Aalto University
School of Science
Degree Programme of Computer Science and Engineering

Yiping Liao

Interactive Intent Modeling Based on Probabilistic Sparse Models

Master's Thesis
Espoo, Jan. 19, 2017

Supervisor:	Professor Samuel Kaski
Instructor:	Pedram Daee M.Sc. (Tech.)

Aalto University
School of Science
Degree Programme of Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author:	Yiping Liao		
Title:	Interactive Intent Modeling Based on Probabilistic Sparse Models		
Date:	Jan. 19, 2017	**Pages:**	56
Professorship:	Data Communication Software	**Code:**	T-110
Supervisor:	Professor Samuel Kaski		
Instructor:	Pedram Daee M.Sc. (Tech.)		

In the exploratory search system, the user interacts with the system by providing feedback on the relevance of the recommended documents and keywords. It is often that the user is unfamiliar with the topic she is investigating, so the system should be able to help her form a precise query and explore the information space.

Typically, the exploratory search process is modeled as a contextual bandit problem, a sequential learning algorithm which adopts the recommendation strategy based on user's feedback, aiming at suggesting more precise keywords and retrieving the most relevant documents with minimum user interactions. One big challenge in the exploratory search is that the corpus in which a bandit algorithm explores is huge while the feedback from the user is always scarce, leading to a non-trivial learning problem with large dimensionality and limited observations. In this thesis, I tackle this challenge by adopting Bayesian linear regression with spike and slab priors which enforce sparsity on the feature space, so the bandit algorithm could narrow down the search to the most relevant documents. I incorporate the Expectation Propagation algorithm to approximate the posterior distribution of the sparse model, Thompson sampling to address the exploration-exploitation dilemma, and Topic model to discover the structure of the documents which could provide group information that can further constrain the search space to specific topics.

To assess the models, I simulate the user behavior in an exploratory search process and compare the model coefficients learned by linear models using Gaussian prior and, spike-and-slab prior with or without group information. Several performance metrics are also evaluated. Empirically, the spike-and-slab with or without group information perform similarly and outperform Gaussian prior which does not encourage sparsity. The learned model coefficients justify the assumption that most of the coefficients do not contribute to the model. Besides, the model of group spike-and-slab prior has fewer coefficients needed to be estimated than spike-and-slab prior without group information, and potentially could be applied to larger corpora.

Keywords: Bayesian sparse linear model, exploratory search, intent modeling, multi-armed bandit, spike and slab priors, Thompson sampling

Language: English

Acknowledgements

Firstly I feel very grateful to be receiving funding from Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN) and MindSee project to support me completing the Master's thesis. Secondly, it is such my pleasure and honor to work under the supervision of Pedram Daee and Professor Samuel Kaski. I had learned many regarding how to conduct a machine learning research as well as how to develop a probabilistic model for a given machine learning problem. Thanks to this opportunity, I have experienced a great Finnish Summer no matter in Aalto University to discuss with other researchers, to attend meetings where the others were always supportive and insightful during the discussions in Probabilistic Machine Learning Group. It was really a pleasant experience I have ever had.

There are just too many thank-yous to say to other people on my mind, so please allow me to skip that and leave a thank you here to you all.

Espoo, Jan. 19, 2017

Yiping Liao

Abbreviations and Acronyms

BO	Bayesian Optimization
EP	Expectation Propagation
HCI	Human-Computer Interaction
IR	Information Retrieval
LASSO	Least Absolute Shrinkage and Selection Operator
LDA	Latent Dirichlet Allocation
LSI	Latent Semantic Indexing

Contents

Chapter 1

Introduction

1.1 Motivation

Information retrieval is an important task for the computer users in their daily life. Although most of the search tasks can be completed with the existing search engines, in the situation that the user is uncertain about the search target, or unfamiliar with the search topic, forming the query becomes very difficult. When the user tries to conduct a more complicated task, such as research or investigation, the information needs may evolve with iterations of verifying the search results and forming the refined queries. Therefore, the search engine should be able to keep track of the user's search history, summarize the overview of the search results, recommend more precise queries as well as present relevant and diverse documents matched to user's potential search intent.

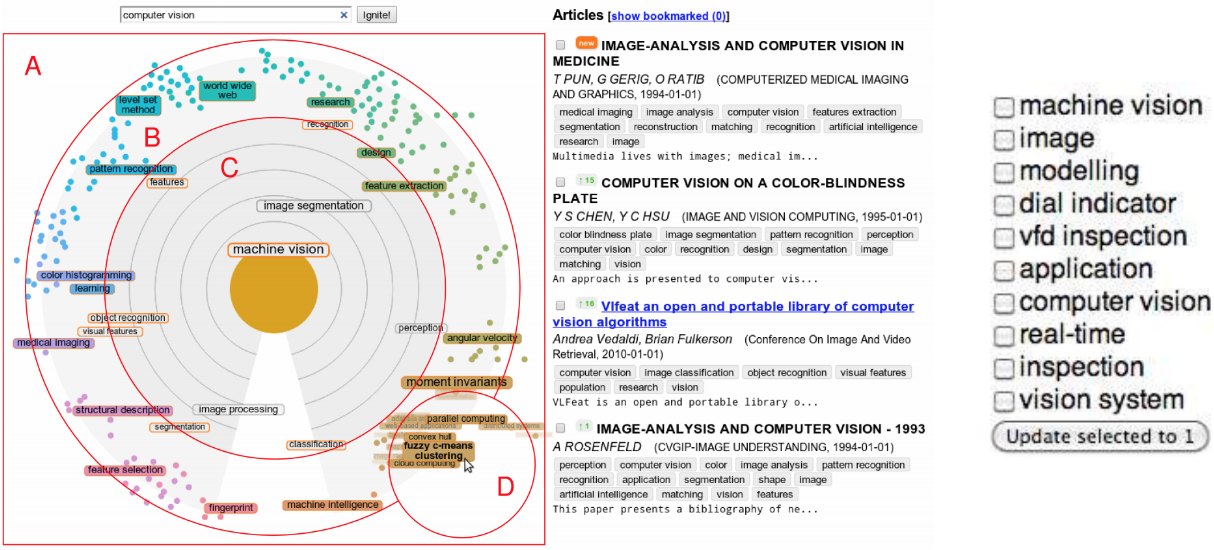
The research focus in this thesis is the *exploratory search algorithms* that are capable of learning and modeling user's search intent through a limited number of iterations. Such exploratory search algorithm takes user's feedback on the search results to make inference on user's search intent, refining the search results in the subsequent iterations.

Challenges have been pointed out in designing algorithms for such exploratory search systems. Firstly, systems taking relevance feedback often have *context trap* problem [4]. The reason for that problem can be that the user's query is imprecise, so the search results can easily fall into the certain context where the user's information need may not reside in. Thereby, the system should present search results that are relevant to user's current query in the meantime provide diverse results that are potentially relevant to the search target. This addresses the well-known *exploitation and exploration dilemma* in designing information retrieval systems. Secondly, the exploratory search systems should be able to model user's search intent and provide the recommendations of relevant keywords and documents during the interactive search process. However, in practice, the feedback from the user is often scarce compared to the number of documents in the database, increasing the difficulty of identifying relevant documents that match to user's information need from a large corpus. In short, the motivation of this thesis is to propose algorithms for efficiently and effectively tackling aforementioned difficulties. In the next section, we describe different types of user activities involved in exploratory search and how to distinguish between exploratory and conventional search activities.

1.2 Exploratory Search

Exploratory search is a topic that intersects the field of information retrieval (IR) and human-computer interaction (HCI). One can define exploratory search to be a search activity that involves interactions between human and the system. First specifically defined by Gary Marchionini in [1], three main types of search activities, i.e. *look-up*, *learning*, and *investigating*, characterize an exploratory search system. *Look-up* search activities are involved in most of the conventional search system in which the system returns well-structured objects, e.g. short excerpt of the document, keywords presented in the documents, dates being indexed etc., to answer user's (often clearly defined) query or other information needs. *Learning* and *investigating* are important features in an exploratory system to distinguish it from conventional search system in which a user discovers the new interest, acquires new knowledge, comprehends, interpret the topic involved in the searching task, and fills up the knowledge gap so as to identify what explicitly answers her information need. These two activities typically involve multiple rounds of interactions between a user and search system. Thus the system is able to return the search results that have been evolved and refined over iterations according to user's need.

However, in the exploratory search scenario, the user may not have concrete and clear thoughts in mind to precisely express her search targets when the search is initiated. This is especially common when user is doing research or investigation on an unfamiliar topic, making forming a proper and precise keywords especially difficult. The search engine thus has to present user diverse and relevant enough search results to help the user explore her all possible objects of interest. On the one hand, if the search results are too broad and diverse to be relevant and informative, then the user could get lost in them. On the other hand, if the search results are too narrow in one or two



Articles [show bookmarked (0)]

☐ new **IMAGE-ANALYSIS AND COMPUTER VISION IN MEDICINE**
T PUN, G GERIG, O RATIB (COMPUTERIZED MEDICAL IMAGING AND GRAPHICS, 1994-01-01)
medical imaging | image analysis | computer vision | features extraction | segmentation | reconstruction | matching | recognition | artificial intelligence | research | image
`Multimedia lives with images; medical im...`

☐ ↑15 **COMPUTER VISION ON A COLOR-BLINDNESS PLATE**
Y S CHEN, Y C HSU (IMAGE AND VISION COMPUTING, 1995-01-01)
color blindness plate | image segmentation | pattern recognition | perception | computer vision | color | recognition | design | segmentation | image | matching | vision
`An approach is presented to computer vis...`

☐ ↑16 **Vlfeat an open and portable library of computer vision algorithms**
Andrea Vedaldi, Brian Fulkerson (Conference On Image And Video Retrieval, 2010-01-01)
computer vision | image classification | object recognition | visual features | population | research | vision
`VLFeat is an open and portable library o...`

☐ ↑1 **IMAGE-ANALYSIS AND COMPUTER VISION - 1993**
A ROSENFELD (CVGIP-IMAGE UNDERSTANDING, 1994-01-01)
perception | computer vision | color | image analysis | pattern recognition | recognition | application | segmentation | shape | image | artificial intelligence | matching | vision | features
`This paper presents a bibliography of ne...`

Figure 1.1: Scinet's intent radar interface visualizes the user's intent model consisting of relevant keywords and ranked documents. A user can provide positive feedback by dragging keywords closer to the center of the radar screen or clicking on the relevant documents shown on the right side of the interface [7].

specific categories or topics, the user may not be able to retrieve the object of interest. Meanwhile, the system should engage the users to take control over the information seeking process, provide the friendly interactive search interface which eases the feedback-giving process, and recommend possible next steps for the user to take for narrowing down the topic of interest [2]. It is vital for the system to make reasonable inference on user's search interest, helping her form better and more precise query and presenting possible queries to match her interests. Thus in the next section, we introduce where an *interactive intent modeling algorithm* is placed in an exploratory search system, and a working instance of such system.

1.3 Interactive Intent Modeling

An exploratory search engine should equip the user with an interface that can ease the burden of interacting and providing relevance feedback. In addition, it should also properly learn the user intent to estimate the relevance scores of each object of interest in the database. A practical system inheriting the features that eased the feedback-giving process and attempted to learn user intent was Scinet search engine [7]. Scinet's interface presented an intent radar to ease the users' efforts for offering relevance feedback through a radial layout as shown in Figure 1.1. The user could drag the predicted relevant keywords in/out from the radial center to express the usefulness/unusefulness of the keywords that could lead to finding more relevant documents in the later iterations while the relevant documents were shown aligned with Intent Radar on the right. The exploratory search was initiated when a user inputs a query for seeking relevant information followed by that the search system suggests relevant information according to user's queries. Afterwards, the user could offer a feedback on the relevance of the presented results, and the system could infer user's information needs by collecting relevant feedback from the user. This process continued iteratively.

The performance of this type of systems relies heavily on built-in user intent modeling algorithms. It is worth mentioning that user's intent can sometimes be misleading and ambiguous when she is investigating an unfamiliar topic [9]. What's worse, the amounts of feedback can be scarce compared to the number of objects in the database, posing a huge challenge for intent modeling algorithm to learn what are the relevant objects of interest. Furthermore, the complexity of the learning algorithm should be well-controlled in order to make system be able to respond spontaneously.

1.4 Scope and The Structure of the Thesis

1.4.1 Scope

This work particularly focuses on seeking relevant scientific articles in an exploratory search setting. The difficulties addressed in previous sections are summarized as follows.

- Exploitation and exploration trade-off in exploratory search algorithms
- User's feedback can be scarce
- User's feedback can be ambiguous and uncertain

Daee et al. tackled the aforementioned challenges and modeled user intent under Bayesian framework and proposed *coupled multi-armed bandit* which coupled two domains of feedback, i.e. keywords and documents, into an unified probabilistic model [5]. It coped with feedback scarcity problem by allowing user giving feedback on both keywords and documents while controlled the trade-off between exploitation and exploration with Thompson sampling which was used to solve the multi-armed bandit problem. However, the model that was adopted for learning user intent faced the challenge of *small n large p* problem, i.e. the size of training data is small, but the dimensions are much larger than the size of training data, resulting in an ill-posed learning problem. To battle against it, this thesis explores the models whose priors are enforced to be *sparse*, i.e. most of the model coefficients are assumed to be irrelevant to the learning targets. The rationale behind it is to assume that the user's targets are usually in a small subset of documents, rather than the whole corpus. In addition to priors that enforce sparsity in each dimension of model coefficient individually, a group sparse prior that enforces sparsity in a group level is also investigated. It is assumed that user would only be interested in a few specific topics in a search session. Specifically, generalized spike-and-slab prior and group spike-and-slab prior are of particular interest in this thesis due to their effectiveness over other sparse priors that had been reported in [19]. Applying group spike-and-slab prior can further specify the group of features believed to be more relevant to the predictions and reduce the number of parameters in the model. The sparse models with spike-and-slab priors can then be approximated with Expectation Propagation (EP) technique which makes the originally intractable model inference process both tractable and efficient. Those sparse models are incorporated in coupled multi-armed bandit algorithm proposed in [5] which employs Thompson sampling to control exploitation and exploration.

