NAIST-IS-DD1261005

Doctoral Dissertation

Latent Variable Models for Discrete Data and the Learning Methods

Takuya Konishi

March 12, 2015

Department of Information Science
Graduate School of Information Science
Nara Institute of Science and Technology

A Doctoral Dissertation
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Doctor of ENGINEERING

Takuya Konishi

Thesis Committee:

Professor Kazushi Ikeda	(Supervisor)
Professor Yuji Matsumoto	(Co-supervisor)
Assistant Professor Takatomi Kubo	(Co-supervisor)
Project Assistant Professor Kohei Hayashi	(National Institute of Informatics)

Latent Variable Models for Discrete Data and the Learning Methods*

Takuya Konishi

Abstract

Latent variable models are probabilistic models that are widely applied in modern data analysis. Latent variable models have unobserved random variables, which are beneficial for revealing hidden structures behind observations and giving meaningful interpretation to complex data. In the last two decades, many researches have shown the efficiency and applicability in a variety of research areas.

While the latent variable models became known as popular methods, there are still research questions remaining. A practical issue is how to design latent variable models according to the properties of tasks. Unless suitable models are used, the users will fail to obtain desirable latent representation. Another issue on latent variable models is how to learn the models when observed data are given. While efficient learning algorithms have been proposed for many models, several models have not been explored enough. Such exploration is important for clarifying the characteristics of learning methods and finding better ones.

On the basis of the above perspectives, this thesis studies two specific problems about the latent variable models. In Chapter 3, we focus on search queries on the Web search engines. A search query consists of a combination of terms and the possible number of them is enormous. However, the search queries can be represented as common low-dimensional patterns. We propose a probabilistic topic model that extracts such patterns as pairs of latent topics. Using two real

*Doctoral Dissertation, Department of Information Science, Graduate School of Information Science, Nara Institute of Science and Technology, NAIST-IS-DD1261005, March 12, 2015.

query datasets, we demonstrate that the obtained topics are intelligible by humans and are highly accurate in keyword recommendation and query generation tasks.

In Chapter 4, we study the variational Bayesian inference methods of the infinite relational model for network data that have not attracted attention. We derive the collapsed variational Bayesian inference that we obtain by marginalizing out the parameters analytically. The collapsed variational Bayesian inference empirically outperforms the standard variational Bayesian inference in many real network datasets. The results also imply the collapsed variational Bayesian inference indicates even better performances in dense networks.

Keywords:

query log, network data, latent variable model, probabilistic topic model, Bayesian nonparametric model

離散データのための潜在変数モデルとその学習法*

小西 卓哉

内容梗概

潜在変数モデルは現代のデータ解析において幅広く利用されている確率モデルである．潜在変数モデルは未知の確率変数を仮定することで，観測データの隠れた構造を明らかにし，複雑なデータに意味ある解釈を与えるために有益である．実際，過去20年ほどで様々な研究分野でその有効性が示され，応用されてきた．

潜在変数モデルはその有用性が認知されてきた一方で，解決すべき問題はいまだ多く残されている．まず実際に用いる際には，扱う問題に応じて潜在変数モデルをどのように設計するか検討する必要がある．適切なモデルを使用しなければ，ユーザは所望の潜在表現を得ることは困難になる．また，与えられたデータからどのようにモデルを学習するかは実用上問題となる．これまで多くのモデルに対して学習アルゴリズムが提案されてきたが，まだ十分検討されていないモデルも存在する．こうした新たな学習アルゴリズムの検討は，そのモデルの学習アルゴリズムの特性を明らかにし，より良いアルゴリズムを発見する上で重要である．

このような観点から，本論文では，潜在変数モデルに関する2つの問題に着目する．3章では，ウェブ検索における検索クエリに焦点を当てる．検索クエリは複数の単語から構成されで，その可能な種類数は無数に存在する．しかし，実際のクエリはより低次元の共通するパターンとして表現できることが期待できる．本論文ではこうしたパターンを潜在トピックの組によって表現するトピックモデルを提案する．数値実験において，提案モデルが既存のトピックモデルよりも解釈性の高いパターンを推定でき，実際の応用を想定した実験で有効であることを確認する．

4章では，ネットワークデータに対する無限関係モデルの変分ベイズ法による学習アルゴリズムを研究する．とくに無限関係モデルのパラメータを解析的に周辺化できる性質を利用することで，周辺化変文ベイズ法と呼ばれる学習アルゴリ

[*] 奈良先端科学技術大学院大学 情報科学研究科 情報科学専攻 博士論文, NAIST-IS-DD1261005, 2015年3月12日.

ズムを導出し，数値実験を通してその性能を評価する．多くのネットワークデータについて，周辺化変分ベイズ法が通常の変分ベイズ法よりも高い性能を示すことを確認し，さらにこの傾向が密なネットワークで顕著に現れることを示す．

キーワード

クエリログ，ネットワークデータ，潜在変数モデル，トピックモデル，ノンパラメトリックベイズモデル

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations and Contributions

Probabilistic modeling is a fundamental approach in modern data analysis. It is a statistical method that views data as random variables and mathematically formulates input/output relations or generative processes of data. Clarifying such unseen mechanisms leads to discover knowledge from complex data. A characteristic of probabilistic models is to account for uncertainty of variables and parameters. Because real-world data are intrinsically noisy and sparse, information about uncertainty enhances the models and is effective in predicting data.

A powerful class of probabilistic models is latent variable models. Latent variable models include latent variables that are unobserved random variables, which represent hidden aspects of data such as cluster structures, semantic patterns and true states of noisy data [14]. Revealing such latent structures gives meaningful interpretation to the observations. A popular example of latent variable models is mixture models [6]. Mixture models represent data by the mixture of base probability distributions (components), and the latent variables map each data point to a component. Mixture models have been applied to a wide variety of problems such as text classification in natural language processing [37], speaker identification in speech recognition [43], and background subtraction in computer vision [48].

While many researches have shown the efficiency and applicability of latent variable models in the last two decades, there are still problems remaining as typified by the following two issues. **Modeling)** A practical issue is how to design latent variable models according to the properties of tasks. Unless suitable models are used, the users will fail to obtain desirable latent representation. In recent years, it is expected to solve more domain-specific tasks where the basic models do not necessarily work well. Hence, developing latent variable models is a significant step for extracting unique and meaningful latent information behind each task. **Learning)** Another issue on latent variable models is how to learn the models when observed data are given. Exact inference of latent variable models is mostly intractable, thus, approximate learning methods are substituted. While the learning methods have been proposed for many models, several models have not been explored enough. Such exploration is important for clarifying the characteristics of the learning methods and finding better ones.

On the basis of above perspectives, this thesis studies two specific problems about the latent variable models. In Chapter 3, we focus on search queries on the Web search engines. A search query consists of a combination of terms and the possible number of them is enormous. However, the search queries can be represented as low-dimensional hidden patterns. For example, queries "NY restaurant" and "boston hotel" are instances of a common semantic pattern "`location service`." To obtain such latent patterns, existing approaches require data preprocessing by humans or limitation of the target query domains, which hinders their applicability.

We propose a probabilistic topic model that extracts such patterns as pairs of latent topics (i.e., latent variables). The key idea is that we consider topic co-occurrence in a query rather than a full combination of topics, which significantly reduces computational cost yet enables us to acquire coherent topics without preprocessing. Using two real query datasets, we demonstrate that the obtained topics are intelligible by humans and are highly accurate in keyword recommendation and query generation tasks.

In Chapter 4, we focus on the learning methods of the infinite relational model (IRM) for network data. The IRM is a latent variable model for discovering cluster structures of relational data. Specifically we study the variational Bayesian (VB) methods for the IRM. The VB methods are a major inference algorithms for Bayesian models with latent variables, however, they have not attracted attention on the IRM. Clarifying the performance of the VB methods leads to understand the learning methods for the IRM more deeply.

We derive the VB inference algorithms of the IRM for network data. After showing the standard VB inference, we derive the collapsed variational Bayesian (CVB) inference and its variant called the zeroth-order collapsed variational Bayesian (CVB0) inference. The CVB and CVB0 inference empirically outperformed the standard VB inference in most real network datasets. The results imply the CVB and CVB0 inference indicates even better performance than in dense networks.

Note that Chapter 3 and 4 are common in that both of them deal with discrete data, which further motivate us to use latent variable models. To obtain latent representation from data, one possible approach is to employ classical dimension reduction methods such as principal component analysis that assume that the observations take continuous values. When these methods are used for discrete data, we need to regard them as continuous ones. However, such assumption may be violate when the methods apply to the prediction of missing values because these methods are allowed to output continuous predicted values. In contrast, latent variable models can be constructed flexibly for the type of observations, e.g., we can use the Bernoulli, multinomial, and Poisson distributions for discrete observations.

1.2 Organization of This Thesis

The remaining contents of this thesis is as follows. In Chapter 2, some statistical methods based on this thesis are introduced. In Chapter 3, we address the problem of extracting search query patterns with topic models. In Chapter 4, we study the VB methods for the IRM on network data. In Chapter 5, we finally summarize this thesis and give future directions for each problem.

Notation

In this thesis, we sometimes describe generative processes of random variables. We write $x \sim P$ as that a random variable x follows distribution P, and also call that x is drawn (generated) from distribution P. For example, if a random variable x is drawn from the multinomial distribution with parameter $\boldsymbol{\theta}$, we write it as $x \sim \text{Multinomial}(\boldsymbol{\theta})$.

Chapter 2

Preliminaries

In this Chapter, we overview statistical methods on the basis of this thesis. Since this thesis adopts Bayesian approaches in the main contents, we first confirm the basic points of Bayesian models with latent variables. Next, we briefly review probabilistic topic models for Chapter 3 and the Dirichlet process for Chapter 4.

2.1 Bayesian Models with Latent Variables

Probabilistic modeling starts from the specification of probability distribution over data points. Assuming that a random variable x_i denotes an observed data point and the distribution is parameterized by $\boldsymbol{\theta}$, the probabilistic model is represented as distribution $p(x|\boldsymbol{\theta})$. Given N observations $\boldsymbol{x} = \{x_i\}_{i=1}^{N}$, a fundamental approach to learn $\boldsymbol{\theta}$ in such parametric models is the maximum likelihood (ML) estimation:

$$\hat{\boldsymbol{\theta}}_{\mathrm{ML}} = \underset{\boldsymbol{\theta}}{\mathrm{argmax}}\, p(\boldsymbol{x}|\boldsymbol{\theta}), \tag{2.1}$$

where $p(\boldsymbol{x}|\boldsymbol{\theta})$ is called the likelihood function. However, it is less likely to obtain a sufficient amount of data in practical analysis. Thus the ML estimator often suffers over-fitting problem where the model fits observations in surplus and degrades generalization performance.

One solution to prevent the over-fitting is to construct Bayesian models that assume parameters are random variables themselves and drawn from distribution

$p(\boldsymbol{\theta})$, which is called prior distribution. In Bayesian models, the objective to evaluate is posterior distribution $p(\boldsymbol{\theta}|\boldsymbol{x})$. Given a model $p(x|\boldsymbol{\theta})$ and prior distribution $p(\boldsymbol{\theta})$, the posterior distribution is derived by Bayes rule:

$$p(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}, \boldsymbol{\theta})}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}. \tag{2.2}$$

where $p(\boldsymbol{x})$ is called the marginal likelihood or model evidence. If we are only interested in a mode of the posterior, the maximum a posteriori (MAP) estimation is used:

$$\hat{\boldsymbol{\theta}}_{\mathrm{MAP}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, p(\boldsymbol{\theta}|\boldsymbol{x}). \tag{2.3}$$

On the other hand, Bayesian estimation requires computing the full posterior distribution, which is used for obtaining important quantities such as Bayesian predictive distribution. However, as discussed later, the computation of the posterior is intractable in many useful models.

Bayesian models often include latent variables, which are unobserved random variables and associate the observations and parameters. Constructing such Bayesian models requires the specification of the joint distribution over variables and parameters. As an example, we consider a mixture model that is formulated as the most simple Bayesian models with latent variables. Consider a K component mixture model having a set of parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_k\}_{k=1}^{K}$. Suppose z_i denotes a discrete latent variable that associates parameters with a data point x_i, and $p(z_i|\boldsymbol{\pi})$ is the multinomial distribution parameterized by $\boldsymbol{\pi}$. We also assume the set of latent variables $\boldsymbol{z} = \{z_i\}_{i=1}^{N}$ and $p(\boldsymbol{\pi})$ as prior distribution. In the mixture model, the joint distribution $p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\pi})$ is written as follows:

$$\begin{aligned} p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\pi}) &= p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta})p(\boldsymbol{z}|\boldsymbol{\pi})p(\boldsymbol{\theta})p(\boldsymbol{\pi}) \\ &= \prod_{i=1}^{N} p(x_i|\boldsymbol{\theta}_{z_i})p(z_i|\boldsymbol{\pi})p(\boldsymbol{\theta})p(\boldsymbol{\pi}). \end{aligned} \tag{2.4}$$

This is derived from the conditional independence assumptions of the mixture model. How to design Bayesian latent variable models is how to define the joint distribution and comes into an essential issue in Bayesian modeling. The joint distribution also provides another view of the Bayesian models as a generative

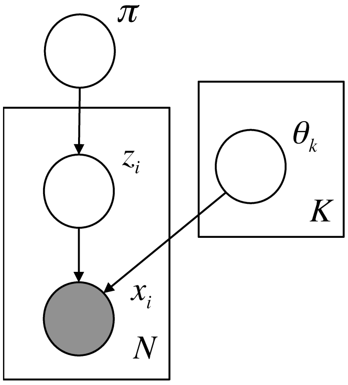

Figure 2.1. The graphical model of the mixture model

model of data. The mixture model represents a sequential generative process of variables; draw $\boldsymbol{\theta}$ and $\boldsymbol{\pi}$, $\boldsymbol{z}$, and $\boldsymbol{x}$ from corresponding distribution in order. To visualize such dependencies among variables, probabilistic graphical models are also used [27]. For example, the graphical model of the above mixture model is described as Figure 2.1. These views are helpful for designing hierarchical Bayesian models, where multiple latent variables and parameters have more complicated and elaborate dependencies. In Chapter 3, we argue a specific example: probabilistic modeling for search query logs.

Bayesian models with latent variables also require to evaluate the posterior distribution. The posterior of the above mixture model is defined on the parameters and the latent variables given observations, i.e., $p(\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\pi}|\boldsymbol{x})$. As in Eq. (2.2), the posterior distribution is derived by Bayes rule:

$$p(\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\pi}|\boldsymbol{x}) = \frac{p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\pi})}{p(\boldsymbol{x})} = \frac{p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta})p(\boldsymbol{z}|\boldsymbol{\pi})p(\boldsymbol{\theta})p(\boldsymbol{\pi})}{\int \sum_{\boldsymbol{z}} p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\theta})p(\boldsymbol{z}|\boldsymbol{\pi})p(\boldsymbol{\theta})p(\boldsymbol{\pi})d\boldsymbol{\theta}d\boldsymbol{\pi}}. \tag{2.5}$$

However, obtaining the full posterior distribution is intractable in many models including mixture models. This is caused by the computation of marginal likelihood $p(\boldsymbol{x})$ where the marginalization with respect to all the latent variables and parameters is required. How to solve such intractable computations is another significant issue about Bayesian models. There are mainly two approaches; Markov chain Monte Carlo (MCMC) methods and VB methods. While MCMC methods realize the posterior by sampling from the distribution, VB methods approximate the posterior by assuming mean-field approximation. In Chapter 4, we study the VB methods for the IRM of network data.

2.2 Probabilistic Topic Model

Probabilistic topic models are latent variable models to discover topics from a collection of documents [9,20]. In general, a document is written about one or more themes (i.e., topics), and such topics provide a clue to searching the relevant documents and visualizing the contents. Topic models extract the topical structures behind document collections by probabilistic modeling. In the following sections, we briefly introduce two topic models that are closely related to Chapter 3.

2.2.1 Latent Dirichlet Allocation

Nowadays, latent Dirichlet allocation (LDA) is known as the most basic topic model [9]. LDA is a hierarchical Bayesian model that incorporate the prior distribution over parameters into the model and a probabilistic generative model that represents the generative process of terms in the documents. In LDA, documents are regarded as "`the bag of words`" where a document is represented by an exchangeable sequence of terms, i.e., the order of terms in the document is ignored. The idea of LDA is available to not only document collections but also any grouped data assuming the bag of words.

In the following, we describe the detail of LDA. K is the number of topics, D is the number of documents in the collection, and V is the vocabulary size. LDA assumes that each term in a document is assigned a topic. Let $\boldsymbol{w}_d = \{w_{d,i}\}_{i=1}^{N_d}$ denote N_d terms in document d, $\boldsymbol{z}_d = \{z_{d,i}\}_{i=1}^{N_d}$ denote the corresponding topics. Topics are drawn from document-specific multinomial distribution whose parameter denotes $\boldsymbol{\theta}_d = (\theta_{d,1}, \ldots, \theta_{d,K})$. Once the topic is specified, the term is drawn from multinomial distribution associated with the topic, i.e., the term is conditionally independent to other terms given the topic. Suppose $\boldsymbol{\phi}_k = (\phi_{k,1}, \ldots, \phi_{k,V})$ is a parameter of multinomial distribution over terms associated with topic k, and $\boldsymbol{\phi} = \{\boldsymbol{\phi}_k\}_{k=1}^{K}$ is a set of the multinomial parameters. The joint distribution over

$\boldsymbol{w}_d$ and $\boldsymbol{z}_d$ is modeled as

$$\begin{aligned} p(\boldsymbol{w}_d, \boldsymbol{z}_d|\boldsymbol{\phi}, \boldsymbol{\theta}_d) &= p(\boldsymbol{w}_d|\boldsymbol{z}_d, \boldsymbol{\phi})p(\boldsymbol{z}_d|\boldsymbol{\theta}_d) \\ &= \prod_{i=1}^{N_d} p(w_{d,i}|z_{d,i}, \boldsymbol{\phi})p(z_{d,i}|\boldsymbol{\theta}_d) \\ &= \prod_{i=1}^{N_d} \phi_{z_{d,i},w_{d,i}}\theta_{d,z_{d,i}}. \end{aligned} \tag{2.6}$$

LDA assumes that the Dirichlet distribution over parameters as follows:

$$p(\boldsymbol{\phi}_k|\boldsymbol{\beta}) = \frac{\Gamma(\sum_{v=1}^{V}\beta_v)}{\prod_{v=1}^{V}\Gamma(\beta_v)}\prod_{v=1}^{V}\phi_{k,v}^{\beta_v-1}, \tag{2.7}$$

$$p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}\theta_{d,k}^{\alpha_k-1}. \tag{2.8}$$

where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_K)$ and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_V)$ are parameter vectors of the corresponding Dirichlet distribution. In total, for all D documents, the joint distribution over terms $\boldsymbol{w} = \{\boldsymbol{w}_d\}_{d=1}^{D}$, topics $\boldsymbol{z} = \{\boldsymbol{z}_d\}_{d=1}^{D}$, $\boldsymbol{\theta} = \{\boldsymbol{\theta}_d\}_{d=1}^{D}$ and $\boldsymbol{\phi}$ is

$$\begin{aligned} p(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\phi}, \boldsymbol{\theta}|\boldsymbol{\alpha}, \boldsymbol{\beta}) &= p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\phi})p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})p(\boldsymbol{\phi}|\boldsymbol{\beta}) \\ &= \prod_{d=1}^{D} p(\boldsymbol{w}_d|\boldsymbol{z}, \boldsymbol{\phi})p(\boldsymbol{z}_d|\boldsymbol{\theta}_d)p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})\prod_{k=1}^{K} p(\boldsymbol{\phi}_k|\boldsymbol{\beta}). \end{aligned} \tag{2.9}$$

From a view of mining knowledge from a document collection, parameters of LDA provides the interpretable latent structures. $\boldsymbol{\phi}_k$ denotes the term occurrence probabilities in topic k, which represents the relationship among terms on topic k. $\boldsymbol{\theta}_d$ is the proportion of topics in document d. This realizes the representation of multiplicity of topics in a document.

2.2.2 The Biterm Topic Model

The biterm topic model (BTM) is a topic model for short texts which appear frequently on the Internet services, e.g., Twitter [13, 55]. As mentioned above, LDA assumes the bag of words representation for each document, and models the document-specific topic proportions. However, these assumptions are not necessarily suitable for short texts. Because short texts consists of the less number of terms than that of standard documents, LDA suffers from the sparsity in documents. To overcome this problem, the BTM has been proposed. The key idea is to construct the generative model for not a collection of documents but term pairs.

