A Study of Using Chinese Restaurant Process Mixture Models in Information Retrieval

by

Wu Lin

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Statistics

Waterloo, Ontario, Canada, 2015

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Retrieval systems help users to isolate relevant information from massive data collections. Usually, a user obtains useful information by submitting a query to such a system. One critical issue is that a query could have many subtopics. A Web query "apple products" is a case. The query may indicate that a user wants to find Web pages related to iPhones or products made from the fruit "apple". Determining which is relevant is difficult without feedback from the user.

Query-specific clustering is one approach used to discover relevant aspects of a query by grouping relevant documents into clusters. In this approach, each cluster represents a relevant aspect of the query. We study Chinese restaurant process mixture models as clustering algorithms in this approach. To the best of our knowledge, our work is the first that studies such models in this context. Classical clustering models such as K-means and K-mixture Gaussian models have to first guess the number of clusters, K, and then estimate clusters from data. Chinese restaurant process mixture models can simultaneously learn the number of clusters and the actual clusters from data.

This thesis first reviews K-means, K-mixture Gaussian models and Bayesian K-mixture models. Then we review Chinese restaurant process mixture models. The Chinese restaurant process mixture models are extensions of the Bayesian models where K is not required to be finite. Among these mixture models, we pay attention to distance-dependent Chinese restaurant process mixture models since external pairwise measures can be used in modeling. Then, we propose two similarity-like measures used for the Chinese restaurant process mixture models in information retrieval. Finally, a Gibbs sampling scheme for both types of models is reviewed. Then the models' performance in the pseudo-relevance feedback via query expansion tasks is tested through experiments. In this task, top-retrieved documents are considered as relevant documents, and here we use a collection of documents from the Robust track of TREC 2004. We investigate the effectiveness of these Chinese restaurant process mixture models in three query sets, each of which contains 50 queries and relevance judgments. To confirm the robustness of these models, sensitivity analysis of the hyper-parameters is conducted. Results show that the Chinese restaurant process mixture models perform better than baseline models used in the feedback task, and are not sensitive when their hyper-parameters are reasonably selected. The proposed measures used in the distance-dependent Chinese restaurant process mixture models perform comparably. On the other hand, the proposed measures barely help these models to outperform the standard Chinese restaurant process mixture models.

Acknowledgments

The idea of this study came from my research interests. I discussed the idea with Professor Mu Zhu, my supervisor, during a regular meeting. He gave me supportive suggestion about this idea. What is more, he helped me clarify a lot of confused concepts during this study. I would like to thank Professor Mu to guide me along a right path toward the kingdom of mathematics and inspire me to think about interesting research topics. I also want to thank Professor Paul Marriott and Professor Martin Lysy to give me the valuable suggestions.

I would like to thank my friend, Eric Lin. He helped me solve math puzzles and we discussed many interesting mathematical topics. Without his help, I could not finish this thesis smoothly.

I owe a lot of debt to my parents, who always stand by me and encourage me to explore the world of mathematics.

Finally, I want to thank all my classmates. They make my life colorful.

To my parents

Table of Contents

List of Tables viii

List of Figures ix

1 Introduction 1

1.1 Introduction . . . 1

1.2 Background of Information Retrieval . . . 2

1.2.1 Evaluation Metrics . . . 3

1.2.2 Ranking Algorithm . . . 5

1.2.3 The Pseudo-relevance Feedback Task . . . 8

1.2.4 Feedback Model Estimation Methods . . . 9

2 Methodology 11

2.1 Problem Formulation . . . 11

2.2 Finite Models . . . 13

2.2.1 K-means . . . 13

2.2.2 K-Mixture Models . . . 14

2.2.3 Bayesian K-Mixture Models . . . 15

2.3 Chinese Restaurant Process Prior Distributions . . . 16

2.3.1 Standard Chinese Restaurant Processes . . . 17

2.3.2 Distance-dependent Chinese Restaurant Processes . . . 19

2.4 Model Estimation . . . 22
2.4.1 Exchangeability . . . 23
2.4.2 Collapsed Gibbs Samplers for CRPMs . . . 27
2.5 Hierarchical K-Mixture Model . . . 35
2.5.1 Latent Dirichlet Allocation . . . 35
2.5.2 Collapsed Gibbs Sampler for LDA . . . 36
2.6 Pairwise Similarity-like Measures . . . 38
2.6.1 Global Latent Measure . . . 39
2.6.2 Oracle Measure . . . 39