1.4.2 Structure

Apart from the introduction chapter, the rest of the thesis are arranged as follows.

- Chapter 2 introduces background knowledge that is required to read the thesis, including the concepts and developments in sparse linear models, Bayesian sparse linear models, topic modeling with Latent Dirichlet Allocation (LDA).

- Chapter 3 introduces the models used in the thesis and their EP inference processes.

- Chapter 4 provides the descriptions of the data used throughout the experiments, setting of the experiments, how experiments were conducted, how simulations were designed, the metrics which were used for evaluation, comparisons between the models in terms of the introduced metrics, and finally the analyses of the experimental results.

- Chapter 5 summarizes the proposed methods and findings presented in this thesis.

Chapter 2

Background

The chapter provides the literature review regarding the algorithms and techniques used in this thesis, presenting the essential knowledge for one to comprehend the contents in the following chapters. Section 2.1 and 2.2 introduce sparse and group sparse linear models, which have become popular due to their effectiveness on the problem involving high-dimensional instances. Section 2.3 introduces the Bayesian optimization framework, contextual multi-armed bandit problem and Thompson sampling. Finally, topic modeling techniques (and in particular, LDA) for information retrieval purposes are discussed in the final section of this chapter.

2.1 Sparse Linear Model

Consider the general problem of regression (2.1) with p dimensions and n samples,

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \boldsymbol{\epsilon}, \tag{2.1}$$

where $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the feature matrix, $\mathbf{y} \in \mathbb{R}^n$ is the target vector, $\mathbf{w} \in \mathbb{R}^p$ is the unknown weight vector and $\boldsymbol{\epsilon}$ is the additive noise. The possible challenges of the regression problems are unsatisfying predicting accuracy due to high variance and the difficulty of interpreting the models. Especially in the underdetermined scenario when the number of observations is much smaller than the dimensionality of the feature space, fitting a large number of coefficients leads to an ill-posed problem. The reason is that there are infinite sets of coefficients that explain the data equally well. Besides, over-fitting is a potential problem when the number of observations is very limited [10]. The general principle to address this problem is to enforce the sparsity in the models, that is, to regularize the models by placing constraints on the coefficients and restricting the values of coefficients to be relatively small or even zero. The general form of this idea is formulated in (2.2) where f is the defined model, which may be parameterized by w as shown in (2.1), $L(\cdot)$ represents a loss function defined on f and input set $\{(x_i, y_i)|i = 1, ..., n\}$, $R(\cdot)$ is some regularization function only depending on model f, and λ is for controlling the level of regularization:

$$f^* = \text{argmin}_f \Sigma_{i=1}^n L(y_i, f(x_i)) + \frac{\lambda}{2} \cdot R(f), \tag{2.2}$$

Considering regularized linear model parameterized by $\mathbf{w}$, we can rewrite (2.2) as

$$\mathbf{w}^* = \text{argmin}_{\mathbf{w}} ||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 + \frac{\lambda}{2} \cdot R(\mathbf{w}) \tag{2.3}$$

The regularization term can take different forms. The particular interest is the form that leads to sparsity, i.e., most of the entries in $\mathbf{w}$ which are zero. The advantages of the sparsity include inducing automatically feature selection by identifying important or negligible dimensions in the model, increasing the interpretability and preventing over-fitting. One category of the well-known approaches to encode sparsity is $L1$ regularization [15], where

$$R(\mathbf{w}) = ||\mathbf{w}||_1,$$

$||\cdot||_1$ is L1-norm which adds penalty on entries of $\mathbf{w}$ are not zero. The reason that L1-norm leads to sparsity (and also why L2-norm when $R(\mathbf{w}) = ||\mathbf{w}||_2^2$ doesn't) can be seen in the left of Figure 2.1, the solution $\mathbf{w}^*$ is intersected by contours of $||\mathbf{w}||_1 \leq t_1$ and $||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 \leq t_2$, and it lies on one axis, leaving its value in another dimension to be zero. Expanding this idea to higher dimensions, one can expect that the solution $\mathbf{w}^*$ are sparse as it resides in only a few number of axes and most of other entries are hence zero. (t_1 and t_2 are arbitrary numbers)

Another category of approaches are Bayesian sparse models, which put the a sparsity-inducing prior on the coefficients $\mathbf{w}$, e.g. $w_j \sim \frac{\lambda}{2} e^{-\lambda \cdot |w_j|}$ (a Laplacian prior), to enforce the sparsity that the majority values in $\mathbf{w}$ are close to zero. To achieve this, different sparse priors have been proposed. Mitchell and Beauchamp proposed spike-and-slab prior which was a mixture distribution that divided the coefficients into two components; slab represented non-zero coefficients, and spike represented the coefficients to be zero or close to zero [11]. Tipping proposed hierarchical prior with zero-mean Gaussian prior distribution and Gamma distribution as the hyperprior shown

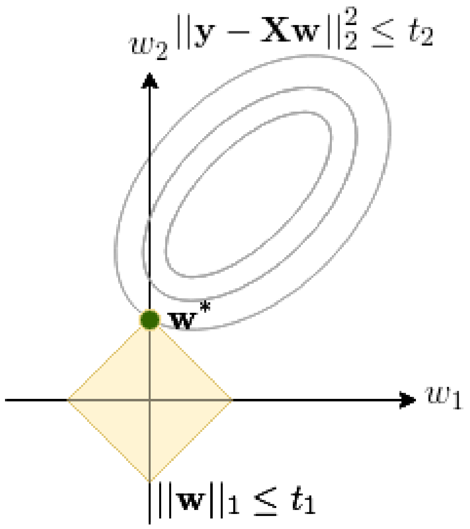

w_2
$||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 \leq t_2$
$\mathbf{w}^*$
w_1
$||\mathbf{w}||_1 \leq t_1$
w_2
$||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 \leq t_2$
$\mathbf{w}^*$
w_1
$||\mathbf{w}||_2^2 \leq t_1$

Figure 2.1: Lasso (L1-norm) vs. Ridge (L2-norm) regression: The reason that L1-norm leads to sparsity (and also why L2-norm when $R(\mathbf{w}) = ||\mathbf{w}||_2^2$ doesn't) can be seen here. The solution $\mathbf{w}^*$ is intersected by contours of $||\mathbf{w}||_1 \leq t_1$ and $||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 \leq t_2$, and it lies on one axis and hence is sparse (t_1 and t_2 are arbitrary number). On the contrary, $\mathbf{w}^*$ for L2-regularized model lies at somewhere close to an axis but not on it, hence it fails to enforce sparsity. The figure has been re-plotted and the original plots are from [12].

in (2.4),

$$\begin{aligned} p(\mathbf{w}|\boldsymbol{\alpha}) &= \prod_{j=1}^{d} \mathcal{N}(w_j|0, \alpha_j^{-1}) \\ p(\boldsymbol{\alpha}) &= \prod_{j=1}^{d} \text{Gamma}(\alpha_j|a, b)) \end{aligned} \tag{2.4}$$

where $\text{Gamma}(\alpha|a,b) = \frac{b^a\alpha^(a-1)}{\Gamma(a)}e^{-b\alpha}$ is a Gamma distribution parameterized by a and b. This brings sparsity to $\mathbf{w}$ since $p(w_j)$ is a Student-t distribution (shown in (2.5) by marginalizing out α_j) whose probability mass was concentrated both at the origin and along the axes [13].

$$\begin{aligned} p(\mathbf{w}) &= \prod_{j=1}^{d} p(w_j) \\ p(w_j) &= \int p(w|\alpha_j)p(\alpha_j)d\alpha_j \\ &= \frac{b^a\Gamma(a+\frac{1}{2})}{(2\pi)^{\frac{1}{2}}\Gamma(a)}(b+\frac{w_j^2}{2})^{-(a+\frac{1}{2})} \end{aligned} \tag{2.5}$$

Seeger focused on Laplace prior, a non-Gaussian sparsity prior shown in (2.6), which puts more weight on the values close to zero than Gaussian prior while having higher probabilities of setting larger values for some coefficients [14].

$$\begin{aligned} p(\mathbf{w}) &= \prod_{j=1}^{d} p(w_j) \\ p(w_j) &= \frac{\tau}{2\sigma}e^{-\frac{\tau}{\sigma}|w_j|} \end{aligned} \tag{2.6}$$

The model was intractable, so EP algorithm [17] was applied to approximate the posterior distribution. Hernandez-Lobato proposed EP algorithm with spike-and-slab prior (shown in (2.7)) in linear model for the reasons that, first, spike-and-slab prior was more effective in enforcing the sparsity comparing to Laplace (2.6), and Student-t distribution (2.5). Second, the proposed EP method had the advantage in computational efficiency [19].

$$\begin{aligned} p(\mathbf{w}|\mathbf{z}) &= \prod_{j=1}^{d} \left[z_j\mathcal{N}(w_j|0, v) + (1-z_j)\delta(w_j)\right] \\ p(\mathbf{z}) &= \text{Bern}(\mathbf{z}|\mathbf{p}) = \prod_{j=1}^{d} \left[p_j^{z_j}(1-p_j)^{1-z_j}\right] \end{aligned} \tag{2.7}$$

One promising extension of sparse linear model in the literature is group sparse model where it assumes that there exists structure in the model coefficients. We describe the group sparse linear models in the next section.

2.2 Group Sparse Linear Model

In the case that the sparsity pattern is observed, the cluster or structure information can facilitate the learning process by assuming the coefficients in each group are collectively relevant or irrelevant to the predicting model. If the correct group information is available, the estimation of coefficients can be improved given fewer observations [16][18][19].

Group Lasso [16] proposed by Yuan et al. was one of the pioneering work that extended classic Lasso to Lasso with group sparsity. Its objective function shown in (2.8) shows that instead of regularizing each w_j individually it regularizes $\mathbf{w}$ in a group level by partitioning $\mathbf{w}$ into G disjoint groups such that $\mathbf{w} = (\mathbf{w}_1^T, ..., \mathbf{w}_g^T, ..., \mathbf{w}_G^T)^T \in \mathbb{R}^{d \times 1}$, where $G \leq d$, $\mathbf{w}_g \in \mathbb{R}^{d_g \times 1}$, and $\sum_{g=1}^{G} d_g = d$.

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2} \left\| \sum_{g=1}^{G} \mathbf{X}_k \mathbf{w}_k - \mathbf{y} \right\|_2^2 + \lambda \sum_{g=1}^{G} \sqrt{l_g} \, \|\mathbf{w}_g\|_2 \tag{2.8}$$

where G is total number of groups and l_g is the length of $\mathbf{w}_g$. If G is equal to the dimension of the data matrix $\mathbf{X}$, (2.8) reduces to the Lasso problem. Group Lasso enforces the sparsity on group of parameters, however group Lasso is not adequate if one is also interested in finding relevant groups and additionally, relevant features *within* each group [22]. Sparse-group Lasso [22], regularizing the model with not only group sparsity and also within-group sparsity, was later proposed to address this problem. Its formulation (2.9) adds up L2-penalty on $\mathbf{w}_g$ at the group level (same as in 2.8) and L1-penalty on $\mathbf{w}$. α controls the balance between group sparsity or individual sparsity, and λ governs the overall sparsity level.

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2} || \sum_{g=1}^{G} \mathbf{X}_k \mathbf{w}_k - \mathbf{y} ||_2^2 + (1-\alpha)\lambda \sum_{g=1}^{G} \sqrt{l_g} || \mathbf{w}_g ||_2 + \alpha\lambda || \mathbf{w} ||_1 \tag{2.9}$$

Laurent et al. generalized the idea of group Lasso by allowing overlaps in groups and also proposed graph Lasso where the model coefficients are assumed to have the graph structure [24].

Despite that several works regarding group Lasso had been proposed, those group Lasso estimators lack meaningful variance estimates for model coefficients. Sudhir Raman et al. hence proposed the Bayesian approach for group Lasso using a hierarchical model where the prior on $\mathbf{w}_g$ was modeled by a Multi-Laplacian distribution shown in (2.10) [25].

$$\text{M-Laplace}(\mathbf{w}_g|\mathbf{0}, c^{-1}) \propto c^{d_g/2} exp(-c||\mathbf{w}_g||_2) \tag{2.10}$$

Several works on different priors have been proposed ever since. Daniel Hernández-Lobato et al. proposed generalized spike-and-slab priors for group feature selection [19]. Similar as (2.7), instead of imposing a prior on individual w_j, (2.11) imposes spike-and-slab prior at the group of model parameters, i.e. $\mathbf{w}_g$.

$$\begin{aligned} p(\mathbf{w}|\mathbf{z}) &= \prod_{g=1}^{G} \left[z_g \mathcal{N}(\mathbf{w}_g|\mathbf{0}, v\mathbf{I}) + (1 - z_g)\delta(\mathbf{w}_g) \right] \\ &= \prod_{d=1}^{D} z_{g(j)} \mathcal{N}(w_j|0, v) + (1 - z_{g(j)})\delta(w_j) \\ p(\mathbf{z}) = \text{Bern}(\mathbf{z}|\mathbf{p}) &= \prod_{g=1}^{G} \left[p_g^{z_g} (1 - p_g)^{(1-z_g)} \right] \end{aligned} \tag{2.11}$$

$g(j)$ is the group identifier indicating w_j belongs to group $g(j)$, z_g is thus the latent variable that is of Bernoulli distribution parameterized by p_g, where p_g is the prior probability or knowledge to inform the model whether the values of $\mathbf{w}_g$ are away from zero (not being sparse) or close to zero (being sparse). They have shown linear models with generalized group spike-and-slab priors outperformed group Lasso [16], Bayesian group Lasso [25], and some other variants on various datasets [19]. Michael R. Anderson et al. extended [19]'s model and proposed structured spike-and-slab priors that allow prior information of the group sparsity pattern be encoded on spike-and-slab prior through a generic covariance function [18].