We describe the BTM (We continue to use the notation of LDA with respect to the same symbols). For the BTM, all term pairs in a document are extracted and aggregated among the collection of documents. In the BTM, one shared topic is assumed for one term pair, and the topic is drawn from collection-level multinomial distribution whose parameter is $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_K)$. Once the topic is specified, both terms in the pair are drawn from the same multinomial distribution associated with the topic. Let $b_i = (w_{i,1}, w_{i,2})$ denote ith term pair, and z_i denote the corresponding topic. The joint distribution over b_i and z_i is

$$\begin{aligned} p(b_i, z_i|\boldsymbol{\phi}, \boldsymbol{\theta}) &= p(b_i|z_i, \boldsymbol{\phi})p(z_i|\boldsymbol{\theta}) \\ &= p(w_{i,1}|z_i, \boldsymbol{\phi})p(w_{i,2}|z_i, \boldsymbol{\phi})p(z_i|\boldsymbol{\theta}) \\ &= \phi_{w_{i,1},z_i}\phi_{w_{i,2},z_i}\theta_{z_i}. \end{aligned} \tag{2.10}$$

For all N term pairs, the joint distribution over $\boldsymbol{b} = \{b_i\}_{i=1}^N$, $\boldsymbol{z} = \{z_i\}_{i=1}^N$, $\boldsymbol{\phi}$, and $\boldsymbol{\theta}$ is as follows:

$$\begin{aligned} p(\boldsymbol{b}, \boldsymbol{z}, \boldsymbol{\phi}, \boldsymbol{\theta}|\boldsymbol{\alpha}, \boldsymbol{\beta}) &= p(\boldsymbol{b}|\boldsymbol{z}, \boldsymbol{\phi})p(\boldsymbol{z}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha})p(\boldsymbol{\phi}|\boldsymbol{\beta}) \\ &= \prod_{i=1}^N p(w_{i,1}|z_i, \boldsymbol{\phi})p(w_{i,2}|z_i, \boldsymbol{\phi})p(z_i|\boldsymbol{\theta})p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \prod_{k=1}^K p(\boldsymbol{\phi}_k|\boldsymbol{\beta}), \end{aligned} \tag{2.11}$$

where $p(\boldsymbol{\theta}|\boldsymbol{\alpha})$ is the Dirichlet distribution over parameters of topic distribution as in Eq. (2.8).

In spite of the simple assumption, Yan *et al.* showed that the BTM outperformed LDA in some empirical experiments [13, 55]. LDA arises in the sparsity of the topic distribution because of the shortness of documents and this hurts the performance. In contrast, the BTM avoids modeling the document-specific topic distribution, and describe the generative process of term pairs that have primitive term co-occurrence information in the document collection. While the BTM has no document-specific topic distribution, it is possible to recover to the topic proportions on a document if the term occurrence information in the document collection is stored in advance.

2.3 The Dirichlet Process

The Dirichlet process is a stochastic process that is often used in Bayesian nonparametric models. Recall that Bayesian models assume prior distribution over parameters. Bayesian nonparametric models particularly use prior distribution defined on infinite-dimensional space with stochastic processes such as the Dirichlet process.

The basic application of the Dirichlet process is to extend the mixture models, which are called the Dirichlet process mixture models. In model selection of mixture models, setting the number of components is essential task. Dirichlet process mixture models extend the mixture models with incorporating the Dirichlet process. This modeling does not assume the fixed number of components, and the number is also estimated according to the complexity of observations.

The definition of the Dirichlet process has been introduced by Ferguson [15]. The Dirichlet process is distribution over probability measures, which draws discrete random probability measure G over measurable space $(\Omega, \mathcal{B})$, where Ω is sample space and $\mathcal{B}$ is σ algebra of Ω. Suppose a finite partition $(A_1, \ldots, A_K)$ of Ω, then $(G(A_1), \ldots, G(A_K))$ indicates a random vector. For any finite partition $(A_1, \ldots, A_K)$, G according to the Dirichlet process satisfies following property:

$$(G(A_1), \ldots, G(A_K)) \sim \text{Dirichlet}(\alpha G_0(A_1), \ldots, \alpha G_0(A_K)), \tag{2.12}$$

where $\text{Dirichlet}(\cdot)$ is the Dirichlet distribution, α is concentration parameter, and G_0 is base measure. We write that G is drawn from the Dirichlet process as $G \sim \text{DP}(\alpha, G_0)$.

2.3.1 The Stick-breaking Process

A more intuitive representation of the Dirichlet process is the stick-breaking process [46]. Suppose the sets of random variables $\{v_k\}_{k=1}^{\infty}$, $\{\pi_k\}_{k=1}^{\infty}$, and $\{\theta_k\}_{k=1}^{\infty}$. A random measure G is specified with the stick-breaking process as follows:

$$\begin{aligned} v_k &\sim \mathrm{Beta}(1, \alpha), \qquad\qquad \theta_k \sim G_0, \\ \pi_k &= v_k \prod_{m=1}^{k-1} (1 - v_m), \\ G &= \sum_{k}^{\infty} \pi_k \delta_{\theta_k}, \end{aligned} \tag{2.13}$$

where $\mathrm{Beta}(\cdot)$ is the Beta distribution, and δ_{θ_k} is Dirac measure centered on θ_k. The construction satisfies $\sum_k^\infty \pi_k = 1$ and is equivalent to $G \sim \mathrm{DP}(\alpha, G_0)$. Intuitively, this process provides infinite discrete distribution $\{\pi_k\}_{k=1}^\infty$ associated with random variables $\{\theta_k\}_{k=1}^\infty$ drawn from base measure. In the Dirichlet process mixture models, base measure corresponds to prior distribution over parameters, and the process is incorporated as a generative process on parameters of components and their proportions whose number is countably infinite.

Chapter 3

Extracting Search Query Patterns via Topic Model

3.1 Introduction

When people want to find new information on the Internet, they commonly use search engines. Users interact with search engines by queries, and their intents and personal backgrounds are expected to be preserved in query logs. Thus, data mining and analysis of query logs are important problems for many applications.

To obtain user information from diverse and unorganized queries, extracting the low-dimensional hidden structures behind the queries is an essential task. This is motivated by our intuition that, while the number of possible queries are nearly infinite, most real queries fall into a combination of a few terms from multiple categories. For example, a query "NY restaurant" consists of the term "NY", which indicates a `location`, and the term "restaurant", which specifies the type of `service`. If we know such *is-a* relationship, we can infer that the user intends to use that `service` and lives in or will go to the `location` in the near future. Such knowledge gives us a rich interpretation of user needs, which is beneficial to better search experiences in related applications. Keyword recommendation in Internet advertising is a typical example [2, 41]; by using the obtained structure of query patterns, we can generate a query for advertising clients such that the query is long-tail (i.e., not popular and low advertising rate) but has the same structure of some hot keywords and is expected to have a similar advertising effectiveness. Furthermore, meaningful query patterns would potentially applicable to predicting click through rate [25], profiling users [18], and improving search results [47].

The identification of such categories and *is-a* relationships has emerged as a new area in information retrieval, which is referred to as query templates [1] and several approaches has been proposed to solve the query template task [1, 17, 40]. Although such approaches have demonstrated the usefulness of query templates for real applications, they have several limitations. First, they need the specification of target domain beforehand [1, 40], i.e., we must collect queries that are related to a certain theme, such as automobiles, travels, and movies. This preprocessing reduces the number of queries and the diversity drastically; however, it loses the generality and much useful information, such as cross-domain knowledge. Another issue is that the need for human assistance [17]. While humans can identify the term-category relationships with high accuracy, its expensive resource cost and low throughput hampers the applicability to large-scale query logs. In addition, due to the expensive computational cost, the number of the category (K) is considerably restricted to be small such as $K \simeq 6$.

The use of topic models [8, 9, 55] can be an alternative approach for the query template task. Topic models learn the relationships between a term and a category as a *topic* from a large number of documents (i.e., queries) in an unsupervised manner. A desirable property of this approach is that they are feasible with $K \geq 100$ categories. However, they typically assume that each document contains many terms; therefore, topic models are not suitable for sparse data, such as query logs [8, 9]. In addition, there is no straightforward way to incorporate information from a combination of topics, which plays a significant role in search queries. For example, a topic combination (`location`, `service`) appears frequently in real queries, but (`service`, `service`) is very rare.

To address the above issues, we propose pairwise coupled topic model (PCTM)—a probabilistic topic model for query logs. The PCTM approximates the combinational information over queries by co-occurrences of topics, which significantly enriches the model and overcomes the shortness and sparseness of queries. On the basis of the approximation, we derive a fully-Bayesian inference algorithm with collapsed Gibbs sampling (CGS), which allows us to handle queries as a collection of term pairs, which significantly reduces computational cost from $O(K^M)$ to $O(M^2K^2)$ where M is the length of a query; in the PCTM, M is essentially very small and we can manage sufficient sizes of $K \simeq 100$. Our contributions are summarized as follows.

Versatility of the model: The PCTM can handle queries of any domains, and it is not necessary to specify them. In addition, the PCTM does not require human assistance.

Validity of pairwise approximation: We derive the PCTM as an approximation of a simple fully-dependent query model, which provides a legitimate procedure to estimate a query pattern while drastically reducing the computational cost.

Extraction of sparse cross-domain relation: The PCTM estimates topic co-occurrence as a sparse covariance matrix, which gives us cross-domain knowledge as an interpretable network of topics.

Applicability to real data: We evaluate the PCTM with two query logs in different languages and show that (1) the obtained topics are coherent and natural for humans, which we examined by crowdsourcing, (2) the PCTM is highly accurate in terms of query recommendation, and (3) the PCTM can generate the nearest queries to the real queries.

3.2 Modeling Query Logs

3.2.1 Problem Definition

Given a set of queries, we would like to obtain the following knowledge: (a) a set of distinct categories of terms used in query logs, (b) *is-a* relationships between the categories and terms, and (c) a mapping from a query to query pattern represented by a combination of the categories. Let us explain the detail by using an example. Suppose we have the query, “chicago hotel cheapest” where “chicago”, “hotel”, and “cheapest” are instances of the categories `location`, `service`, and `condition`, respectively. Such categories are related to many other terms, e.g., `location` is associated with a variety of terms that indicate the names of places. Note that a term can possibly belong to multiple categories, e.g., “chicago” may be associated with a `movie` category in other queries. With such many-to-many relationships, the query pattern is recovered as a combination of categories (`location`, `service`, `condition`).

We notice that it is difficult to obtain the category information from any other external resources in advance (e.g., dictionaries and thesauruses) because the concepts defined by external resources do not always correspond to desirable categories that reflect the actual query activities. For example, both “houston” and “miami” are names of places; however, queries containing “houston” and queries containing “miami” could be very different in terms of user intent and personal backgrounds. The former is in a residential area, and would likely be used with `dailylife` keywords, such as “houston apartment” or “houston job.” In contrast, the latter is in a resort area and would likely be used with `travel` keywords, such as “miami hotel” or “miami restaurant.” Treating these two terms as the same category loses such query-specific information.

3.2.2 A Naive Query Model

To obtain knowledge from query logs, we consider a simple probabilistic model of a query based on a topic model. Suppose we have an M-long query $\boldsymbol{q} = (w_1, \ldots, w_M)$ where each term w_m has a latent topic z_m and these terms are conditionally independent given the topics. Under this assumption, the joint probability of $\boldsymbol{q}$ and $\boldsymbol{z} = (z_1, \ldots, z_M)$ is expressed as follows:

$$\begin{aligned} p(\boldsymbol{q}, \boldsymbol{z}) &= p(\boldsymbol{q}|\boldsymbol{z})p(\boldsymbol{z}) \\ &= \prod_{w_m \in \boldsymbol{q}} p(w_m|z_m)p(z_1, \ldots, z_M). \end{aligned} \tag{3.1}$$

This formulation satisfies the requirements mentioned in Section 3.2.1: (a) categories (i.e., topics) are represented as the latent variables $\boldsymbol{z}$, (b) *is-a* relationships between w_m and z_m are represented as term distribution $p(w_m|z_m)$ that indicates the probability of term occurrences given a topic, and (c) a combination of the categories is represented as a topic combination $(z_1, \ldots, z_M)$, and $p(z_1, \ldots, z_M)$ is the distribution that indicates the probability of the occurrences of the topic combination. Note that representing *is-a* relationships by the term distribution enables that one term can be associated with multiple topics.

Without loss of generality, model (3.1) is rewritten as a product of the multinomial distribution for terms and the tensor-variate multinomial distribution for a topic combination as follows:

$$p(\boldsymbol{q}, \boldsymbol{z}) = \prod_{w_m \in \boldsymbol{q}} \phi_{w_m, z_m} \pi_{z_1, \ldots, z_M}, \tag{3.2}$$

where ϕ_{w_m, z_m} is the probability of the term w_m given topic z_m, and $\pi_{z_1,\ldots,z_M}$ is the probability of topic combination $(z_1, \ldots, z_M)$; thus, $\pi_{z_1,\ldots,z_M}$ can be considered as an M-th order tensor having K^M elements, which represent the probabilities of all possible combinations of topics $z_1, \ldots, z_M$.

Model (3.2) is general and flexible; however, it requires huge computational cost in parameter estimation. Bayesian inference needs to marginalize the latent variables $\boldsymbol{z}$, which is equivalent to computing a summation of all possible values of $\pi_{z_1,\ldots,z_M}$ in Eq. (3.2) and this requires $O(K^M)$ computation. Even though query length M is relatively small compared to that of normal documents, practically M could be more than 10 in real query logs and it would easily make parameter estimation computationally infeasible.

3.3 The PCTM

As discussed above, the main computational cost of model (3.2) arises from the full dependency of $p(z_1, \ldots, z_M)$. In this section, we consider relaxation of this assumption.

3.3.1 Pairwise Decomposition of Topics

Our key idea is that, in most queries, the full dependency of a topic combination is sufficiently approximated by a collection of pairwise relationships among topics. For example, let us reconsider query "chicago hotel cheapest." Here, we can recognize that "chicago" is used as `location` rather than `movie` because it is jointly used with "hotel." Similarly, "hotel cheapest" sufficiently determines the context of both "hotel" and "cheapest." Therefore, in this example, two word pairs “chicago hotel” and “hotel cheapest” are sufficient to estimate the intent of this query.

By following this observation, we assume that $p(z_1, \ldots, z_M)$ is decomposed as the product of the second-order tensors with respect to z. For example, if $M = 3$, the joint probability of topics is written as:

$$p(z_1, z_2, z_3) = (\Psi(z_1, z_2)\Psi(z_1, z_3)\Psi(z_2, z_3))^{\frac{1}{2}}, \tag{3.3}$$

where $\Psi(\cdot, \cdot)$ is a potential function representing a pairwise interaction of topics. This approximation can be generalized as follows:

$$p(z_1, \ldots, z_M) = \left(\prod_{(i,j) \in B_M} \Psi(z_i, z_j) \right)^{\frac{1}{M-1}}, \tag{3.4}$$

where B_M is the set of all pair indices in an M-long sequence. Note that we introduce exponential weight $\frac{1}{M-1}$ for later convenience. Substituting decomposition (3.4) into model (3.1) yields:

$$\begin{aligned} p(\boldsymbol{q}, \boldsymbol{z}) &= \prod_{w_m \in \boldsymbol{q}} p(w_m|z_m) \left(\prod_{(i,j) \in B_M} \Psi(z_i, z_j) \right)^{\frac{1}{M-1}} \\ &= \left(\prod_{(i,j) \in B_M} p(w_i|z_i) p(w_j|z_j) \Psi(z_i, z_j) \right)^{\frac{1}{M-1}}. \end{aligned} \tag{3.5}$$

3.3.2 Modeling Term and Topic Pairs

Interestingly, Eq. (3.5) implies that pairwise decomposition of topics (3.3) allows us to deal with the entire model as the product of the sub model with respect to term pairs (w_i, w_j) and topic pairs (z_i, z_j) for $(i, j) \in B_M$. By inspiring this notion, we directly model these pairs instead of a query and a topic combination. Let $\boldsymbol{b} = \{b_i\}_{i=1}^{|B_M|} = \{(w_{i,1}, w_{i,2})\}_{i=1}^{|B_M|}$ denote $|B_M|$ term pairs[1] and $\boldsymbol{x} = \{x_i\}_{i=1}^{|B_M|} =$

[1] Clearly, $|B_M| = M(M-1)/2$.

$\{(z_{i,1}, z_{i,2})\}_{i=1}^{|B_M|}$ denote the corresponding topic pairs, and we model them as

$$\begin{aligned} p(\boldsymbol{b}, \boldsymbol{x}) &= \prod_{i=1}^{|B_M|} p(b_i|x_i)p(x_i), \qquad (3.6)\\ \text{where} \quad p(b_i|x_i) &= p(w_{i,1}|z_{i,1})p(w_{i,2}|z_{i,2}) \\ &= \phi_{w_{i,1},z_{i,1}}\phi_{w_{i,2},z_{i,2}}, \\ p(x_i) &= p(z_{i,1}, z_{i,2}) \\ &= \psi_{z_{i,1},z_{i,2}}. \end{aligned}$$

Note that $\psi_{\cdot,\cdot}$ denotes the probability of the $K \times K$ matrix-variate multinomial distribution, which represents co-occurrences of topics.

While both $p(\boldsymbol{q}, \boldsymbol{z})$ in Eq. (3.5) and $p(\boldsymbol{b}, \boldsymbol{x})$ in Eq. (3.6) look similar, there exists a major difference in terms of a treatment of the latent variables. In model (3.5), z_i is shared over the product, i.e., z_i in (z_i, z_j) is the same latent variable of z_i in (z_i, z_k) if both pairs belong to the same query. However, model (3.6) ignores the information of a query, and all zs are considered as different latent variables.

Inconsistency of the latent variables in $p(\boldsymbol{b}, \boldsymbol{x})$ provides a tradeoff between accuracy and computational cost. Since we deal with z_is in (z_i, z_j) and (z_i, z_k) are different variables, $p(\boldsymbol{b}, \boldsymbol{x})$ possibly assigns different topics into the z_is; thus, accuracy of topic estimation may suffer. However, since all latent variables are independent in $p(\boldsymbol{b}, \boldsymbol{x})$, we can separately take the marginalization with respect to $\boldsymbol{x}$, which significantly reduces computational cost from $O(K^M)$ to $O(M^2K^2)$.

3.3.3 A Hierarchical Bayesian Model

On the basis of the above idea, we construct the PCTM as a hierarchical Bayesian model. Given query logs, Let $\boldsymbol{b} = \{b_i\}_{i=1}^N = \{(w_{i,1}, w_{i,2})\}_{i=1}^N$ be a set of term pairs[2] converted from the query logs and $\boldsymbol{x} = \{x_i\}_{i=1}^N = \{(z_{i,1}, z_{i,2})\}_{i=1}^N$ be the corresponding topic pairs. Note that N is all the number of term pairs in the query logs. In the PCTM, term pairs in a query are not distinguished from those in the other queries no longer. As explained above, a topic pair is regarded as a single

[2] In the following, we assume that each query contains more than two terms and ignore single-term queries. We also assume that a query does not include the same terms; thus, we eliminate duplicate terms, e.g., we deal with "NY NY restaurant" as "NY restaurant.")

Algorithm 1 Generative process of the PCTM

```
Draw $\boldsymbol{\psi} \sim$ Dirichlet($\boldsymbol{\gamma}$)
for all topics $k$ =1...$K$ do
    Draw $\boldsymbol{\phi}_k \sim$ Dirichlet($\boldsymbol{\beta}$)
end for
for all term pairs $b_i$ $i$ =1...$N$ do
    Draw topic pair $x_i = (z_{i,1}, z_{i,2}) \sim$ Multinomial($\boldsymbol{\psi}$)
    Draw term $w_{i,1} \sim$ Multinomial($\boldsymbol{\phi}_{z_{i,1}}$)
    Draw term $w_{i,2} \sim$ Multinomial($\boldsymbol{\phi}_{z_{i,2}}$)
end for
```

latent variable; a topic pair is drawn from a multinomial distribution having a K^2-dimensional parameter vector $\boldsymbol{\psi}$. Once a topic pair x_i is generated, individual topic assignments $z_{i,1}$ and $z_{i,2}$ are determined. If topics $z_{i,1} = k$ and $z_{i,2} = l$, then $w_{i,1}$ and $w_{i,2}$ are drawn from multinomial distributions that are associated with the kth and lth topics, respectively. These K multinomials are parameterized by V-dimensional parameter vectors $\boldsymbol{\phi} = \{\boldsymbol{\phi}_k\}_{k=1}^{K}$ where V denotes the number of words in the vocabulary.

