u_2, but contributes a table count to $t_{i,j',k}$ for $u_2 < j' \leq j$; and $u_1 = 1$ denotes $w_{i,j,l}$ contributes a table count to each $t_{i,j',k}$ for $u_2 \leq j' \leq j$. Now, we are ready to compute the conditional probabilities for jointly sampling topics and table indicators from the model likelihood function (7.2).

7.3.2 Removing the Current Topic

Before sampling a new topic for $w_{i,j,l}$, we first need to remove its current value ($z_{i,j,l} = k'$) from the related statistics according to its table indicator $u_{i,j,l}$. However, table indicators for all words are not recorded. Therefore, the table indicators need to be randomly assigned by sampling. Given $z_{i,j,l} = k'$ and $u_2 = j'$ $(1 \leq j' \leq j)$, the probabilities of a word $w_{i,j,l}$ (*i.e.*, a customer in the restaurant) being a table head at restaurant j' (*i.e.* j'-th segment) are respectively:

$$p(u_1 = -1 \,|\, u_2 = j', z_{i,j,l} = k') = \frac{s^*_{i,j',k'}}{n_{i,j',k'} + t^*_{i,j'+1,k'}} \tag{7.5}$$

$$p(u_1 = 1 \,|\, u_2 = j', z_{i,j,l} = k') = \frac{t^*_{i,j',k'}}{n_{i,j',k'} + t^*_{i,j'+1,k'}} \tag{7.6}$$

$$p(u_1 = 0 \,|\, u_2 = j', z_{i,j,l} = k') = \frac{(n_{i,j',k'} + t^*_{i,j'+1,k'}) - (s_{i,j',k'} + t^*_{i,j',k'})}{n_{i,j',k'} + t^*_{i,j'+1,k'}} \tag{7.7}$$

The challenge here is to handle the two constraints (7.3) and (7.4) to make sure they are always satisfied after removing a topic. It is very interesting that the three probabilities have implicitly guaranteed that sampling to remove a topic according to Equations (7.5), (7.6) and (7.7) will not violate the two constraints. Specifically, the following cases at each restaurant j' (for $1 \leq j' \leq j$) are considered.

Let j' iterate from j to 1,

1. If $n_{i,j',k'} + t^*_{i,j'+1,k'} = s^*_{i,j',k'} + t^*_{i,j',k'} > 1$, removing a customer implies that we must remove a table count from either $s_{i,j',k'}$ or $t_{i,j',k'}$. It is easy to see that $p(u_1 = 0 \,|\, u_2 = j', z_{i,j,l} = k')$ is always equal to zero in this case. Therefore, if this equation holds, removing a table is guaranteed by either $p(u_1 = 1 \,|\, u_2 = j', z_{i,j,l} = k') > 0$ or $p(u_1 = -1 \,|\, u_2 = j', z_{i,j,l} = k') > 0$, or both. Thus, u_2 is set to j'. The value of u_1 depends on whether $s^*_{i,j',k'}$ or $t^*_{i,j',k'}$ is sampled. If $t^*_{i,j',k'}$ is sampled, *i.e.* $u_1 = 1$, we need to continue

the constraint check in restaurant $j'-1$ (*i.e.*, the parent restaurant of j'), because the table removed from $t^*_{i,j',k'}$ is a proxy customer in the parent restaurant.

2. If $n_{i,j',k'} + t^*_{i,j'+1,k'} > s^*_{i,j',k'} + t^*_{i,j',k'}$, it is a bit more complex than the above case when they are equal. We have to consider all the following three cases:

   (a) If $s^*_{i,j',k'} + t^*_{i,j',k'} > 1$, a table could either be removed or not. It depends on the value of $u_{i,j,l}$ sampled according to Equations (7.5), (7.6) and (7.7). If a table was sampled to be removed, u_2 will be set to j', and u_1 will be set to either -1 or 1. If u_1 is 1, which means the table will be removed from $t^*_{i,j',k'}$, then we need to recursively do the check at the parent restaurant $j'-1$.

   (b) If $s^*_{i,j',k'} + t^*_{i,j',k'} = 1$, a table must not be removed. This is because there are other customers (*i.e.*, words) sitting at that table and sharing the dish (*i.e.*, a topic) with $w_{i,j,l}$. Although the table was contributed by $w_{i,j,l}$, it cannot be removed. The recursive constraint check can be terminated.

   (c) If $t^*_{i,j',k'} = 0$, $p(u_1 = 1 \,|\, u_2 = j', z_{i,j,l} = k') = 0$. The customer does not contribute a table count to $t^*_{i,j',k'}$. We do not need to recursively check constraints at the parent restaurant $j'-1$.

It is clear that, if $u_1 = 1$, the constraint check should be done recursively towards the first segment indexed by 1 until u_1 changes to 0. Algorithm 6 shows how to sample the table indicators to remove a topic. It is a concrete example of the table indicator sampling algorithm for the CPDP embedded in a DAG structure, as introduced in Section 3.3.

7.3.3 Sampling a New Topic

Now consider a new topic k is sampled for $w_{i,j,l}$, denoted by $z_{i,j,l} = k$. In order to satisfy the constraints (7.3) and (7.4), for each node j' ($1 \le j' \le j$), we have to do the recursive constraint check as done in removing a topic. The following cases are considered: similar to removing a topic, let j' start from j to 1,

1. If $n_{i,j',k} + t^*_{i,j'+1,k} = 0$, which means $s^*_{i,j',k} + t^*_{i,j',k} = 0$, adding a customer eating the k-th dish means a new table must be created. The new table can be either contributed to $s^*_{i,j',k}$ or $t^*_{i,j',k}$, which is according to Equation (7.9). If

Algorithm 6 Sample to remove a word $w_{i,j,l}$ in AdaTM

```
initialise $u_{i,j,l}$ with $u_1 = 0$, $u_2 = j$
for $j' = j$ to 1 do
    $T = s^*_{i,j',k} + t^*_{i,j',k}$
    $N = n_{i,j',k} + t^*_{i,j'+1,k}$
    if $T = 1$ & $N > T$ then
        return $u_{i,j,l}$
    else
        sample $u'_1$ according to Equations (7.5), (7.6) and (7.7);
        if $u'_1 = 0$ then
            return $u_{i,j,l}$
        else
            if $u'_1 = -1$ then
                $u_1 = -1$, $u_2 = j'$
                return $u_{i,j,l}$
            else
                $u_1 = 1$, $u_2 = j'$
            end if
        end if
    end if
end for
if $u_1 = 1$ then
    Decrement $t^*_{i,j',k}$ where $u_2 \leq j' \leq j$
else
    if $u_1 = -1$ then
        Decrement $s^*_{i,u_2,k}$ and $t^*_{i,j',k}$ where $u_2 < j' \leq j$
    end if
end if
Decrement $n_{i,j,k}$ and update other related statistics
```

Algorithm 7 Sample a new topic for $w_{i,j,l}$ in AdaTM

```
for $k = 1$ to $K$ do
    $p(z_{i,j,l} = k) = 0$;
    Find the least integer $u'$, otherwise $u' = -1$;
    if $u' = -1$ then
        $p(z_{i,j,l} = k) \mathrel{+}= p(z_{i,j,l} = k, u_1 = 0, u_2 = j)$ with formula (7.8);
    end if
    for $j' = 1$ to $j$ do
        if $j' \le u'$ & $j' > 1$ then
            $p(z_{i,j,l} = k) \mathrel{+}= p(z_{i,j,l} = k, u_1 = 1, u_2 = j')$ with formula (7.9);
        end if
        $p(z_{i,j,l} = k) \mathrel{+}= p(z_{i,j,l} = k, u_1 = -1, u_2 = j')$ with formula (7.10);
    end for
end for
sample a topic $k'$ according to the computed probabilities $p(z_{i,j,l} = k)$, $1 \le k \le K$;
sample $u_{i,j,l}$ according the computed probabilities, conditioned on $z_{i,j,l} = k'$;
if $u_1 = -1$ then
    increase $s_{i,u_2,k'}$, and all $t_{i,j'',k'}$ for $u_2 < j'' \le j$;
else
    if $u_1 = 1$ then
        increase $t_{i,j',k'}$ for $u_2 \le j'' \le j$;
    end if
end if
$n_{i,j,k'} = n_{i,j,k'} + 1$;
```

it is sampled to contribute the table to $t^*_{i,j',k}$, a recursive constraint check is needed in the parent restaurant $j' - 1$, since this new table will be sent as a proxy customer to the parent restaurant.

2. If $n_{i,j',k} + t^*_{i,j'+1,k} > 0$, adding a customer may or may not increase the table count (either $t^*_{i,j',k}$ or $s^*_{i,j',k}$) by one. It will depend on the value of $u_{i,j,l}$ sampled according to Equations (7.8), (7.9) and (7.10). Similar to the first case, if $t^*_{i,j',k}$ is sampled, we need to do the recursive check up to the parent restaurant $j' - 1$.