Several studies have shown group sparsity can be useful in practice and number of applications. Yuwon Kim et al. showed that when the features contain a categorical observation which are then encoded into a collection (or group) of binary values, then the collection should share the same level of sparsity, i.e. either the collection is being left out or considered as a whole [21]. Besides, in gene expression data, genes would not be functioning individually as groups of genes may belong to the same pathway. Hence it is natural to regularize the regression model with group sparsity assumption when conducting pathway analysis [22], such as breast cancer study [24].

2.3 Bayesian Optimization

Consider the following optimization problem,

$$x^* = arg \max_{x \in X} f(x), \tag{2.12}$$

where we want to maximize the objective function f within space of interest X. In many realistic scenarios, the objective function has no closed form, but can be evaluated at any samples from X. Therefore, it can be considered as a sequential search algorithm. In the sequence of iterations, the optimizer decides the next location within X to sample based on the output y from f, and after several observations, the optimizer should decide the optimal estimate x^*. The principle of Bayesian optimization is to utilize historic information available from previous observations of f to enhance the data efficiency, so it is suitable in the situation when evaluating f is expensive. There are two major ingredients in Bayesian optimization (Algorithm 1)[26]). The first one is to select the prior distribution which represents the belief about the unknown objective function. The second one is to decide the acquisition function $\alpha(\cdot)$, which is used to determine an optimal sequence of samples to evaluate by maximizing the utility or minimizing the regret[27]. Bayesian optimization has been widely applied to wide range of applications, including inferring navigation policy for robots [28], preference learning [29], automatic model selection (e.g. hyperparameter selection in machine learning algorithms or architecture configuration in deep neural net [30][31]), portfolio selection and allocation [32], and interactive user intent modeling [33] etc.

Algorithm 1 Bayesian Optimization

```
for n=1:N do
    Determine next $x_{n+1} \in$ X to evaluate by optimizing
        $x_{n+1} = \arg\max_x \alpha(x; x_{1:n}, y_{1:n})$
    Sample $x_{n+1}$ from objective function, and observe $y_{n+1}$
    Update the model
end for
```

We next introduce contextual multi-armed bandit problem, a particular problem that could be tackled under the Bayesian optimization framework in 2.3.1. In 2.3.2, we introduce Thompson sampling that can naturally control exploration and exploitation for solving contextual multi-armed bandit.

2.3.1 Contextual Multi-armed Bandits

The goal of the optimization is to minimize the cumulative regrets (2.13) after a sequence of N actions, and it can be seen as a contextual multi-armed bandit problem that the learning algorithm should optimize the trade-off between exploration and exploitation. That is, based on the observations of contextual information and payoff of each action, the learning algorithm sequentially chooses an action to take, while at the same time, it searches the potentially optimal action that can minimize the overall cumulative regret.

$$\text{cumulative regret} = N \cdot \mathbf{E}[f(x^*)] - \sum_{n=1}^{N} \mathbf{E}[f(x_n)], \tag{2.13}$$

The regret minimization problem has been widely studied and applied in varying applications. To cite a classic example, considering that according to the historical information of user's clicks, the ads system would like to suggest advertisements when a user visits an entry page of a website. However, the room for displaying the ads is very limited, so the ads system should take the right action, i.e. selecting the advertisements that interest the user most from the large pool of advertisements to maximize the click-through rate, hence minimizing the regret of not recommending the most interesting ads to the users. To cite another example, Li et al. [34] modeled personalized recommendation of news articles as a contextual bandit problem, also aiming at maximizing the click through rate of the news articles on Yahoo! webpages. Other application areas of contextual bandit are click-through rate prediction of sponsored search in Microsoft Bing Search [35], experimental design, control of complex system [36]. In short, contextual bandit were found useful in such contexts as the resource is always limited but the hope is that the system should be able to take the right action depending on the given context to minimize the regret (or equivalently, maximizing the sum of the rewards).

2.3.2 Thompson Sampling with Contextual Information

One particular algorithm to solve contextual multi-armed bandit problem is Thompson sampling [37]. The steps of Thompson sampling are, first, decide the prior distribution $P(\boldsymbol{\theta})$ which is the assumption of the model. Second, calculate the likelihood distribution $P(D_n|\boldsymbol{\theta})$, where $D_n = \{(r_i, c_i, a_i)\}_{i=1}^{n}$ represents the observed reward r_i and contextual information c_i of iteration i by pulling arm a_i. Then, based on the Bayesian inference, the prior and likelihood distribution induce the posterior distribution, shown as (2.14), which can be seen as the summary of previous observations and the uncertainty of the model as well as represents the probability of improvement. Therefore, in each iteration, the algorithm draws a sample of the parameter for each arm from the posterior distribution and ranks the arms based on the sample. According to the observations of playing the best arms, the model is updated. By the randomized process of sequentially drawing samples from the posterior and updating the model, Thompson sampling algorithm attempts to balance between exploration and exploitation and identify the optimal arm of the bandit. Recently, Thompson sampling has drawn many attentions because it has been demonstrated to achieve comparable regret with other methods, such as upper confidence bound (UCB) algorithm. [38] presented the competitive empirical results of applying Thompson sampling to display advertisements.

$$P(\boldsymbol{\theta}|D_n) \propto P(D_n|\boldsymbol{\theta})P(\boldsymbol{\theta}), \tag{2.14}$$

2.4 Topic Model

2.4.1 Topic Model for Information Retrieval

A modern information retrieval system often consists of a massive collection of documents, so an effective way to organize, manage and search the information contents is very crucial. Topic modeling techniques are capable of finding semantic representation in massive amounts of documents, discovering themes in words and documents, organizing the documents and keywords based on the relationships and patterns between them without human labeling the data. Incorporating topic models in the information retrieval system, the search engine can narrow down the search fields by “zoom in” to specific topics related to user’s search target, or “zoom out” to discover more diverse results [39]. Therefore, topic models have been applied widely in analyzing large-scale text database, exploring, and retrieving information. For instance, they have been considered as a vital feature in digitization [41], and been shown successfully applicable in scientific paper finding [42], modeling themes in huge newspaper dataset such as *The New York Times* or in *Wikipedia* [43].

2.4.2 Latent Dirichlet Allocation

One particularly promising approach for topic modeling is Latent Dirichlet Allocation (LDA) [40]. It is a generative and unsupervised probabilistic topic modeling algorithm that discovers the frequent co-occurrence of words in different topics and summarizes the topics of the documents by the words belonging to them. The topics in LDA are the distributions over a fixed set of words. For example, under the topic of "data analysis", the words of high probability may be "accuracy" and "algorithm". Besides, the documents related to similar topics may consist of the same set of words. For instance, if the documents are about "president election", they are very likely to use "candidate", "vote" or some other related words. The topic distribution of each document can be identified by discovering the structure of all the documents inside corpus without labeling the data beforehand. Thereby, LDA is useful to be applied to the large collection of unlabeled text data.

In generative model of LDA, the topics distribution over words is denoted as $\beta_{1:K}$, where K is the number of topics in the corpus and the topic distribution of each document is denoted as $\theta_{1:D}$. For each document d, a topic $z_d \in z$ is sampled from the corresponding distribution θ_d. A word is sampled from distribution β_k associated with topic z_d. Figure 2.2 shows the plate diagram of the probabilistic model, which describes the dependency of parameters. The joint distribution of hidden and observed variables in the probabilistic model is shown as (2.15). We are interested in computing the posterior distribution (2.16) which infers the structure of topics given the observed documents. However the posterior distribution is intractable and cannot be computed directly. Several posterior inference methods have been used to approximate the posterior, e.g. variational methods, or to draw sample from it, e.g. Gibbs sampling.

$$\begin{aligned} &p(\beta_{1:K}, \theta_{1:D,}, z_{1:D}, w_{1:D}) \\ &= \prod_{i=1}^{K} p(\beta_i) \prod_{i=d}^{D} p(\theta_i) (\prod_{n=1}^{N} p(z_{d,n}|\theta_i) p(w_{d,n}|\beta_{1:K}, z_{d,n})) \end{aligned} \tag{2.15}$$

$$p(\beta_{1:K}, \theta_{1:D}, z_{1:D}|w_{1:D}) = \frac{p(\beta_{1:K}, \theta_{1:D}, z_{1:D}, w_{1:D})}{p(w_{1:D})} \tag{2.16}$$

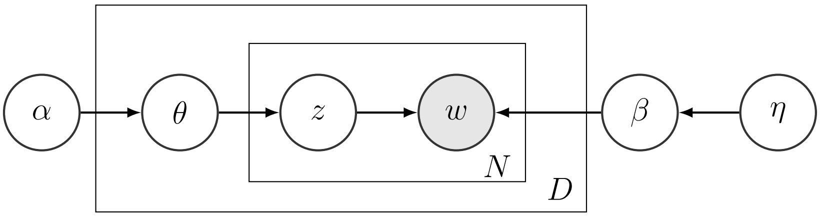


Figure 2.2: Graphic representation of LDA [39]

Chapter 3

Probabilistic User Model

In this chapter, we walk through the core algorithms applied in this thesis. We first introduce coupled multi-armed bandit algorithm [5] for exploratory search and the probabilistic user model incorporated with it. The work in this thesis aims to improve the user intent modeling in [5] in terms of enhancing estimated accuracy and maximizing accumulated rewards on user's feedback in rounds of interactions with the user. This is achieved by replacing the prior distribution with two sparse priors, namely, spike-and-slab prior with and without group information. The formulations and descriptions of the model with those two alternative sparse priors are discussed subsequently. Finally, we explain how to solve the model with those two prior models with efficient Expectation Propagation (EP) algorithm.

3.1 Coupled Multi-armed Bandit

In a practical exploratory search system [2], the user tends to provide limited feedback, so it is difficult for the system to accurately estimate user's intent model. To tackle this problem, [5] introduced the coupled multi-armed bandit algorithm where the user is allowed to give feedback on the arms (documents) and features of the arms (keywords). Given a set of document D in corpus and keywords K extracted from D, the relevance of the documents r_d and keyword r_k to user's search intent are defined as random variables over [0,1] with the expected values $E_D[d]$ and $E_K[k]$, where document $d \in D$ and keyword $k \in K$. The expected relevance of keywords and documents are bridged by $E_D[D] = \mathbf{M} \cdot E_K[K]$ under the assumption that a document-keyword matrix $\mathbf{M} \in \mathbb{R}^{|D| \times |K|}$ exists, where $E_D[D] = (E_D[d_1], ..., E_D[d_{|D|}])^T$, $E_K[K] = (E_K[k_1], ..., E_K[k_{|K|}])^T$, and $\mathbf{M}$ is defined as

$$\mathbf{M} = \begin{pmatrix} P(k_1|d_1) & \dots & P(k_{|K|}|d_1) \\ P(k_1|d_2) & & P(k_{|K|}|d_2) \\ \vdots & \ddots & \vdots \\ P(k_1|d_{|D|}) & \cdots & P(k_{|K|}|d_{|D|}) \end{pmatrix}_{|D|\times|K|} \tag{3.1}$$

where $P(k_i|d_j)$ is the likelihood of document d_j generating keyword k_i. It is assumed that the expected relevance of keywords is linearly related to the feature vectors by the unknown weight vector $\boldsymbol{\theta} \in \mathbf{R}^{|D|}$. Under this assumption, one can define the expected relevances of keywords and documents as (3.2),

$$\begin{aligned} E_K[K] &= \mathbf{M}^T\boldsymbol{\theta} \\ E_D[D] &= \mathbf{M}\mathbf{M}^T\boldsymbol{\theta}. \end{aligned} \tag{3.2}$$

Following how expected relevances of keywords and documents are defined in (3.2), in [5] the random variables r_k and r_d defining the user's relevance feedback on keywords and documents are modeled as Gaussian distributions shown as equation 3.3.

$$\begin{aligned} r_k &= N(\mathbf{x}_k^T\boldsymbol{\theta}, \beta_K^2) \\ r_d &= N(\mathbf{x}_d^T\boldsymbol{\theta}, \beta_D^2), \end{aligned} \tag{3.3}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{|D|}$ is shared among two linear models, $\mathbf{x}_k$ is the k^{th} column of $\mathbf{M}$, $\mathbf{x}_d$ is the d^{th} column of $\mathbf{M}\mathbf{M}^T$. β_K^2 and β_D^2 are the variance of Gaussian noise. The likelihood of user feedback is modeled as

$$\begin{aligned} P(\mathbf{r}_{\mathbf{n}_D}, \mathbf{r}_{\mathbf{n}_K}|\mathbf{x}_{\mathbf{n}_D}, \mathbf{x}_{\mathbf{n}_K}, \boldsymbol{\theta}) &= \prod_{d\in\mathbf{n}_D} P(r_d|\mathbf{x}_d, \boldsymbol{\theta}) \prod_{k\in\mathbf{n}_K} P(r_k|\mathbf{x}_k, \boldsymbol{\theta}) \\ &= \prod_{d\in\mathbf{n}_D} N(r_d|\mathbf{x}_d^T\boldsymbol{\theta}, \beta_D^2) \prod_{k\in\mathbf{n}_K} N(r_k|\mathbf{x}_k^T\boldsymbol{\theta}, \beta_K^2) \end{aligned} \tag{3.4}$$

where $\mathbf{n}_K$ and $\mathbf{n}_D$ are sets of feedback on the keywords and documents received.