3 Experiments **43**
3.1 Experimental Setup . . . 43
3.1.1 Dataset and Text Pre-processing . . . 45
3.1.2 Estimation in Query Expansion . . . 47
3.2 Results . . . 51
3.3 Sensitivity Analysis . . . 53

4 Conclusion and Future Work **56**
4.1 Conclusion . . . 56
4.2 Future Work . . . 56

Acronyms **59**

References **60**

List of Tables

2.1 Equivalent notions in different contexts 17

3.1 Fixed parameters used in experiments 44
3.2 Fields in a TREC query 46
3.3 Statistics of query sets 46
3.4 Results for query set 1, where $\alpha = 1$ and $v = 0.1$ 52
3.5 Results for query set 2, where $\alpha = 1$ and $v = 0.1$ 52
3.6 Results for query set 3, where $\alpha = 1$ and $v = 0.1$ 52
3.7 Results for query set 1, where $\alpha = 1$ 53
3.8 Results for query set 2, where $\alpha = 1$ 53
3.9 Results for query set 3, where $\alpha = 1$ 53
3.10 Results for query set 1, where $\alpha = 1$ 54
3.11 Results for query set 2, where $\alpha = 1$ 54
3.12 Results for query set 3, where $\alpha = 1$ 54

List of Figures

2.1 Framework for the feedback task 13
2.2 Customer a comes in the order (a, b, c) 24
2.3 Customer b comes in the order (a, b, c) 24
2.4 Customer c comes in the order (a, b, c) 24
2.5 Customer b comes in the order (b, c, a) 24
2.6 Customer c comes in the order (b, c, a) 24
2.7 Customer a comes in the order (b, c, a) 24
2.8 Edge candidates in order (a, b, c) 25
2.9 Edge candidates in order (a, c, b) 25
2.10 $\Pr(e_1 = 1, e_2 = 1, e_3 = 2)$ 26
2.11 $\Pr(e_1 = 1, e_2 = 1, e_3 = 1)$ 26
2.12 $\Pr(e_1 = 1, e_2 = 2, e_3 = 1)$ 26
2.13 $\Pr(e_1 = 1, e_2 = 1, e_3 = 1)$ 27
2.14 $\Pr(e_1 = 1, e_2 = 1, e_3 = 2)$ 27
2.15 $\Pr(e_1 = 1, e_2 = 1, e_3 = 3)$ 27
2.16 $(z_1^{(t+1)}, z_2^{(t+1)}, z_3^{(t)}, z_4^{(t)}, z_5^{(t)}) = (1, 2, 3, 2, 2)$ 29
2.17 $(z_1^{(t+1)}, z_2^{(t+1)}, -, z_4^{(t)}, z_5^{(t)}) = (1, 2, -, 2, 2)$ 30
2.18 $(z_1^{(t+1)}, z_2^{(t+1)}, z_3^{(t+1)}, z_4^{(t)}, z_5^{(t)}) = (1, 2, 1, 2, 2)$ 30
2.19 $(z_1^{(t+1)}, z_2^{(t+1)}, z_3^{(t+1)}, z_4^{(t)}, z_5^{(t)}) = (1, 2, 2, 2, 2)$ 31
2.20 $(z_1^{(t+1)}, z_2^{(t+1)}, z_3^{(t+1)}, z_4^{(t)}, z_5^{(t)}) = (1, 2, 4, 2, 2)$ 31

2.21 $(e_1^{(t+1)}, e_2^{(t+1)}, e_3^{(t)}, e_4^{(t)}, e_5^{(t)}) = (1, 2, 2, 3, 3)$ 33
2.22 $(e_1^{(t+1)}, e_2^{(t+1)}, \_, e_4^{(t)}, e_5^{(t)}) = (1, 2, \_, 3, 3)$ 33
2.23 $(e_1^{(t+1)}, e_2^{(t+1)}, e_3^{(t+1)}, e_4^{(t)}, e_5^{(t)}) = (1, 2, 1, 3, 3)$ 34
2.24 $(e_1^{(t+1)}, e_2^{(t+1)}, e_3^{(t+1)}, e_4^{(t)}, e_5^{(t)}) = (1, 2, 2, 3, 3)$ 34
2.25 $(e_1^{(t+1)}, e_2^{(t+1)}, e_3^{(t+1)}, e_4^{(t)}, e_5^{(t)}) = (1, 2, 3, 3, 3)$ 35