As a consequence, adding a customer $w_{i,j,l}$ to the current restaurant with $z_{i,j,l} = k$ could create a new table in each restaurant j' for $1 \le j' \le j$. However, to guarantee the table is created recursively, if $n_{i,j',k} + t^*_{i,j'+1,k} = 0$ and $t^*_{i,j',k}$ is sampled to increase, we must find the least integer u' so that $n_{i,j',k} + t^*_{i,j'+1,k} = 0$ for $u' \leq j' \leq j$. All nodes between u' (exclusive) and j (inclusive) should only consider two options, $u_1 = -1$ and $u_1 = 0$, because a recursion is needed if $u_1 = 1$. Moreover, the special case is when $j' = u'$, u_1 now can be chosen to be 1. After considering all the cases discussed above, we can derive the joint conditional probabilities of a topic assignment $z_{i,j,l}$ and the corresponding table indicator $u_{i,j,l}$ as follows.

$$\begin{aligned}
&p(z_{i,j,l} = k,\, u_1 = 0,\, u_2 = j) \ \propto \qquad (7.8)\\
&\quad \frac{1}{b + N_{i,j} + T_{i,j+1}} \frac{n_{i,j,k} + t^*_{i,j+1,k} + 1 - (t^*_{i,j,k} + s^*_{i,j,k})}{n_{i,j,k} + t^*_{i,j+1,k} + 1} \frac{S^{n_{i,j,k}+t^*_{i,j+1,k}+1}_{t^*_{i,j,k}+s^*_{i,j,k},a}}{S^{n_{i,j,k}+t^*_{i,j+1,k}}_{t^*_{i,j,k}+s^*_{i,j,k},a}}\\
&\quad \frac{\gamma_{w_{i,j,l}} + M_{k,w_{i,j,l}}}{\sum_w (\gamma_w + M_{k,w})} \,.
\end{aligned}$$

$$\begin{aligned}
&p(z_{i,j,l} = k,\, u_1 = 1,\, u_2 = j') \ \propto \qquad (7.9)\\
&\quad \frac{1}{b + N_{i,j'-1} + T_{i,j'}} \prod_{j''=j'}^{j} \left(\frac{T_{i,j''} + \lambda_T}{S_{i,j''} + T_{i,j''} + \lambda_S + \lambda_T} \frac{b + a(T_{i,j''} + S_{i,j''})}{b + N_{i,j''} + T_{i,j''+1}} \right)\\
&\quad \frac{n_{i,j'-1,k} + t^*_{i,j',k} + 1 - (t^*_{i,j'-1,k} + s^*_{i,j'-1,k})}{n_{i,j'-1,k} + t^*_{i,j',k} + 1} \prod_{j''=j'}^{j} \frac{t^*_{i,j'',k} + s^*_{i,j'',k} + 1}{n_{i,j'',k} + t^*_{i,j''+1,k} + 1}\\
&\quad \frac{S^{n_{i,j'-1,k}+t^*_{i,j',k}+1}_{t^*_{i,j'-1,k}+s^*_{i,j'-1,k},a}}{S^{n_{i,j'-1,k}+t^*_{i,j',k}}_{t^*_{i,j'-1,k}+s^*_{i,j'-1,k},a}} \prod_{j''=j'}^{j} \frac{S^{n_{i,j'',k}+t^*_{i,j''+1,k}+1}_{t^*_{i,j'',k}+s^*_{i,j'',k}+1,a}}{S^{n_{i,j'',k}+t^*_{i,j''+1,k}}_{t^*_{i,j'',k}+s^*_{i,j'',k},a}} \frac{\gamma_{w_{i,j,l}} + M_{k,w_{i,j,l}}}{\sum_w (\gamma_w + M_{k,w})} \,.
\end{aligned}$$

$$\begin{aligned}
&p(z_{i,j,l} = k,\, u_1 = -1,\, u_2 = j') \ \propto \qquad (7.10)\\
&\quad \frac{\alpha_k + \sum_j s_{i,j,k}}{\sum_k \alpha_k + \sum_{j,k} s_{i,j,k}} \frac{S_{i,j'} + \lambda_S}{T_{i,j'} + \lambda_T} \prod_{j''=j'}^{j} \frac{T_{i,j''} + \lambda_T}{S_{i,j''} + T_{i,j''} + \lambda_S + \lambda_T}\\
&\quad \prod_{j''=j'}^{j} \left(\frac{b + a(T_{i,j''} + S_{i,j''})}{b + N_{i,j''} + T_{i,j''+1}} \frac{t^*_{i,j'',k} + s^*_{i,j'',k} + 1}{n_{i,j'',k} + t^*_{i,j''+1,k} + 1} \frac{S^{n_{i,j'',k}+t^*_{i,j''+1,k}+1}_{t^*_{i,j'',k}+s^*_{i,j'',k}+1,a}}{S^{n_{i,j'',k}+t^*_{i,j''+1,k}}_{t^*_{i,j'',k}+s^*_{i,j'',k},a}} \right)\\
&\quad \frac{\gamma_{w_{i,j,l}} + M_{k,w_{i,j,l}}}{\sum_w (\gamma_w + M_{k,w})} \,.
\end{aligned}$$

Algorithm 7 shows how to sample to add a new topic based on Equations (7.8), (7.9) and (7.10). The implementation is quite easy and straightforward.

7.3.4 Estimating Topic/Word Distributions

From statistics obtained after the burn-in of the Markov chain, we can estimate document topic distributions $\boldsymbol{\mu}$, segment topic distributions $\boldsymbol{\nu}$, and topic-word distributions $\boldsymbol{\phi}$. Like STM and SeqLDA, they can be approximated from the following posterior expected values via sampling:

$$\widehat{\mu}_{i,k} \;=\; \mathbb{E}_{\boldsymbol{z}_{i,1:J_i},\boldsymbol{t}^*_{i,1:J_i},\boldsymbol{s}^*_{i,1:J_i},\,|\,\boldsymbol{w}_{i,1:J_i},\boldsymbol{\alpha},a,b,\lambda_S,\lambda_T}\left[\frac{\alpha_k+\sum_{j=1}^{J_i}s^*_{i,j,k}}{\sum_{k=1}^{K}\left(\alpha_k+\sum_{j=1}^{J_i}s^*_{i,j,k}\right)}\right] \tag{7.11}$$

$$\begin{aligned}\widehat{\nu}_{i,j,k} \;=\; & \mathbb{E}_{\boldsymbol{z}_{i,1:J_i},\boldsymbol{t}^*_{i,1:J_i},\,\boldsymbol{s}^*_{i,1:J_i},\,|\,\boldsymbol{w}_{i,1:J_i},\boldsymbol{\alpha},a,b,\lambda_S,\lambda_T}\left[\frac{(n_{i,j,k}+t^*_{i,j+1,k})-a\times(t^*_{i,j,k}+s^*_{i,j,k})}{b+N_{i,j}+T_{i,j+1}}\right.\\ & \left.+\frac{a(T_{i,j}+S_{i,j})+b}{b+N_{i,j}+T_{i,j+1}}\left(\frac{\mu_{i,k}(S_{i,j}+\lambda_S)+\nu_{i,j-1,k}(T_{i,j}+\lambda_T)}{T_{i,j}+S_{i,j}+\lambda_S+\lambda_T}\right)\right]\end{aligned} \tag{7.12}$$

$$\widehat{\phi}_{k,w} \;=\; \mathbb{E}_{\boldsymbol{z}_{1:I,1:J},\boldsymbol{t}^*_{1:I,1:J},\boldsymbol{s}^*_{1:I,1:J}\,|\,\boldsymbol{w}_{1:I,1:J},\boldsymbol{\alpha},\boldsymbol{\gamma},a,b,\lambda_S,\lambda_T}\left[\frac{\gamma_w+M_{k,w}}{\sum_{w'=1}^{W}(\gamma_{w'}+M_{k,w'})}\right] \tag{7.13}$$

7.4 Experimental Results

As done in experiments in Chapters 5 and 6, I implemented AdaTM in C, and ran it on a desktop with Intel Core i5 CPU (2.8GHz×4), although the code is not multi-threaded. In the following sets of experiments, there are three objectives:

1. To explore different setting of hyper-parameters.
2. To compare AdaTM with the earlier STM, SeqLDA and the standard LDA (on either the document level or the segment level) in terms of per-word predictive likelihood.
3. To view the results in detail on a number of characteristic problems.

The first objective is to study how hyper-parameters can affect the performance of AdaTM; the second is to show the superiority of AdaTM over the other three models with respect to document modelling accuracy; The last is to demonstrate that AdaTM can be a promising tool for structured document analysis, which could be useful for other ad-hoc document analysis techniques, such as structured information retrieval, document summarisation, and topical segmentation.

Following the standard way of doing evaluation in topic modelling, we use perplexity, a standard measure of dictionary-based compressibility, in performance comparison. When reporting test perplexities, the held-out perplexity measure [Rosen-Zvi et al., 2004] is used to evaluate the generalisation capability to the unseen data. This is known to be unbiased. To compute the held-out perplexity, 20% of patents in each data set was randomly held out from training to be used for testing. For this, 1000 Gibbs cycles were done for burn-in followed by 500 cycles with a lag for 100 for parameter estimation.