In [5], for computational simplicity, the prior of $\boldsymbol{\theta}$ is defined as a conjugate prior with Gaussian distribution of zero mean and η^2 being its variance shown as equation (3.5), so the posterior distribution of $\boldsymbol{\theta}$ is also a Gaussian distribution with closed form solution shown as equation (3.6).

$$\pi_0(\boldsymbol{\theta}) = N(\boldsymbol{\theta}|0, \eta^2\mathbf{I}) \tag{3.5}$$

$$
\begin{aligned}
&\pi(\boldsymbol{\theta}) = N(\boldsymbol{\theta}|\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \\
&\boldsymbol{\Sigma}_t^{-1} = \beta_D^{-2}\mathbf{X}_{\mathbf{n}_D}^T\mathbf{R}_{\mathbf{n}_D} + \beta_K^{-2}\mathbf{X}_{\mathbf{n}_K}^T\mathbf{R}_{\mathbf{n}_K} + \eta^2\mathbf{I} \\
&\boldsymbol{\mu}_t = \Sigma_t(\beta_D^{-2}\mathbf{X}_{\mathbf{n}_D}^T\mathbf{R}_{\mathbf{n}_D} + \beta_K^{-2}\mathbf{X}_{\mathbf{n}_K}^T\mathbf{R}_{\mathbf{n}_K}),
\end{aligned}
\tag{3.6}
$$

where $\mathbf{X}_{\mathbf{n}_D}$ and $\mathbf{X}_{\mathbf{n}_K}$ are $|\mathbf{n}_D|\times|D|$ and $|\mathbf{n}_K|\times|D|$ design matrices constructed by the feature vectors for observed documents in $\mathbf{n}_D$ and observed keywords in $\mathbf{n}_K$, respectively. $\mathbf{R}_{\mathbf{n}_D}$ and $\mathbf{R}_{\mathbf{n}_K}$ are $|\mathbf{n}_D| \times 1$ and $|\mathbf{n}_K| \times 1$ matrices of the observed relevancies, respectively. The search engine should recommend documents d^* and keywords k^* with highest expected relevance $E_D[d^*]$ and $E_K[k^*]$ to the user. Meanwhile, for utilizing user's limited feedback well, the search engine should decide which keywords and documents to show to the user, so the feedback on those keywords and documents can achieve the maximum gains of decreasing the uncertainty of user's model and improving the accuracy of estimating $\boldsymbol{\theta}$, leading to a more accurate estimate of relevance scores in earlier iterations.

The exploratory search system should choose the optimal documents and keywords to the user and balance between exploiting the predicting models and exploring the corpus space. The target is to minimize cumulative regret $cum\_regret = |\mathbf{n}_D|E_D[d^*] - \sum_{d\in\mathbf{n}_D} E_D[d]$ after receiving a set of feedback on documents $\mathbf{n}_D$. To achieve this, Thompson Sampling was applied to guide the exploration by the uncertainty of the posterior. As shown in Algorithm 2, in each iteration $\boldsymbol{\theta}$ was sampled from (3.6) to calculate the relevance scores for every document and keyword. The top relevant documents and keywords were recommended to the user. Based on user's feedback, the posterior distribution was updated. From the simulation results and user studies in the paper, it illustrated that considering user feedback on both keywords and documents improved the prediction accuracy and quality of exploratory search compared to the earlier methods.

Algorithm 2 Thompson Sampling for Coupled Multi-armed Bandits [5]

```
for $t = 1 : T$ do
    Draw $\boldsymbol{\theta}$ from posterior distribution (3.6)
    for document bandit: select $d^+ = \arg\max_{d\in D} x_D^T\boldsymbol{\theta}$
    for keyword bandit: select $k^+ = \arg\max_{k\in K} x_K^T\boldsymbol{\theta}$.
    Update the the posterior (3.6) based on user feedback and observed
  feature vectors.
end for
```

3.2 Probabilistic User Models under Sparse and Group Sparse Priors

3.2.1 Curse of Dimensionality

When modeling user's search intent on the large corpus, over thousands of model coefficients need to be estimated to predict the relevance of documents of interest. In practice, the feedback from the user is scarce and of some uncertainty. A linear predictor is usually applied to model this problem. However, the feature matrix M shown as (3.1) is in high dimensional space when the number of documents in the corpus is very large because the dimension of feature matrix M is modeled by all the documents, leading to underdetermined and ill-posed learning problem. Under this circumstance, there is no unique solution for the weight vector $\boldsymbol{\theta}$, and the estimation of $\boldsymbol{\theta}$ becomes difficult and possibly leads to over-fitting.

Besides, the drawback of the coupled multi-armed bandit [5], along with earlier methods(e.g. LinRel algorithm [6]) used in user's intent modeling [8] is the computational complexity of the inference being approximately cubic with respect to the size of the corpus. This makes the regret minimization part of those algorithms hard to scale up when the number of documents is large.

To address those difficulties, in the thesis, the models that enforce sparsity on the coefficients $\boldsymbol{\theta}$ under spike-and-slab priors are employed to allow only a few of all the coefficients or features become relevant to the prediction. Besides, an approximate algorithm, Expectation Propagation (EP) algorithm, with the time complexity $O(n^2d)$ is favored to approximate the posterior distribution. In the following sections, we describe how we incorporate the sparse model and the structure of the documents to improve the coupled multi-armed bandit [5]. The formulation of EP algorithm is also described accordingly.

3.2.2 Sparse Prior and Group Sparse Prior

In Bayesian inference, when the number of samples is way smaller than the number of features, the prior distribution will become the dominant factor in the posterior. Thereby, extracting meaningful prior knowledge about the modeled problem significantly affects the accuracy of the inference. Our proposed framework for enhancing predicting accuracy and scalability is based on this idea, and it imposes the assumption that the prior of $\boldsymbol{\theta}$ comes from a family of sparse priors in the prior distribution of coupled multi-armed bandit algorithm. This is motivated by the fact that in a big corpus, most of the documents are usually not relevant to the current information need of the user. In particular, generalized spike-and-slab prior shown in (3.7) is applied here due to its effectiveness over other sparse priors that had been reported in [19]. $\delta(\cdot)$ is a point probability mass centered at the origin, and $\mathbf{p}_0$ is the hyperparameter.

$$P(\boldsymbol{\theta}|\mathbf{z}) = \prod_{i=1}^{|D|}[z_i N(\boldsymbol{\theta}_i|\mathbf{0}, \eta\mathbf{I}) + (1 - z_i)\delta(\boldsymbol{\theta}_i)] \tag{3.7}$$

$$P(\mathbf{z}) = Bern(\mathbf{z}|\mathbf{p_0}) = \prod_{i=1}^{|D|}[p_{0,i}^{z_i}(1 - p_{0,i})^{1-z_i}] \tag{3.8}$$

Following the linear coupled bandit model described in Chapter 3.1, the relationship of model coefficients, observed variables and the hyperparameters can be seen in the plate diagram Figure 3.1.

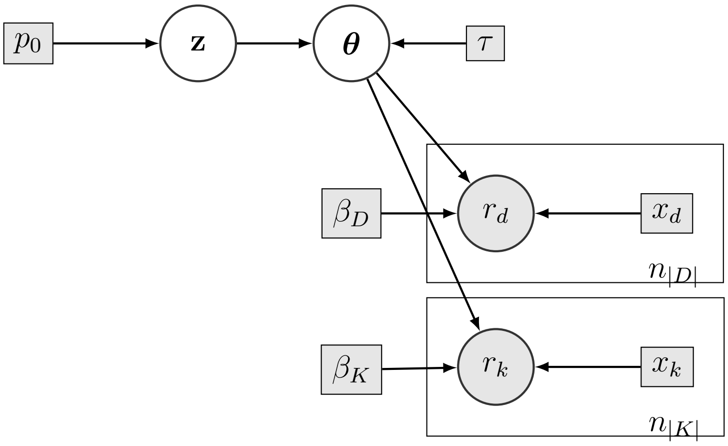


Figure 3.1: Probabilistic user model based on spike-and-slab prior.

In addition, the user intent modeled by previous methods was defined on documents and keywords, but it did not use the information on how keywords and documents can be represented in higher level cluster, e.g. topics. Therefore, we group the documents in topics based on the document-keyword relationships in the corpus. The prior distribution can use this topic information to generalize user feedback on individual keywords and documents, to the similar keywords and documents in the topic. To some extent, the clustering scales the intent learning problem from document space to a more abstract topic space in which the prior lies. By modeling the prior at the topic level, when the corpus size increases, the number of topics would not grow linearly with the corpus size. It is beneficial when one would like to apply the model in the larger scale of corpus, because the number of latent variables $\mathbf{z}$ shown in (3.10) to be estimated does not increase linearly with respect to corpus size. Besides, it is assumed that only specific topics are related to the user's search target. When the feedback is very limited, group information can help to model narrow down to few topics by identifying the most relevant ones matched to user's interest.

To implement the mentioned assumptions, we encode the group sparsity [19] to the coefficients of $\boldsymbol{\theta}$. That is, the documents are grouped into topics with LDA algorithm and from user's feedback, we can identify the topics which are related to user's search target. It is assumed that user's search target in one search session is confined to limited topics which indicates that the coefficients of $\boldsymbol{\theta}$ belonging to those topics should be different from zero. On the contrary, for the dimensions belonging to irrelevant topics and having no significant influence to the prediction, the coefficients should be close to zero. Group sparsity is encoded into $\boldsymbol{\theta}$ through formulating its prior as group spike-and-slab model [19] defined as follows.

$$P(\boldsymbol{\theta}|\mathbf{z}) = \prod_{i=1}^{G}[z_g N(\boldsymbol{\theta}_g|\mathbf{0}, \eta\mathbf{I}) + (1 - z_g)\delta(\boldsymbol{\theta}_g)] \tag{3.9}$$

where $\boldsymbol{\theta}_g, g = 1, ..., G$ are the disjoint partitions of $\boldsymbol{\theta}$ such that $\boldsymbol{\theta} = (\boldsymbol{\theta}_1^T, \boldsymbol{\theta}_2^T, ..., \boldsymbol{\theta}_G^T)^T$, and $\delta(\cdot)$ is a point probability mass centered at the origin. The prior for $\mathbf{z}$ is modeled as multivariate Bernoulli distribution,

$$P(\mathbf{z}) = Bern(\mathbf{z}|\mathbf{p_0}) = \prod_{g=1}^{G}[p_{0,g}^{z_g}(1 - p_{0,g})^{1-z_g}] \tag{3.10}$$

where the hyperparameter $p_{0,g}$ is the probability that the coefficients of g^{th} group are different from zero. In the next section, we introduce Expectation Propagation algorithm for estimating the model parameters in (3.9) and (3.10).

3.3 Expectation Propagation Approximation

This section introduces EP algorithm for estimating parameters of the model incorporated with group spike-and-slab prior. We directly discuss the EP approximation process for group spike-and-slab prior and skip that for spike-and-slab prior. Since one can notice that if the number of group is equal to the size of the corpus, the formulation of group spike-and-slab prior becomes equivalent to that of spike-and-slab prior and so does the EP approximation process.

The likelihood function (3.3) and the prior distribution (3.9) induce the joint posterior distribution $P(\boldsymbol{\theta}, \mathbf{z}, \mathbf{p}|\mathbf{r_k}, \mathbf{r_d})$, shown as Equation (3.11), which is intractable, so it is the target of approximation. Considering the time complexity and estimate accuracy, one can apply the deterministic EP algorithm [17], which uses simpler distribution Q from the exponential family to approximate the intractable Bayesian inference. Distributions of the exponential family, e.g. Bernoulli, Beta, or Gaussian distribution, can be referred to Appendix A in [20] for more details.

$$\begin{aligned} P(\boldsymbol{\theta}, \mathbf{z}, \mathbf{p}|\mathbf{r}) &\propto \prod_{d \in n_D} P(\mathbf{r_d}|\boldsymbol{\theta}) \prod_{k \in n_K} P(\mathbf{r_k}|\boldsymbol{\theta}) \prod_{i=1}^{d} P(\theta_i|z_i) \prod_{g=1}^{G} P(z_g|p_g) \\ &= \prod_{n=1}^{3} f_n(\boldsymbol{\theta}, \mathbf{z}, \mathbf{p}|\mathbf{r}) \end{aligned} \tag{3.11}$$

$$\begin{aligned} f_1(\boldsymbol{\theta}, \mathbf{z}) &= \prod_{d \in n_D} N(r_d|\mathbf{x}_d^T\boldsymbol{\theta}, \beta_D^2) \prod_{k \in n_K} N(r_k|\mathbf{x}_k^T\boldsymbol{\theta}, \beta_K^2) \\ f_2(\boldsymbol{\theta}, \mathbf{z}) &= \prod_{i=1}^{|D|} [(1 - z_{g(i)})\delta(\theta_i) + z_{g(i)} N(\theta_i|0, \eta)] \\ f_3(\boldsymbol{\theta}, \mathbf{z}) &= \prod_{g=1}^{G} [Bern(z_g|p_{0,g})] \end{aligned} \tag{3.12}$$

The joint posterior distribution can be written as the product of three factors, shown as (3.11) and (3.12), EP algorithm approximates each of the factors f_i in (3.12) by selecting a distribution $\hat{f}_i$ from exponential family. The distribution Q is the product of approximate factors $\hat{f}_1$, $\hat{f}_2$ and $\hat{f}_3$, shown as (3.13), which is analytically tractable and flexible enough to fit the approximating targets.

where Z is normalization constant and $\sigma(z) = \frac{1}{1+\exp(-z)}$ is the sigmoid function.

$$\begin{aligned}\hat{f}_1(\boldsymbol{\theta}, \mathbf{z}) &= \prod_{i=1}^{d} N(\theta_i|\hat{m}_{1,i}, \hat{v}_{1,i}) \\ \hat{f}_2(\boldsymbol{\theta}, \mathbf{z}) &= \prod_{i=1}^{d} N(\theta_i|\hat{m}_{2,i}, \hat{v}_{2,i}) Bern(z_i|\sigma(\hat{p}_{2,i)})) \\ \hat{f}_3(\boldsymbol{\theta}, \mathbf{z}) &= \prod_{g=1}^{G} Bern(z_g|\sigma(\hat{p}_{3,g}))\end{aligned} \tag{3.13}$$

In the EP algorithm, the parameters of $\hat{f}_1$, $\hat{f}_2$ and $\hat{f}_3$ are updated iteratively by minimizing the Kullback-Leibler (KL) divergence between $f_n(\boldsymbol{\theta}, \mathbf{z})Q^{\setminus n}(\boldsymbol{\theta}, \mathbf{z})$ and $\hat{f}_n(\boldsymbol{\theta}, \mathbf{z})Q^{\setminus n}(\boldsymbol{\theta}, \mathbf{z})$, where $Q^{\setminus i}(\boldsymbol{\theta}, \mathbf{z})$ is defined as Equation (3.14). This optimization is a convex problem in which a single solution can be found by expected sufficient statistics [17]. That is, the parameters of $\hat{f}_i$ are chosen to let the approximation distribution $\hat{f}_i(\boldsymbol{\theta}, \mathbf{z})Q^{\setminus i}(\boldsymbol{\theta}, \mathbf{z})$ closer to $f_n(\boldsymbol{\theta}, \mathbf{z})Q^{\setminus i}(\boldsymbol{\theta}, \mathbf{z})$. After updated approximate distribution Q^* is calculated, the i-th approximate factor $\hat{f}_n^*$ can be updated by (3.15). The new $Q(\boldsymbol{\theta}, \mathbf{z})$ can then be updated with (3.16). The iterations of optimizing the parameters of each approximate factor $\hat{f}_n^*$ and updating $Q^{\setminus i}(\boldsymbol{\theta}, \mathbf{z})$ continue until $Q(\boldsymbol{\theta}, \mathbf{z})$ converges.