The PCTM assumes multinomial parameters are drawn from the Dirichlet distributions. $\boldsymbol{\beta}$ is a V-dimensional parameter vector for the Dirichlet over $\boldsymbol{\phi}_k$ $(k = 1, \ldots, K)$, and $\boldsymbol{\gamma}$ is a K^2-dimensional parameter vector for the Dirichlet over $\boldsymbol{\psi}$. We assume the symmetric Dirichlet distribution for $\boldsymbol{\phi}_k$ and $\boldsymbol{\psi}$; thus, $\beta_v = \beta$ $(v = 1, \ldots, V)$ and $\gamma_{k,l} = \gamma$ $(k, l = 1, \ldots, K)$. The generative process and graphical model are shown in Algorithm 1 and Figure 3.1(a), respectively.

3.3.4 An Inference Algorithm

The joint distribution of the PCTM is written as:

$$\begin{aligned}
&p(\boldsymbol{b}, \boldsymbol{x}, \boldsymbol{\phi}, \boldsymbol{\psi}|\boldsymbol{\beta}, \boldsymbol{\gamma}) \\
&= \prod_{i=1}^{N} p(b_i|x_i, \boldsymbol{\phi}) p(x_i|\boldsymbol{\psi}) \prod_{k=1}^{K} p(\boldsymbol{\phi}_k|\boldsymbol{\beta}) p(\boldsymbol{\psi}|\boldsymbol{\gamma}) \\
&= \prod_{i=1}^{N} p(w_{i,1}|z_{i,1}, \boldsymbol{\phi}) p(w_{i,2}|z_{i,2}, \boldsymbol{\phi}) p(x_i|\boldsymbol{\psi}) \prod_{k=1}^{K} p(\boldsymbol{\phi}_k|\boldsymbol{\beta}) p(\boldsymbol{\psi}|\boldsymbol{\gamma}). \qquad (3.7)
\end{aligned}$$

Then, the distribution after marginalizing out parameters $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ is given as:

$$\begin{aligned}
p(\boldsymbol{b}, \boldsymbol{x}|\boldsymbol{\beta}, \boldsymbol{\gamma}) &= p(\boldsymbol{b}|\boldsymbol{x}, \boldsymbol{\beta}) p(\boldsymbol{x}|\boldsymbol{\gamma}) \\
&= \int p(\boldsymbol{b}|\boldsymbol{x}, \boldsymbol{\phi}) p(\boldsymbol{\phi}|\boldsymbol{\beta}) d\boldsymbol{\phi} \int p(\boldsymbol{x}|\boldsymbol{\psi}) p(\boldsymbol{\psi}|\boldsymbol{\gamma}) d\boldsymbol{\psi} \\
&= \left(\frac{\Gamma(V\beta)}{\Gamma(\beta)^V} \right)^K \prod_{k=1}^{K} \frac{\prod_{v=1}^{V} \Gamma(n_{v|k} + \beta)}{\Gamma(n_{\cdot|k} + V\beta)} \\
&\quad \frac{\Gamma(K^2\gamma)}{\Gamma(\gamma)^{K^2}} \frac{\prod_{k=1}^{K} \prod_{l=1}^{K} \Gamma(n_{k,l} + \gamma)}{\Gamma(N + K^2\gamma)}, \qquad (3.8)
\end{aligned}$$

where $\Gamma(\cdot)$ is the gamma function, $n_{v|k}$ is the number of terms v assigned to topic k, $n_{\cdot|k}$ is the total number of terms assigned to topic k, and $n_{k,l}$ is the number of term pairs assigned to the pair of topic k and l, which are written as follows:

$$n_{v|k} = \sum_{i=1}^{N} \mathbb{I}_v[w_{i,1}]\mathbb{I}_k[z_{i,1}] + \mathbb{I}_v[w_{i,2}]\mathbb{I}_k[z_{i,2}], \qquad (3.9)$$

$$n_{\cdot|k} = \sum_{v=1}^{V} n_{v|k}, \qquad (3.10)$$

$$n_{k,l} = \sum_{i=1}^{N} \mathbb{I}_k[z_{i,1}]\mathbb{I}_l[z_{i,2}], \qquad (3.11)$$

where $\mathbb{I}_a[b]$ is an indicator function that takes 1 when $a = b$.

We infer the parameters by CGS algorithm. Samples are obtained by the

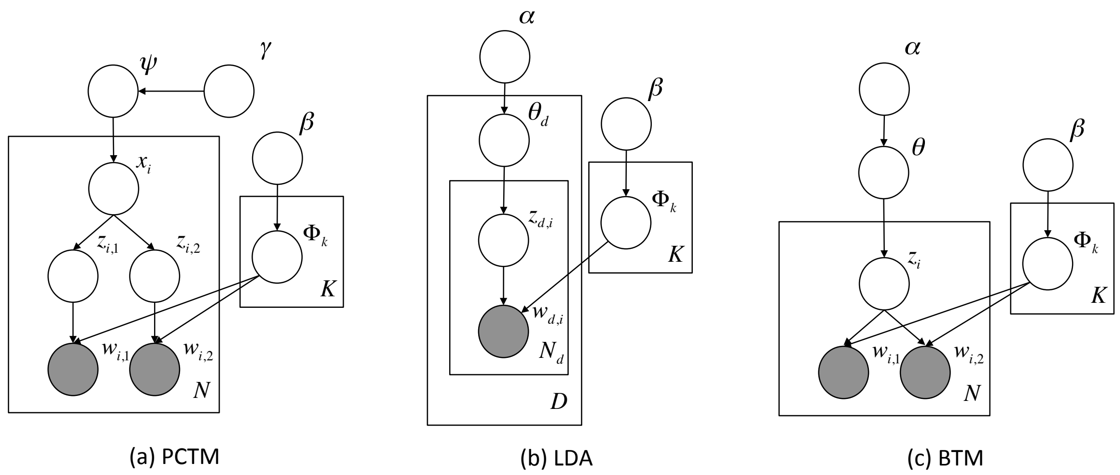

Figure 3.1. Graphical models of the PCTM, LDA, and the BTM. For the LDA and BTM, we follow the notation from Chapter 2.

following conditional distribution:

$$p(x_i = (k,l)|\boldsymbol{b}, \boldsymbol{x}^{-i})$$
$$\propto \begin{cases} (n_{k,l}^{-i} + \gamma)\frac{(n^{-i}_{w_{i,1}|k}+\beta)(n^{-i}_{w_{i,2}|l}+\beta)}{(n^{-i}_{\cdot|k}+V\beta)(n^{-i}_{\cdot|l}+V\beta)} & (k \neq l) \\ (n_{k,k}^{-i} + \gamma)\frac{(n^{-i}_{w_{i,1}|k}+\beta)(n^{-i}_{w_{i,2}|k}+\beta)}{(n^{-i}_{\cdot|k}+1+V\beta)(n^{-i}_{\cdot|k}+V\beta)} & (k = l) \end{cases}, \tag{3.12}$$

where the notation $-i$ indicates the set of variables or the counts excluding the variable at the ith position. The formula changes slightly when $z_{i,1}$ and $z_{i,2}$ take the same topic because two terms are excluded from this particular count. From the obtained samples of topic assignments, we estimate the integrated parameters $\boldsymbol{\phi}$ and $\boldsymbol{\psi}$ as follows:

$$\hat{\phi}_{k,v} = \frac{n_{v|k} + \beta}{n_{\cdot|k} + V\beta}, \tag{3.13}$$

$$\hat{\psi}_{k,l} = \frac{n_{k,l} + \gamma}{N + K^2\gamma}. \tag{3.14}$$

This inference requires K^2 computations to check the probabilities of each topic pair, and the total time complexity is $O(NK^2)$.

3.3.5 Recovery of Topic Combination

The PCTM also gives us an approximate way to recover the most probable topic combination $\boldsymbol{z} = (z_1, \ldots, z_M)$ from the original query $\boldsymbol{q} = (w_1, \ldots, w_M)$. In the PCTM, the posterior of the topic pair $x = (z_i, z_j)$ given the term pair $b = (w_i, w_j)$ and the estimated parameters, i.e., $\hat{\boldsymbol{\phi}}$ and $\hat{\boldsymbol{\psi}}$ is written as follows:

$$p(x|b)_{kl} = p(z_i, z_j|w_i, w_j)_{kl} = \frac{\hat{\psi}_{k,l}\hat{\phi}_{k,w_i}\hat{\phi}_{l,w_j}}{\sum_{k'=1}^{K}\sum_{l'=1}^{K}\hat{\psi}_{k',l'}\hat{\phi}_{k',w_i}\hat{\phi}_{l',w_j}}. \tag{3.15}$$

By combining this and Bayes' rule, we approximate the joint posterior of the topics by the following Markov chain:

$$\begin{aligned}
&p(\boldsymbol{z}|\boldsymbol{q}) \\
&\approx p(z_1|\boldsymbol{q})p(z_2|z_1, \boldsymbol{q})p(z_3|z_2, \boldsymbol{q}) \ldots p(z_M|z_{M-1}, \boldsymbol{q}) \\
&\approx p(z_1|w_1, w_2)p(z_1|z_2, w_1, w_2)p(z_3|z_2, w_2, w_3) \ldots p(z_M|z_{M-1}, w_{M-1}, w_M)
\end{aligned} \tag{3.16}$$

where

$$p(z_i|z_j, w_i, w_j)_{kl} = \frac{p(z_i, z_j|w_i, w_j)_{kl}}{p(z_j|w_i, w_j)_l} = \frac{p(z_i, z_j|w_i, w_j)_{kl}}{\sum_{k'} p(z_i, z_j|w_i, w_j)_{k'l}} \tag{3.17}$$

denotes the conditional posterior of z_i given z_j. The last line of Eq. (3.16) enables us to compute the most probable combination of topics by forward algorithms, such as the Viterbi algorithm [50]. In the Viterbi algorithm, we can compute the most probable combination $\boldsymbol{z}^{max} = (z_1^{max}, \ldots, z_M^{max})$ as in Algorithm 2.

Note that z_m has K-different topics, and the original joint probability $p(\boldsymbol{z}|\boldsymbol{q})$ is represented by the M-th order tensor where each dimension is given by K. In this representation, there is no efficient way to find the most probable combination, and the naive computation requires $O(K^M)$ complexity. In contrast, the proposed approach is computable by multiplication and summation on $K \times K$ matrices, which requires only $O(MK^2)$ complexity.

3.4 Related Works

3.4.1 Topic Models for Search Logs

Topic models have been widely used for modeling search log datasets with different motivations. One motivation has been to improve search results based on

Algorithm 2 The Viterbi algorithm for the recovery of the topic combination

```
Input: Conditional probabilities (3.16)
for $z_1 = 1 \dots K$ do
    $t(1, z_1) = \log p(z_1|w_1, w_2)$
end for
for $m = 2 \dots M$ do
    for $z_m = 1 \dots K$ do
        $t(m, z_m) = \max_{z_{m-1}} [\log p(z_m|z_{m-1}, w_{m-1}, w_m) + t(m-1, z_{m-1})]$
        $s(m, z_m) = \mathrm{argmax}_{z_{m-1}} [\log p(z_m|z_{m-1}, w_{m-1}, w_m) + t(m-1, z_{m-1})]$
    end for
end for
$z_M^{max} = \mathrm{argmax}_z\, t(M, z)$
for $m = M - 1 \dots 1$ do
    $z_m^{max} = s(m+1, z_{m+1}^{max})$
end for
Output: $\boldsymbol{z}^{max}$
```

search log histories [11,18,47]. For example, Harvey *et al.* proposed a topic model that incorporates search engine users for user profiling to improve personalized searches [18]. Other researchers have explored topic models for traditional search log tasks, such as predicting click-through rates [25, 26]. Jiang *et al.* proposed a topic model for some information included in search logs, such as URLs and timestamps [25]. Search logs are valuable sources of information for obtaining web-based knowledges. Xu *et al.* proposed a topic model for named entity mining with an efficient semi-supervised learning algorithm [54].

As in the above models, topic models have been used for modeling search log datasets that contain not only queries and other information, such as users, time stamps, URLs, and click-through logs. On the other hand, we focus on modeling the queries themselves. In our setting, we require only raw queries, which extends the range of application.

3.4.2 Query Templates

Query template methods have been explored to obtain query patterns [1,17,40]. Agarwal *et al.* advocated concepts of query templates and proposed a probabilistic inference framework for mining templates based on tripartite graphs among queries, websites, and templates [1]. Han *et al.* studied a human-assisted method for analyzing query templates that incorporates crowdsourcing for query interpretation [17].

Pandey and Punera proposed a probabilistic generative model for queries and query templates [40]. In that model, templates are defined as a topic combination without duplication, which is not an actual topic assignment for each term. Given a template per query, the model generates the number of terms assigned to the topics included in the template with the Poisson distribution and assigns the topics for each term. For example, the query "toyota 2002 seat cover" is assumed to have a template (`brand`, `year`, `parts`) and is generated by the topic assignments (`brand`, `year`, `parts`, `parts`), that is, the model generates "toyota" from `brand`, "2002" from `year`, and "seat" and "cover" from `parts`.

Although these methods have been demonstrated to extract meaningful query patterns, they only dealt with queries in a particular domain. This restriction improves the quality of query templates for that domain; however, it must perform preprocessing to limit target domains by using such as query classification. For example, one proposed method [1] assumes that *is-a* relationships in a domain are given *a priori* using other methods, such as named entity mining, and they only focused on query pattern extraction. Another proposed model [40] does not employ assumptions about domains; however, it also requires domain limitation due to the computational cost that arises from the dependencies among topics. Compared to those methods, we tackle a more challenging task, i.e., extraction of category and query patterns simultaneously without domain limitation and other preprocessing methods.

3.4.3 Relation to Other Topic Models

LDA

The PCTM is related to other topic models. One major topic model is LDA [9]. A graphical model of LDA is shown in Figure 3.1(b). We can apply LDA to query logs by considering a single query as a document; however this may fail due to the following reasons. First, although LDA assumes that a topic assigned to a term is drawn from a document-specific topic distribution, short texts such as queries have no more than a few terms in a document. This would cause the sparsity problem, i.e., LDA suffers from the lack of information for each query in estimating the topic distribution [55]. Moreover, if a dataset contains D documents, LDA has a parameter of size $D \times K$, and since D is normally very large in query logs, we must consider memory usage for parameter inference carefully. In contrast, the PCTM has only one distribution that generates topic pairs over the dataset. This avoids the sparsity problem, and the topic pair distribution is represented by a K^2 matrix, which does not depend on the number of documents. In the parameter inference, LDA requires $O(K)$ computation to check the probability of a single term. Therefore, if the average length of documents is L, the total time complexity is $O(DLK)$.

The BTM

Yan *et al.* proposed the BTM [55]. The graphical model of the BTM is shown in Figure 3.1(c). While the BTM and PCTM can be considered topic models for term pairs, the key difference is that the BTM assumes that a term pair shares the same topic; this assumption leads a great success of the BTM in application to short texts because term pairs in general short texts would rarely take completely different topics. However this assumption is not suitable for modeling queries because a query consists of terms having a variety of topics, and it is violent to assume only one shared topic.

To clarify this difference, we illustrate the generative processes of the BTM and PCTM in Figure 3.2. Recall the example “NY restaurant” (Section 3.1). The BTM generates this term pair from one term distribution associated with a topic. Because the location names and service names, such as “NY” and “restaurant,”

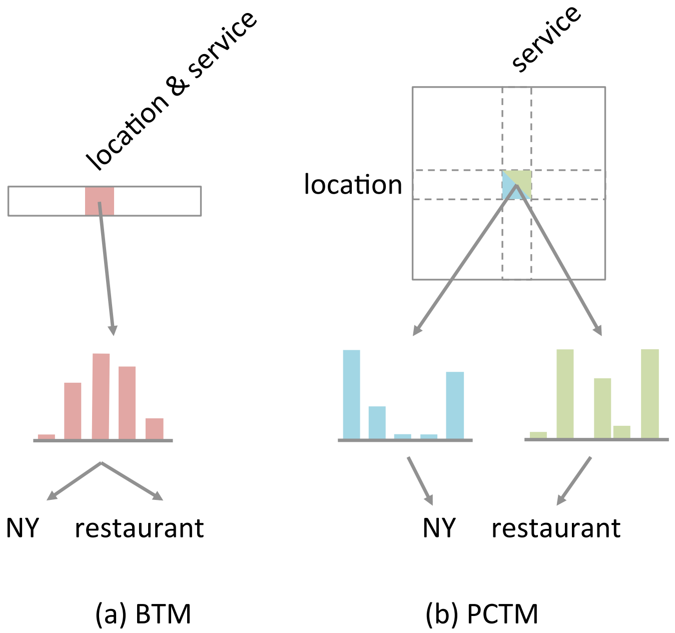

Figure 3.2. Generative processes of (a) BTM and (b) PCTM

have high probabilities in this term distribution, the BTM estimates them as an integrated topic of `service` and `location`. In the PCTM, "NY" and "restaurant" is generated by term distributions associated with different topics. This leads to learn topics that separately indicate `location` with high probability for "NY" and `service` with high probability for "restaurant." Such separated topic representation is more intuitive for humans and is suitable for our task. In Section 3.5.5, we show topics estimated by the BTM and PCTM from real query logs.

The Correlated Topic Model

The correlated topic model (CTM) has been proposed to extract topics and their correlations from documents [8]. The CTM assumes the logistic normal distribution to model the correlation and has been shown to be successful meaningful topic correlation from the academic journal dataset. However, since the CTM also models topic distribution for each document, it would suffer from the sparsity problem and use huge amount of memory space as discussed in LDA. Moreover, while a VB inference algorithm has been proposed, there is no analytical solution because of the non-conjugacy between the logistic normal and the multinomial distributions. Thus, it requires to perform the conjugate gradient method, which consumes more memory space.

The Product Space Mixture Model

Note that the PCTM has the same structure as the product space mixture model (PMM) [21]. Compared to the PMM, we provide the following contributions: (1) we associate the PMM with a fully-dependent topic model (3.2), (2) we demonstrate applicability to query log modeling, and (3) we formulate a fully-Bayesian framework and derive an efficient inference algorithm with CGS rather than an annealed expectation-maximization algorithm [21].

3.5 Experiments

3.5.1 Experimental Setting

We prepared two real query logs: AOL and Yahoo! Japan datasets. The AOL dataset consists of approximately 20 million queries [42]. The Yahoo! Japan dataset consists of approximately 600 million queries sampled from one week search logs. Before conducting our experiments, we performed the following preprocessing. We eliminated queries from the datasets that included low frequency terms or stop words. We then performed random sampling to reduce the data size. This processing allowed us to select queries that occur often in the dataset; however, the datasets were not limited by the query genre. In addition, for the AOL dataset, we performed stemming with the Porter's algorithm[3]. Table 3.5.1 summarizes the preprocessed datasets.

Throughout the experiments, we compared the PCTM with the LDA and BTM. As explained in Section 3.4.3, LDA handles a query as a document. Note that we also attempted to evaluate the CTM with an R implementation[4]; however,

[3] http://tartarus.org/martin/PorterStemmer/

[4] http://cran.r-project.org/web/packages/topicmodels/index.html

Table 3.1. Information about query log datasets. D is the number of documents, V is the number of words in the vocabulary.

	D	V	Average number of terms per query	Language
Yahoo	996K	15K	2.31	Japanese
AOL	583K	14K	2.77	English

this did not work in our experimental environment[5] due to memory overflow. All models were inferred with CGS. We iterated 1,000 samples for the LDA and BTM, and 500 samples for the PCTM. We obtained the last sample in a Markov chain and made use of this sample to estimate parameters, which was required in some experiments. In all experiments and the models, the number of topics was fixed to 100, and β, which is a common hyperparameter, was fixed to 0.1. Since α of LDA is a sensitive parameter [51], we assumed an asymmetric prior and estimated by using Minka's fixed point iteration [34]. For α of the BTM and γ of the PCTM, we assumed a symmetric prior and used fixed values; we set α of the BTM to $\frac{50}{K}$ and γ of the PCTM to $\frac{50}{K^2}$[6] [55].