7.4.1 Datasets

For objectives one and two, five patent datasets are randomly selected from U.S. patents granted in 2009 and 2010. Patents in Pat-A are selected from international patent class (IPC) "A", which is about "HUMAN NECESSITIES"; those in Pat-B are selected from class "B60" about "VEHICLES IN GENERAL"; those in Pat-H are selected from class "H" about "ELECTRICITY"; those in Pat-F are selected from class "F" about "MECHANICAL ENGINEERING; LIGHTING; HEATING; WEAPONS; BLASTING"; and those in Pat-G are selected from class "G06" about "COMPUTING; CALCULATING; COUNTING". All the patents in these five datasets are split into paragraphs that are taken as segments, and the sequence of paragraphs in each patent is reserved in order to maintain the original layout. All the stop words, the top 10 common words, the uncommon words (i.e., words in less than five patents) and numbers have been removed.

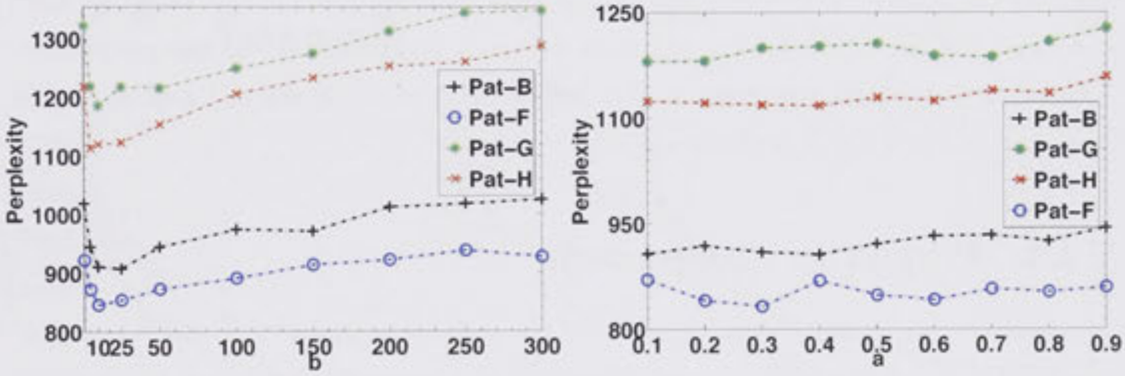
Two books used for more detailed investigation are "The Prince" by Niccolò Machiavelli and "Moby Dick" by Herman Melville. They are split into chapters and/or paragraphs which are treated as segments, and only stop-words are removed. Table 7.2 shows in detail the statistics of these datasets after preprocessing. The statistics of the two books can be find in Table 6.3. For "The Prince", there are 192 paragraphs.

7.4.2 Hyper-parameters Investigation

Experiments on the impact of the hyper-parameters on the patent data sets are as follows. First, fixing $K = 50$, the Beta parameters $\lambda_T = 1$ and $\lambda_S = 1$, optimise symmetric α, and do two variations *fix-a:* $a = 0.0$, trying $b = 1, 5, 10, 25, ..., 300$, and *fix-b:* $b = 10$, trying $a = 0.1, 0.2, ..., 0.9$. Second, *fix-*λ_T *(fix-*λ_S*)*: fix $a = 0.2$ and $\lambda_T(\lambda_S) = 1$, optimise b and α, change $\lambda_S(\lambda_T) = 0.1, 1, 10, 50, 100, 200$. Fig-

Table 7.2: Datasets

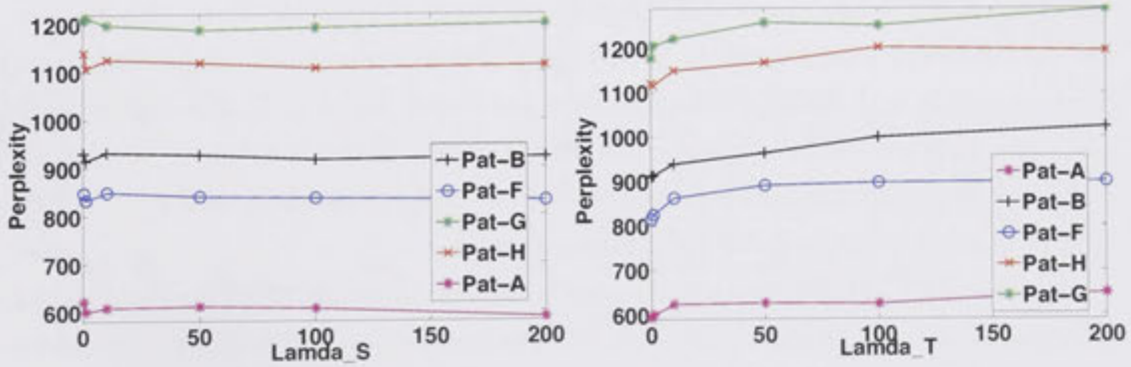
	#docs	#segs	#words	vocab
Pat-A	500	51,748	2,146,464	16,573
Pat-B	397	9,123	417,631	7,663
Pat-G	500	11,938	655,694	6,844
Pat-H	500	11,662	562,439	10,114
Pat-F	140	3,181	166,091	4,674



(a) fix $a = 0$

(b) fix $b = 10$

Figure 7.3: Analysis of parameters of Poisson-Dirichlet process. (a) shows how perplexity changes with b; (b) shows how it changes with a.



(a) fix $\lambda_T = 1$

(b) fix $\lambda_S = 1$

Figure 7.4: Analysis of the two parameters for Beta distribution. (a) shows how perplexity changes with λ_S; (b) shows how it changes with λ_T.

Table 7.3: P-values for one-tail paired t-test on the five patent datasets.

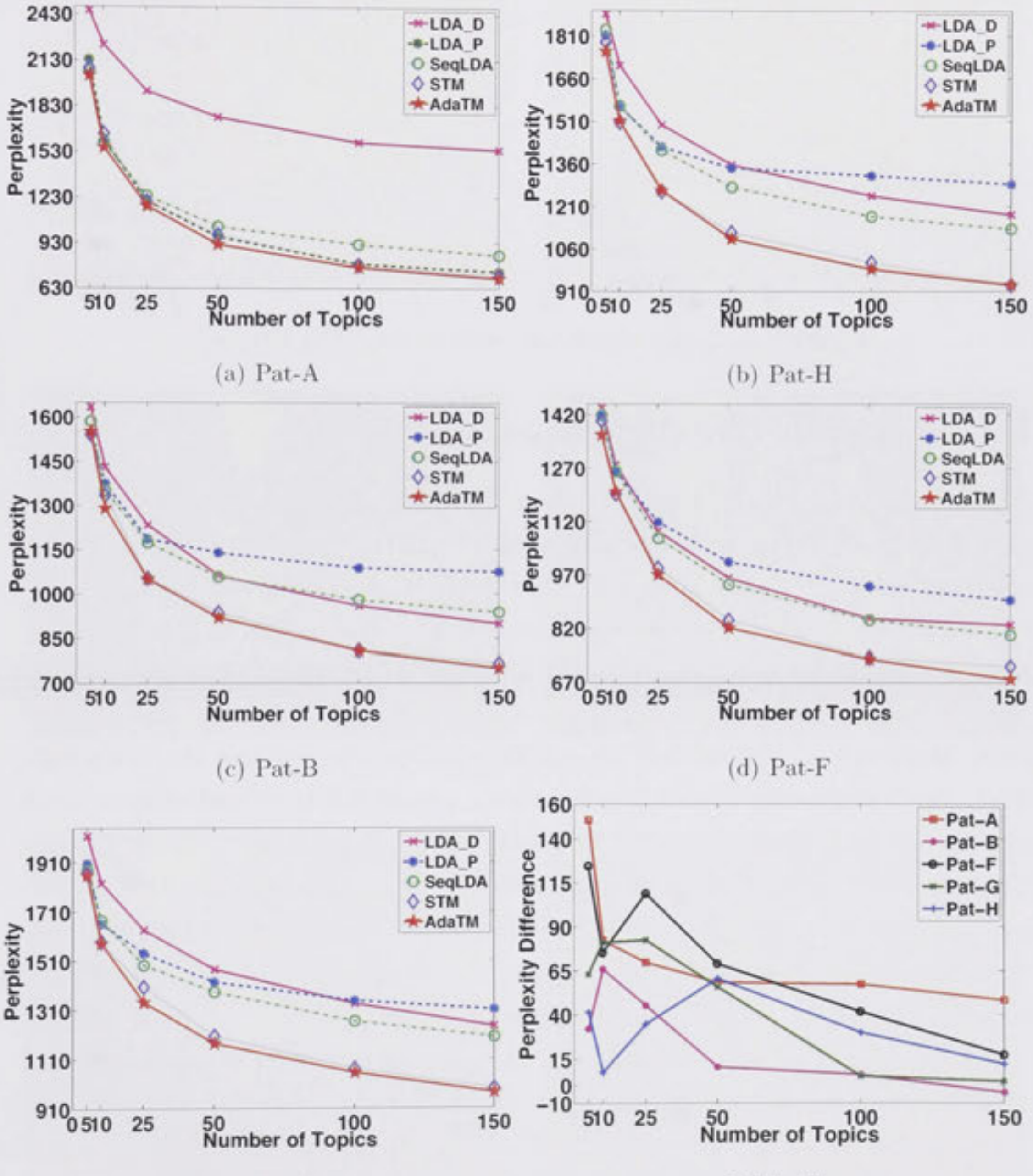
	AdaTM				
	Pat-G	Pat-A	Pat-F	Pat-H	Pat-B
LDA_D	.0001	.0001	.0002	.0001	.0001
LDA_P	.0041	.0030	.0022	.0071	.0096
SeqLDA	.0029	.0047	.0003	.0012	.0023
STM	.0220	.0066	.0210	.0629	.0853

ures 7.3 and 7.4 show the corresponding plots. Figures 7.3(b) and 7.4(a) show that varying the values of a and λ_S does not significantly change the perplexity. In contrast, Figure 7.3(a) shows different b values significantly change perplexity. Therefore, I sought to optimise b. The experiment of fixing $\lambda_S = 1$ and changing λ_T shows a small λ_T is preferred.