3.1 LDA of 200 latent base models 47
3.2 For TREC query 310 48
3.3 For TREC query 310 49
3.4 For TREC query 310 50

Chapter 1

Introduction

1.1 Introduction

Retrieval systems such as Web search engines and library retrieval systems are commonly used in our daily life. A common scenario is that a user submits a query to a retrieval system in order to find relevant records. Usually, unless sufficient details are given, a query may be ambiguous. Without users' feedback, it is challenging to design a retrieval system that discovers relevant aspects of the query and sorts documents properly. For example, a user could submit a query "apple product" to a Web search engine in order to find Web pages related to products from the Apple company. There are many aspects of the query, some more useful than other. For instance, finding Web pages about iPhones is relevant. Seeking Web pages about the actual fruit "apple" is irrelevant. Without interaction with the user, it is almost impossible for a Web search engine to return documents that are only about products from the Apple company.

One approach of discovering relevant aspects of a query is to group relevant documents into clusters [8]. In this approach, each cluster represents one relevant aspect of the query. In the literature, this approach is referred to as query-specific clustering. Many clustering algorithms are proposed to group documents into clusters. Most of these algorithms, such as K-means and K-mixture Gaussian models, have to first fix the number of clusters, K, and then learn these clusters. In other words, the number of relevant aspects is fixed across all queries in this context. However, in reality, the number of relevant aspects of a query can vary from query to query. Classical hierarchical clustering algorithm has been proposed to address this issue. One issue of this algorithm is how to cut the hierarchical tree into clusters [16]. To address these issues mentioned above, we propose to use Chinese restaurant process

mixture models as clustering algorithms. These models can simultaneously learn clusters and the number of clusters from data. What is more, according to our experiments, the hyper-parameters of these models are relatively easy to set. Lastly, a Bayesian extension of hierarchical clustering is a special case of such models [10]. Another issue of the approach is how to identify relevant documents without user's feedback. An idea to deal with the issue is that top-retrieved documents are considered as relevant documents. It is the assumption used in the pseudo-relevance feedback task.

In this thesis, we study the Chinese restaurant process mixture models used in query-specific clustering for the pseudo-relevance feedback task. The main contribution of this thesis is that our work, to the best of our knowledge, is the first investigation of Chinese restaurant process mixture models in this context. The thesis is organized as below.

Fundamental concepts in information retrieval are first introduced in Chapter 1. With these concepts in hand, the main research question is then formulated in Chapter 2. Related models and Gibbs samplers are also discussed in Chapter 2. The experiment setup is given and results of the experiments are shown in Chapter 3. Finally, Chapter 4 concludes this work and proposes future research.

1.2 Background of Information Retrieval

First, let us recall the apple example. A user may submit a query, "apple product", to a Web engine such as Google if the user wants to find Web pages discussing products from the Apple company. A ranked document list about the query is returned to the user. Usually, the user reads only a few documents at the top of the list to find out about Apple products. This example illustrates key concepts in information retrieval.

The Web search engine Google is an information retrieval system. The following definition of information retrieval used in this thesis is taken from [16].

Definition 1. *An information need is a desire to locate and obtain information to satisfy a specific need.*

Definition 2. *Information Retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers).*

According to Definition 2, documents in the sorted list reflect the material that the user seeks, and finding Web pages related to Apple products satisfies user's information need in the apple example.