3.5.2 Human Evaluation of Topic Quality

In our first experiment, we investigated about the interpretability of estimated topics for humans, i.e., how natural and meaningful topics are compared to human knowledge. We conducted three evaluation tasks via crowdsourcing: word intrusion and topic intrusion tasks proposed in [12], and query selection task, which we explain in detail in the following subsections. In these experiments, we used the Yahoo! dataset, and the tasks were performed through using the Yahoo! Japan crowdsourcing service[7]. All jobs contained ten tasks and were assigned to eight different workers [12].

[5] CPU:Intel Corei7-3770 3.40GHz, Memory:16GB RAM

[6] While we did not estimate α of the BTM and γ of the PCTM, these prior effects will be relatively less than that of LDA because topic (topic pair) distribution is shared among all the latent variables.

[7] http://crowdsourcing.yahoo.co.jp/

Word Intrusion

The word intrusion task evaluated the cohesion of a group of terms belonging to the same topic. In this task, several terms that belong to the same topic, except for one term (an intruder), were displayed and, a crowdworker attempted to locate the intruder. The intruder was selected randomly at low probability in the topic, but at high probability in another topic. The other displayed terms were selected with high probability in the topic. Therefore, if the topic was significantly cohesive, the intruder was clearly isolated from the other terms, and the crowdworker could find the intruder easily. We performed this task by varying the number of displayed terms (6 and 8). We used 50 topics, from which a large number of terms were assigned in the last samples of the inference. Performance was measured according to the precision of intruders detection. We compute the fraction of intruder detection by 8 workers for each topic.

The results are shown in Figure 3.3. At first glance, we see that LDA is significantly worse than the BTM and PCTM; actually, there was significant difference[8] between LDA and the others. Although the PCTM was slightly worse than the BTM, the difference was not significant. We discuss the reason why the performances of the BTM and PCTM are nearly the same in Section 3.6.

Topic Intrusion

The topic intrusion task examined how relevant topics were assigned to queries. For each task, we displayed a query and four topics such that three of the topics were the most relevant to the query but the remaining topic was chosen randomly from the top-10 most irrelevant topics. Then, similar to the word intrusion task, a crowdworker attempted to identify the irrelevant topic. Each topic was represented as T terms having the top-T highest probability in the topic. We performed this task with varying T (3, 5, and 8). Note that as the number of displayed terms increased, topics became more identifiable. We used 50 queries in total, which were chosen randomly from the dataset. As the same manner of the word intrusion task, we evaluated the performance by the precision. Note that we excluded LDA from this experiment because the estimated topic distributions

[8] In terms of the one side paired t-test with 95% confidence.

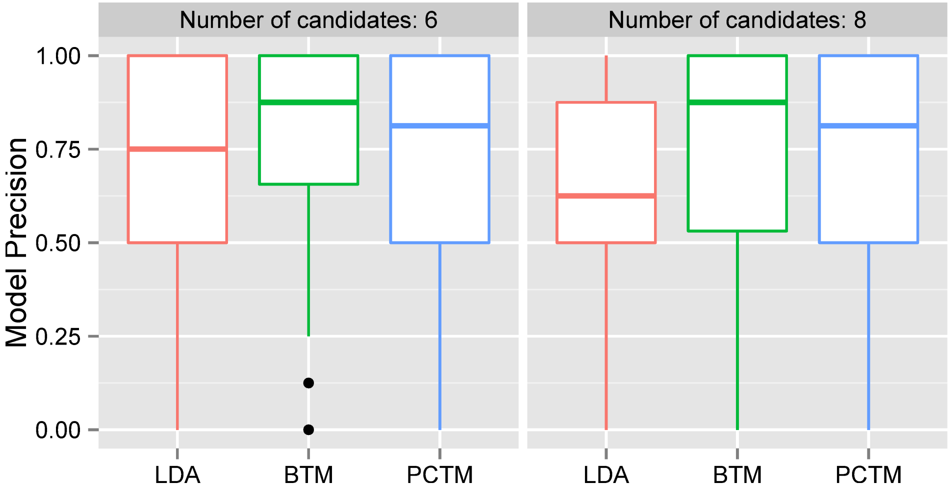

Figure 3.3. Results of the word intrusion task. Each panel shows the results for the number of displayed terms.

were very sparse; thus, we could not select three relevant topics.

For the BTM and PCTM, to collect relevant topics for a given query, we estimate topic probability of the query by taking the average of the topic probability of the term pairs. In the BTM, the probability of topic z_q in an M-long query q is expressed as follows:

$$p(z_q|q)_k = \frac{2}{M(M-1)} \sum_{b_i \in B(q)} p(z_i|b_i)_k, \tag{3.18}$$

where $B(q)$ is the set of term pairs in q and $p(z_i|b_i)$ is the posterior probability of topic z_i given term pair b_i, which can be estimated in the same manner as Eq. (3.15). For the PCTM, we first compute the probability of topic pair x_q in query q:

$$p(x_q|q)_{kl} = \frac{2}{M(M-1)} \sum_{b_i \in B(q)} p(x_i|b_i)_{kl}, \tag{3.19}$$

where $p(x_i|b_i)$ is given by Eq. (3.15). To obtain the probability of individual topic z_q, we then compute the following probability as follows:

$$p(z_q|q)_k = \frac{1}{2} \sum_{l=1}^{K} p(x_q|q)_{kl} + p(x_q|q)_{lk}, \tag{3.20}$$

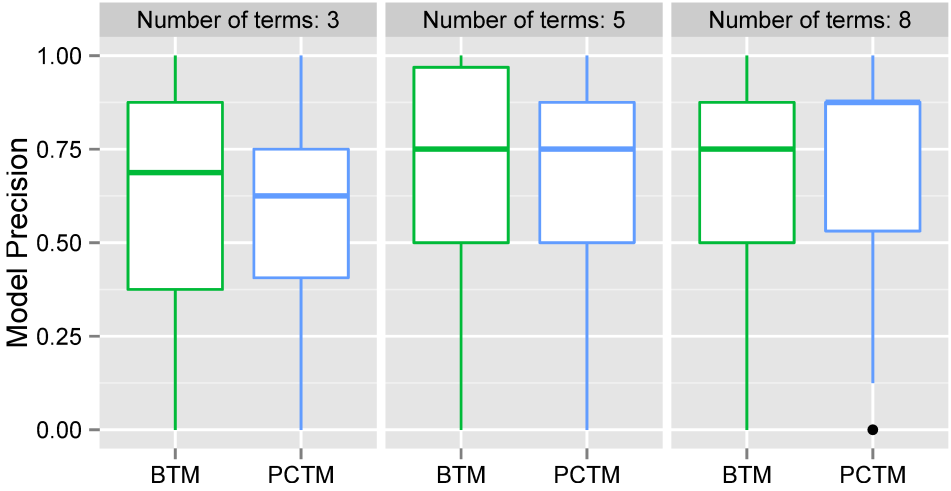

Figure 3.4. Results of the topic intrusion task

i.e., we take the average for each element and its diagonal element and marginalize out topic l.

The results are shown in Figure 3.4. We observe that there was no significant difference[8] between the BTM and PCTM; however, the performances of the PCTM tends to increase as the number of displayed terms increases. This is in contrast to the BTM, in which performance does not change with respect to the number of displayed terms. This implies that the PCTM obtained more coherent topics; thus, crowdworkers could use the increased terms as meaningful information effectively.

Query Selection

The query selection task evaluates how topic models recover a user intent as a query pattern. In each task, we display an actual (target) query and three groups of artificial queries, which are respectively generated by LDA, the BTM, and the PCTM. We let crowdworkers infer the search intent of the target query and then select a group having the most number of queries that have the same intent. For example, we consider a target query as “NY restaurant” in which the intent is assumed to be (`location service`). If a crowdworker infer the intent correctly, a group consisting of (`location service`)-queries, such as “chicago lottery” and “florida hotel”, should be selected.

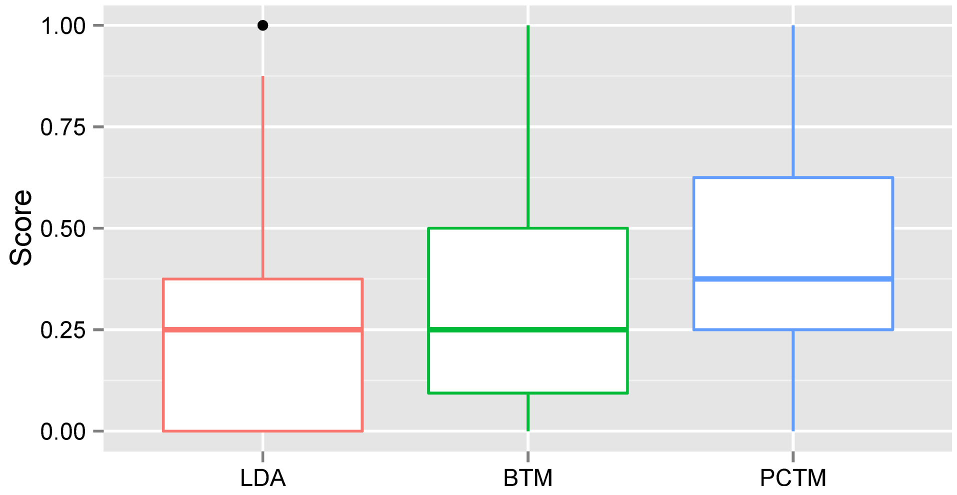

Figure 3.5. Results of the query selection task

We randomly selected 300 2-long queries from the training dataset and used as the target queries. For these 2-long target queries, we estimated the topic pair which takes maximum posterior probability at the last of the inference. For each model, we fixed the length of generated queries to 2 and selected by the following procedure. Given the estimated topic pair in a target query, we compute the posterior probability of 2-long queries, i.e., term pairs. For the PCTM, we computed the probability of term pair $b = (w_1, w_2)$ given topic pair $x = (z_1, z_2)$ and the estimated parameter $\hat{\boldsymbol{\phi}}$ as follows:

$$p(b = (w_1, w_2)|x = (z_1, z_2), \hat{\boldsymbol{\phi}}) = \hat{\phi}_{w_1,z_1}\hat{\phi}_{w_2,z_2}. \tag{3.21}$$

The BTM and LDA computed the probabilities in the same manner. Finally, we used the top-5 term pairs taking the most high probability as the displayed queries. We measured the score which is the fraction of the selected by 8 workers for each target query.

The results are shown in Figure 3.5. The PCTM was significant better[8] than the BTM and LDA. This means that the PCTM could obtain topic pairs where the intents are closer to ones interpreted by humans than the others.

3.5.3 Keyword Recommendation

In this experiment, we evaluated prediction performance in terms of keyword recommendation, i.e., given several terms as a part of a query, we attempted to predict the term that a user will input next. From the whole dataset, we randomly selected 10% of the queries and picked one term for each query as test data. We repeated this procedure 10 times and created training and testing datasets. We randomly initialized the latent variables for each trial. Then, after learning the parameters with the training dataset, we calculated the perplexity for each dataset using the last 100 samples for prediction.

While LDA can perform this experiment simply without modifications, the BTM and PCTM could not. Therefore, we computed the predictive probability for a single term rather than for a term pair. We computed this by assuming that positions of missing terms in the test queries were known in advance, and that the set of term pairs preserved the missing positions. We regarded missing terms as latent variables in the models and performed CGS on these variables in the parameter inference. The sample of the jth term in the ith term pair was obtained by following the conditional distribution in the PCTM:

$$p(w_{i,j} = v | \boldsymbol{b}^{-i,j}, \boldsymbol{x}^{-i}, x_i = (k,l)) = \frac{n^{-i}_{w_{i,j}|k} + \beta}{n^{-i}_{\cdot|k} + V\beta}, \tag{3.22}$$

where $\boldsymbol{b}^{-i,j}$ denotes that the set of term pairs excluding only $w_{i,j}$. The samples of the BTM were obtained in the same manner. After parameter inference, the predictive probability of missing term $w_{i,j}$ was calculated by Eq. (3.22). We then took the average of the terms, which was contained in separate term pairs but was identical in a query[9].

The results are shown in Figure 3.6. For both datasets, the PCTM outperformed the BTM. Indeed, the PCTM was significantly better[8] than the BTM in both datasets.

[9] Compared to the training dataset of LDA, that of the BTM and PCTM includes the position information about the missing terms. Since LDA does not require such information in the traditional way to compute the perplexity, we did not perform the sampling of the missing terms for LDA.

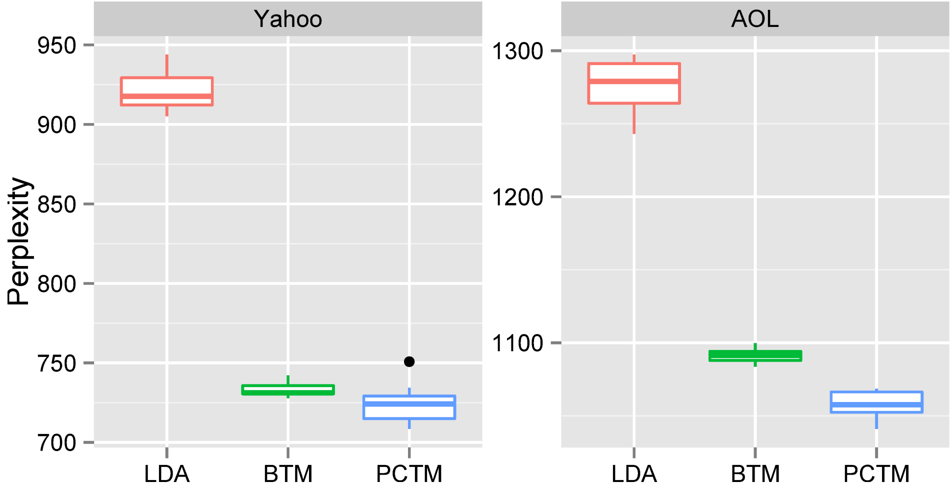

Figure 3.6. Results of the keyword recommendation task. Lower perplexity indicates the better performance.

3.5.4 Query Generation

In this experiment, we investigated the ability of the topic models as query generators, which is important for query recommendation application in Internet advertising. To evaluate the performance, we learned each model by using training data and computed the top-10,000 most frequent 2-long queries. Then, we measured AUC against the test queries, which we randomly selected another 100,000 2-long queries[10]. In addition, we investigated the total number of generated queries detected in the test data. Note that, while we also performed query generation in the query selection task (Section 3.5.2), the evaluation criterion is different in terms of that the query selection takes the user intent into account; this experiment purely evaluate how the topic models can generate natural queries, regardless of the user intents. We computed the probability of term pair $b = (w_1, w_2)$ whose topic pair is marginalized out as follows:

$$p(b = (w_1, w_2)|\hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\psi}}) = \sum_{z_1} \sum_{z_2} \hat{\phi}_{w_1,z_1} \hat{\phi}_{w_2,z_2} \hat{\psi}_{z_1,z_2}. \tag{3.23}$$

The BTM and LDA computed the probability as well as in the query selection task.

[10] We allowed the test data to include queries consisting of duplicate terms, such as “NY NY” and “restaurant restaurant”.

Table 3.2. Results of AUC in the query generation task

	LDA	BTM	PCTM
Yahoo	0.5743	0.5729	0.6080
AOL	0.5634	0.6021	0.6144

Table 3.3. Results of the total number of detected queries in the query generation task

	LDA	BTM	PCTM
Yahoo	645	2,173	3,196
AOL	632	1,437	1,880

The results are shown in Table 3.2 and 3.3. The AUC of the PCTM was higher than the BTM and LDA in both datasets. This result implies that the occurrence probability of a query computed by the PCTM reflects the occurrence of the real query logs. Moreover, the PCTM detected the most queries from test data in both datasets. The result indicates the PCTM has more natural generative process for real query logs than the BTM and LDA.

3.5.5 Estimated Topics and Topic Co-occurrences

Here, we compare the obtained topics as a qualitative evaluation using the AOL dataset. We show the `location` topic in Table 3.4; for each model, we selected a topic such that the most number of terms related to locations in the top-20 topic terms were used. The result shows that the PCTM obtained the most cohesive topic that only contains the terms related to location, while the BTM and LDA topics are corrupted by some non-geographical terms, such as “lottery” and “hotel.” This is a reasonable result for LDA because it counts the topic probability for every document, (i.e., query) and a document is extremely short; thus, terms appearing in the same query were likely to be assigned to the same topic. A similar interpretation holds for the BTM. As described in Section 3.4.3, the BTM mitigates the shortness problem of queries and assumes that term pairs must be assigned to the same topic, which causes contamination of “not-geographical-but-jointly-used” terms, such as “hotel”.

Figure 3.7 shows $\hat{\boldsymbol{\psi}}$, which represents the relationships among topics, i.e., $\hat{\psi}_{kl}$ indicates the probability of how often topics k and l are jointly used in term pairs. Note that in Figure 3.7 the elements are shown in gray where $n_{k,l}$ in Eq. (3.14)

Table 3.4. Estimated topics about location

	Top 20 terms
LDA	lottery state universe florida lotto california ny texas unit ohio map michigan job pa georgia result nj illinois virginia service
BTM	san la vegas diego hotel francisco antonio casino california nevada jose nv mission citi cabo reno valley mexico grand lo
PCTM	florida ny texas san nj ohio chicago nc virginia nyc houston fl va pa lo orlando michigan angel la california

takes 0, which shows that $\hat{\boldsymbol{\psi}}$ is actually very sparse. This indicates that each topic co-occurs with only a few other topics. This result agrees with common sense; people possibly search "NY hotel" but not "lottery hotel."

An unique advantage of the PCTM compared to the LDA and BTM is that we obtain the relationship among topics as $\hat{\boldsymbol{\psi}}$. Figure 3.8 represents relationships among "public" topics, which we see that topic 3 represents public facilities and is associated with topics about location (topic 4), public service (topic 0), and job (topic 1). Figure 3.9 shows "leisure" topics: a topic about service (topic 5) is associated with topics about location names (topic 6), resorts (topic 8), and leisure activities (topic 9). We clearly observe that the PCTM could obtain the network of topics that is very reasonable to our general knowledge. This is a surprising result because the PCTM is a fully unsupervised approach and we did not use any human resources.

3.6 Discussions

3.6.1 The Performance in the Crowdsourcing Tasks

The PCTM did not significantly outperform the BTM in the word and topic intrusion tasks. While the PCTM can obtain separated topic representation as

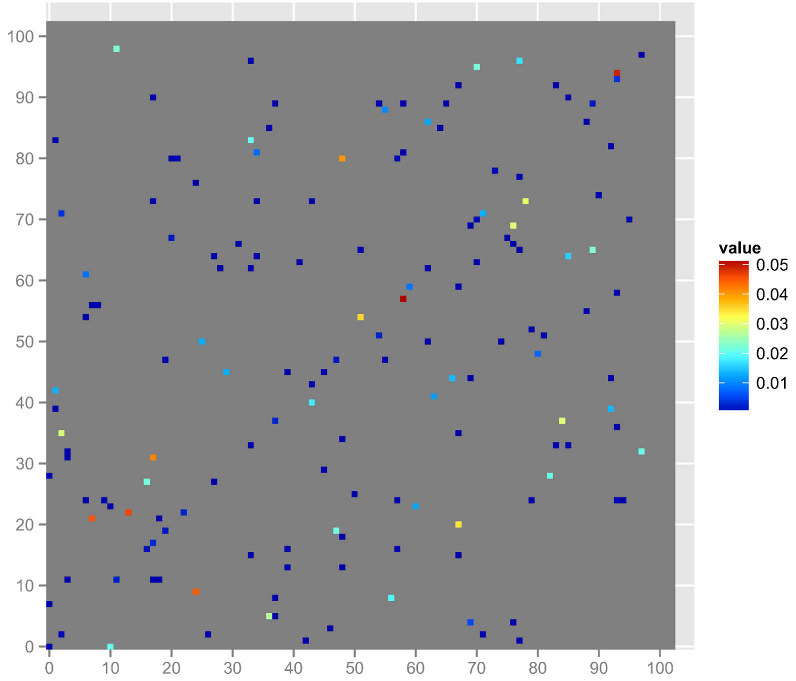

Figure 3.7. Estimated topic co-occurrence probabilities

described in Section 3.4.3, we found that this property did not seem to contribute to the improvement in the both tasks. For example, in the word intrusion task, even though we displayed “hotel” and “florida” included in an integrated topic `location & service` estimated by the BTM, crowdworkers possibly infer that these terms are associated with each other more than the other intruder term, because people frequently use location and service names in web search. Conversely, in the query selection task, the PCTM significantly outperformed the BTM because this task requires the separability of topics for what crowdworkers can recognize query intents.