In following sections 3.4.1, 3.4.2, 3.4.3, and 3.4.4, we follow the steps in [19] to compute the updates of the parameters in $\hat{f}_1(\boldsymbol{\theta}, \mathbf{z})$, $\hat{f}_2(\boldsymbol{\theta}, \mathbf{z})$, $\hat{f}_3(\boldsymbol{\theta}, \mathbf{z})$, and $Q(\boldsymbol{\theta}, \mathbf{z})$.

$$Q^{\setminus n}(\boldsymbol{\theta}, \mathbf{z}) = \frac{Q(\boldsymbol{\theta}, \mathbf{z})}{\hat{f}_n(\boldsymbol{\theta}, \mathbf{z})} \tag{3.14}$$

$$\hat{f}_n^*(\boldsymbol{\theta}, \mathbf{z}) \propto \frac{Q^*(\boldsymbol{\theta}, \mathbf{z})}{Q^{\setminus i}(\boldsymbol{\theta}, \mathbf{z})} \tag{3.15}$$

$$Q^{new}(\boldsymbol{\theta}, \mathbf{z}) \propto \hat{f}_n^*(\boldsymbol{\theta}, \mathbf{z}) \cdot Q^{\setminus n}(\boldsymbol{\theta}, \mathbf{z}) \tag{3.16}$$

3.4 Expectation Propagation Update Operations

3.4.1 Estimating Parameters for $\hat{f}_1$

From (3.12), $f_1(\boldsymbol{\theta}, \mathbf{z})$ is the product of two Gaussian distributions which is another Gaussian distribution. Because the estimated factor $\hat{f}_1(\boldsymbol{\theta}, \mathbf{z})$ is also in the form of Gaussian distribution parameterized by a mean $\hat{\mathbf{m}}_{\mathbf{1}}$ and a covariance matrix $\hat{\mathbf{v}}_{\mathbf{1}}$ which can be calculated directly. The parameters of $\hat{f}_1$ are

$$
\begin{aligned}
\hat{\mathbf{v}}_{\mathbf{1}} &= (\beta_D^{-2}\mathbf{X}_{\mathbf{n}_D}^T\mathbf{X}_{n_D} + \beta_K^{-2}\mathbf{X}_{\mathbf{n}_K}^T\mathbf{X}_{\mathbf{n}_K})^{-1} \\
\hat{\mathbf{m}}_{\mathbf{1}} &= \hat{\mathbf{v}}_1(\beta_D^{-2}\mathbf{X}_{\mathbf{n}_D}^T\mathbf{R}_{\mathbf{n}_D} + \beta_K^{-2}\mathbf{X}_{\mathbf{n}_K}^T\mathbf{R}_{\mathbf{n}_K})
\end{aligned}
\tag{3.17}
$$

3.4.2 Estimating Parameters for $\hat{f}_2$

Following the step in [19], to estimate parameters for $\hat{f}_2(\boldsymbol{\theta}, \mathbf{z})$, one requires, starting with computing $Q^{\setminus 2}(\boldsymbol{\theta}, \mathbf{z})$, going through (3.15) and (3.16). First, the term $\hat{f}_2(\boldsymbol{\theta}, \mathbf{z})$ can be factorized as

$$
\begin{aligned}
\hat{f}_2(\boldsymbol{\theta}, \mathbf{z}) &= \prod_{i=1}^{d} \hat{f}_{2,i}(\theta_i, z_i) \\
&= \prod_{i}^{d} N(\theta_i|\hat{m}_{2,i}, \hat{v}_{2,i}) Bern(z_i|\sigma(\hat{p}_{2,i})),
\end{aligned}
\tag{3.18}
$$

Each component of $\hat{f}_{2,i}$ consists of a univariate Gaussian distribution $N(\theta_i|\hat{m}_{2,i}, \hat{v}_{2,i})$ and a Bernoulli distribution $Bern(z_i|\sigma(\hat{p}_{2,i}))$. The values $\hat{m}_{2,i}$, $\hat{v}_{2,i}$ and $p_{2,i}$ are the targets of the approximation. $Q^{\setminus 2,i}(\theta_i, z_i)$ can be calculated as follows.

$$
\begin{aligned}
Q^{\setminus 2,i}(\theta_i, z_i) &= \frac{Q(\theta_i, z_i)}{\hat{f}_2(\theta_i, z_i)} \\
&\propto N(\theta_i|m_i^{\setminus 2,i}, v_i^{\setminus 2,i}) Bern(z_g(i)|\sigma(p_g^{\setminus 2,i}))
\end{aligned}
\tag{3.19}
$$

$$
\begin{aligned}
m_i^{\setminus 2,i} &= \hat{m}_{1,i} \\
v_i^{\setminus 2,i} &= \hat{v}_{1,i} \\
p_g^{\setminus 2,i} &= p_g - \hat{p}_{2,i}
\end{aligned}
\tag{3.20}
$$

Second, the target is to update $\hat{f}_2$ so that the KL divergence between $f_{2,i}(\theta_i, z_i)Q^{\setminus 2,i}(\theta_i, z_i)$ and $\hat{f}_{2,i}(\theta_i, z_i)Q^{\setminus 2,i}(\theta_i, z_i)$ is minimized. Finding the solution of this minimization is a convex optimization problem, and can be done by matching the expected sufficient statistics.

We define the updated $Q^*(\theta_i, z_i)$ as

$$Q^*(\theta_i, z_i) \propto N(\theta_i|m^*_{2,i}, v^*_{2,i})Bern(z_g(i)|\sigma(p^*_{2,i})), \tag{3.21}$$

For updating Q^*, one needs to calculate its parameters by finding the central moments of Q^* and first moment of z_i governed by

$$\Theta_m = \sum_{z_i} \int \theta_i^m f_{2,i}(\theta_i, z_i)Q^{\backslash 2}(\theta_i, z_i)d\theta_i, m = 0, 1, 2 \tag{3.22}$$

$$Z_1 = \sum_{z_i} \int z_i f_{2,i}(\theta_i, z_i)Q^{\backslash 2}(\theta_i, z_i)d\theta_i, \tag{3.23}$$

where Θ_0, Θ_1, and Θ_2 correspond to 0^{th}, 1^{st}, and 2^{nd} moments of Q^* with respect to θ_i while Z_1 is the 1^{st} moment of z_i with respect to θ_i. The update rule for the parameters of Q^*, $m^*_{2,i}$, $v^*_{2,i}$ and $p^*_{2,i}$ are given in (3.24).

$$\begin{aligned} m^*_{2,i} &= E[\theta_i] = \frac{\Theta_1}{\Theta_0} \\ v^*_{2,i} &= Var[\theta_i] = \frac{\Theta_2}{\Theta_0} - (\frac{\Theta_1}{\Theta_0})^2 \\ p^*_{2,i} &= \sigma^{-1}(E[z_i]) = \log \frac{E[z_i]}{1 - E[z_i]} = \log \frac{\frac{Z_1}{\Theta_0}}{1 - \frac{Z_1}{\Theta_0}}, \end{aligned} \tag{3.24}$$

where $E[z_i] = \frac{Z_1}{\Theta_0}$. The exact computation for evaluating Θ_0, Θ_1, Θ_2, and Z_1 are given as follows.

$$\begin{aligned} Z_1 &= \sum_{z_i} \int z_i f_{2,i}(\theta_i, z_i)Q^{\backslash 2}(\theta_i, z_i)d\theta_i \\ &= \sum_{z_i} \int z_i[(1 - z_{g(i)})\delta(\theta_i) + z_{g(i)}N(\theta_i|0, \eta)]N(\theta_i|\hat{m}_{1,i}, \hat{v}_{1,i})Bern(z_g(i)|\sigma(p_g^{\backslash 2,i}))d\theta_i \\ &= \int N(\theta_i|0, \eta_i)N(\theta_i|\hat{m}_{1,i}, \hat{v}_{1,i})\sigma(p_g^{\backslash 2,i})d\theta_i \\ &= \sigma(p_g^{\backslash 2,i})N(0|\hat{m}_{1,i}, \hat{v}_{1,i} + \eta_i) \end{aligned} \tag{3.25}$$

$$
\begin{aligned}
\Theta_0 &= \sum_{z_i} \int f_{2,i}(\theta_i, z_i) Q^{\backslash 2,i}(\theta_i, z_i) d\theta_i \\
&= \sum_{z_i} \int [(1 - z_{g(i)})\delta(\theta_i) + z_{g(i)} N(\theta_i|0,\eta)] N(\theta_i|\hat{m}_{1,i}, \hat{v}_{1,i}) Bern(z_g(i)|\sigma(p_g^{\backslash 2,i})) d\theta_i \\
&= \int [N(\theta_i|0,\eta_i) N(\theta_i|\hat{m}_{1,i}, \hat{v}_{1,i}) \sigma(p_g^{\backslash 2,i}) + \delta(\theta_i) N(\theta_i|\hat{m}_{1,i}, \hat{v}_{1,i}) \sigma(-p_g^{\backslash 2,i})] d\theta_i \\
&= \sigma(p_g^{\backslash 2,i}) N(0|\hat{m}_{1,i}, \hat{v}_{1,i} + \eta_i) + \sigma(-p_g^{\backslash 2,i}) N(0|\hat{m}_{1,i}, \hat{v}_{1,i}) \\
&= Z_1 + \sigma(-p_g^{\backslash 2,i}) N(0|\hat{m}_{1,i}, \hat{v}_{1,i})
\end{aligned}
\tag{3.26}
$$

$$
\begin{aligned}
\Theta_1 &= \sum_{z_i} \int \theta_i f_{2,i}(\theta_i, z_i) Q^{\backslash 2,i}(\theta_i, z_i) d\theta_i \\
&= \sum_{z_i} \int \theta_i [(1 - z_{g(i)})\delta(\theta_i) + z_{g(i)} N(\theta_i|0,\eta)] N(\theta_i|\hat{m}_{1,i}, \hat{v}_{1,i}) Bern(z_g(i)|\sigma(p_g^{\backslash 2,i})) d\theta_i \\
&= \int \theta_i N(\theta_i|0,\eta_i) N(\theta_i|\hat{m}_{1,i}, \hat{v}_{1,i}) \sigma(p_g^{\backslash 2,i}) d\theta_i + \int \theta_i \delta(\theta_i) N(\theta_i|\hat{m}_{1,i}, \hat{v}_{1,i}) \sigma(-p_g^{\backslash 2,i}) d\theta_i \\
&= \sigma(p_g^{\backslash 2,i}) N(0|\hat{m}_{1,i}, \hat{v}_{1,i} + \eta_i) \frac{\eta_i \hat{v}_{1,i}}{\eta_i + \hat{v}_{1,i}} \frac{\hat{m}_{1,i}}{\hat{v}_{1,i}} \\
&= Z_1 \frac{\eta_i \hat{v}_{1,i} \hat{m}_{1,i}}{(\eta_i + \hat{v}_{1,i}) \hat{v}_{1,i}}
\end{aligned}
\tag{3.27}
$$

$$
\begin{aligned}
\Theta_2 &= \sum_{z_i} \int \theta_i^2 f_{2,i}(\theta_i, z_i) Q^{\backslash 2,i}(\theta_i, z_i) d\theta_i \\
&= \sum_{z_i} \int \theta_i^2 [(1 - z_{g(i)})\delta(\theta_i) + z_{g(i)} N(\theta_i|0,\eta)] N(\theta_i|\hat{m}_{1,i}, \hat{v}_{1,i}) Bern(z_g(i)|\sigma(p_g^{\backslash 2,i})) d\theta_i \\
&= \int \theta_i^2 N(\theta_i|0,\eta_i) N(\theta_i|\hat{m}_{1,i}, \hat{v}_{1,i}) \sigma(p_g^{\backslash 2,i}) d\theta_i + \int \theta_i^2 \delta(\theta_i) N(\theta_i|\hat{m}_{1,i}, \hat{v}_{1,i}) \sigma(-p_g^{\backslash 2,i}) d\theta_i \\
&= \sigma(p_g^{\backslash 2,i}) N(0|\hat{m}_{1,i}, \hat{v}_{1,i} + \eta_i) [\frac{\eta_i + \hat{v}_{1,i}}{\eta_i \hat{v}_{1,i}} + (\frac{\eta_i \hat{v}_{1,i} \hat{m}_{1,i}}{(\eta_i + \hat{v}_{1,i}) \hat{v}_{1,i}})^2] \\
&= Z_1 [\frac{\eta_i + \hat{v}_{1,i}}{\eta_i \hat{v}_{1,i}} + (\frac{\eta_i \hat{v}_{1,i} \hat{m}_{1,i}}{(\eta_i + \hat{v}_{1,i}) \hat{v}_{1,i}})^2]
\end{aligned}
\tag{3.28}
$$

Finally, we update $f_{2,i}$ by

$$\begin{aligned}\hat{f}^{new}_{2,i} &= \frac{Q^*(\theta_i, z_i)}{Q^{\backslash 2,i}(\theta_i, z_i)} \\ &\propto N(\theta_i|\hat{m}^{new}_{2,i}, \hat{v}^{new}_{2,i})Bern(z_g(i)|\sigma(\hat{p}^{new}_{2,i}))\end{aligned} \tag{3.29}$$

where

$$\begin{aligned}\hat{v}^{new}_{2,i} &= [v^{*-1}_{2,i} - \hat{v}^{-1}_{1,i}]^{-1} \\ \hat{m}^{new}_{2,i} &= v^{new}_{2,i}[v^{*-1}_{2,i}m^*_{2,i} - \hat{v}^{-1}_{1,i}\hat{m}_{1,i}] \\ p^{new}_{2,i} &= p^*_{2,i} - \hat{p}^{\backslash 2,i}_g.\end{aligned} \tag{3.30}$$

3.4.3 Estimating Parameters for $\hat{f}_3$

The approximate factor $\hat{f}_3$ has the same form of exact factor f_3, and they are products of Bernoulli distributions for each component of $\mathbf{z}$. Hence, the parameter $\hat{p}_{3,g}$ of $\hat{f}_3$ can be computed directly by

$$\begin{aligned}\sigma(\hat{p}_{3,g}) &= p_{0,g} \\ \hat{p}_{3,g} &= \sigma^{-1}(p_{0,g}) = \log\frac{p_{0,g}}{1 - p_{0,g}}\end{aligned} \tag{3.31}$$

where $\sigma^{-1}(z) = \log\frac{z}{1-z}$ is the logit function, the inverse of the sigmoid function.