In our example, "apple product" is a query submitted to Google and a Web page is a document in a collection. The terms "query", "document", and "collection" are respectively defined as:

Definition 3. *Typically, a query is a sequence of words, which is submitted to an information retrieval system. A query is used to represent an information need.*

Definition 4. *A document is a sequence of words, which is archived and indexed in an information retrieval system.*

Definition 5. *A collection is a set of documents that indexed in an information retrieval system. The collection is static. In other words, the number of documents and the content of documents in the collection do not change.*

Definition 6. *Relevance of a document with respect to a query is defined as how well the document satisfies the information need that the query represents.*

Given a query, Google first computes a relevance score for each document in a collection based on its ranking algorithm, then sorts documents in the collection by relevance score, and returns a rated document list that contains only the top-1000 scored documents. A ranking algorithm is defined as follows:

Definition 7. *A ranking algorithm is a procedure to estimate the relevance of documents with respect to a query.*

1.2.1 Evaluation Metrics

Sometimes a user may not be satisfied with the retrieved results. In this example, the user is not satisfied if most of top-retrieved documents in the list are about the fruit "apple". Even if documents related to the Apple company do appear at the bottom of the list, the user feels unhappy about having to browse most of documents in the list. In other words, the document order of a retrieved list matters. Many evaluation metrics have been proposed to address the order. Among them, we pay attention to three metrics—Precision at position N (P@N), Average Precision (AP), and Normalized Discounted Cumulative Gain at position N (NDCG@N). These metrics are used to measure the effectiveness of a ranking algorithm given one query. Note that N usually is not greater than 1000 since a sorted document list that only contains the top-1000 rated documents, is returned. In order to compute these metrics, we assume that the relevance judgments made by humans per query are known. All definitions of evaluation metrics are taken from [4] and [5].

Precision at position N and average precision are defined based on binary relevance judgments.

Definition 8. *Rel is defined as a set of documents assessed as relevant by human judgment.*

Definition 9. *Ret is defined as an ordered list of documents returned by a retrieval system. $Ret[1,\ldots,N]$ is a truncated list of Ret, which contains only the top-N retrieved documents.*

Definition 10.

$$P@N = \frac{|Ret[1,\ldots,N] \cap Rel|}{N} \tag{1.1}$$

Definition 11.

$$AP = \frac{\sum_{i=1}^{|Ret|} I_{rel}(Ret[i]) \times P@i}{|Ret|} \tag{1.2}$$

where $I_{rel}(Ret[i]) \in \{0,1\}$ is a relevant indicator function. Its value is 0 unless the i-th document ($Ret[i]$) in a ranked list is relevant.

According to Definition 11, average precision is a weighed combination of precision at position Ns, where weights are calculated according to the order of documents and binary relevance judgments.

On the other hand, normalized discounted cumulative gain at position N can make use of graded relevance judgments. In graded relevance judgments, an irrelevant document is marked as 0 and a relevant document is marked as a positive value which represents the degree of relevance. Of course, a binary relevance judgment is a 0-1 graded relevance judgment.

First, discounted cumulative gain at position N (DCG@N) is defined as

Definition 12.

$$DCG@N = \sum_{j=1}^{N} \frac{rel(Ret[j])}{log_2(j+1)} \tag{1.3}$$

where $rel(Ret[j])$ is a degree of relevance for the j-th document ($Ret[j]$) in a ranked list.

If documents are ordered in descending order based on their graded relevance judgments and the top N documents are evaluated, it can be easily verified that DCG@N has an attainable upper bound.

Definition 13. *Ideal discounted cumulative gain at position N (iDCG@N) is an attainable upper bound of DCG@N.*

Normalized discounted cumulative gain at position N then is defined as

Definition 14.

$$nDCG@N = \frac{DCG@N}{iDCG@N} \tag{1.4}$$

Note that all evaluation metrics mentioned above are used to assess the effectiveness of a ranking algorithm with respect to one query. In order to summarize performance across queries, the arithmetic means of precision at position N, average precision and normalized discounted cumulative gain at position N are used. The means of precisions at position N, average precisions and normalized discounted cumulative gains at position N are referred to as MP@N, MAP and MNDCG@N respectively.

Usually effectiveness of an algorithm in information retrieval is evaluated based on the arithmetic mean of a metric across test queries. However, the mean alone is not enough. In statistics, the mean is not robust to outlier. Hypothesis tests are often used to justify a superior algorithm. In this context, a "population" used in these tests is a set of queries submitted to an information retrieval system. Similarly, a "sample" used in these tests is a set of test queries used to represent the population. Usually, test queries are obtained by analyzing searching history. In this thesis, we use the following tests: t-test and randomized test.