7.4.3 Perplexity Comparison

Perplexity comparisons were done with the default settings $a = 0.2$, $\alpha = 0.1$, $\gamma = 0.01$, $\lambda_S = 1$, $\lambda_T = 1$ and b optimised automatically using the scheme discussed in Chapter 6. Moreover, LDA has been run on both the document level (LDA_D) and the paragraph level (LDA_P). The different numbers of topics I have run are 5, 10, 25, 50, 100, and 150. Figures 7.5(a) to 7.5(e) show the results on these five patent datasets for different number of topics. Table 7.3 gives the p-values of a one-tail paired t-test for AdaTM versus the others, where lower p-value indicates AdaTM has statistically significant lower perplexity. From this we can see that AdaTM is significantly better than SeqLDA and LDA, and better than or comparable with STM. I observed that for Pat-B and Pat-H, the hierarchical structure dominates the sequential structure, given that the relative weights on edges between μ and ν_j are usually larger than those between ν_j and ν_{j-1}, which results in that AdaTM and STM are comparable.

In addition, I ran another set of experiments by randomly shuffling the order of paragraphs in each patent several times before running AdaTM. Then, I calculate the difference between perplexities with and without random shuffle. Figure 7.5(f) shows the plot of differences in each data sets. The positive difference means randomly shuffling the order of paragraphs indeed increases the perplexity. It can further prove that there does exist sequential topic structure in patents, which confirms the finding in Chapter 6.



(a) Pat-A

(b) Pat-H

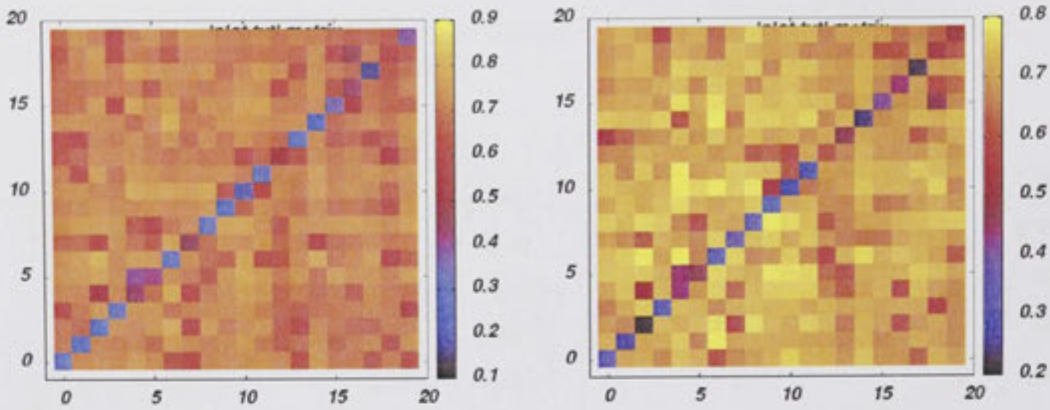
(c) Pat-B

(d) Pat-F

(e) Pat-G

(f) Shuffle

Figure 7.5: Perplexity comparisons.



(a) LDA versus AdaTM for chapters

(b) LDA versus AdaTM for paragraphs

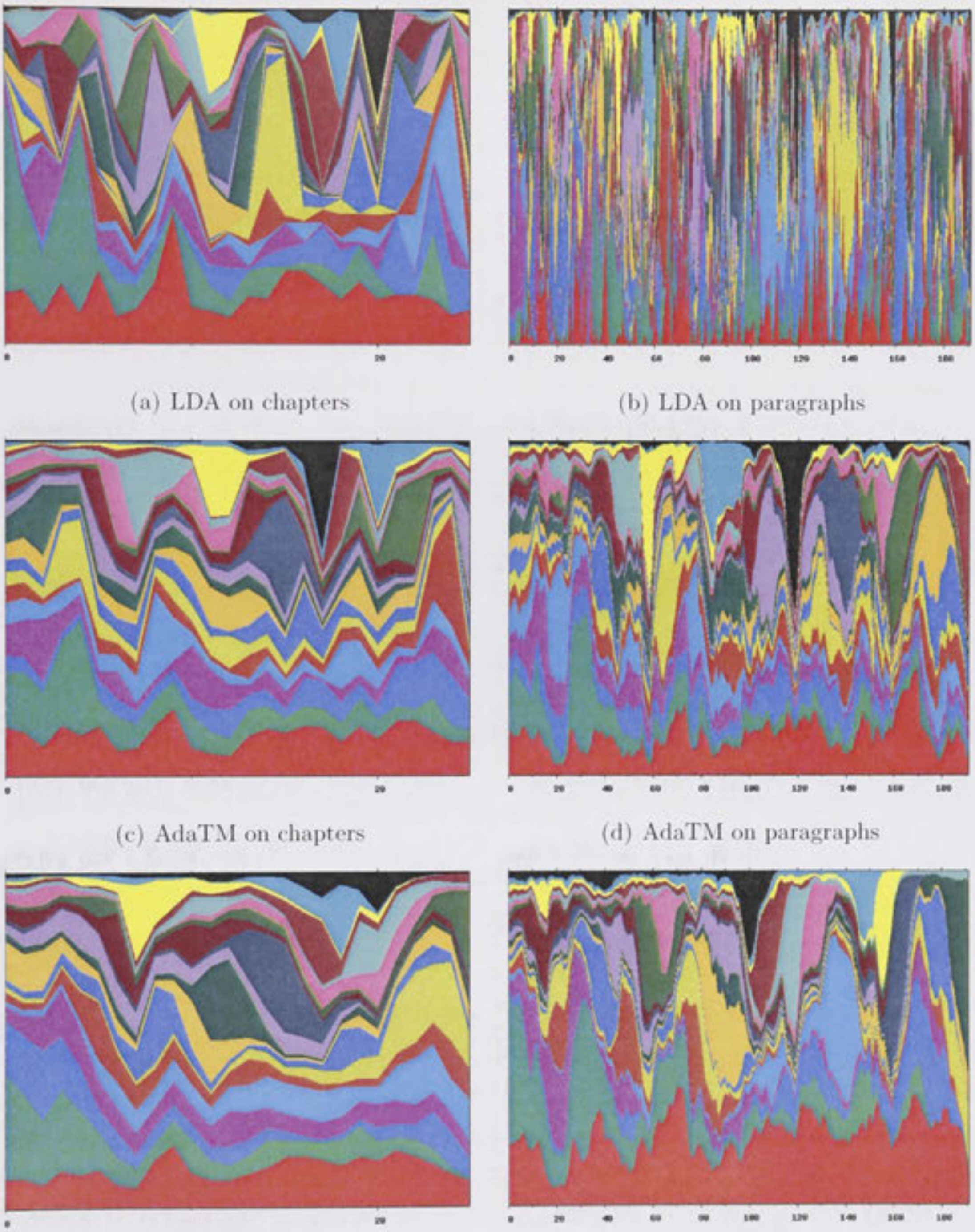
Figure 7.6: Topic alignment analysis on "The Prince".

7.4.4 Topic Evolution Comparisons

All experiments reported in this section are run with 20 topics, the upper limit for easy visualisation, and without optimising any parameters. The Dirichlet Priors are fixed as $\alpha_k = 0.1$ and $\gamma_w = 0.01$. For AdaTM, SeqLDA, and STM, $a = 0.0$ and $b = 100$ for "The Prince" and $b = 200$ for "Moby Dick". These settings have proven robust in experiments. To align the topics so visualisations match, the sequential models are initialised using an LDA model built at the chapter level. Moreover, all the models are run at both the chapter and the paragraph level. With the common initialisation, both paragraph level and chapter level models can be aligned. Figure 7.6 shows the alignment of topics between the initialising model (LDA on chapters) and AdaTM run on chapters/paragraphs. Each point in the matrix gives the Hellinger distance between the corresponding topics, color coded. The plots for the other models, chapters or paragraphs, are similar so plots like Figure 7.7 can be meaningfully compared.

To visualise topic evolution, I use a plot with one colour per topic displayed over the sequence, as done in Chapter 6. Figures 7.7(a) and 7.7(b) show these for LDA run on chapters/paragraphs of "The Prince". The proportion of 20 topics is the Y-axis, spread across the unit interval. The chapters/paragraphs run along the X-axis, so the topic evolution is clearly displayed. One can see there is no clear sequential structure in these derived by LDA, especially in paragraphs, and similar plots result from "Moby Dick" for LDA.