3.4.4 Estimating Parameters for Q

After evaluating $\hat{f}_1$, $\hat{f}_2$, and $\hat{f}_3$, the parameters of Q in (**??**) can be obtained by the following rules.

$$\begin{aligned}\mathbf{V} &= (\hat{\mathbf{v}}_1^{-1} + \hat{\mathbf{v}}_2^{-1})^{-1} \\ &= [(\beta_D^{-2}\mathbf{X}^T_{\mathbf{n}_D}\mathbf{X}_{\mathbf{n}_D} + \beta_K^{-2}\mathbf{X}^T_{\mathbf{n}_K}\mathbf{X}_{\mathbf{n}_K})^{-1} + \hat{\mathbf{v}}_2^{-1}]^{-1}\end{aligned} \tag{3.32}$$

$$\begin{aligned}\mathbf{m} &= \mathbf{V}(\hat{\mathbf{v}}_1^{-1}\hat{\mathbf{m}}_1 + \hat{\mathbf{v}}_2^{-1}\hat{\mathbf{m}}_2) \\ &= \mathbf{V}[(\beta_D^{-2}\mathbf{X}^T_{n_D}\mathbf{R}_{n_D} + \beta_K^{-2}\mathbf{X}^T_{n_K}\mathbf{R}_{n_K}) + \hat{\mathbf{v}}_2^{-1}\hat{\mathbf{m}}_2]\end{aligned} \tag{3.33}$$

$$p_g = \sum_{g(i)=g} \hat{p}_{2,i} + \hat{p}_{3,g}, g = 1, ..., G. \tag{3.34}$$

The complexity to calculate (3.32) requires the inversion of $d \times d$ matrix. Woodbury formula can be applied for faster evaluating the inversion at the cost of $O(n^2 d)$ which is much lower than standard inversion, $O(d^3)$. Hence by applying Woodbury formula on (3.32), it becomes

$$\mathbf{V} = \mathbf{V}' + \mathbf{V}'\mathbf{X}_{nD}^T(\mathbf{I}_k + \mathbf{X}_{nD}\mathbf{V}'\mathbf{X}_{nD}^T)^{-1}\mathbf{X_{nD}}, \tag{3.35}$$

where

$$\mathbf{V}' = \hat{\mathbf{v}}_\mathbf{2} + \hat{\mathbf{v}}_\mathbf{2}\mathbf{X}_{nK}^T(\mathbf{I}_k + \mathbf{X}_{nK}\hat{\mathbf{v}}_\mathbf{2}\mathbf{X}_{nK}^T)^{-1}\mathbf{X_{nK}} \tag{3.36}$$

3.5 Thompson Sampling with Sparse Priors

Recall that for minimizing the regrets by utilizing Thompson sampling as shown in Algorithm 2, instead of drawing θ with posterior distribution under Gaussian prior as it did in [5], the proposed methodology draws posterior distribution under the spike-and-slab priors presented in the above sections. One can possibly learn the model under such sparse priors with fewer samples and be able to make the predictions mostly based on only the relevant coefficients, which is tailored to the use case in this thesis where the feedback from users is usually scarce. Accordingly, it is expected that drawn θ can represent user intent better in the way of eliminating irrelevant dimensions which might affect the prediction. The control between exploration and exploitation is still governed by Thompson sampling and it is expected that Thompson sampling incorporating spike-and-slab priors can conduct more meaningful exploration and exploitation on documents that could be more relevant to the user.

To be exact on how our proposed framework is differed from Algorithm 2, we modify line 2 in Algorithm 2 by replacing the posterior used to draw $\boldsymbol{\theta}$ with the marginal distribution of $Q(\boldsymbol{\theta}, \mathbf{z})$ for each components of $\boldsymbol{\theta}$ (where $\mathbf{z}$ is marginalized out), which is a Gaussian distribution $N(\boldsymbol{\theta}|\mathbf{m}, \mathbf{v})$ in (**??**). The other parts of algorithm stay unchanged.

Chapter 4

Simulation and Experiments

This chapter discusses simulations and experiments. First, we demonstrate how to simulate user behavior in the exploratory search context, e.g. the procedure explaining how would user score the relevance for each keyword and document presented by the system. Second, we explain the evaluation metrics for comparing models' performance, and discuss the comparison scenario set up for investigation. To be specific, we mainly compare three scenarios which apply models under distinct priors, i.e. coupled multi-armed bandit with Gaussian prior in [5], coupled multi-armed bandit with spike-and-slab prior, coupled multi-armed bandit with group spike-and-slab prior. In addition, we would like to validate that using coupled multi-armed bandit algorithm under spike-and-slab prior and group spike-and-slab prior can also lead us to the conclusion stated in [5] that it can improve the performance and quality of exploratory search comparing to those use single source of feedback, e.g. feedback only on documents or keywords. Finally, we define experiment parameters, present and discuss the experimental results.

4.1 Simulation Setting

4.1.1 Dataset Description

Data extracted from arXiv repository (`http://arxiv.org/`), an on-line open access repository for scientific reports and journals were used throughout the simulations and all of the experiments conducted in this thesis. ArXiv, hosted by Cornell University, is a open repository consistently growing and currently consisting of over one million scientific papers across six main categories, i.e. Physics, Mathematics, Computer Science, Quantitative Biology, Quantitative Finance, and Statistics. Our particular interest lies in Computer Science category consisting of 40 subcategories, such as Artificial Intelligent, Information Retrieval, Machine Learning, and Computation Language etc. Author lists, abstracts, and full-text documents are made completely accessible.

Document-keyword matrix $\mathbf{M}$ defined in (3.1) is constructed with keywords extracted in the abstracts of documents of interest. To be more specific, keywords are extracted by treating texts in the abstracts as bag-of-words and thus $\mathbf{M}$ can be constructed with normalized tf-idf representation, where $|K|$ is the size of bag-of-words and $|D|$ is the number of total documents of interest.

4.1.2 Relevance scores of documents and keywords

In the simulation, the relevance scores treated as ground truth are generated by scoring the relevance of all the documents and keywords in the corpus. The steps are as follows:

1. Select a target document and generate a set of corresponding target keywords. For example, if the simulated search target is the article "Reinforcement Learning: A Survey", the related keywords are "reinforcement", "learning", "exploration", "exploitation" and so on.

2. Tokenize the documents and keywords and remove the stop words from the corpus. The documents are represented as "bag-of-word", and the stop words which occur with high frequency but are irrelevant to the searching purpose are filtered out.

3. Calculate tf-idf weighting for every word. The words are weighted by considering the term frequency and inverse document frequency. A word which occurs frequently in one document but rarely in other documents is considered to be more important, so it is assigned with higher weight.

4. Transform text to vector by *Latent Semantic Indexing* (LSI)[44]. The documents and keywords are transformed to a space where the keywords and documents of similar concepts and semantic meanings have similar representations.

5. Score the relevance of all the documents and keywords in $[0, 1]$ by measuring the similarities of them to the target documents and keywords. The cosine similarity measure is used in the simulation.

After calculating the relevance scores of documents $E_D[d]$ and keywords $E_K[k]$, we simulate that a user gives feedback on recommended documents and keywords with probabilities r_d and $|r_k - 0.5|$, respectively, where $r_d \sim N(E_D[d], \beta_D^2)$ and $r_k \sim N(E_K[k], \beta_K^2)$. The rationale of assigning $|r_k - 0.5|$ as the probability of giving feedback on keywords is due to user usually giving feedback to keywords which are highly relevant or irrelevant.

4.1.3 Model and System Parameters

In the simulation, 2,000 computer science arXiv articles are used in the search pool, and 50 of them are the documents with high relevant scores in user's intent model. The number of topics for those 2000 documents is fixed as 100. The model parameters are set as $\beta_D = \beta_K = 0.3$, which represent the uncertainties of user's intent model. The parameter η for the prior of Gaussian distribution in eq.(3.5) is set to 0.5. For spike and slab prior shown as (3.7), the parameter τ is set to 1 and the hyperparameter $p_{0,g}$ is set to 0.2. We simulate that the user interacts with the search system for 15 rounds, and in each round, 5 documents and 5 keywords are shown to the user for the feedback giving process.

4.1.4 Topic groups

For exploiting the group structure of the documents to improve the estimate accuracy, all of the documents in the corpus are divided into non-overlapping groups according to the distribution over topics of each document. That is, documents are assigned to a mixture of topics with different probabilities after applying LDA clustering algorithm on them, and each document is assigned to only one topic which is of highest probability. The number of topics in the corpus is pre-defined as 100. Figure 4.1 shows the number of documents under these 100 topics. It can be observed that the number of documents in each topic is varying, so the group sizes in the model are uneven. Besides, in the generated ground truth, the relevant documents distribute over around 30 percent of the total topics.

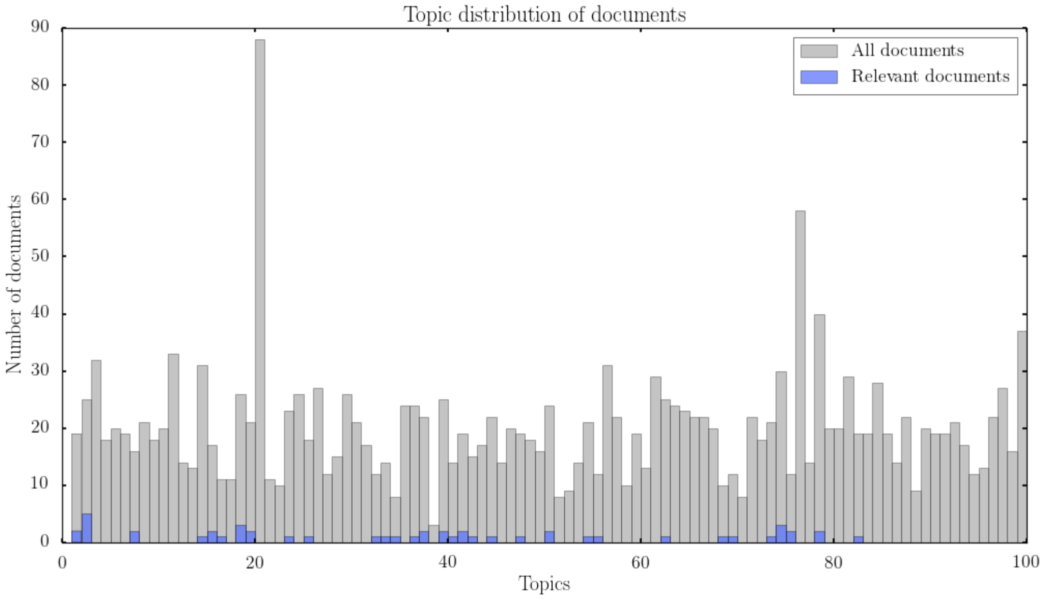


Figure 4.1: The histogram of number of documents under 100 topics.

4.2 Evaluation Methodology

4.2.1 Evalution Metrics

In the information retrieval system, precision and recall are widely used to evaluate the system performance. *Precision* is defined as the fraction of retrieved documents that are relevant to user's search target and *Recall* is defined as the fraction of relevant documents that are retrieved by the search engine. The equations for calculating recall and precision are shown in (4.1) and (4.2).

$$\text{Precision} = \frac{\text{No. of relevant documents retrieved}}{\text{No. of documents retrieved}} \tag{4.1}$$

$$\text{Recall} = \frac{\text{No. of relevant documents retrieved}}{\text{No. of relevant documents in the database}} \tag{4.2}$$

There usually is a trade-off between precision and recall. The search engine can increase recall by retrieving more documents. However, in the meantime, more irrelevant documents could also be retrieved so as to decrease precision. Therefore, the search engine typically can not achieve high recall and precision altogether. Different applications could have different preferences on optimizing precision and recall. For instance, if a user hopes the retrieved information to be short but precise, the search system should maintain high precision. In this simulation, precision and recall are used to evaluate how relevant the retrieved documents are to user's search intent, and how many relevant documents are retrieved after interacting with the search system.

Another evaluation method used in the simulation is accumulated expected relevance of documents. It is assumed that according to user's search intent, each document has a relevance score ranged from 0 to 1. Accumulated expected relevance score is calculated by summing over the relevance score of all the documents selected by the user over the iterations that have been through.

$$\text{Accumulated expected relevance of documents} = \sum_{d \in n_D} E_D[d] \tag{4.3}$$

4.2.2 Comparison Scenarios

Three different scenarios are compared in the simulation. The first one is Bayesian linear regression model with Gaussian prior. Because the likelihood function is also in the form of Gaussian distribution, using conjugate prior shares the benefit of existing closed form solution for the posterior distribution. However, the model does not encourage sparsity. The second one is to use spike and slab prior, shown as (3.7), which assumes most of the features are irrelevant to the predicting model and puts most of the coefficients to be zero or close to zero. The third one is spike and slab prior with group information, shown as (3.9). The prior leverages the observed topic structure (inferred by LDA in our case) to group documents into different topic groups.