1.2.2 Ranking Algorithm

Definition 15. *A vocabulary is a set of distinct words, each of which appears in any document in a collection.*

There are many ranking algorithms studied in the literature. To address our purpose, we use the Kullback-Leibler (KL) ranking algorithm. The Kullback-Leibler ranking algorithm has been proposed in [22]. This algorithm assumes that a multinomial distribution over words in a vocabulary can model a sequence of words. Under this assumption, words in a sequence are exchangeable, which means the order of words is neglected. We then define the terms "query model" and "document model" as:

Definition 16. *A multinomial distribution over words in a vocabulary used to model a query is called a query model,* θ_{query} .

Definition 17. *Similarly, a multinomial distribution over words in a vocabulary is used to model a document is called a document model,* θ_{doc} .

Definition 18. *tf_w is called a term frequency of a word w in a sequence of words. The raw frequency of the word in the sequence is used.*

Definition 19. *A background noise model θ_{back} is a known multinomial distribution over words in a vocabulary, usually obtained from domain knowledge.*

In the literature, a maximum likelihood estimator is commonly used to estimate a query model and a Dirichlet smoothed maximum likelihood estimator is used to estimate a document model [22]. Definitions of these estimators are given as below.

Definition 20. *The maximum likelihood estimators of a query model for word w is:*

$$\Pr_{query}(w) = \frac{tf_w}{L} \tag{1.5}$$

where tf_w is a term frequency of word w in a query, and $L = \sum_w tf_w$, is the length of the query.

Note that the probability of any word that does not appear in the query is estimated as 0. In the apple example, the probability of any word that is neither "apple" nor "product" is 0 in the query model.

Definition 21. *The Dirichlet smoothed maximum likelihood estimator of a document model for word w is defined as:*

$$\Pr_{doc}(w) = \frac{tf_w + u \times \Pr_{back}(w)}{L+u} \tag{1.6}$$

where tf_w is a term frequency of word w in a document, $L = \sum_w tf_w$, is the length of the document, $\Pr_{back}(w)$ is the probability of word w in a background noise model, θ_{back}, and u is a Dirichlet smoothing parameter.

Note that the probability of a word that does not appear in the document is estimated as non-zero. In fact, the Dirichlet smoothed maximum likelihood estimator is a maximum a posterior estimator of the document model with a Dirichlet prior belief.

The Kullback-Leibler ranking algorithm uses the Kullback-Leibler divergence, which measures the difference between a query and a document. The divergence estimates relevance of the document with respect to the query. The Kullback-Leibler divergence is defined as:

Definition 22.

$$KL(\theta_{query}||\theta_{doc}) = \sum_{word\ w} \Pr_{query}(w) \times (log(\Pr_{query}(w)) - log(\Pr_{doc}(w))) \tag{1.7}$$

Note that the divergence is asymmetric. Efficiency is the main reason why the asymmetric divergence is used. Usually, a query is shorter than a document. What is more, the content of a document do not change but distinct queries are submitted to a retrieval system. With the help of the data structure, inverted index, the divergence can be efficiently computed. On the other hand, computing a symmetric divergence is slow. The same argument is also applied to the reason why the maximum likelihood estimator is used to estimate a query model.

The Kullback-Leibler ranking algorithm computes a relevance score of a document with respect to a query as:

Definition 23.

$$score(\theta_{doc}|\theta_{query}) = \sum_{word\ w\ in\ query} \Pr_{query}(w) \times log(\Pr_{doc}(w)) \tag{1.8}$$

According to Definition 22 and Definition 23, the relationship between the Kullback-Leibler divergence and the Kullback-Leibler ranking algorithm is

$$\begin{aligned} -\mathrm{KL}(\theta_{\text{query}}||\theta_{\text{doc}}) &= \sum_{\text{word } w \text{ in query}} \Pr_{\text{query}}(w) \times (\log(\Pr_{\text{doc}}(w)) - \log(\Pr_{\text{query}}(w))) \\ &= \mathrm{score}(\theta_{\text{doc}}|\theta_{\text{query}}) + \mathrm{Const} \end{aligned}$$

It is obvious that exact matching of keywords is not enough. In the apple example, a Web page about iPhones should be considered as relevant even if not all keywords from the query "apple product" appear in the page. In this case, smoothing is critical. If a document model is estimated by a maximum likelihood estimator, Equation 1.8 becomes undefined since $\Pr_{\text{doc}}(w) = 0$ and $\Pr_{\text{query}}(w) > 0$, where w is a keyword from a query. In Statistics, a smoothing technique is used against over-fitting related to the maximum likelihood estimator. This is why the Dirichlet smoothed maximum likelihood estimator is used to estimate a document model.