Figure 7.7 then shows the corresponding evolution plots for AdaTM and SeqLDA on chapters and paragraphs. The contrast of these with LDA is stark. The



(a) LDA on chapters

(b) LDA on paragraphs

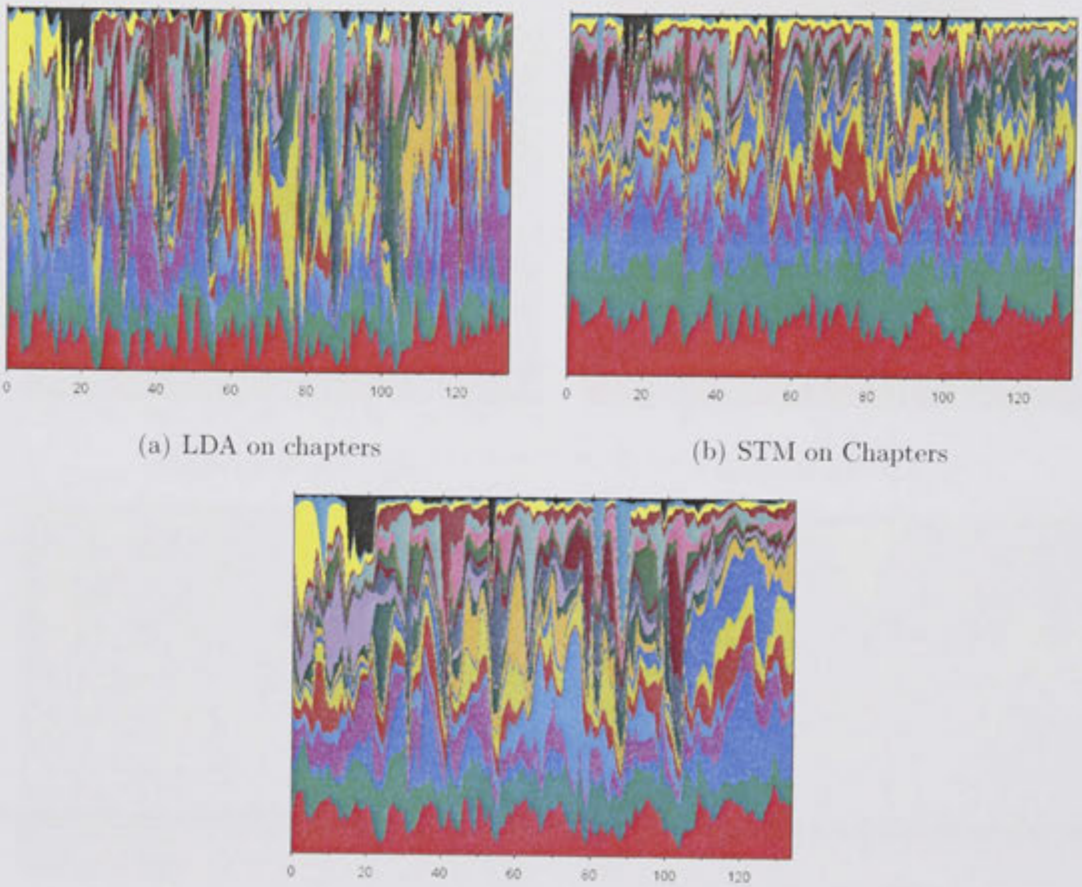
(c) AdaTM on chapters

(d) AdaTM on paragraphs

(e) SeqLDA on chapters

(f) SeqLDA on paragraphs

Figure 7.7: Topic Evolution on "The Prince".

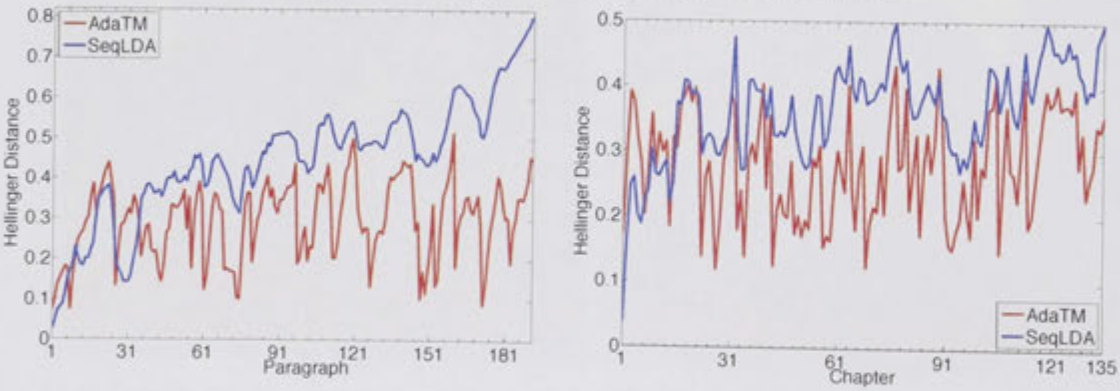


(a) LDA on chapters

(b) STM on Chapters

(c) AdaTM on Chapters

Figure 7.8: Topic Evolution on "Moby Dick".



(a) "The Prince"

(b) "Moby Dick"

Figure 7.9: Topic evolution analysis based on Hellinger Distance

large improvement in perplexity for AdaTM (see Section 7.4.3) along with no change in lexical coherence means that the sequential information is actually beneficial statistically. Note that SeqLDA, while exhibiting slightly stronger sequential structure than AdaTM in these figures has significantly worse test perplexity, so its sequential affect is too strong and harming results. Also, note that some topics have different time sequence profiles between AdaTM and SeqLDA. Indeed, inspection of the top words for each show these topics differ somewhat. So while LDA to AdaTM/SeqLDA topic correspondences are quite good due to the use of LDA initialisation, the correspondences between AdaTM and SeqLDA have degraded. We see that AdaTM has nearly as good sequential characteristics as SeqLDA. Furthermore, segment topic distribution $\nu_{i,j}$ of SeqLDA are gradually deviating from the document topic distribution μ_i, which is not the case for AdaTM.

Results for "Moby Dick" on chapters are comparable. Figure 7.8 shows similar topic evolution plots for LDA, STM and AdaTM. In contrast, the AdaTM topic evolutions are much clearer for the less frequent topics, as shown in Figure 7.8(c). Various parts of this are readily interpreted from the storyline. Here I briefly discuss topics by their colour: *black:* Captain Peleg and the business of signing on; *yellow:* inns, housing, bed; *mauve:* Queequeg; *azure:* (around chapters 60-80) details of whales *aqua:* (peaks at 8, 82, 88) pulpit, schools and mythology of whaling. We see that AdaTM can be used to understand the topics with regards to the sequential structure of a book. In contrast, the sequential nature for LDA and STM is lost in the noise.

7.4.5 Further comparison between AdaTM and SeqLDA

In previous section, I have shown the topic profile analysis for LDA, STM SeqLDA and AdaTM on the two books. In order to further compare AdaTM with SeqLDA, here, I do analysis on the two models by using the Hellinger Distance (HD). Specifically, the methodology used is to compute, for each segment j, the Hellinger distance between document topic distribution $\boldsymbol{\mu}$ and $\boldsymbol{\nu}_j$, denoted by $HD(\boldsymbol{\mu}, \boldsymbol{\nu}_j)$. Figure 7.9 shows the plots of $HD(\boldsymbol{\mu}, \boldsymbol{\nu}_j)$ on the paragraph level of "The Prince" and the chapter level of "Moby Dick".

It is interesting that, for SeqLDA, $HD(\boldsymbol{\mu}, \boldsymbol{\nu}_j)$ increases as the paragraph index becomes large. This phenomena may be due to the Markov chain used by SeqLDA, see Figure 6.2. In such a chain structure, $\boldsymbol{\nu}_j$ is likely to be concentrated on less support as j grows, which results in that $\boldsymbol{\nu}_j$ becomes less dependent on

$\boldsymbol{\mu}$. However, allowing some hierarchical influence in a strict sequential structure can reduce the sequential affect of Markov chain, and it can balance the support that $\boldsymbol{\nu}_j$ is concentrated on. Therefore, if the distance between $\boldsymbol{\nu}_j$ and $\boldsymbol{\mu}$ becomes large, AdaTM can pull $\boldsymbol{\nu}_j$ back, as shown in red in Figure 7.9.

7.5 Summary

In this chapter, I have proposed an adaptive topic model (AdaTM) that models the document structure by embedding the CPDP in a simple DAG structure. This DAG structure is motivated by both the hierarchical and the sequential subject structures embedded in the document layout, *i.e.*, a segment sequence. It can be taken as a generalisation of STM introduced in Chapter 5 and SeqLDA introduced in Chapter 6. Specifically, if the mixture weight ρ is set to 1, AdaTM reduces to SeqLDA; if ρ is set to 0, it reduces to STM. In order to do posterior inference for AdaTM, I have developed a blocked table Indicator Gibbs sampling algorithm based on BTIGS introduced in Chapter 3.