In contrast to the word and topic intrusion tasks, the PCTM significantly outperformed the LDA and BTM with respect to keyword recommendation, which is a similar characteristic to the CTM; The CTM may select an intruder from highly correlated topics in word intrusion. This results in degenerating the performance because the intruder from the correlated topic confuses crowdworkers [12]. The PCTM seems to have the same problem since it incorporates such correlations between topics.

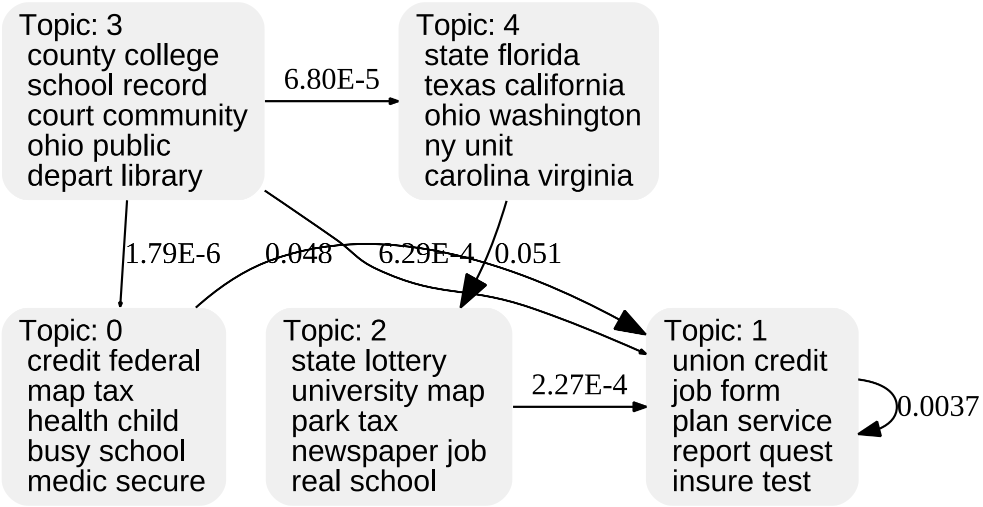

Figure 3.8. A subgraph of topic relationships estimated by the PCTM. We put an edge between the topics if the corresponding entry of $\hat{\boldsymbol{\psi}}$ has larger probability than $\frac{\gamma}{N+K^2\gamma}$. For each topic, the top-10 terms are displayed. For each edge, we show the value of $\hat{\boldsymbol{\psi}}$ and the larger value indicates the strong connectivity.

3.6.2 Further Reduction of the Computational Cost

The issue of the PCTM is the computational cost. CGS for the PCTM (Section 3.3.4) requires $O(NK^2)$ time complexity. On the other hand, the LDA and BTM require $O(DLK)$ and $O(NK)$ complexity, respectively. When comparing to LDA, the difference between K and K^2 is dominant because $O(DL)$ of LDA is comparable to the $O(N)$ of the PCTM in real query dataset. Although this issue didn't matter in our used datasets, this will become more problematic in real application.

One promising approach to reduce the complexity is to use a property that the PCTM learns sparse $\hat{\boldsymbol{\psi}}$. As described in Figure 3.7, the number of active topic pairs is considerably smaller than all possible number of them. This implies that current CGS checks many meaningless topic pairs. Interestingly, according to our preliminary experiment, the active number of topic pairs is constant regardless of K. By using this property, we consider an approximate sampling algorithm that skips unnecessary computations in the sampling procedure as follows. In CGS of

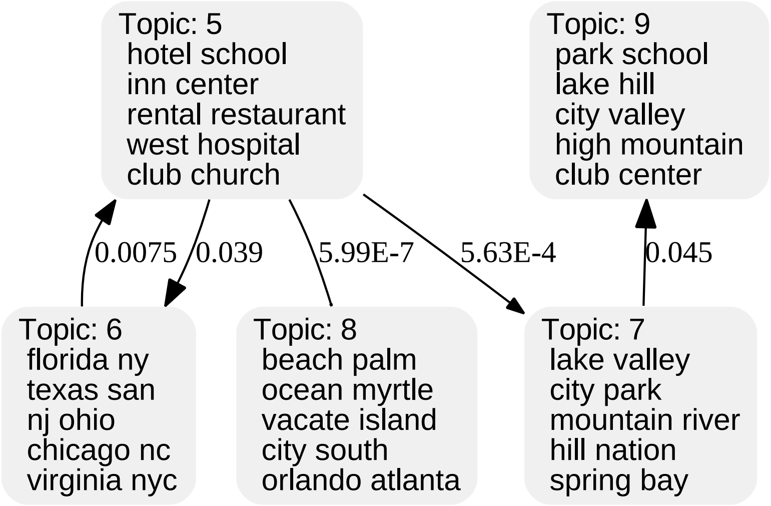

Figure 3.9. Another subgraph of topic relationships estimated by the PCTM

the PCTM, we must check all possible assignments of topic pairs for each term pair; thus, the PCTM requires $O(K^2)$ complexity to obtain one sample to a topic pair. We divide this sampling procedure into the following two steps: (1) we first sample a binary variable to determine which active or non-active topic pairs to sample in accordance with the current proportion of the active/non-active topic pairs, and then (2) the topic pair is sampled from only among active/non-active topic pairs. This computational cost amounts to $O(pL+(1-p)(K^2-L))$, where p is the proportion of the active topic pairs, and A is the number of current active topic pairs. This computation requires $O(K^2)$ complexity in general; however, if A is at most $O(K)$ and $1-p$ is less than $O(\frac{1}{K})$, the complexity is reduced from $O(K^2)$ to $O((1-\frac{1}{K})K + \frac{1}{K}(K^2-K)) = O(K)$. While this approximated sampling is not guaranteed to converge to the stationary distribution and it would not be suitable for accurate prediction tasks such as computing the perplexity, we empirically confirmed the quality of obtained topics are almost the same as the exact approach.

3.7 Summary

We have proposed a new probabilistic topic model for query logs. The PCTM can capture topic co-occurrences in a query, which make topics more coherent without limiting the target domain of queries. For model learning, we derived a fully-Bayesian inference algorithm with collapsed Gibbs sampling. We have examined three types of experiments, i.e., crowdsourcing, keyword recommendation, and query generation tasks.

Chapter 4

VB Inference of the IRM for Network Data

4.1 Introduction

Network data are the most basic relational data that consist of only one kind of objects. These networks appear in many situations such as social networks, hyperlinks on Web pages, citation networks, gene interactions, and brain region interactions. To obtain knowledges from networks, there are following two tasks. One is to analyze latent structures behind relations. Revealing these structures gives us summarized representations of complex interactions between objects. The other is to predict missing relationships from observations. Relation prediction methods can be widely used in practical applications such as friend recommender systems in social networking services. To face these tasks, various statistical models have contributed. As an early work, Nowicki and Snijders have proposed the stochastic block model (SBM), which is a basic probabilistic model for relational data including networks [38]. Compared to other graph clustering algorithms such as The SBM is general framework for relational data such as spectral clustering.

As a Bayesian nonparametric extension of the SBM, Kemp *et al.* proposed the IRM [29]. The IRM identifies cluster structures behind networks, and automatically determines the number of clusters. It can be used in various relational data, e.g., analyzing the functional connectivity of neural elements [36]. The IRM can be also extended as incorporating side information such as time-series information [22], and more sophisticated models [10,23]. One of the advantages of the IRM is less computational complexity compared to other Bayesian nonparametric relational models [32,33].

To compute the posterior distribution for probabilistic models such as the IRM, MCMC methods are used. MCMC methods approximate the posterior distribution by multiple samples drawn from stationary distribution, and a practical choice for the IRM is to use CGS. However, in principle, MCMC methods need to assess convergence and to identify coherent latent variables across multiple samples [49].

Another choice for the posterior inference is the VB inference methods, which we focus on in this work. The VB inference methods are deterministic algorithms that transform the inference problem into an optimization problem with some approximations for computational tractability [4,5,28]. The methods maximize a lower bound of the log marginal likelihood of the model, and make the lower bound converge to local optima by iteration. We can easily diagnose the convergence by monitoring this lower bound. The VB inference methods have been incorporating more efficient algorithms and dealing with wider problems, e.g., the online learning setting [19,45].

As a special case of the VB inference, Teh *et al.* have proposed the CVB inference [49]. The idea is to relax the assumption of the mean-field approximation by marginalizing parameters, and can find better local optima than the standard VB inference. The CVB inference has been applied to some statistical models, and its efficiency has been reported [49,52].

In this Chapter, we focus on the IRM of network data, and derive the CVB inference algorithm and its variant called the CVB0 inference algorithm [3]. We validated the performance of these algorithms through six real network datasets. In the experiment, the CVB inference outperformed the VB one in many datasets, and the tendencies remarkably appeared in dense networks.

4.2 The IRM for Network Data

The IRM is a probabilistic model for general relational data such as purchasing histories and user rating data [29]. Relational data are specified by objects and observed relations. The IRM models the cluster structures of the objects based on the observed relations. For example, user rating data consist of two kinds of objects: users and items, and a observed relation indicates a rating of a user to a item. In this case, the IRM assigns cluster indexes to both users and items based on the observed ratings. Note that the IRM has slightly different generative process according to observations. For modeling user ratings, the Poisson distribution is used for generating the ratings as non-negative integers, while the Bernoulli distribution is used for generating binary logs in purchasing histories.

In this thesis we consider the IRM for network data that consist of one kind of objects. The objects correspond to the nodes in the networks such as humans in social networks and proteins in protein-protein interactions. The observed relations are represented by the edges that are relations between objects and this thesis considers undirected binary observations, e.g., whether links or not among humans in social networks. Given the observations, the IRM assigns a cluster index for each object according to the generative process. Intuitively, in the IRM, the objects assigned to the same cluster tend to have similar links to other objects. For example, if objects A and B link to the same objects C, D, and E, A is similar to B in the sense that both objects have links to the same objects in the network. As a result, A and B tend to the same cluster, and otherwise C, D, E also tend to the same cluster.

We describe the generative model of the IRM with the stick-breaking process (SBP) representation [46]. Let us assume that we observe network data consisting of N objects without self-link observations. Firstly, the set of link probabilities $\boldsymbol{\eta} = \{\eta_{kl}\}_{k=1,l=k}^{\infty,\infty}$ are generated as

$$\eta_{kl} \sim \mathrm{Beta}(\alpha, \beta), \tag{4.1}$$

where η_{kl} denotes the link probability between cluster k and cluster l. $\mathrm{Beta}(\cdot)$ is the Beta distribution, and α and β are its parameters. Note that because η_{kl} is identical to η_{lk} for undirected networks, we unify these variables as one, and use only η_{kl} whose subscript index $k \leq l$. Next discrete hidden variables $\boldsymbol{z} = \{\boldsymbol{z}_i\}_{i=1}^{N}$

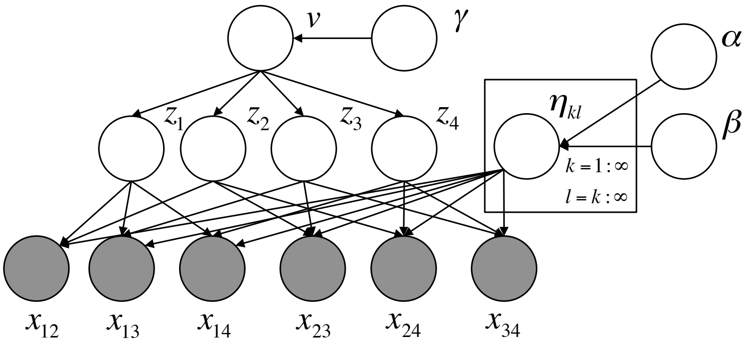

Figure 4.1. Graphical model of the IRM for network data when $N = 4$.

are drawn from a multinomial distribution with a vector $\boldsymbol{\pi}$,

$$\boldsymbol{z}_i \sim \boldsymbol{\pi}, \tag{4.2}$$

where $\boldsymbol{z}_i$ represents a vector of object i where only one element corresponding to a cluster is 1 and the others 0. $\boldsymbol{\pi}$ is generated by the stick-breaking process,

$$\pi_k = v_k \prod_{m=1}^{k-1}(1 - v_m),\ v_k \sim \text{Beta}(1, \gamma) \quad (k = 1, ...), \tag{4.3}$$

where γ is the concentration parameter. At last observations $\boldsymbol{x} = \{x_{ij}\}_{i=1,j=i+1}^{N,N}$ are generated as

$$x_{ij} \sim \text{Bern}(\eta_{z_i,z_j}), \tag{4.4}$$

where η_{z_i,z_j} is a parameter corresponding to the cluster assignments of object i and j, and x_{ij} is a binary variable, which means whether object i and j link or not. Bern$(\cdot)$ denotes the Bernoulli distribution. In undirected networks, we only use x_{ij} whose subscript index $i < j$. Therefore the total number of observations is $N(N-1)/2$. In Fig. 4.1, we instantiate the graphical model when $N = 4$. The dependencies between $\boldsymbol{x}$ and $\boldsymbol{z}$ prohibit a simple plate notation.

4.3 The VB Inference Algorithms of the IRM

We derive three VB inference algorithms of the IRM presented in Section 4.2. We describe update equations for the standard VB inference, the CVB inference, and the CVB0 inference.

4.3.1 The VB Inference of the IRM

For the tractable VB inference, we replace the stick-breaking process (4.3) with the truncated stick-breaking process [7],

$$\pi_k = v_k \prod_{m=1}^{k-1} (1 - v_m),\ v_k \sim \mathrm{Beta}(1, \gamma),\ \ v_T = 1. \tag{4.5}$$

T is the truncation level of this process, and does not necessarily mean the number of clusters. If T is larger than effective number of clusters K, some clusters will remain not used, and this process approximates the original stick-breaking process. By using this representation, the joint distribution of the IRM is written as

$$\begin{aligned}
&p(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{\eta}, \boldsymbol{v}|\alpha, \beta, \gamma) \\
&= p(\boldsymbol{x}|\boldsymbol{z}, \boldsymbol{\eta}) p(\boldsymbol{z}|\boldsymbol{v}) p(\boldsymbol{\eta}|\alpha, \beta) p(\boldsymbol{v}|\gamma) \\
&= \prod_{i=1}^{N-1} \prod_{j=i+1}^{N} p(x_{ij}|\boldsymbol{z}_i, \boldsymbol{z}_j, \boldsymbol{\eta}) \prod_{i=1}^{N} p(\boldsymbol{z}_i|\boldsymbol{v}) \prod_{k=1}^{T} \prod_{l=k}^{T} p(\eta_{kl}|\alpha, \beta) \prod_{k=1}^{T-1} p(v_k|\gamma).
\end{aligned} \tag{4.6}$$

Each distribution is as follows:

$$\begin{aligned}
p(x_{ij}|\boldsymbol{z}_i, \boldsymbol{z}_j, \boldsymbol{\eta}) &= \prod_{k=1}^{T} \prod_{l=k+1}^{T} \left\{ {\eta_{kl}}^{x_{ij}} (1 - \eta_{kl})^{(1-x_{ij})} \right\}^{(z_{ik} z_{jl} + z_{il} z_{jk})} \\
&\qquad \left\{ {\eta_{kk}}^{x_{ij}} (1 - \eta_{kk})^{(1-x_{ij})} \right\}^{z_{ik} z_{jk}}, && (4.7) \\
p(\boldsymbol{z}_i|\boldsymbol{v}) &= \prod_{k=1}^{T} {\pi_k}^{z_{ik}} = \prod_{k=1}^{T} \left\{ v_k \prod_{m=1}^{k-1} (1 - v_m) \right\}^{z_{ik}}, && (4.8) \\
p(\eta_{kl}|\alpha, \beta) &= \frac{1}{\mathrm{B}(\alpha, \beta)} {\eta_{kl}}^{\alpha-1} (1 - \eta_{kl})^{\beta-1}, && (4.9) \\
p(v_k|\gamma) &= \frac{1}{\mathrm{B}(1, \gamma)} {v_k}^{1-1} (1 - v_k)^{\gamma-1}, && (4.10)
\end{aligned}$$

where B($\cdot$) is the Beta function. By Jensen's inequality, we obtain a lower bound of the marginal likelihood for the IRM:

$$
\begin{aligned}
\log p(\boldsymbol{x}|\alpha,\beta,\gamma) &= \log \int \sum_{\boldsymbol{z}} p(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v}|\alpha,\beta,\gamma) \\
&= \log \int \sum_{\boldsymbol{z}} q(\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v}) \frac{p(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v}|\alpha,\beta,\gamma)}{q(\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v})} \\
&= \log \mathbb{E}_q \left[\frac{p(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v}|\alpha,\beta,\gamma)}{q(\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v})} \right] \\
&\geq \mathbb{E}_q \left[\log \frac{p(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v}|\alpha,\beta,\gamma)}{q(\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v})} \right] \\
&\equiv \mathcal{L}[q(\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v})]. \qquad (4.11)
\end{aligned}
$$

where $q(\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v})$ is the variational posterior distribution. This is defined as

$$
q(\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v}) = \prod_{i=1}^{N} q(\boldsymbol{z}_i|\boldsymbol{\phi}_i) \prod_{k=1}^{T} \prod_{l=k}^{T} q(\eta_{kl}|\mu_{kl},\nu_{kl}) \prod_{k=1}^{T-1} q(v_k|\kappa_k,\lambda_k). \qquad (4.12)
$$

We assume the mean-field approximation which limits the variational posterior to fully factorized distributions. $q(\boldsymbol{z}_i|\boldsymbol{\phi}_i)$ follows the multinomial distribution, and $q(\eta_{kl}|\mu_{kl},\nu_{kl})$ and $q(v_k|\kappa_k,\lambda_k)$ follow the Beta distribution. $\boldsymbol{\phi} = \{\boldsymbol{\phi}_i\}_{i=1}^{N}$, $\boldsymbol{\mu} = \{\mu_{kl}\}_{k=1,l=k}^{T,T}$, $\boldsymbol{\nu} = \{\nu_{kl}\}_{k=1,l=k}^{T,T}$, $\boldsymbol{\kappa} = \{\kappa_k\}_{k=1}^{T-1}$, and $\boldsymbol{\lambda} = \{\lambda_k\}_{k=1}^{T-1}$ are variational parameters. By optimizing these parameters, the lower bound approaches the log marginal likelihood. It is equivalent to minimizing KL divergence from the variational posterior to the true posterior KL($q(\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v})|p(\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v}|\boldsymbol{x})$). The lower bound $\mathcal{L}[q(\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v})]$ consists of following five expectation terms:

$$
\begin{aligned}
\mathcal{L}[q(\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v})] &= \mathbb{E}_q \left[\log p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\eta})\right] + \mathbb{E}_q \left[\log p(\boldsymbol{z}|\boldsymbol{v})\right] \\
&\quad + \mathbb{E}_q \left[\log p(\boldsymbol{\eta}|,\alpha,\beta)\right] + \mathbb{E}_q \left[\log p(\boldsymbol{v}|\gamma)\right] - \mathbb{E}_q \left[\log q(\boldsymbol{z},\boldsymbol{v},\boldsymbol{v})\right]. \qquad (4.13)
\end{aligned}
$$

The detail of each term is shown in Appendix A. By maximizing the lower bound (4.13) with respect to the parameters of the variational posterior (4.12), the update equations of the VB inference algorithm are derived as follows:

$$\begin{aligned}\phi_{ik} \propto \exp\Bigg(& \sum_{l=k+1}^{T} \left\{ n_{il}\psi\left(\mu_{kl}\right) + \overline{n}_{il}\psi\left(\nu_{kl}\right) - \left(n_{il} + \overline{n}_{il}\right)\psi\left(\mu_{kl} + \nu_{kl}\right) \right\} \\ & + \sum_{l=1}^{k} \left\{ n_{il}\psi\left(\mu_{lk}\right) + \overline{n}_{il}\psi\left(\nu_{lk}\right) - \left(n_{il} + \overline{n}_{il}\right)\psi\left(\mu_{lk} + \nu_{lk}\right) \right\} \\ & + \psi\left(\kappa_k\right) - \psi\left(\kappa_k + \lambda_k\right) + \sum_{m=1}^{k-1} \left\{ \psi\left(\lambda_m\right) - \psi\left(\kappa_m + \lambda_m\right) \right\} \Bigg), \end{aligned} \tag{4.14}$$

$$\mu_{kl} = \begin{cases} \alpha + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left(\phi_{ik}\phi_{jl} + \phi_{il}\phi_{jk}\right) x_{ij} & (l \neq k), \\ \alpha + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \phi_{ik}\phi_{jk} x_{ij} & (l = k), \end{cases} \tag{4.15}$$

$$\nu_{kl} = \begin{cases} \beta + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \left(\phi_{ik}\phi_{jl} + \phi_{il}\phi_{jk}\right) \left(1 - x_{ij}\right) & (l \neq k), \\ \beta + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \phi_{ik}\phi_{jk}(1 - x_{ij}) & (l = k), \end{cases} \tag{4.16}$$

$$\kappa_k = 1 + \sum_{i=1}^{N} \phi_{ik}, \tag{4.17}$$

$$\lambda_k = \gamma + \sum_{i=1}^{N} \sum_{m=k+1}^{T} \phi_{im}, \tag{4.18}$$

where

$$n_{il} = \sum_{j=i+1}^{N} \phi_{jl} x_{ij} + \sum_{j=1}^{i-1} \phi_{jl} x_{ji}, \tag{4.19}$$

$$\overline{n}_{il} = \sum_{j=i+1}^{N} \phi_{jl}(1 - x_{ij}) + \sum_{j=1}^{i-1} \phi_{jl}(1 - x_{ji}), \tag{4.20}$$

and $\psi(\cdot)$ represents the digamma function. Note that the term $\psi\left(\kappa_k\right) - \psi\left(\kappa_k + \lambda_k\right)$ in (4.14) does not appear in the update of ϕ_{iT}. The algorithm repeats updating equations alternately until converging to a local maximum.