We compare these three models in terms of precision, recall, accumulated expected relevance of documents, and also investigate the model coefficients learned by each model. These comparisons can offer us insight on the influence of different sparsity assumptions imposed to the models, i.e. non-sparse, sparse at the individual level, and sparse at group level.

Furthermore, the results of considering user's feedback on keywords, documents and both of them (coupled method) are compared. Following the simulation results in [5], we would like to see if the coupled method can still improve the predicting accuracy of the model using spike-and-slab prior with or without group information.

4.3 Evaluation Results

4.3.1 Estimates of Model Parameters

Figure 4.2 shows the estimated coefficients $\boldsymbol{\theta}$ of using Gaussian prior, sparse prior and group sparse prior for $\boldsymbol{\theta}$ in the Bayesian linear model after 15 rounds of interactions with search system. We would like to discuss the properties of different priors, the feature selection effect of using spike-and-slab prior, and the benefits and drawbacks of enforcing group sparsity.

For facilitating the comparison, the indices of the model coefficients shown in horizontal axis are ordered by the relevance of corresponding documents. The fifty leftmost dimensions on the figures are considered to be more relevant to the predicting model than others. In principle, the coefficients estimated in those fifty dimensions are expected to be larger than others. In other words, one would not like to obtain the result where the coefficients in dimensions corresponding to irrelevant documents are as close to zero as possible. Furthermore, one can expect that the discrepancy between the estimated coefficients of relevant and irrelevant features should be visibly large if the model has learned to identify the most relevant dimensions.

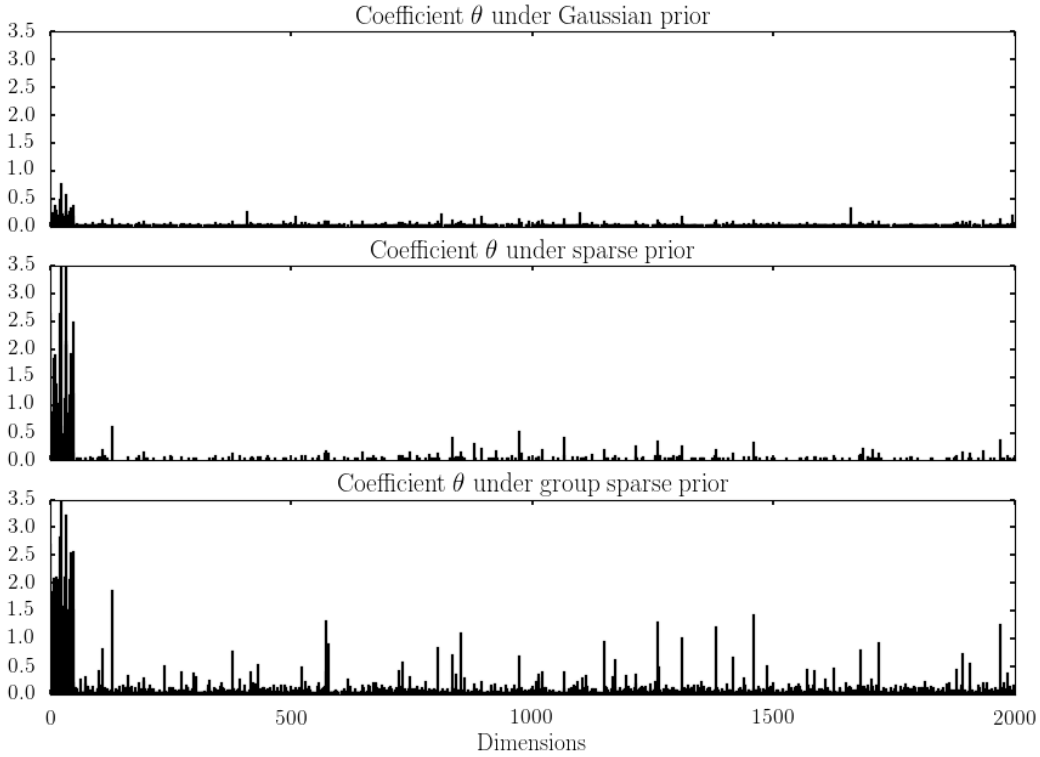


Figure 4.2: The estimated expected coefficients $\boldsymbol{\theta}$ under Gaussian prior, sparse prior, and group sparse prior after 15 rounds of interactions

For the model with Gaussian distribution with zero mean as the prior, most coefficients of the irrelevant features are close to zero. However, the discrepancy between the estimated coefficients of relevant and irrelevant features is not as obvious as the other two scenarios. This is especially the case in the first few iterations when the feedback from the user is scarce, and the dimensions are much higher than available observed data, it is difficult for the non-sparse model to learn the coefficients and identify the important and relevant features.

When using spike-and-slab prior, it enlarges the discrepancy between the estimated coefficients of relevant and irrelevant features as shown in the middle of Figure 4.2. On the one hand, the slab part of the mixture model allows the coefficients of relevant features to have larger values. On the other hand, the spike part of the mixture model enables the coefficients of unimportant features remains small and close to zero. It can be seen as an automatic feature selection process which emphasizes the features that are more relevant, and eliminates the influence of irrelevant features by putting small values on those coefficients, preventing over-fitting when available observations are limited.

In the case of using spike-and-slab prior with group information, the grouping effect allows the model to identify small subset of features that are possibly relevant to the model in fewer observations as it considers the additional information that the features in the same groups with the relevant features are the potential candidates of important features to the model. Therefore, from Figure 4.2 one can find that the coefficients $\boldsymbol{\theta}$ of relevant features are higher than those learned with the other two scenarios. However, some irrelevant features have the coefficients different from zero because they are likely in the same group with the relevant features. Overall, the discrepancy of coefficients $\boldsymbol{\theta}$ between relevant and irrelevant features is the largest comparing to other scenarios when using spike-and-slab prior with group information. It can be seen as a group feature selection process that the model selects some groups of features to be relevant to the predicting model and to have a higher probability of coefficients different from zero.

To further discuss the influence of using group spike-and-slab prior, the estimated expected values of the latent variable $\mathbf{z}$, the indicator of how likely the group is relevant to the prediction, for selecting the relevant topics are shown as Figure 4.3. The expected $\mathbf{z}$ of 100 groups indicate the probability of coefficients $\boldsymbol{\theta}$ within each topic group are different from zero. From Figure 4.3 the estimated $\mathbf{z}$ is mostly proportional to the number of relevant documents across 100 topics. For the topics that are more relevant to user's intent, the expected values of $\mathbf{z}$ are higher. Comparing to using spike-and-slab prior without group information, there are fewer parameters needed to be estimated for the model by grouping features according to the similarity, so the model complexity is lower.

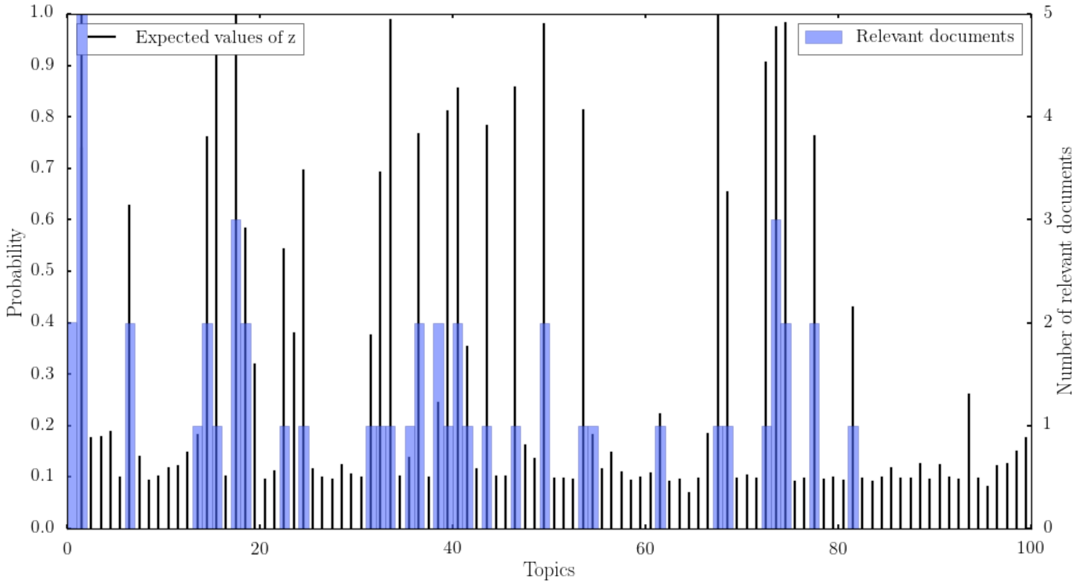


Figure 4.3: Estimated expected coefficients $\mathbf{z}$ and topic distribution of relevant documents. The horizontal axis indexes the topics. The blue shaded bars whose scale is shown in right vertical axis represent the number of relevant documents across 100 topics. The black thinner bars whose scale is shown in the left vertical axis represent the value of estimated $\mathbf{z}$.

4.3.2 Expected Rewards

In this section, we would like to evaluate the performance of the retrieval system in the perspective of how well does the interactive model predict the user's intent and recommend the relevant documents to the user accordingly. We run the simulation 20 times for each scenario, and plot the histogram over sum of expected relevances in the independent runs. Figure 4.4 shows the sum of expected relevances of selected documents that are predicted with Gaussian prior, spike-and-slab prior and spike and slab prior with group information. Among the results, the model with Gaussian prior performs the worst as the peak of the distribution is at around 13, the lowest among the three. The other two methods perform similarly while the model with group information performs slightly better than that without group information. The variances of those with spike-and-slab priors, with and without group information are also similar while that with Gaussian prior varies more in terms of expected relevances.

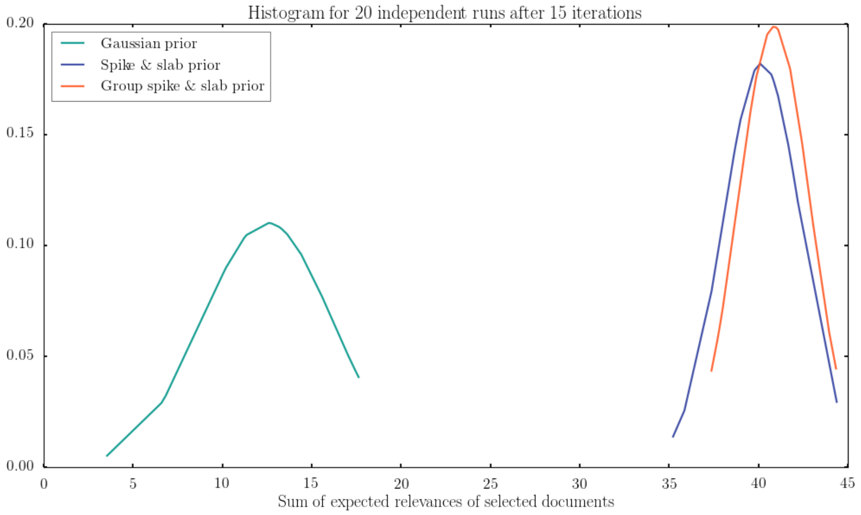


Figure 4.4: Histogram for 20 independent runs of the models over Gaussian prior, spike-and-slab prior as well spike-and-slab prior with group information.

Figure 4.5 compares the accumulated expected relevance of selected documents of considering the feedback on both keywords and documents, keywords only and documents only under Gaussian prior, spike-and-slab prior, as well as spike-and-slab prior with group information. Among three comparing scenarios, allowing user to give feedback on both keywords and documents outperforms the other two in terms of higher accumulated expected rewards after the user interact with the system for 15 iterations. Therefore, leveraging the feedback on both keywords and documents for the model can achieve a more accurate predicting accuracy of user's intent model in earlier iterations. Besides, it is observed that the feedback on keywords is more crucial than the feedback on documents, because it significantly increases the relevant documents selected by the user in the the earlier rounds.

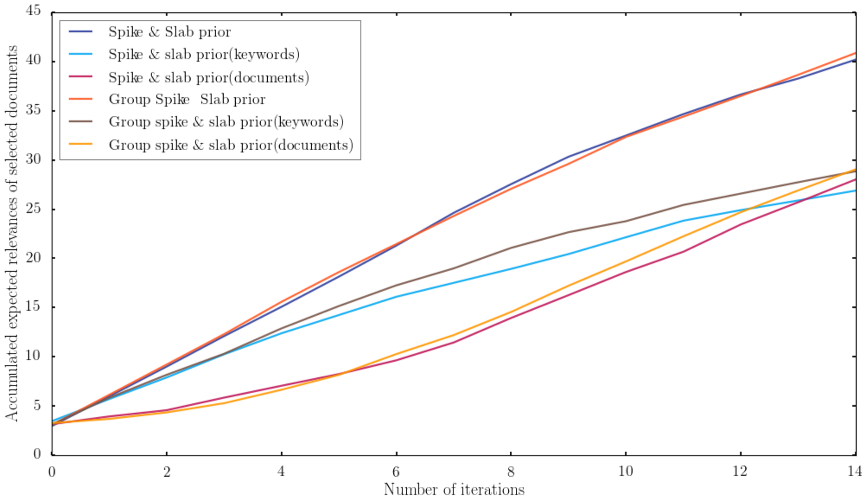


Figure 4.5: Accumulated expected relevances of selected documents in each iteration with feedback on both keywords and documents, keywords only, and documents only in two scenarios under spike-and-slab prior with and without group information.