The experimental results on five sets of patents show that the average predictive accuracy of AdaTM on unseen words is significantly better than SeqLDA and LDA, and somewhat better than STM; the topic evolution analysis shows that with AdaTM, one could extract meaningful topics from a book like Herman Melville's "Moby Dick" and concurrently gain their sequential profile. In the future, I would like to study how AdaTM can be used in ad-hoc document analysis. For example, It can be very interesting to apply AdaTM to topic segmentation, summarisation, and semantic title evaluation. Currently, the code runs fairly slow due to the procedure of sampling new topics discussed in Section 7.3.3. The development of a more effective and efficient sampling algorithm is one possible future research direction, such as particle filtering [Canini et al., 2009].

Chapter 8

Conclusions and Future Work

Topic models, as promising unsupervised learning approaches, have gained significant momentum recently in machine learning, data mining and natural language processing communities. They have gained wide applications in, for example, information retrieval, sentiment analysis, and text analysis. Related techniques such as NMF are also widely used in images analysis for codebook/dictionary optimisation. In particular, the standard LDA has been extended by relaxing its underlying assumptions to incorporate beyond the "*bag-of-words*" information, such as supervised information (*e.g.*, class labels) or meta-data (*e.g.*, authors or citations).

Despite various topic models that have been proposed in the literature, the field of topic modelling still needs to be further developed. One promising area in topic modelling that has been introduced in this thesis is to directly consider the document structure ranging from semantically high-level segments (*e.g.*, chapters or sections) to low-level segments (*e.g.*, sentences or words). The layout of these segments in a document is usually represented jointly with the document subject structure. Exploring the document structure can be very useful in exploratory and predictive text analytics.

This thesis presented a family of structured topic models by taking advantage of non-parametric Bayesian methods, *i.e.*, the two-parameter Poisson-Dirichlet process (PDP). These models take into consideration document structure directly by looking at the original layout of each document as a guide to structure. Three Bayesian topic models were introduced, each capturing different types of document structures: the hierarchical document structure, the sequential document structure, and a mixture of the two. The experimental results from applying the three models to several real-world document collections have demonstrated that it is beneficial to jointly model the document structure with the latent topic variables.

In chapter 3, I introduced two new Gibbs sampling methods for doing posterior inference for the PDP in finite discrete space. One is a two-stage Gibbs sampling algorithm, called a Collapsed Multiplicity Gibbs Sampler (CMGS), which is based on the table *multiplicity* representation for the PDP. Different from Sampling for Seating Arrangement (SSA) sampler most commonly used with the hierarchical DP and PDP modelling, CMGS does not need to dynamically record the customer count at each table. The other is a Blocked Table Indicator Gibbs Sampler (BTIGS). In BTIGS, a new auxiliary latent variable, called *table indicator*, is introduced to record the table contribution of customers. Unlike recording the customer-table assignment, table indicators can be randomly assigned in Gibbs cycles. Note from the *table indicator* assignments, we can reconstruct the table *multiplicity* representation, and *vice versa*. The results of experiments run in a simply controlled environment of multinomial sampling have shown that both CMGS and BTIGS converge faster than SSA.

Chapter 5 presented a Segmented Topic Model (STM) that directly models the document structure with a four-level hierarchy. It maps the layout of segments in a document to a hierarchical subject structure. I developed for STM an effective collapsed Gibbs sampling algorithm based on CMGS. Using several real-world document collections, I compared it with the standard LDA and other segmented topic models, demonstrating that STM performs better than other models in terms of per-word predictive perplexity. For example, STM gains 28% improvement over LDA running on document level and 18% on paragraph level, when 100 topics are used for the patent dataset. The concentration parameter b is optimised for the case when $a = 0$. The primary benefit of STM is that it allows us to model document structure by simultaneously modelling document and segment topic distributions in the same latent topic space.

In Chapter 6, I considered another document structure, the sequential document structure, by introducing a novel Sequential Latent Dirichlet Allocation (SeqLDA) model. This model relaxes the exchangeability assumption on the segments, which is made by STM. SeqLDA uses a simple first-order Markov chain to simulate the segment sequence in a document, the first node in the chain corresponds to the document subject, subsequent nodes correspond to segment subtopics. I adapted CMGS in a multi-level hierarchy context to do posterior inference for SeqLDA. In addition to the better predictive accuracy on unseen words, the ability of SeqLDA to explore the topic evolution in individual documents has been demonstrated by topic evolution analysis on several story books. Furthermore, I modified an adaptive rejection sampling method to optimise b for $a > 0$. It has been shown that this optimisation algorithm works as well as manual optimisation.

Chapter 7 further considered the full document structure, which is a mixture of the two modelled respectively by STM and SeqLDA. I introduced an Adaptive Topic Model (AdaTM) for modelling the full document structure by embedding the compound Poisson-Dirichlet process in a simple DAG structure. The topic distribution of each segment is now an admixture of the document topic distribution and its preceding segment topic distribution. Each document can exhibit both the hierarchical and sequential structures. The experimental results showed that the performance of AdaTM is better than the earlier models: compared with STM, AdaTM can uncover clear sequential topic structures in documents without harming the perplexity; compared with SeqLDA, AdaTM can gain much lower perplexity.

In addition, understanding and applying CMGS and BTIGS to complex models are quite challenging. Careful attention should be paid to the implementation, especially, to handle the constraints (*e.g.*, Constraint 3.5) on table and customer counts in a recursive way. Therefore, another important contribution of this thesis is the implementation of CMGS and BTIGS in the context of a hierarchy, a Markov chain and a DAG structure to do posterior inference for STM, SeqLDA and AdaTM respectively.

8.1 Future Work

Possible future work is how to extend the three structured topic models presented in this thesis to consider more complex document structures. For example, a scientific article consists of sections, each of which contains paragraphs, and each paragraph is composed of sentences. This gives us an article-section-paragraph structure. One promising research is to extend STM to a multi-level hierarchy, since the PDPs can be easily extended to full trees, and the proposed Gibbs sampling algorithms still apply. In addition, it would be interesting to learn document structure automatically without taking the segment layout as a guide to structure, which is closely related to structured learning [Lee et al., 2007; Yehezkel and Lerner, 2009]. To find a good Bayesian network structure that matches the document subject structure, we could do heuristic search or MCMC sampling over the space of network structures.

Text analysis is one of the important application areas of those models, *e.g.*, document summarisation and segmentation. The former aims at finding a short set of words or paragraphs that can adequately represent the main subject of a text document or a collection of documents. The latter is the task of dividing a given text data into semantically coherent parts. Topic models have been applied to both summarisation [Arora and Ravindran, 2008a,b] and segmentation [Blei and Moreno, 2001; Purver et al., 2006; Misra et al., 2009; Nguyen et al., 2012]. It would be worth exploring how document structure can assist in both kinds of analysis by taking advantage of the three models presented in this thesis.

The inference methods I proposed in this thesis are Gibbs sampling based on the Chinese restaurant process (CRP) presentation for the PDP, since the CRP provides an elegant analogy of incremental sampling for the posterior of the PDP. They are good enough to test the three proposed topic models. However, it is still worth studying other algorithms for DP/PDP mixture models, such as Gibbs sampling for the stick-breaking construction [Ishwaran and James, 2001], and variational inference [Blei and Jordan, 2005; Teh et al., 2008], and indeed variants of the existing algorithms could also prove superior. In particular, to analyse unseen documents for the purpose of, for example, topic segmentation, instead of using Gibbs sampling, one could also consider leveraging the forward-backward algorithm [Yu and Kobayashi, 2006] to find the most likely state for each segment, especially for SeqLDA that has a simple Markov chain. Thus, developing forward-backward algorithms for the proposed models can be an interesting research topic.

Moreover, to further investigate capabilities of three topic models presented in Chapters 5 to 7, it would be important to compare them, especially AdaTM in Chapter 7, with other dynamic models, such as Dynamic Topic Models (DTM) [Blei and Lafferty, 2006b], dynamic HDPs [Ren et al., 2008], graphical Pitman-Yor process [Wood and Teh, 2009] and Evolutionary HDPs [Zhang et al., 2010] with more extensive experiments.

Bibliography

A. Ahmed and E. Xing. Dynamic non-parametric mixture models and the recurrent Chinese restaurant process: with applications to evolutionary clustering. In *Proceedings of The Eighth SIAM International Conference on Data Mining (SDM2008)*, 2008.

A. Ahmed and E. Xing. Timeline: A dynamic hierarchical Dirichlet process model for recovering birth/death and evolution of topics in text stream. In *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, pages 20–29, 2010.

D. Aldous. Exchangeability and related topics. In *Ecole d'Ete de Probabilities de Saint-Flour XIII 1983*, pages 1–198. Springer, 1985.

L. AlSumait, D. Barbará, and C. Domeniconi. On-line LDA: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 3–12, 2008.

C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.

R. Arora and B. Ravindran. Latent Dirichlet allocation and singular value decomposition based multi-document summarization. In *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 713–718, 2008a.

R. Arora and B. Ravindran. Latent Dirichlet allocation based multi-document summarization. In *AND '08: Proceedings of the second workshop on Analytics for noisy unstructured text data*, pages 91–97, 2008b.

L. Azzopardi, M. Girolami, and C. van Rijsbergen. Topic based language models for ad hoc information retrieval. In *Proceedings of 2004 IEEE International Joint Conference on Neural Networks*, pages 3281–3286, 2004.