We estimate the hyperparameters α and β according to Minka's fixed point iteration method [34]. We also optimize γ by maximizing the lower bound with respect to it.

4.3.2 The CVB Inference of the IRM

Teh *et al.* proposed the CVB inference for LDA [49]. Kurihara *et al.* also proposed the CVB inference of the Dirichlet process mixture model [30]. On the basis of these works, the CVB inference of the IRM are derived by marginalizing out the parameters $\boldsymbol{\eta}$ and $\boldsymbol{v}$. The joint distribution after marginalizing out $\boldsymbol{\eta}$ and $\boldsymbol{v}$ is

$$\begin{aligned}
p(\boldsymbol{x},\boldsymbol{z}|\alpha,\beta,\gamma) &= \int p(\boldsymbol{x},\boldsymbol{z},\boldsymbol{\eta},\boldsymbol{v}|\alpha,\beta,\gamma)d\boldsymbol{\eta}d\boldsymbol{v} \\
&= \int p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\eta})p(\boldsymbol{\eta}|\alpha,\beta)d\boldsymbol{\eta}\int p(\boldsymbol{z}|\boldsymbol{v})p(\boldsymbol{v}|\gamma)d\boldsymbol{v} \\
&= \prod_{k=1}^{T}\prod_{l=k}^{T}\frac{1}{\mathrm{B}(\alpha,\beta)}\int \eta_{kl}^{n_{kl}+\alpha-1}(1-\eta_{kl}^{\overline{n}_{kl}+\beta-1})d\eta_{kl} \\
&\quad \prod_{k=1}^{T-1}\frac{1}{\mathrm{B}(1,\gamma)}\int v_k^{n_k+1-1}(1-v_k^{n_{>k}+\gamma-1})dv_k \\
&= \prod_{k=1}^{T}\prod_{l=k}^{T}\frac{\mathrm{B}\left(n_{kl}+\alpha,\overline{n}_{kl}+\beta\right)}{\mathrm{B}\left(\alpha,\beta\right)}\prod_{k=1}^{T-1}\gamma\frac{\Gamma\left(n_k+1\right)\Gamma\left(n_{>k}+\gamma\right)}{\Gamma\left(n_{\geq k}+\gamma+1\right)}, \quad (4.21)
\end{aligned}$$

where

$$n_{kl} = \begin{cases} \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left(z_{ik}z_{il}+z_{il}z_{jk}\right)x_{ij} & (l \neq k), \\ \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} z_{ik}z_{jk}x_{ij} & (l = k), \end{cases}$$

$$\overline{n}_{kl} = \begin{cases} \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\left(z_{ik}z_{jl}+z_{il}z_{jk}\right)\left(1-x_{ij}\right) & (l \neq k), \\ \sum_{i=1}^{N-1}\sum_{j=i+1}^{N} z_{ik}z_{jk}(1-x_{ij}) & (l = k), \end{cases}$$

$$
\begin{aligned}
n_k &= \sum_{i=1}^{N} z_{ik}, \\
n_{>k} &= \sum_{i=1}^{N} \sum_{m=k+1}^{T} z_{im}, \\
n_{\geq k} &= n_k + n_{>k}.
\end{aligned}
$$

In this derivation, we used a property of the Beta distribution: the integrals are analytically computed as a product of the Gamma function. The conditional distribution $p(z_{ik} = 1|\boldsymbol{x}, \boldsymbol{z}^{-i}, \alpha, \beta, \gamma)$ is also written as follows:

$$
\begin{aligned}
p(z_{ik} = 1|\boldsymbol{x}, \boldsymbol{z}^{-i}, \alpha, \beta, \gamma) \propto \prod_{l=k+1}^{T} \frac{\mathrm{B}\left(n_{kl} + \alpha, \overline{n}_{kl} + \beta\right)}{\mathrm{B}\left(n_{kl}^{-i} + \alpha, \overline{n}_{kl}^{-i} + \beta\right)} \prod_{l=1}^{k} \frac{\mathrm{B}\left(n_{lk} + \alpha, \overline{n}_{lk} + \beta\right)}{\mathrm{B}\left(n_{lk}^{-i} + \alpha, \overline{n}_{lk}^{-i} + \beta\right)} \\
\frac{n_k^{-i} + 1}{n_{\geq k} + \gamma + 1} \prod_{m=1}^{k-1} \frac{n_{>m}^{-i} + \gamma}{n_{\geq m} + \gamma + 1}. \qquad (4.22)
\end{aligned}
$$

The superscripts "$-i$" such as n_{kl}^{-i} mean to exclude the counts with respect to z_i. In the CVB inference, we consider a lower bound of the marginal likelihood after integrating out $\boldsymbol{\eta}$ and $\boldsymbol{v}$:

$$
\begin{aligned}
\log p(\boldsymbol{x}|\alpha, \beta, \gamma) &= \log \sum_{\boldsymbol{z}} p(\boldsymbol{x}, \boldsymbol{z}|\alpha, \beta, \gamma) \\
&= \log \sum_{\boldsymbol{z}} q(\boldsymbol{z}) \frac{p(\boldsymbol{x}, \boldsymbol{z}|\alpha, \beta, \gamma)}{q(\boldsymbol{z})} \\
&= \log \mathbb{E}_q \left[\frac{p(\boldsymbol{x}, \boldsymbol{z}|\alpha, \beta, \gamma)}{q(\boldsymbol{z})} \right] \\
&\geq \mathbb{E}_q \left[\log \frac{p(\boldsymbol{x}, \boldsymbol{z}|\alpha, \beta, \gamma)}{q(\boldsymbol{z})} \right] \\
&\equiv \hat{\mathcal{L}}[q(\boldsymbol{z})], \qquad (4.23)
\end{aligned}
$$

where $q(\boldsymbol{z})$ is the variational posterior distribution in the CVB inference. This is defined as

$$
q(\boldsymbol{z}) = \prod_{i=1}^{N} q(\boldsymbol{z}_i|\boldsymbol{\phi}_i). \qquad (4.24)
$$

The variational posterior only assumes mean-field approximation to $\boldsymbol{z}$. This relaxation of independence assumption allows the variational posterior to get closer to the true posterior. We maximize the lower bound (4.23) with respect to variational parameters $\boldsymbol{\phi} = \{\boldsymbol{\phi}_i\}_{i=1}^N$ by using the Lagrange multipliers method. When maximizing ϕ_{ik}, the Lagrangian function $L_{\phi_{ik}}$ is defined as follows:

$$L_{\phi_{ik}} = \phi_{ik}\mathbb{E}_{q(\boldsymbol{z}^{-i})}\left[p(\boldsymbol{x}, z_{ik} = 1, \boldsymbol{z}^{-i}|\alpha, \beta, \gamma)\right] - \phi_{ik}\log\phi_{ik} - \rho\left(\sum_{k'=1}^{T}\phi_{ik'} - 1\right), \tag{4.25}$$

where ρ is the Lagrangian multiplier. we set this to zero, and obtain the following update equation:

$$\begin{aligned}\phi_{ik} &\propto \exp\left(\mathbb{E}_{q(\boldsymbol{z}^{-i})}\left[\log p(z_{ik} = 1|\boldsymbol{x}, \boldsymbol{z}^{-i}, \alpha, \beta, \gamma)\right]\right)\\ &\propto \exp\left(\mathbb{E}_{q(\boldsymbol{z}^{-i})}\left[\sum_{l=k+1}^{T}\log\frac{\Gamma\left(n_{kl}+\alpha\right)}{\Gamma\left(n_{kl}^{-i}+\alpha\right)}\frac{\Gamma\left(\overline{n}_{kl}+\beta\right)}{\Gamma\left(\overline{n}_{kl}^{-i}+\beta\right)}\frac{\Gamma\left(n_{kl}^{-i}+\overline{n}_{kl}^{-i}+\alpha+\beta\right)}{\Gamma\left(n_{kl}+\overline{n}_{kl}+\alpha+\beta\right)}\right.\right.\\ &\qquad + \sum_{l=1}^{k}\log\frac{\Gamma\left(n_{lk}+\alpha\right)}{\Gamma\left(n_{lk}^{-i}+\alpha\right)}\frac{\Gamma\left(\overline{n}_{lk}+\beta\right)}{\Gamma\left(\overline{n}_{lk}^{-i}+\beta\right)}\frac{\Gamma\left(n_{lk}^{-i}+\overline{n}_{lk}^{-i}+\alpha+\beta\right)}{\Gamma\left(n_{lk}+\overline{n}_{lk}+\alpha+\beta\right)}\\ &\qquad \left.\left. + \log\frac{n_k^{-i}+1}{n_{\geq k}^{-i}+\gamma+1} + \sum_{m=1}^{k-1}\log\frac{n_{>m}^{-i}+\gamma}{n_{\geq m}^{-i}+\gamma+1}\right]\right).\end{aligned} \tag{4.26}$$

Although (4.26) is intractable because of the expensive computations of expectation terms, we can use the Taylor series approximation. In general, the expectation of function f is approximated by

$$\begin{aligned}\mathbb{E}\left[f(x)\right] &\simeq \mathbb{E}\left[f(\mathbb{E}\left[x\right])\right] + \mathbb{E}\left[f^{'}(\mathbb{E}\left[x\right])(x - \mathbb{E}\left[x\right])\right] + \mathbb{E}\left[\frac{1}{2}f^{''}(\mathbb{E}\left[x\right])(x - \mathbb{E}\left[x\right])^2\right]\\ &= f(\mathbb{E}\left[x\right]) + \frac{1}{2}f^{''}(\mathbb{E}\left[x\right])\mathrm{var}\left[x\right].\end{aligned} \tag{4.27}$$

Note that the first term vanishes. By using this approximation, the above expectation terms are computed like the following manners:

$$\begin{aligned}\mathbb{E}_q\left[\log\Gamma\left(n_{kl}^{-i}+\alpha\right)\right] &\simeq \log\Gamma\left(\mathbb{E}_q\left[n_{kl}^{-i}\right]+\alpha\right) + \frac{1}{2}\psi^{(1)}\left(\mathbb{E}_q\left[n_{kl}^{-i}\right]+\alpha\right)\mathrm{var}_q\left[n_{kl}^{-i}\right]\\ &\equiv \mathrm{a}\left(n_{kl}^{-i}, \alpha\right),\end{aligned} \tag{4.28}$$

$$\begin{aligned}\mathbb{E}_q\left[\log\left(n_k^{-i}+1\right)\right] &\simeq \log\left(\mathbb{E}_q\left[n_k^{-i}\right]+1\right) - \frac{\mathrm{var}_q\left[n_k^{-i}\right]}{2\left(n_k^{-i}+1\right)^2}\\ &\equiv \mathrm{b}\left(n_k^{-i}, 1\right),\end{aligned} \tag{4.29}$$

where $\psi^{(1)}(\cdot)$ represents the trigamma function. Expected/variance counts with the superscript $-i$ are

$$\mathbb{E}_q\left[n_{kl}^{-i}\right] = \begin{cases} \displaystyle\sum_{\substack{i'=1\\i'\neq i}}^{N-1}\sum_{\substack{j'=i'+1\\j'\neq i}}^{N}\left(\phi_{i'k}\phi_{j'l}+\phi_{i'l}\phi_{j'k}\right)x_{i'j'} & (l\neq k),\\ \displaystyle\sum_{\substack{i'=1\\i'\neq i}}^{N-1}\sum_{\substack{j'=i'+1\\j'\neq i}}^{N}\phi_{i'k}\phi_{j'k}x_{i'j'} & (l=k),\end{cases} \tag{4.30}$$

$$\mathrm{var}_q\left[n_{kl}^{-i}\right] = \begin{cases} \displaystyle\sum_{\substack{i'=1\\i'\neq i}}^{N-1}\sum_{\substack{j'=i'+1\\j'\neq i}}^{N}\left\{\phi_{i'k}\phi_{j'l}\left(1-\phi_{i'k}\phi_{j'l}\right)x_{i'j'}+\phi_{i'l}\phi_{j'k}\left(1-\phi_{i'l}\phi_{j'k}\right)x_{i'j'}\right\} & (l\neq k),\\ \displaystyle\sum_{\substack{i'=1\\i'\neq i}}^{N-1}\sum_{\substack{j'=i'+1\\j'\neq i}}^{N}\phi_{i'k}\phi_{j'k}\left(1-\phi_{i'k}\phi_{j'k}\right)x_{i'j'} & (l=k),\end{cases} \tag{4.31}$$

$$\mathbb{E}_q\left[n_{k}^{-i}\right] = \sum_{\substack{i'=1\\i'\neq i}}^{N}\phi_{i'k}, \tag{4.32}$$

$$\mathrm{var}_q\left[n_{k}^{-i}\right] = \sum_{\substack{i'=1\\i'\neq i}}^{N}\phi_{i'k}\left(1-\phi_{i'k}\right). \tag{4.33}$$

Because $z_{ik}=1$ and $z_{i\cdot}=0$ $(\cdot\neq k)$, the following counts are written as

$$\mathbb{E}_q\left[n_{kl}\right] = \mathbb{E}_q\left[n_{kl}^{-i}\right]+\sum_{j'=i+1}^{N}\phi_{j'l}x_{ij'}+\sum_{i'=1}^{i-1}\phi_{i'l}x_{i'i}, \tag{4.34}$$

$$\mathrm{var}_q\left[n_{kl}\right] = \mathrm{var}_q\left[n_{kl}^{-i}\right]+\sum_{j'=i+1}^{N}\phi_{j'l}(1-\phi_{j'l})x_{ij'}+\sum_{i'=1}^{i-1}\phi_{i'l}(1-\phi_{i'l})x_{i'i}. \tag{4.35}$$

Other counts are listed in Appendix B. By using these, equation (4.26) is approximated as follows:

$$
\begin{aligned}
(4.26) \simeq \exp\Bigg(& \sum_{l=k+1}^{T} \Big\{ \mathrm{a}\left(n_{kl}, \alpha\right) - \mathrm{a}\left(n_{kl}^{-i}, \alpha\right) + \mathrm{a}\left(\overline{n}_{kl}, \beta\right) - \mathrm{a}\left(\overline{n}_{kl}^{-i}, \beta\right) \\
& \qquad - \mathrm{a}\left(n_{kl} + \overline{n}_{kl}, \alpha + \beta\right) + \mathrm{a}\left(n_{kl}^{-i} + \overline{n}_{kl}^{-i}, \alpha + \beta\right) \Big\} \\
& + \sum_{l=1}^{k} \Big\{ \mathrm{a}\left(n_{lk}, \alpha\right) - \mathrm{a}\left(n_{lk}^{-i}, \alpha\right) + \mathrm{a}\left(\overline{n}_{lk}, \beta\right) - \mathrm{a}\left(\overline{n}_{lk}^{-i}, \beta\right) \\
& \qquad - \mathrm{a}\left(n_{lk} + \overline{n}_{lk}, \alpha + \beta\right) + \mathrm{a}\left(n_{lk}^{-i} + \overline{n}_{lk}^{-i}, \alpha + \beta\right) \Big\} \\
& + \mathrm{b}\left(n_{k}^{-i}, 1\right) - \mathrm{b}\left(n_{\geq k}^{-i}, \gamma + 1\right) + \sum_{m=1}^{k-1} \Big\{ \mathrm{b}\left(n_{>m}^{-i}, \gamma\right) - \mathrm{b}\left(n_{\geq m}^{-i}, \gamma + 1\right) \Big\} \Bigg) .
\end{aligned}
\tag{4.36}
$$

Note that the CVB inference gains flexibility by relaxation of independence assumption in return for losing accuracy by Taylor approximation.

Next, we derive the CVB0 inference that has firstly proposed for LDA [3]. The difference from the CVB inference is to use only the zeroth-order term in Taylor approximation. Equation (4.26) is approximated as follows:

$$
\begin{aligned}
& (4.26) \simeq \\
& \exp\Bigg(\sum_{l=k+1}^{T} \log \frac{\Gamma\left(\mathbb{E}_q\left[n_{kl}\right] + \alpha\right)}{\Gamma\left(\mathbb{E}_q\left[n_{kl}^{-i}\right] + \alpha\right)} \frac{\Gamma\left(\mathbb{E}_q\left[\overline{n}_{kl}\right] + \beta\right)}{\Gamma\left(\mathbb{E}_q\left[\overline{n}_{kl}^{-i}\right] + \beta\right)} \frac{\Gamma\left(\mathbb{E}_q\left[n_{kl}^{-i}\right] + \mathbb{E}_q\left[\overline{n}_{kl}^{-i}\right] + \alpha + \beta\right)}{\Gamma\left(\mathbb{E}_q\left[n_{kl}\right] + \mathbb{E}_q\left[\overline{n}_{kl}\right] + \alpha + \beta\right)} \\
& \quad + \sum_{l=1}^{k} \log \frac{\Gamma\left(\mathbb{E}_q\left[n_{lk}\right] + \alpha\right)}{\Gamma\left(\mathbb{E}_q\left[n_{lk}^{-i}\right] + \alpha\right)} \frac{\Gamma\left(\mathbb{E}_q\left[\overline{n}_{lk}\right] + \beta\right)}{\Gamma\left(\mathbb{E}_q\left[\overline{n}_{lk}^{-i}\right] + \beta\right)} \frac{\Gamma\left(\mathbb{E}_q\left[n_{lk}^{-i}\right] + \mathbb{E}_q\left[\overline{n}_{lk}^{-i}\right] + \alpha + \beta\right)}{\Gamma\left(\mathbb{E}_q\left[n_{lk}\right] + \mathbb{E}_q\left[\overline{n}_{lk}\right] + \alpha + \beta\right)} \\
& \quad + \log \frac{\mathbb{E}_q\left[n_{k}^{-i}\right] + 1}{\mathbb{E}_q\left[n_{\geq k}^{-i}\right] + \gamma + 1} + \sum_{m=1}^{k-1} \log \frac{\mathbb{E}_q\left[n_{>m}^{-i}\right] + \gamma}{\mathbb{E}_q\left[n_{\geq m}^{-i}\right] + \gamma + 1} \Bigg) .
\end{aligned}
\tag{4.37}
$$

This is more computationally efficient than (4.36) because it does not require to calculate and maintain variance counts.

We also estimate the hyperparameters α, β and γ by using Minka's fixed point iteration method [34].