1.2.3 The Pseudo-relevance Feedback Task

However, the smoothing technique alone cannot help an information retrieval system to retrieve relevant documents that contain no keywords used in a query. Let us recall the apple example. A Web about iPhones, which does not contain any keyword from the query "apple product", should be considered as relevant. The pseudo-relevance feedback technique has been proposed to deal with this issue. First, we define the "relevance feedback" technique as:

Definition 24. *The relevance feedback technique is an interactive two-stage procedure to improve a retrieval result. At the first stage, a user is asked to mark relevant documents from a sorted list with respect to an original query. At the second stage, these documents are used to form a new expanded query.*

The "pseudo-relevance feedback" technique and "pseudo-relevance feedback task" are then defined as:

Definition 25. *The pseudo-relevance feedback technique is an automatic one-stage procedure to improve a retrieval result. In the procedure, an information retrieval system uses the top-retrieved documents to expand an original query. These documents can be viewed as relevant documents marked by pseudo-user feedback.*

Definition 26. *The pseudo-relevance feedback task is a task to evaluate the performance of information retrieval systems using the pseudo-relevance feedback technique. Given an information need and an initial query, an information retrieval system evaluated in this task uses the technique to expand the query and returns sorted documents to meet the information need.*

In the apple example, the information need is to find Web pages about products from the Apple company, and the initial query is "apple product". Using the pseudo-relevance feedback technique, an information retrieval system is likely to find pages such as pages about iPhones even though these pages contain no "apple product" keywords.

Definition 27. *A feedback document is a document marked by user or pseudo-user feedback.*

Definition 28. *A feedback document model, θ_d, is a multinomial distribution over words in a vocabulary, which is used to model feedback document d.*

Definition 29. *A feedback model, $\theta_{feedback}$, is a multinomial distribution over words in a vocabulary, which is used to model all feedback documents. The model is used to expand a query.*

Once the feedback model is known, a query is expanded by the following way:

Definition 30.

$$\theta_{expand} = \pi^{(q)} \times \theta_{query} + (1 - \pi^{(q)}) \times \theta_{feedback} \tag{1.9}$$

where θ_{expand} is a query model of an expanded query. $\theta_{feedback}$ is a feedback model, $\pi^{(q)}$ is an interpolation weight for the expanded query model, and θ_{query} is a query model of the original query.

1.2.4 Feedback Model Estimation Methods

In the feedback task, the key issue is to deal with noise in the feedback model estimation. Noise is mainly introduced from irrelevant documents since the top-ranked documents are indeed not always relevant. The noise must be taken into account since the estimated feedback model is used to expand a query. Ideally, only informative words in the feedback model are used in query expansion. Various solutions were studied in the estimation. From among them, we have selected the simple mixture method and the relevance method as most suitable for our purposes.

The Simple Mixture method (SM) was proposed in [23]. The assumption behind it is that the noise are generated from a background noise model and, therefore the noise can be reduced if the background model reasonably models the noise.

The simple mixture method assumes that words in feedback documents are generated as below:

1. Given two models θ_0 and θ_1
2. Given a mixing coefficient, $\overrightarrow{\pi} = (1.0 - \pi^{(SM)}, \pi^{(SM)})$
3. For the j-th word in a feedback document i
   (a) independently generate a latent model indicator, $z_{ji} \sim \text{Bern}(z|\overrightarrow{\pi})$

(b) independently generate a word, $w_{ji} \sim d(w|\theta_{z_{ji}})$

Where "Bern" denotes a Bernoulli distribution (Bern), $\overrightarrow{\pi}$ is usually given, $d(\cdot)$ is a family of modeling distributions (say, the multinomial family), θ_0 is a feedback model, $\theta_{\text{feedback}}^{(\text{SM})}$, to be estimated, and θ_1 is a background noise model, θ_{back}.