R. Barzilay and L. Lee. Catching the drift: Probabilistic content models, with applications to generation and summarization. In *HLT-NAACL 2004: Main Proceedings*, pages 113–120. Association for Computational Linguistics, 2004.

J. M. Bernardo and A. F. M. Smith. *Bayesian Theory*. Wiley, 1994.

I. Bhattacharya and L. Getoor. A latent Dirichlet model for unsupervised entity resolution. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 47–58, 2006.

I. Bíró, J. Szabó, and A. A. Benczúr. Latent Dirichlet allocation in web spam filtering. In *Proceedings of the 4th international workshop on Adversarial information retrieval on the web*, pages 29–32, 2008.

D. Blackwell and J. B. Macqueen. Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1:353–355, 1973.

D. Blei. Introduction to probabilistic topic models. In *Communications of the ACM*. 2011.

D. Blei and M. Jordan. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1:121–144, 2005.

D. Blei and J. Lafferty. Correlated topic models. In *Advances in Neural Information Processing Systems 18*, pages 147–154. 2006a.

D. Blei and J. Lafferty. Dynamic topic models. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 113–120, 2006b.

D. Blei and J. Lafferty. Topic models. In *Text Mining: Classification, Clustering, and Application*. Taylor & Francis, 2009.

D. Blei and J. McAuliffe. Supervised topic models. In *Advances in Neural Information Processing Systems 20*, pages 121–128. 2007.

D. Blei and P. Moreno. Topic segmentation with an aspect hidden Markov model. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 343–348, 2001.

D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

D. Blei, T. Griffiths, and M. Jordan. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *J. ACM*, 57:1–30, 2010.

J. Boyd-Graber, D. Blei, and X. Zhu. A topic model for word sense disambiguation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1024–1033, 2007.

S. Brody and N. Elhadad. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 804–812, 2010.

W. Buntine. Operations for learning with graphical models. *Journal of Artificial Intelligence Research 2*, 96:159–225, 1994.

W. Buntine. Estimating likelihoods for topic models. In *The first Asian Conference on Machine Learning*, pages 51–64, 2009.

W. Buntine and M. Hutter. A Bayesian review of the Poisson-Dirichlet process. Technical Report arXiv:1007.0296v2, *ArXiv*, Cornell, July 2010.

W. Buntine and A. Jakulin. Discrete components analysis. In *Subspace, Latent Structure and Feature Selection Techniques*. Springer-Verlag, 2006.

W. Buntine, J. Lofstrom, J. Perkio, S. Perttu, V. Poroshin, T. Silander, H. Tirri, A. Tuominen, and V. Tuulos. A scalable topic-based open source search engine. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 228–234, 2004.

W. Buntine, L. Du, and P. Nurmi. Bayesian networks on Dirichlet distributed vectors. In *Proceedings of the Fifth European Workshop on Probabilistic Graphical Models (PGM-2010)*, pages 33–40, 2010.

W. L. Buntine. Variational Extensions to EM and Multinomial PCA. In *ECML '02: Proceedings of the 13th European Conference on Machine Learning*, pages 23–34, 2002.

K. R. Canini, L. Shi, and T. L. Griffiths. Online inference of topics with latent Dirichlet allocation. In *In Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pages 65–72, 2009.

J. Canny. GaP: a factor model for discrete data. In *Proceeding of the 27th Annual ACM SIGIR Conference on Research and development in information retrieval*, pages 122–129, 2004.

L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. In *Proceedings of IEEE Intern. Conf. in Computer Vision (ICCV).*, 2007.

G. Casella and C. P. Robert. Rao-Blackwellisation of sampling schemes. *Biometrika*, 83:81–94, 1996.

C. Chemudugunta, P. Smyth, and M. Steyvers. Modeling general and specific aspects of documents with a probabilistic topic model. In *Advances in Neural Information Processing Systems 19*, pages 241–248. 2007.

C. Chen, L. Du, and W. Buntine. Sampling for the Poisson-Dirichlet process. In *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Database*, pages 296–311, 2011.

H. Chen, S. Branavan, R. Barzilay, and D. Karger. Global models of document structure using latent permutations. In *Proceedings of Human Language Technologies: The 2009 Annual Conf. of the North American Chapter of the Association for Computational Linguistics*, pages 371–379, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

D. Cohn and T. Hofmann. The missing link - a probabilistic model of document content and hypertext connectivity. In *Advances in Neural Information Processing Systems 20*. 2001.

M. Cowles and B. Carlin. Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91: 883–904, 1996.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

L. Du, W. Buntine, and H. Jin. Sequential latent Dirichlet allocation: Discover underlying topic structures within a document. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, pages 148–157, 2010a.

L. Du, W. Buntine, and H. Jin. A segmented topic model based on the two-parameter Poisson-Dirichlet process. *Machine Learning*, 81:5–19, 2010b.

L. Du, W. Buntine, and H. Jin. Modelling sequential text with an adaptive topic model. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 535–545, July 2012a.

L. Du, W. Buntine, H. Jin, and C. Chen. Sequential latent Dirichlet allocation. *Knowledge and Information Systems*, 31(3):475–503, 2012b.

J. A. Duan, M. Guindani, and A. E. Gelfand. Generalized spatial Dirichlet process models. *Biometrika*, 94:809–825, 2007.

J. Eisenstein and R. Barzilay. Bayesian unsupervised topic segmentation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 334–343, 2008.

T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1:209–230, 1973.

P. Flaherty, G. Giaever, J. Kumm, M. I. Jordan, and A. P. Arkin. A latent variable model for chemogenomic profiling. *Bioinformatics*, 21(15):3286–3293, 2005.

B. A. Frigyik, A. Kapila, and M. R. Gupta. Introduction to the Dirichlet distribution and related processes. Technical report, Department of Electrical Engineering, University of Washington, 10 2010.

S. Geman and D. Geman. *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, pages 452–472. 1990.

W. R. Gilks and P. Wild. Adaptive rejection sampling for Gibbs sampling. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 41:337–348, 1992.

M. Girolami and A. Kabán. On an equivalence between PLSI and LDA. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 433–434, 2003.

S. Goldwater, T. Griffiths, and M. Johnson. Interpolating between types and tokens by estimating power-law generators. In *Advances in Neural Information Processing Systems 18*, pages 459–466. 2006.

S. Goldwater, T. L. Griffiths, and M. Johnson. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54, 2009.

T. Griffiths, M. Steyvers, D. Blei, and J. Tenenbaum. Integrating topics and syntax. In *Advances in Neural Information Processing Systems 17*, pages 537–544. 2005.

T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proc Natl Acad Sci USA*, 101 Suppl 1:5228–5235, 2004.

A. Gruber, Y. Weiss, and M. Rosen-Zvi. Hidden topic Markov models. *Journal of Machine Learning Research - Proceedings Track*, 2:163–170, 2007.

Q. He, B. Chen, J. Pei, B. Qiu, P. Mitra, and L. Giles. Detecting topic evolution in scientific literature: how can citations help? In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 957–966, 2009.

X. He and R. S. Zemel. Latent topic random fields: Learning using a taxonomy of labels. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008.*, pages 1–8, 2008.

G. Heinrich. Parameter estimation for text analysis. Technical report, University of Leipzig, 2008. URL `http://www.arbylon.net/publications/text-est.pdf`.

N. L. Hjort, C. Holmes, P. Müller, and S. G. Walker. *Bayesian Nonparamerics: Principles and Practice*. Cambridge University Press, Cambridge, MA, USA, 2010.

T. Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57, 1999.

T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Mach. Learn.*, 42(1-2):177–196, 2001.

L. C. Hsu and P. J.-S. Shiue. A unified approach to generalized Stirling numbers. *Adv. Appl. Math.*, 20:366–384, April 1998.

H. Ishwaran and L. James. Gibbs sampling methods for stick breaking priors. *Journal of the American Statistical Association*, 96:161–173, 2001.

M. Johnson, T. Griffiths, and S. Goldwater. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In *Advances in Neural Information Processing Systems 19*, pages 641–648. 2007.

N. L. Johnson and S. Kotz. *Urn models and their application*. Wiley, 1977.

M. I. Jordan. Dirichlet processes, Chinese Restaurant Processes and All That. In *Tutorial presentation at the NIPS conference*. 2005.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37:183–233, 1999.

V. Kandylas, S. Upham, and L. Ungar. Finding cohesive clusters for analyzing knowledge communities. *Knowledge and Information Systems*, 17:335–354, 2008.

D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using l_1-regularization. In *Advances in Neural Information Processing Systems 19*, pages 817–824. 2007.

D. Lewis, Y. Yand, T. Rose, and F. Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

F.-F. Li and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 524–531, 2005.

W. Li and A. McCallum. Pachinko allocation: DAG-structured mixture models of topic correlations. In *Proceedings of the 23rd international conference on Machine learning*, pages 577–584, 2006.

W. Li, D. Blei, and A. McCallum. Nonparametric Bayes Pachinko allocation. In *Proceedings of the Proceedings of the Twenty-Third Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-07)*, pages 243–250, 2007.

C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 375–384, 2009. ISBN 978-1-60558-512-3.

D. Lin, E. Grimson, and J. Fisher. Construction of dependent Dirichlet processes based on Poisson processes. In *Advances in Neural Information Processing Systems 23*, pages 1396–1404. 2010.