4.4 Experimental Comparisons

In section 4.3, we presented three types of VB inference algorithms for the IRM. In this section, we call these algorithms "VB", "CVB", and "CVB0", and compare these VB inference algorithms and CGS. Note that for CGS we also use the Minka's fixed point iteration method [34] for estimating the hyperparameters α and β, and the Slice sampler for the hyperparameter γ.

We confirmed effectiveness of these algorithms to six real datasets. We prepared the NIPS coauthorship network dataset [16]. This consists of information about which each author published the paper together with another author in the 1st-17th NIPS conferences[1]. We made a dataset with only most connected 225 authors from original data as in [33, 39]. Note that we omitted the self-link observations from the data. We also prepared five collaboration networks in arXiv [31][2]. They consist of collaboration information about authors submitted in arXiv from January 1993 to April 2003, and are categorized to Astro Physics (AstroPh), Condense Matter Physics (CondMat), General Relativity and Quantum Cosmology (GrQc), High Energy Physics - Phenomenology (HepPh), and High Energy Physics - Theory (HepTh). We also made datasets with most connected authors in the same way as the NIPS dataset. Table 4.1 shows some statistics of these six datasets including the number of objects, ratio of links, and average clustering coefficient for each dataset.

We used the test log likelihood and the AUC, the area under the ROC (Receiver Operating Characteristic) curve, as evaluation metrics. We used 80% as training data and 20% as test data, and repeated 25 times while changing the test data, and took the average of them. For one training procedure, we repeated the update until the difference of the lower bound gets smaller than 10^{-6} and set the maximum number of iteration to 3,000 for VB algorithms. We also set the truncation level T to 50. For CGS, we repeated the update by 20,000 times, and used the last 500 samples for the test.

Results about the test log likelihood and the AUC are shown in Table 4.2 and 4.3. The higher test log likelihood and AUC indicate better performances. We also examined the Wilcoxon signed rank test with p-value of 0.01. We highlighted

[1] http://ai.stanford.edu/~gal/data.html

[2] http://snap.stanford.edu/data/

Table 4.1. Information about 6 network datasets. These are sorted by ratio of links for comparison.

	Number of objects	Ratio of links (%)	Average clustering coefficient
NIPS	225	2.36	0.573
CondMat	230	5.33	0.329
HepTh	205	5.52	0.417
GrQc	236	11.11	0.800
AstroPh	220	15.74	0.500
HepPh	400	43.30	0.825

Table 4.2. Results of the test log likelihood ($T = 50$)

	CGS	VB	CVB	CVB0
NIPS	**-353.3(38.0)**	-400.4(34.5)	-405.6(46.9)	-432.9(51.6)
CondMat	-1073.6(42.8)	-922.6(41.3)	**-888.2(39.5)**	**-882.8(46.7)**
HepTh	-555.4(27.1)	-483.9(24.9)	**-466.5(25.2)**	**-458.8(26.6)**
GrQc	-299.9(20.8)	-384.1(187.9)	**-219.9(22.7)**	**-202.6(23.5)**
AstroPh	-1307.6(39.6)	-1488.7(56.0)	**-1285.9(43.0)**	**-1274.0(31.8)**
HepPh	-2272.2(47.6)	-1949.1(58.7)	-1819.5(176.8)	**-1671.1(74.8)**

the best results and those not significantly worse than them.

VB outperformed CVB and CVB0 in the NIPS dataset which is the most sparse dataset in our experiment. On the other hand, CVB and CVB0 outperformed VB in other five datasets. The difference is especially larger in dense datasets. The tendencies of the performances with respect to the AUC is similar to those of the test log likelihood.

We also validate the effects of the truncation level T. We set T to 10, 20, 80, and 100 besides 50, and evaluated them as in a previous experiment. The results with box plots are shown in Fig. 4.2 and 4.3. Titles in each graph denote the name of dataset, its ratio of links (Links), and estimated average number of clusters for CGS (CGS_K).

Table 4.3. Results of the AUC ($T = 50$)

	CGS	VB	CVB	CVB0
NIPS	**0.8939(0.0301)**	0.8736(0.0146)	0.8573(0.0273)	0.8351(0.0388)
CondMat	0.5738(0.0203)	0.7908(0.0175)	**0.8018(0.0216)**	**0.8077(0.0162)**
HepTh	0.7689(0.0562)	0.8983(0.0169)	**0.9062(0.0152)**	**0.9102(0.0169)**
GrQc	0.9933(0.0014)	0.9877(0.0105)	**0.9954(0.0010)**	**0.9956(0.0013)**
AstroPh	**0.9011(0.0077)**	0.8465(0.0094)	0.8886(0.0069)	0.8910(0.0060)
HepPh	0.9843(0.0011)	0.9861(0.0010)	0.9872(0.0022)	**0.9888(0.0010)**

Although it is ideally desirable that increasing T does not degrade the performances, the results in some datasets and algorithms did not work so. This is remarkable in datasets estimated as relatively small number of clusters by CGS. It seems that they were unsuccessful in too large truncation levels because they are liable to fall into local maxima. Bottom two datasets whose estimated numbers of clusters were relatively large did not degrade the performances even when $T = 80$ and 100 except for VB. One solution to this problem will be to use the Kurihara's label reordering [30] although the method requires more computational costs.

To compare actual runtime, we show the evolution of the test log likelihood for each algorithm in Fig. 4.4. We showed the results of NIPS and AstroPh, and randomly selected 5 trials from 25 trials. For VB algorithms, we calculate the test log likelihood for every 5 iterations, and plotted those by 300 iterations. For CGS, we calculated the test log likelihood for every 200 iterations by using last 200 samples, and plotted those by 20,000 iterations.

CGS is much faster per iteration than other VB algorithms because it does not require calculating the log Gamma, digamma and trigamma functions except for when estimating hyperparameters. However the values of CGS sometimes vary in time, and its convergence diagnosis may not be easy. On the other hand, VB algorithms converge to a maximum for a few tens of iterations. CVB is slower than VB and CVB0 since CVB requires more computational costs.

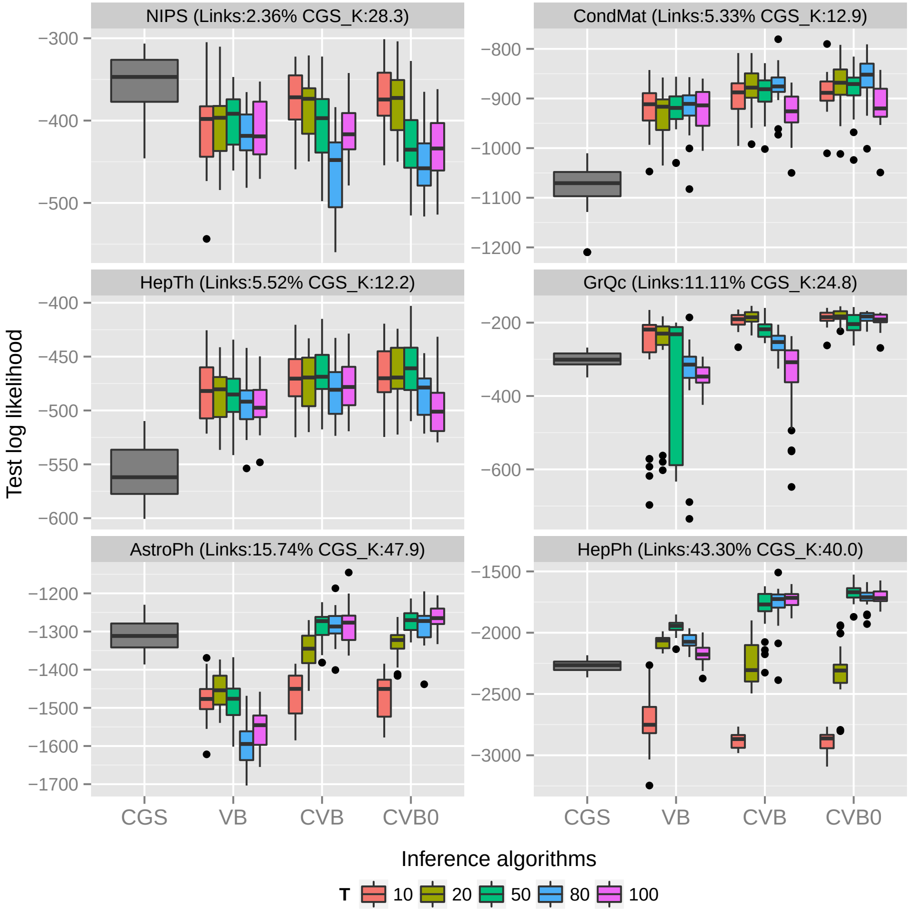

Figure 4.2. Results of the test log likelihood changing the T to 10, 20, 50, 80, and 100.

4.5 Discussions

In Table 4.2 and 4.3, we compared empirical performance of the algorithms and showed CVB and CVB0 outperformed the VB in most dense networks. The results follow our intuition. Denser datasets will tend to present more complex relations into networks. Complex relations reflect strong dependencies of parameters of clusters. In this situation, we will receive the benefit from CVB inference to relax the independence assumption about the parameters by marginalizing out $\boldsymbol{\eta}$ and $\boldsymbol{v}$. Though the performances of CVB and CVB0 are competitive, CVB0 slightly outperforms CVB in the most of datasets in spite of its rough approxi-

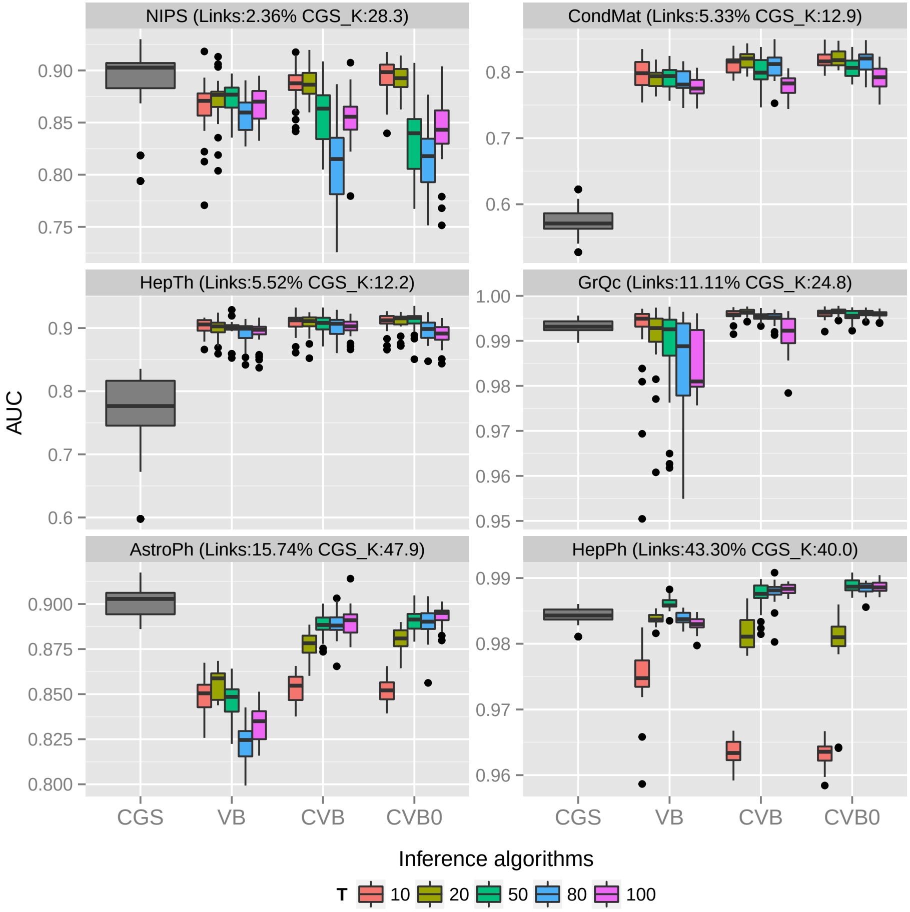

Figure 4.3. Results of the AUC changing the T to 10, 20, 50, 80, and 100.

mation. CGS does not necessarily perform better than other algorithms. Since CGS is a MCMC method, in principle, this provides the samples from the true posterior distribution after sufficiently many iterations. However, in practice, the Gibbs sampling can be slow to converge and mix poorly, since it is possible to stay around a local mode [24].

In the experiments, CVB0 was slightly better than CVB in spite of the more rough Taylor approximation. Asuncion *et al.* reported CVB0 outperformed CVB in LDA [3]. Sato and Nakagawa gave a theoretical analysis to the result: they explained the good performance of CVB0 in LDA by using the α-divergence [44]. Sato and Nakagawa showed that CVB0 is composed of the $(\alpha = 1, -1)$-divergence

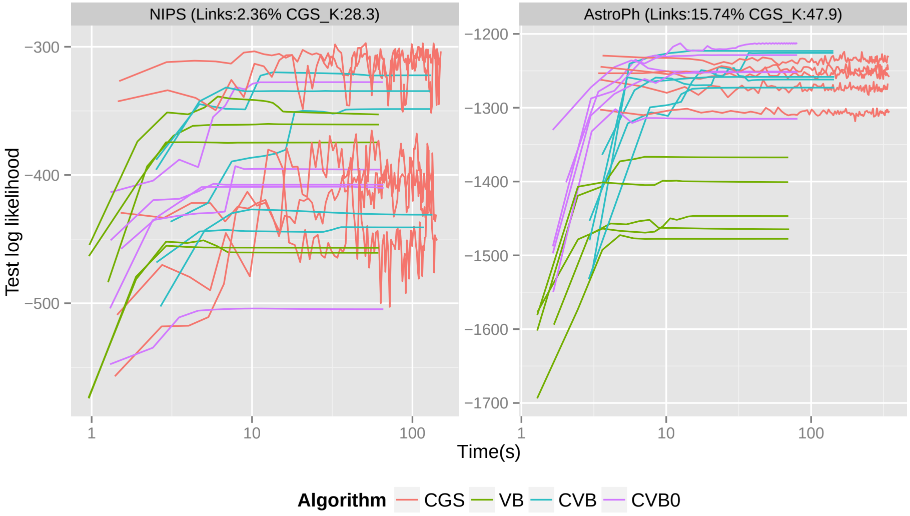

Figure 4.4. Evolution of the test log likelihood in NIPS and AstroPh datasets.

projections and that $\alpha = -1$ is similar to $\alpha = 1$ in LDA. The virtue of $\alpha = 1$ was discussed in Minka's work [35], which supports the analysis of Sato and Nakagawa. However, their work does not cover the general CVB0 including the case of the IRM. To clarify our results, further theoretical analysis will be required.

In this Chapter, we focused on the case when observations are represented by an undirected network. The IRM can be used in more general relational data, e.g., in directed networks or bipartite graphs. For modeling these observations, we need to extend the generative process to those with more latent variables and parameters. These generative processes will have more strong dependencies of parameters $\boldsymbol{\eta}$ and $\boldsymbol{v}$. It is expected that, in these cases, the CVB inference algorithms can be more effective, and it is possible that CVB and CVB0 outperform VB in more datasets.

Note that the IRM suffers from falling into the local maximum. While this causes slower mixing in CGS, we confirmed that the VB and CVB inference also converged to worse local maximum in preliminary experiments with artificial datasets. To avoid the problem, some measures will be taken. For example, split-merge techniques will be effective for CGS [24], and annealing methods is another choice for the VB and CVB inference.

4.6 Summary

In summary, we derived three VB inference algorithms of the IRM for network data which have not been shed light on, and compared performances of these algorithms in real six datasets. Our contributions have two points: 1) we derived the CVB inference algorithms of the type of block model, e.g., the SBM and the IRM, and 2) we confirmed the performances for six real undirected networks, and discussed the relationships between performances of algorithms and data sparsity of the datasets.

Chapter 5

Conclusion

This thesis dealt with latent variable models that have been successful in many fields but still open to research questions. Latent variable models are flexible frameworks to obtain knowledge from complex data and the efficiencies have been shown with the spread of the machine learning approaches in data analysis. However, individual data often have distinctive properties that is significant to capture richer latent structures, thus, domain-specific modeling and learning are required for obtaining more valuable knowledge.

In particular, this thesis tackled two topics about the latent variable models. In Chapter 3, we address the issue of extracting search query patterns with topic models. After showing computational problems of a naive query model, we proposed the PCTM that models topic co-occurrence patterns on queries. The PCTM showed the higher empirical performance than standard topic models with respect to human evaluation of estimated topics and some tasks supposing real applications. In Chapter 4, we studied the VB methods of the IRM for networked data. We derived the CVB inference algorithms of the IRM for networks, and compared empirical performance of the VB methods and CGS algorithm. The results implied the CVB inference algorithms are particularly superior to other methods in dense networks.

5.1 Future Directions

Finally we give some future directions as follows. In Chapter 3, we proposed the PCTM as an approximated model of a naive query model. While this approximation brings computationally efficiencies, the consistencies among topic pairs on a query preserve no longer. Moreover, since the PCTM is not a fully generative model for queries, we need extra computations that is included in the assumption of the PCTM to obtain the query-specific topic information. It is significant to develop a topic model for queries while keeping the computationally efficiencies. As another direction, it will be beneficial to improve the quality of estimated low-frequent topics. To compensate observations about such topics, other text resources are available. Fortunately, the PCTM is formulated as a fully-Bayesian model, thus, it will be relatively easy to incorporate such information into the PCTM. The extended model will be expected to enhance the quality of all the topics.

In Chapter 4, from our experiments, it is inferred that the relative performances of the inference algorithms for the IRM seem to depend on the density of networks. As a future work, more theoretical analyses are needed about the behaviors. One approach is to use the asymptotic analysis for the VB inference algorithms [53]. This approach clarifies the asymptotic behaviors of the VB lower bounds in the large sample size and may provide good indicator for comparing the VB inference algorithms. Moreover, the analysis of the CVB inference is significant. As discussed in Section 4.5, the relationship between the CVB and CVB0 inference is not clarified. Providing the further theoretical implication of the CVB inference will be also contributed to not only the IRM but also other models. As another direction, developing the faster VB inference algorithms will be beneficial. The VB inference of the IRM needs to compute the gamma and polygamma functions that is computationally expensive. The reduction of such calculations will be important in applying large-scale data.