4.3.3 Precision and Recall

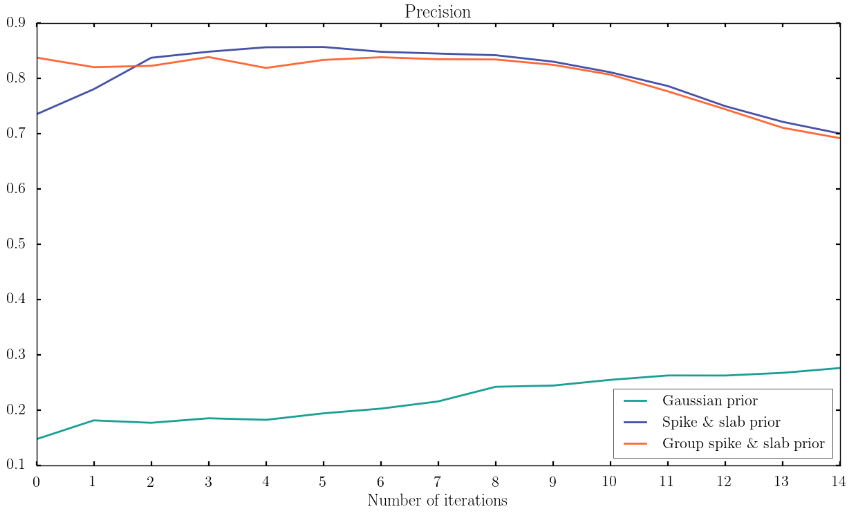


Figure 4.6: The average precision of 20 independent runs in each iteration obtained from the model under Gaussian prior, spike-and-slab prior as well spike-and-slab prior with group information

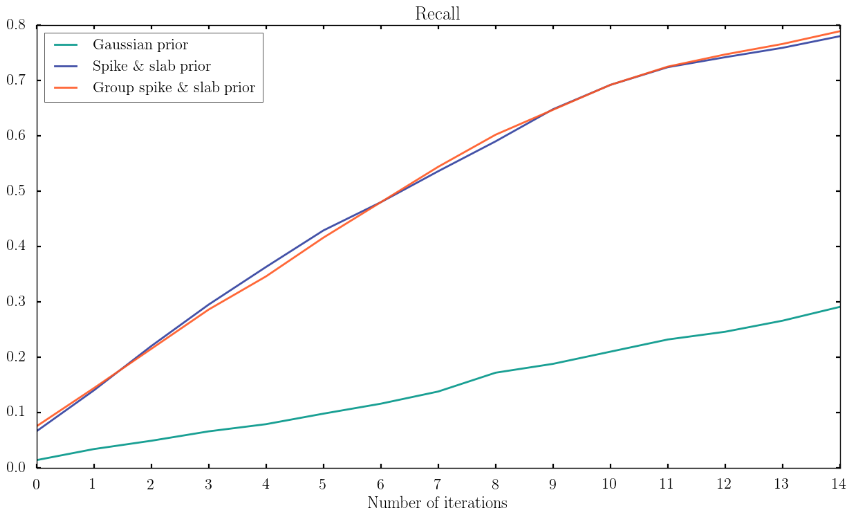


Figure 4.7: The average recall of 20 independent runs in each iteration obtained from the model under Gaussian prior, spike-and-slab prior as well spike-and-slab prior with group information

Figure 4.6 and 4.7 show the precision and recall of the retrieval system. Observing precision and recall allows us to capture if the search system tends to "explore" to less relevant documents or "exploit" highly relevant documents when it iterates. The precision and recall are very similar no matter if group information is used or not. Besides, models using spike-and-slab prior with or without group information which encourages sparsity outperform the model using Gaussian prior. Generally, in the scenarios of spike-and-slab prior with and without group information, the precision values decrease after around eight iterations, because it becomes more difficult to identify the remained relevant documents when most of the relevant ones have already been found. However, the model keeps exploring the documents space to recommend potential relevant documents to the user, so the recall keeps increasing. After fifteen iterations, around 80 percent of relevant documents are retrieved. Overall, using spike and slab prior with group information achieves slightly lower precision. However, in terms of retrieving the possible relevant documents, it performs slightly better than that without group information.

Chapter 5

Conclusions

The aim of this thesis was to model the user's search intent in an exploratory search task. Within a few iterations of interaction with the user, the search system needs to learn user's search intent from the feedback on keywords and documents given by the user. Following the previous work [**?**], we improved the predicting accuracy and the time complexity of the model by employing spike-and-slab priors which could automatically select relevant features or coefficients for the model.

We incorporated the EP algorithm to approximate the posterior distribution, Thompson sampling to address the exploration-exploitation dilemma, and Topic model to discover the structure of the documents which could provide group information to the model.

In the simulation, we compared the model coefficients learned in the models of adopting Gaussian prior, spike-and-slab prior with or without group information and analyze the performance of them. Overall, the simulation demonstrated when increasing the size of documents in the corpus, leveraging sparse prior guaranteed the predicting accuracy. The spike-and-slab with or without group information performed similarly and outperformed Gaussian prior which did not encourage sparsity.

Although spike-and-slab prior with or without group information performed similarly, one advantage of group spike-and-slab prior was that the model complexity was lower than that of spike-and-slab prior without group information due to fewer coefficients needed to be estimated, so potentially group spike-and-slab prior could be generalized to a larger corpus. Besides, the sparse model matched our assumption that most of the coefficients did not have the contribution to the model, so the sparse models outperformed the model using Gaussian prior which did not encourage sparsity.

The limitation of our current model was that we had to learn the group structure of the documents and user's intent model separately, which meant the group structure stayed static during the model learning process. It would be ideal that the model could be learned end-to-end and on-the-fly, so the clustering information could be dynamically adapted according to different user intents.

It would be interesting to implement sparse probabilistic user model to the real exploratory search system. e.g. Scinet. This allows us to conduct the user study in a more realistic setting and provides us more insights on users' real behaviors, helping us collect more results to validate and justify our model.

Bibliography

[1] Gary Marchionini, Exploratory search: from finding to understanding, *Commun. ACM*, (49):4, 41-46, 2006.

[2] Tuukka Ruotsalo, Giulio Jacucci, Petri Myllymäki, and Samuel Kaski, Interactive intent modeling: information discovery beyond search, *Commun. ACM*, (58):1, 86-92, 2014.

[3] Pernilla Qvarfordt, Gene Golovchinsky, Tony Dunnigan, and Elena Agapie, Looking ahead: query preview in exploratory search, In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval*, 2013.

[4] Alan Medlar, Kalle Ilves, Ping Wang, Wray Buntine, and Dorota Głowacka, PULP: A System for Exploratory Search of Scientific Literature, In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2016.

[5] Pedram Daee, Joel Pyykkö, Dorota Głowacka, and Samuel Kaski, Interactive Intent Modeling from Multiple Feedback Domains, In *Proceedings of the 21st International Conference on Intelligent User Interfaces*, 71-75, 2016.

[6] Peter Auer, Using confidence bounds for exploitation-exploration tradeoffs, *Journal of Machine Learning Research*, 397-422, 2002.

[7] Tuukka Ruotsalo, Jaakko Peltonen, Manuel Eugster, Dorota Głowacka, Ksenia Konyushkova, Kumaripaba Athukorala, Ilkka Kosunen, Aki Reijonen, Petri Myllymäki, Giulio Jacucci, and Samuel Kaski, Directing exploratory search with interactive intent modeling, In *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*, 1759-1764, 2013.

[8] Dorota Głowacka, Tuukka Ruotsalo, Ksenia Konuyshkova, Kumaripaba Athukorala, Samuel Kaski, and Giulio Jacucci, Directing exploratory

search: reinforcement learning from user interactions with keywords, In *Proceedings of the 2013 International Conference on Intelligent User Interfaces*, 117-128, 2013.

[9] Antti Kangasrääsiö, Yi Chen, Dorota Głowacka, and Samuel Kaski, Interactive Modeling of Concept Drift and Errors in Relevance Feedback, In *Proceedings of the 2016 Conference on User Modeling Adaptation and Personalization, ACM*, 185-193, 2016.

[10] Andrew Y. Ng. Feature selection, L1 vs. L2 regularization, and rotational invariance. *In Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.

[11] T. J. Mitchell and J. J. Beauchamp, Bayesian variable selection in linear-regression, *Journal of the American Statistical Association*, 83(404):1023-1036, 1988.

[12] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition, *Springer*, 2009.

[13] Michael E. Tipping, Sparse Bayesian Learning and the Relevance Vector Machine, *Journal of Machine Learning Research*, 1(Jun): 211-244, 2001.

[14] Matthias W. Seeger, Bayesian Inference and Optimal Design for the Sparse Linear Model, *Journal of Machine Learning Research*, 9:759-813, 2008.

[15] Robert Tibshirani, Regression shrinkage and selection via the lasso, *Journal of the Royal Statistical Society: Series B*, 58:267-288, 1996.

[16] Yuan Ming and Lin Yi, Model selection and estimation in regression with grouped variables, *Journal of the Royal Statistical Society: Series B*, 68:49-67, 2006.

[17] Thomas P. Minka, Expectation propagation for approximate Bayesian inference, *In Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, 362-369, 2001.

[18] Michael R. Andersen, Ole Winther and Lars K. Hansen, Bayesian Inference for Structured Spike and Slab Priors, *Advances in Neural Information Processing Systems 27*, 1745-1753, 2014.

[19] Daniel Hernández-Lobato, José M. Hernández-Lobato and Pierre Dupont, Generalized Spike-and-Slab Priors for Bayesian Group Feature Selection Using Expectation Propagation, *Journal of Machine Learning Research*, 14:1891-1945, 2013.

[20] Daniel Hernández-Lobato, Prediction Based on Averages over Automatically Induced Learners: Ensemble Methods and Bayesian Techniques, *Phd Thesis, Universidad Autónoma de Madrid*, 2010.

[21] Yuwon Kim, Jinseog Kim and Yongdai Kim, *Statistica Sinica*, 16: 375-390, 2006.

[22] Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani, A Sparse-Group Lasso, *Journal of Computational and Graphical Statistics*, 22(2): 231-245, 2013.

[23] Junzhou Huang and Tong Zhang, The Benefit of Group Sparsity, *The Annals of Statistics*, 38(4): 1978-2004, 2010.

[24] Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert, Group lasso with overlap and graph lasso, In *Proceedings of the 26th Annual International Conference on Machine Learning*, 433-440, 2009.

[25] Sudhir Raman, Thomas J. Fuchs, Peter J. Wild, Edgar Dahl, and Volker Roth, The Bayesian group-Lasso for analyzing contingency tables. *In Proceedings of the 26th Annual International Conference on Machine Learning*, 881-888, 2009.

[26] Bobak Shahriari, Kevin Swersky, Ziyu Wang,Ryan P. Adams, andNando de Freitas, Taking the human out of the loop: a review of Bayesian optimization, *In Proceedings of IEEE*, 148-175, 2016.

[27] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams, Practical Bayesian Optimization of Machine Learning Algorithms, *In Advances in Neural Information Processing Systems 25*, 2960-2968, 2012.

[28] Ruben Martinez-Cantin, Nando de Freitas, Eric Brochu, Jose Castellanos and Arnaud Doucet, A Bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot, *Autonomous Robots*, 27(2): 93–103, 2009.

[29] Neil Houlsby, José M. Hernández-Lobato, Ferenc Huszar, and Zoubin Ghahramani, Collaborative Gaussian Processes for Preference Learning, *Advances in Neural Information Processing Systems*, 2096-2104, 2012.

[30] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams, Practical Bayesian Optimization of Machine Learning Algorithms, *Advances in Neural Information Processing Systems 25*, 2951-2959, 2012.

[31] Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown, Auto-WEKA: combined selection and hyperparameter optimization of classification algorithms. In *Proceedings of the 19th ACM SIGKDD international Conference on Knowledge Discovery and Data Mining*, 847-855, 2013.

[32] Favour M. Nyikosa, Michael A. Osborne and Stephen J. Roberts, Adaptive Bayesian Optimisation for Online Portfolio Selection, In *Workshop on Bayesian Optimization at NIPS 2015*, 2015.

[33] Antti Kangasrääsiö, Yi Chen, Dorota Głowacka, and Samuel Kaski. Interactive Modeling of Concept Drift and Errors in Relevance Feedback. In *Proceedings of the Conference on User Modeling Adaptation and Personalization*, 185-193, 2016.

[34] Lihong Li, Wei Chu, John Langford, and Robert E. Schapire, A contextual-bandit approach to personalized news article recommendation, In *Proceedings of the 19th International Conference on World Wide Web*, 661-670, 2010.

[35] Thore Graepel, Joaquin Quiñonero Candela, Thomas Borchert, and Ralf Herbrich, Web-Scale Bayesian Click-Through Rate Prediction for Sponsored Search Advertising in Microsoft's Bing Search Engine, In *Proceedings of the 27th International Conference on Machine Learning*, 13-20, 2010.

[36] Andreas Krause and Cheng S. Ong, Contextual Gaussian Process Bandit Optimization, *Advances in Neural Information Processing Systems 24*, 2447-2455, 2011.

[37] William R. Thompson, On the likelihood that one unknown probability exceeds another in view of the evidence of two samples, *Biometrika* 285-294, 1933.

[38] Shipra Agrawal and Navin Gayal, Thompson sampling for contextual bandits with linear payoffs, In *Proceedings of the 30th International Conference on Machine Learning*, 127-135, 2013.

[39] David M. Blei, Probabilistic topic models, *Commun. ACM*, 55(4), 77-84. 2012.

[40] David M. Blei, Andrew Y. Ng, and Michael I. Jordan, Latent dirichlet allocation, *Journal of Machine Learning Research*, 3(March): 993-1022, 2003

[41] Sharon Block, Doing More with Digitization, *The Journal of Early American Life*, 6(2), 2006.

[42] Thomas L. Griffiths and Mark Steyvers, Finding scientific topics, In *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl. 1): 5228-5235, 2004.

[43] John Paisley, Chong Wang, David M. Blei, and Michael I. Jordan, Nested Hierarchical Dirichlet Processes, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 37(2): 256-270, 2015.

[44] Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien, Using linear algebra for intelligent information retrieval, *SIAM Rev.* 37(4): 573-595, 1995 .