S. MacEachern. Dependent nonparametric processes. In *ASA Proceedings of the Section on Bayesian Statistical Science*, pages 50–55, 1999.

D. J. C. Mackay and L. Peto. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1:1–19, 1995.

A. McCallum. Multi-label text classification with a mixture model trained by EM. In *AAAI 99 Workshop on Text Learning*, 1999.

A. Mccallum, A. Corrada-emmanuel, and X. Wang. The author-recipient-topic model for topic and role discovery in social networks: Experiments with Enron and academic email. Technical report, University of Massachusetts Amherst, 2004.

A. McCallum, X. Wang, and A. Corrada-Emmanuel. Topic and role discovery in social networks with experiments on Enron and academic email. *J. Artif. Int. Res.*, 30:249–272, 2007.

Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180, 2007.

Q. Mei, D. Cai, D. Zhang, and C. Zhai. Topic modeling with network regularization. In *Proceeding of the 17th international conference on World Wide Web*, pages 101–110, 2008.

D. Mimno and A. McCallum. Topic models conditioned on arbitrary features with Dirichlet-multinomial regression. In *Proceedings of the Twenty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence*, pages 411–418, 2008.

D. Mimno, W. Li, and A. McCallum. Mixtures of hierarchical topics with Pachinko allocation. In *Proceedings of the 24th international conference on Machine learning*, pages 633–640, 2007.

T. Minka. Estimating a Dirichlet distribution. Technical report, MIT, 2000.

T. Minka and J. Lafferty. Expectation-propogation for the generative aspect model. In *Proceedings of the Proceedings of the Eighteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)*, pages 352–359, 2002.

H. Misra, F. Yvon, J. M. Jose, and O. Cappe. Text segmentation via topic modeling: an analytical study. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1553–1556, 2009.

D. Mochihashi and Y. Matsumoto. Context as filtering. In *Advances in Neural Information Processing Systems 18*, pages 907–914. 2006.

D. Mochihashi and E. Sumita. The infinite Markov model. In *Advances in Neural Information Processing Systems 20*, pages 1017–1024. 2008.

R. Nallapati, A. Ahmed, E. Xing, and W. Cohen. Joint latent topic models for text and citations. In *Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 542–550, 2008.

R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265, 2000.

V.-A. Nguyen, J. Boyd-Graber, and P. Resnik. Sits: A hierarchical nonparametric model using speaker identity for topic segmentation in multiparty conversations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 78–87, 2012.

J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *Int. J. Comput. Vision*, 79:299–318, 2008.

M. Pennacchiotti and S. Gurumurthy. Investigating topic models for social media user recommendation. In *Proceedings of the 20th international conference companion on World wide web*, pages 101–102, 2011.

X.-H. Phan, L.-M. Nguyen, and S. Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceeding of the 17th international conference on World Wide Web*, pages 91–100, 2008.

J. Pitman. Combinatorial stochastic processes. Technical Report 621, Department of Statistics, University of California, 2002.

J. Pitman and M. Yor. The two-parameter Poisson-Diriclet distribution derived from a stable subordinator. *Annals Probability*, 25:855–900, 1997.

I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed Gibbs sampling for latent Dirichlet allocation. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 569–577, 2008.

J. Pritchard, M. Stephens, and P. Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155:945–959, 2000.

I. Pruteanu-Malinici, L. Ren, J. Paisley, E. Wang, and L. Carin. Hierarchical Bayesian modeling of topics in time-stamped documents. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32:996–1011, 2010.

M. Purver, T. L. Griffiths, K. P. Körding, and J. B. Tenenbaum. Unsupervised topic modelling for multi-party spoken discourse. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 17–24, 2006.

D. Ramage, S. Dumais, and D. Liebling. Characterizing microblogs with topic models. In *4th International AAAI Conference on Weblogs and Social Media*. American Association for Artificial Intelligence, 2010.

L. Ren, D. Dunson, and L. Carin. The dynamic hierarchical Dirichlet process. In *Proceedings of the 25th international conference on Machine learning*, pages 824–831, 2008.

L. Ren, D. Dunson, S. Lindroth, and L. Carin. Dynamic nonparametric Bayesian models for analysis of music. *Journal of the American Statistical Association*, 105:458–472, 2010.

A. Ritter, C. Cherry, and B. Dolan. Unsupervised modeling of Twitter conversations. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 172–180, 2010.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. 2005.

A. Rodríguez. A short course on Bayesian nonparametric. In *Course slides at Universidade Federal Do Rio de Janeiro*. 2011.

A. Rodríguez, D. B. Dunson, and A. E. Gelfand. The nested Dirichlet process. *Journal of the American Statistical Association*, 103:1131–1154, 2008.

M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in Artificial Intelligence*, pages 487–494, 2004.

G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986.

I. Sato and H. Nakagawa. Topic models with power-law using Pitman-Yor process. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 673–682, 2010.

F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34:1–47, 2002.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4: 639–650, 1994.

M. Shafiei and E. Milios. Latent Dirichlet co-clustering. In *Proceedings of the Sixth International Conference on Data Mining*, pages 542–551, 2006.

M. Steyvers and T. Griffiths. Probabilistic Topic Models. In *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, 2007.

M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 306–315, 2004.

J. Tang, J. Zhang, R. Jin, Z. Yang, K. Cai, L. Zhang, and Z. Su. Topic level expertise search over heterogeneous networks. *Mach. Learn.*, 82:211–237, 2011.

Y. W. Teh. A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06, School of Computing, National University of Singapore, 2006a.

Y. W. Teh. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 985–992, 2006b.

Y. W. Teh. Dirichlet processes: Tutorial and practical course. In *Tutorial presentation at the 2007 Machine Learning Summer School*. 2007.

Y. W. Teh. Dirichlet processes. In *Encyclopedia of Machine Learning*. Springer, 2010.

Y. W. Teh and M. I. Jordan. Hierarchical Bayesian nonparametric models with applications. In *Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press, 2010.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581, 2006.

Y. W. Teh, D. Newman, and M. Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems 19*, pages 1353–1360. 2007.

Y. W. Teh, K. Kurihara, and M. Welling. Collapsed variational inference for HDP. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

I. Titov and R. McDonald. Modeling online reviews with multi-grain topic models. In *Proceeding of the 17th international conference on World Wide Web*, pages 111–120, 2008a.

I. Titov and R. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of ACL-08: HLT*, pages 308–316, 2008b.

H. Wallach, C. Sutton, and A. McCallum. Bayesian modeling of dependency trees using hierarchical Pitman-Yor priors. In *Proceedings of the Workshop on Prior Knowledge for Text and Language (in conjunction with ICML/UAI/COLT)*, pages 15–20, 2008.

H. Wallach, D. Mimno, and A. McCallum. Rethinking lda: Why priors matter. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1973–1981, 2009.

H. M. Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd international conference on Machine learning*, pages 977–984, 2006.

C. Wang, D. Blei, and D. Heckerman. Continuous time dynamic topic models. In *Proceedings of the Proceedings of the Twenty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-08)*, pages 579–586, 2008a.

C. Wang, D. Blei, and L. Fei-Fei. Simultaneous image classification and annotation. In *CVPR*, 2009a.

C. Wang, B. Thiesson, C. Meek, and D. Blei. Markov topic models. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, pages 583–590, 2009b.

H. Wang, M. Huang, and X. Zhu. A generative probabilistic model for multi-label classification. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 628–637, 2008b.

H. Wang, D. Zhang, and C. Zhai. Structural topic model for latent topical structure analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1526–1535, 2011.

X. Wang and E. Grimson. Spatial latent Dirichlet allocation. In *Advances in Neural Information Processing Systems 20*, pages 1577–1584. 2008.

X. Wang and A. McCallum. Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 424–433, 2006.

X. Wei and W. B. Croft. LDA-based document models for ad-hoc retrieval. In *SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, 2006.

X. Wei, J. Sun, and X. Wang. Dynamic mixture models for multiple time series. In *Proceedings of the 20th international joint conference on Artificial intelligence*, pages 2909–2914, 2007.

F. Wood and Y. W. Teh. A hierarchical nonparametric Bayesian approach to statistical language model domain adaptation. In *Proceedings of the Twelfth*

International Conference on Artificial Intelligence and Statistics, pages 607–614, 2009.

H. Xiao and T. Stibor. Efficient collapsed Gibbs sampling for latent Dirichlet allocation. In *the 2nd Asian Conference on Machine Learning (ACML)*, 2010.

D. Xing and M. Girolami. Employing latent Dirichlet allocation for fraud detection in telecommunications. *Pattern Recogn. Lett.*, 28:1727–1734, 2007.

R. Yehezkel and B. Lerner. Bayesian network structure learning by recursive autonomy identification. *J. Mach. Learn. Res.*, 10:1527–1570, 2009.

S.-Z. Yu and H. Kobayashi. Practical implementation of an efficient forward-backward algorithm for an explicit-duration hidden markov model. *Signal Processing, IEEE Transactions on*, 54(5):1947 – 1951, 2006.

J. Zhang, Y. Song, C. Zhang, and S. Liu. Evolutionary hierarchical Dirichlet processes for multiple correlated time-varying corpora. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1079–1088, 2010.