Acknowledgements

本研究の遂行にあたり，たくさんの方々にご支援いただきました．

まず池田和司先生に深く感謝いたします．池田先生には，数理的なバックグラウンドの乏しい自分に，後期課程から本学で学ぶ機会をいただきました．研究室の主要なテーマからやや逸れた問題に興味があったにもかかわらず，自由に研究させていただけたことで充実した3年間を過ごすことができました．また，様々な学外での活動を紹介いただいたことで貴重な経験を積むことができました．

松本裕治先生には，副審査委員を引き受けていただきました．元々，松本先生の研究室を見学のために訪ねたことが本学へ進学するきっかけになりました．審査中の有益なご助言と合わせて，深くお礼いたします．

久保孝富先生には，日頃の研究の中で度々ご指導いただきました．手法や実験方法に関するご助言に加え，メンタル面でも様々なアドバイスをいただきました．久保先生を訪ねると，いつも真摯に対応いただだいたことは日々研究を進めていく中で大きな助けになりました．また，普段から研究室で機械学習の問題や手法について議論させていただき，とても刺激になりました．ありがとうございました．

国立情報学研究所の林浩平さんには，副審査委員を引き受けていただきました．約1年半の間共同研究させていただき，所外の所属であるにもかかわらず，手厚いご指導をいただきました．研究の進め方を始め，論文の書き方や発表方法など，たくさんのことを学ばせていただきました．特に打ち合わせでの鋭いご指摘は，より深く問題を考えるきっかけになりました．ありがとうございました．

豊橋技術科学大学の渡辺一帆先生には，主に本学から移動されるまでの間ご指導いただきました．研究のテクニカルな面で質問をさせていただく機会が多かったですが，その度に1つ1つ丁寧にご指導いただけて，とても助かりました．豊橋大へ移動されて以降も，メールやスカイプを通して打ち合わせを続けていただき，お世話になりました．また，自分がロシアに出張に行く際には，現地の情報を色々教えていただきました．ありがとうございました．

為井智也先生には，主に研究室での生活でお世話になりました．自分が研究室でスポーツ担当だったため，先生のご活躍には本当に助けられました．九州工業大学の柴田智広先生と船谷浩之先生には本学から移動されるまでの間，主に研究室のセミナーでご助言いただきました．皆さまに深くお礼いたします．

後期課程在籍中，JST ERATO 河原林巨大グラフプロジェクトに参加させていただきました．河原林健一先生には本プロジェクトに参加する機会をいただきま

した．大輪拓也さんには，林さんとともに定期的な打ち合わせを行っていただき研究に関するアドバイスをいただきました．研究所での長期滞在の際に，林さんと一緒に連れて行っていただいたランチはいつもおいしくて，良い思い出になりました．また，共同研究先であるYahoo Japan!研究所の藤田澄男さんには，データの提供を始め，企業の立場からご助言をいただきました．その他，プロジェクトメンバーならびにプロジェクト研究推進員の皆さまに改めて感謝いたします．

本学への進学前には立命館大学で博士前期課程までを過ごしました．ご指導いただいた前田亮先生，木村文則先生に感謝申し上げます．本学に進学後も，研究発表の打ち合わせや実験について相談させていただき，国際会議での発表の機会を与えていただきました．また，同研究室のBiligsaikhan Batjargalさんには前述の国際会議に同行いただきました．厳しい寒さのロシアを一緒に旅したことは大切な思い出です．筑波大学の手塚太郎先生には，前期課程1年時までご指導いただきました．手塚先生には機械学習に興味をもつきっかけを与えていただき，確率モデルの基礎を教えていただきました．立命館大学での経験が本学での研究の支えになりました．皆さまに深くお礼いたします．

後期課程2年時には京都女子大学にて非常勤講師を務める機会がありました．同大学の小波秀雄先生には，授業に関する打ち合わせを通して度々お世話になりました．初めての経験でわからないことだらけでしたが，いつも丁寧にご対応いただいたこと感謝いたします．

数理情報学研究室の皆さんには日頃から様々な場面で助けていただきました．入学した当時は後期課程から進学したこともあり馴染めるか心配でしたが，皆さんに暖かく迎えていただきました．特に，丸野由希さんには非常勤講師の仕事についてたくさんアドバイスをいただきました．また，スタッフの谷本史さん，足立敏美さんには申請書の手続きなどで何度もお世話になりました．3年間でお世話になったメンバーの皆さまを含め，改めてお礼いたします．

最後になりましたが，長きに渡った学生生活の中で家族の支えは欠かせないものでした．在学中には家族にとって大きな出来事があり，学生を続けるべきか悩んだ時期もありました．それでも，研究を続けるよう背中を押してくれた家族に心より感謝します．

Bibliography

[1] Ganesh Agarwal, Govind Kabra, and Kevin Chen-Chuan Chang. Towards rich query interpretation: Walking back and forth for mining query templates. In *Proceedings of the 19th International Conference on World Wide Web*, pages 1–10, 2010.

[2] Rahul Agrawal, Archit Gupta, Yashoteja Prabhu, and Manik Varma. Multi-label learning with millions of labels: Recommending advertiser bid phrases for Web pages. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 13–23, 2013.

[3] Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. On smoothing and inference for topic models. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pages 27–34, 2009.

[4] Hagai Attias. A variational Baysian framework for graphical models. In *Advances in Neural Information Processing Systems*, 1999.

[5] Matthew J. Beal. *Variational algorithms for approximate Bayesian inference.* PhD thesis, Gatsby Computational Neuroscience Unit, University College London, 2003.

[6] Christopher M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[7] David M. Blei and Michael I. Jordan. Variational inference for Dirichlet process mixtures. *Journal of Bayesian Analysis*, 1(1):121–144, 2005.

[8] David M. Blei and John D. Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 1(1):17–35, 2007.

[9] David M. Blei, Andrew Y. Ng, and Michael. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[10] Charles Blundell, Katherine A. Heller, and Jeffrey M. Beck. Modelling reciprocating relationships with Hawkes processes. In *Advances in Neural Information Processing Systems*, 2012.

[11] Mark J. Carman, Fabio Crestani, Morgan Harvey, and Mark Baillie. Towards query log based personalization using topic models. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 1849–1852, 2010.

[12] Jonathan Chang, Jordan Boyd-Graber, Chong Wang, Sean Gerrish, and David M. Blei. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems*, 2009.

[13] Xueqi Cheng, Xiaohui Yan, Yanyan Lan, and Jiafeng Guo. BTM: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering*, 26(12):2928–2941, 2014.

[14] Alexander P. Dawid A. P. Dawid and Allan M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):20–28, 1979.

[15] Thomas S. Ferguson. A bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230, 1973.

[16] Amir Globerson, Gal Chechik, Fernando Pereira, and Naftali Tishby. Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, 8:2265–2295, 2007.

[17] Jun Han, Ju Fan, and Lizhu Zhou. Crowdsourcing-assisted query structure interpretation. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 2092–2098, 2013.

[18] Morgan Harvey, Fabio Crestani, and Mark J. Carman. Building user profiles from topic models for personalised search. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management*, pages 2309–2314, 2013.

[19] Matthew D. Hoffman, David M. Blei, Chong Wang, and John D. Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

[20] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57, 1999.

[21] Thomas Hofmann and Jan Puzicha. Statistical models for co-occurrence data. Technical report, Massachusetts Institute of Technology Artificial Intelligence Laboratory, 1998.

[22] Katsuhiko Ishiguro, Tomoharu Iwata, Naonori Ueda, and Joshua B. Tenenbaum. Dynamic infinite relational model for time-varying relational data analysis. In *Advances in Neural Information Processing Systems*, 2010.

[23] Katsuhiko Ishiguro, Naonori Ueda, and Hiroshi Sawada. Subset infinite relational models. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, 2012.

[24] Sonia Jain and Radford M. Neal. A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1):158–182, 2004.

[25] Di Jiang, Kenneth Wai Ting Leung, Wilfred Ng, and Hao Li. Beyond click graph: Topic modeling for search engine query log analysis. In *Proceedings of the 18th International Conference on Database Systems for Advanced Applications*, pages 209–223, 2013.

[26] Di Jiang and Wilfred Ng. Mining Web search topics with diverse spatiotemporal patterns. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 881–884, 2013.

[27] Michael I. Jordan. *Learning in Graphical Models.* MIT Press, 1999.

[28] Michael I. Jordan, Zoubin Ghahramani, Tommi S. Jaakkola, and Lawrence K. Saul. An introduction to variational methods for graphical models. *Machine Learning*, 37:183–233, 1999.

[29] Charles Kemp, Joshua B. Tenenbaum, Thomas L. Griffiths, Takeshi Yamada, and Naonori Ueda. Learning systems of concepts with an infinite relational

model. In *Proceeding of the American Association for Artificial Intelligence*, 2006.

[30] Kenichi Kurihara, Max Welling, and Yee Whye Teh. Collapsed variational Dirichlet process mixture models. In *Proceedings of the 20th International Joint Conferences on Artificial Intelligence*, 2007.

[31] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM Transactions on Knowledge Discovery from Data*, 1(1), 2007.

[32] Edward Meeds, Zoubin Ghahramani, Radford M. Neal, and Sam Roweis. Modeling dyadic data with binary latent factors. In *Advances in Neural Information Processing Systems*, 2007.

[33] Kurt T. Miller, Thomas L. Griffiths, and Michael I. Jordan. Nonparametric latent feature models for link prediction. In *Advances in Neural Information Processing Systems*, 2009.

[34] Thomas P. Minka. Estimating a Dirichlet distribution. Technical report, Microsoft, 2000.

[35] Thomas P. Minka. Divergence measures and message passing (MSR-TR-2005-173). Technical report, Microsoft Research, 2005.

[36] Morten Mørup, Kristoffer H. Madsen, Anne M. Dogonowski, Hartwig Siebner, and Lars K. Hansen. Infinite relational modeling of functional connectivity in resting state fMRI. In *Advances in Neural Information Processing Systems*, 2010.

[37] Kamal Nigam, Andrew Kachites Mccallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2-3):103–134, 2000.

[38] Krzysztof Nowicki and Tom A. B. Snijders. Estimation and prediction for stochastic blockstructures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

[39] Konstantina Palla, David A. Knowles, and Zoubin Ghahramani. An infinite latent attribute model for network data. In *Proceedings of the International Conference on Machine Learning*, 2012.

[40] Sandeep Pandey and Kunal Punera. Unsupervised extraction of template structure in Web search queries. In *Proceedings of the 21st International Conference on World Wide Web*, pages 409–418, 2012.

[41] Sandeep Pandey, Kunal Punera, Marcus Fontoura, and Vanja Josifovski. Estimating advertisability of tail queries for sponsored search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 563–570, 2010.

[42] Greg Pass, Abdur Chowdhury, and Cayley Torgeson. A picture of search. In *Proceedings of the 1st International Conference on Scalable Information Systems*, 2006.

[43] Douglas A. Reynolds and Richard C. Rose. Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, 3(1):72–83, 1995.

[44] Issei Sato and Hiroshi Nakagawa. Rethinking collapsed variational bayes inference for LDA. In *Proceedings of the International Conference on Machine Learning*, 2012.

[45] Masaaki Sato. Online model selection based on the variational Bayes. *Neural Computation*, 13(7):1649–1681, 2001.

[46] Jayaram Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[47] Wei Song, Yu Zhang, Ting Liu, and Sheng Li. Bridging topic modeling and personalized search. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 1167–1175, 2010.

[48] Christopher Stauffer and Eric Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 252–258, 1999.

[49] Yee Whye Teh, David Newman, and Max Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, 2007.

[50] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, 1967.

[51] Hanna M. Wallach, David Mimno, and Andrew McCallum. Rethinking LDA: Why priors matter. In *Advances in Neural Information Processing Systems*, 2009.

[52] Pengyu Wang and Phil Blunsom. Collapsed variational Bayesian inference for hidden markov models. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics*, 2013.

[53] Kazuho Watanabe and Sumio Watanabe. Stochastic complexities of gaussian mixtures in variational bayesian approximation. *Journal of Machine Learning Research*, 7:625–644, 2006.

[54] Gu Xu, Shuang-Hong Yang, and Hang Li. Named entity mining from clickthrough data using weakly supervised latent Dirichlet allocation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1365–1373, 2009.

[55] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. A biterm topic model for short texts. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1445–1556, 2013.

Appendix

A The detail of the VB lower bound (4.13)

Each term in the lower bound (4.13) is described as follows:

$$\mathbb{E}_q\left[\log p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\eta})\right] = \sum_{i=1}^{N-1}\sum_{j=i+1}^{N}\sum_{k=1}^{T}\sum_{l=k+1}^{T}$$

$$(\phi_{ik}\phi_{il}+\phi_{il}\phi_{jk})\Bigg\{x_{ij}(\psi(\mu_{kl})-\psi(\mu_{kl}+\nu_{kl}))+(1-x_{ij})(\psi(\nu_{kl})-\psi(\mu_{kl}+\nu_{kl}))\Bigg\}$$

$$+(\phi_{ik}\phi_{jk})\Bigg\{x_{ij}(\psi(\mu_{kk})-\psi(\mu_{kk}+\nu_{kk}))+(1-x_{ij})(\psi(\nu_{kl})-\psi(\mu_{kl}+\nu_{kl}))\Bigg\},$$

$$\mathbb{E}_q\left[\log p(\boldsymbol{z}|\boldsymbol{v})\right] = \sum_{i=1}^{N}\sum_{k=1}^{T-1}\Bigg\{\phi_{ik}(\psi(\kappa_k)-\psi(\kappa_k+\lambda_k))\sum_{m=k+1}^{T}\phi_{im}(\psi(\lambda_k)-\psi(\kappa_k+\lambda_k))\Bigg\},$$

$$\mathbb{E}_q\left[\log p(\boldsymbol{\eta}|,\alpha,\beta)\right] = \frac{1}{2}T(T+1)(\log\Gamma(\alpha+\beta)-\log\Gamma(\alpha)-\log\Gamma(\beta))$$

$$+\sum_{k=1}^{T}\sum_{m=k}^{T}\Bigg\{(\alpha-1)(\psi(\mu_{kl})-\psi(\mu_{kl}+\nu_{kl}))+(\beta-1)(\psi(\nu_{kl})-\psi(\mu_{kl}+\nu_{kl}))\Bigg\},$$

$$\mathbb{E}_q\left[\log p(\boldsymbol{v}|\gamma)\right] = (T-1)(\log\Gamma(1+\gamma)-\log\Gamma(1)-\log\Gamma(\gamma))$$

$$+\sum_{k=1}^{T-1}\Bigg\{(1-1)(\psi(\kappa_k)-\psi(\kappa_k+\lambda_k))+(\gamma-1)(\psi(\lambda_k)-\psi(\kappa_k+\lambda_k))\Bigg\},$$

$$\mathbb{E}_q\left[\log q(\boldsymbol{z},\boldsymbol{v},\boldsymbol{v})\right]$$

$$=\sum_{i=1}^{N}\sum_{k=1}^{T}\phi_{ik}\log\phi_{ik}+\sum_{k=1}^{T}\sum_{l=k}^{T}\Bigg\{\log\Gamma(\mu_{kl}+\nu_{kl})-\log\Gamma(\mu_{kl})-\log\Gamma(\nu_{kl})$$

$$+(\mu_{kl}-1)(\psi(\mu_{kl})-\psi(\mu_{kl}+\nu_{kl}))+(\nu_{kl}-1)(\psi(\nu_{kl})-\psi(\mu_{kl}+\nu_{kl}))\Bigg\}$$

$$+\sum_{k=1}^{T-1}\Bigg\{\log\Gamma(\kappa_k)-\log\Gamma(\kappa_k)-\log\Gamma(\lambda_k)+(\kappa_k-1)(\psi(\kappa_k)-\psi(\kappa_k+\lambda_k))$$

$$+(\lambda_k-1)(\psi(\lambda_k)-\psi(\kappa_k+\lambda_k))\Bigg\}.$$

B Expected/variance counts in the CVB inference of the IRM

We list expected/variance counts that are not listed in Section 4.3.2.

$$
\mathbb{E}_q\left[\overline{n}_{kl}^{-i}\right] = \begin{cases} \displaystyle\sum_{\substack{i'=1\\ i'\neq i}}^{N-1} \sum_{\substack{j'=i'+1\\ j'\neq i}}^{N} \left(\phi_{i'k}\phi_{j'l} + \phi_{i'l}\phi_{j'k}\right)\left(1 - x_{i'j'}\right) & (l \neq k), \\ \displaystyle\sum_{\substack{i'=1\\ i'\neq i}}^{N-1} \sum_{\substack{j'=i'+1\\ j'\neq i}}^{N} \phi_{i'k}\phi_{j'k}(1 - x_{i'j'}) & (l = k), \end{cases}
$$

$$
\mathrm{var}_q\left[\overline{n}_{kl}^{-i}\right] = \begin{cases} \displaystyle\sum_{\substack{i'=1\\ i'\neq i}}^{N-1} \sum_{\substack{j'=i'+1\\ j'\neq i}}^{N} \left\{\phi_{i'k}\phi_{j'l}\left(1 - \phi_{i'k}\phi_{j'l}\right)\left(1 - x_{i'j'}\right)\right. \\ \qquad\qquad \left. +\phi_{i'l}\phi_{j'k}\left(1 - \phi_{i'l}\phi_{j'k}\right)\left(1 - x_{i'j'}\right)\right\} & (l \neq k), \\ \displaystyle\sum_{\substack{i'=1\\ i'\neq i}}^{N-1} \sum_{\substack{j'=i'+1\\ j'\neq i}}^{N} \phi_{i'k}\phi_{j'k}\left(1 - \phi_{i'k}\phi_{j'k}\right)\left(1 - x_{i'j'}\right) & (l = k), \end{cases}
$$

$$
\mathbb{E}_q\left[n_{kl}^{-i} + \overline{n}_{kl}^{-i}\right] = \begin{cases} \displaystyle\sum_{\substack{i'=1\\ i'\neq i}}^{N-1} \sum_{\substack{j'=i'+1\\ j'\neq i}}^{N} \left(\phi_{i'k}\phi_{j'l} + \phi_{i'l}\phi_{j'k}\right) = \mathbb{E}_q\left[n_{kl}^{-i}\right] + \mathbb{E}_q\left[\overline{n}_{kl}^{-i}\right] & (l \neq k), \\ \displaystyle\sum_{\substack{i'=1\\ i'\neq i}}^{N-1} \sum_{\substack{j'=i'+1\\ j'\neq i}}^{N} \phi_{i'k}\phi_{j'k} = \mathbb{E}_q\left[n_{kk}^{-i}\right] + \mathbb{E}_q\left[\overline{n}_{kk}^{-i}\right] & (l = k), \end{cases}
$$

$$
\mathrm{var}_q\left[n_{kl}^{-i} + \overline{n}_{kl}^{-i}\right] = \begin{cases} \displaystyle\sum_{\substack{i'=1\\ i'\neq i}}^{N-1} \sum_{\substack{j'=i'+1\\ j'\neq i}}^{N} \left\{\phi_{i'k}\phi_{j'l}\left(1 - \phi_{i'k}\phi_{j'l}\right) + \phi_{i'l}\phi_{j'k}\left(1 - \phi_{i'l}\phi_{j'k}\right)\right\} \\ = \mathrm{var}_q\left[n_{kl}^{-i}\right] + \mathrm{var}_q\left[\overline{n}_{kl}^{-i}\right] & (l \neq k), \\ \displaystyle\sum_{\substack{i'=1\\ i'\neq i}}^{N-1} \sum_{\substack{j'=i'+1\\ j'\neq i}}^{N} \phi_{i'k}\phi_{j'k}\left(1 - \phi_{i'k}\phi_{j'k}\right) = \mathrm{var}_q\left[n_{kk}^{-i}\right] + \mathrm{var}_q\left[\overline{n}_{kk}^{-i}\right] & (l = k), \end{cases}
$$

$$\mathbb{E}_q\left[\overline{n}_{kl}\right] = \mathbb{E}_q\left[\overline{n}_{kl}^{-i}\right] + \sum_{j'=i+1}^{N} \phi_{j'l}(1 - x_{ij'}) + \sum_{i'=1}^{i-1} \phi_{i'l}(1 - x_{i'i}),$$

$$\mathrm{var}_q\left[\overline{n}_{kl}\right] = \mathrm{var}_q\left[\overline{n}_{kl}^{-i}\right] + \sum_{j'=i+1}^{N} \phi_{j'l}(1 - \phi_{j'l})(1 - x_{ij'}) + \sum_{i'=1}^{i-1} \phi_{i'l}(1 - \phi_{i'l})(1 - x_{i'i}).$$

$$\mathbb{E}_q\left[n_{kl} + \overline{n}_{kl}\right] = \mathbb{E}_q\left[n_{kl}^{-i} + \overline{n}_{kl}^{-i}\right] + \sum_{j'=i+1}^{N} \phi_{j'l} + \sum_{i'=1}^{i-1} \phi_{i'l},$$

$$\mathrm{var}_q\left[n_{kl} + \overline{n}_{kl}\right] = \mathrm{var}_q\left[n_{kl}^{-i} + \overline{n}_{kl}^{-i}\right] + \sum_{j'=i+1}^{N} \phi_{j'l}(1 - \phi_{j'l}) + \sum_{i'=1}^{i-1} \phi_{i'l}(1 - \phi_{i'l}),$$

Publication List

Journal Paper

1. Takuya Konishi, Takatomi Kubo, Kazuho Watanabe, and Kazushi Ikeda. Variational Bayesian inference algorithms for infinite relational model of network data. *IEEE Transactions on Neural Networks and Learning Systems*, in press.

International Conferences

1. Takuya Konishi, Takatomi Kubo, Kazuho Watanabe, and Kazushi Ikeda. Variational Bayesian inference algorithms for network infinite relational model. In *NIPS Workshop on Frontiers of Network Analysis: Methods, Models, and Applications*, 2013.

2. Takuya Konishi, Fuminori Kimura, and Akira Maeda. Topic model for user reviews with adaptive windows. In *Proceedings of the 35th European Conference on Information Retrieval*, pages 730-733, 2013.

Domestic Conferences

1. 小西 卓哉，大輪 拓也，藤田 澄男，池田 和司，林 浩平．共起トピックモデルによる検索クエリのパターン推定．第17回情報論的学習理論ワークショップ，2014．

2. 小西 卓哉，久保 孝富，渡辺 一帆，池田 和司．ネットワークデータに対する無限関係モデルの変分ベイズ学習に関する考察．第16回情報論的学習理論ワークショップ，2013．