Definition 31. *A relevance measure is a non-negative weight that ranges from 0 to 1 to measure the relevance of a document with respect to a query. If a document is relevant, its relevance measure should be close to 1. Similarly, if a document is irrelevant, its relevance measure should be close to 0. A relevance measure usually is a relevance score with proper transformation.*

Unlike the simple mixture method, the Relevance Method (RM) [13] explicitly uses relevance measures in modeling. The key idea behind the relevance method is that relevance measures are used to reduce the noise since this noise comes from non-relevant documents, each of which should have a low relevance measure.

The relevance method is defined as a weighted combination of feedback document models:

$$\theta_{\text{feedback}}^{(\text{RM})} = \frac{\sum_{\text{feedback document d}} w_d^{(\text{RM})} \times \theta_d^{(\text{RM})}}{\sum_{\text{feedback document d}} w_d^{(\text{RM})}} \tag{1.10}$$

where $w_d^{(\text{RM})}$ is a relevance measure of feedback document d, and $\theta_d^{(\text{RM})}$ is a feedback document model.

Chapter 2

Methodology

This chapter first formulates the research question studied in this thesis. Then a classic clustering model (K-means), its probabilistic extension (K-mixture of Gaussian), and a finite Bayesian extension are reviewed. All these models have to specify the number of clusters K, which implies K is known.

In order to deal with the case where the number of clusters is unknown, Chinese Restaurant Process Mixture model (CRPM) has been proposed. In the literature, a Standard Chinese Restaurant Process Mixture model (s-CRPM) is also known as a Dirichlet process mixture model. We also study a Distance Dependent Chinese Restaurant Process Mixture model (dd-CRPM), which uses external pairwise measures to form a prior belief. Finally, we propose two text-specific measures in the information retrieval domain.

2.1 Problem Formulation

As mentioned in Chapter 1, grouping relevant documents with respect to a query is an approach to discover relevant aspects of the query. Without user's feedback, an information retrieval system adapts the approach by using top-retrieved documents as pseudo-relevant documents. It is critical to select an appropriate clustering algorithm for the adapted approach since not all top-retrieved documents are relevant. We study this adapted approach in the pseudo-relevance feedback task. Note that these top-retrieved documents are also called feedback documents in the task. The simple mixture method implicitly assumes that feedback documents together represent one and only one relevant aspect of a query, while the relevance method assumes that each feedback document represents one distinct

relevant aspect of a query. It prompts us to wonder whether the number of relevant aspects of a query plays a role in the feedback task, since the number of aspects should vary from query to query. A Chinese restaurant process mixture model can automatically learn the number of aspects. We proposed to study standard Chinese restaurant process mixture models [18] and distance-dependent Chinese restaurant process mixture models [1] as clustering algorithms to find relevant aspects of a query. We then evaluate the effectiveness of Chinese restaurant process mixture models in the pseudo-relevance feedback task. In other words, our main research question is:

Can Chinese restaurant process mixture models improve retrieval results in the pseudo-relevance feedback task?

Definition 32. *An aspect model of a query, θ_{aspect}, is a multinomial distribution over words in a vocabulary, which is used to model one aspect of the query.*

A detailed framework used a Chinese restaurant process mixture model in the feedback task is described as (Figure 2.1):

1. An initially ranked document list with respect to a query is returned using the KL ranking algorithm.
2. Top retrieved documents from the list are considered as relevant documents from a pseudo-user's feedback. In other words, these documents are considered as pseudo-relevant documents.
3. These documents are grouped into clusters using a Chinese restaurant process mixture model. Distance-dependent Chinese restaurant process mixture models can utilize external information to construct a prior belief. Gibbs sampling scheme is used in the model estimation.
4. Given documents in a cluster, the simple mixture method can be applied to these documents to learn an aspect model of a query.
5. By viewing an aspect model as a document model, the relevance method is adapted to obtain a feedback model. In this thesis, we use relevance scores to obtain relevance measures used in the relevance method. Recall that given a query model and a document model, a document relevance score with respect to the corresponding query is computed based on the KL ranking algorithm.

6. An expanded query model is then created by linearly combining the original query model and the feedback model.

7. Finally, a new ranked documents list with respect to the expanded query model is returned.

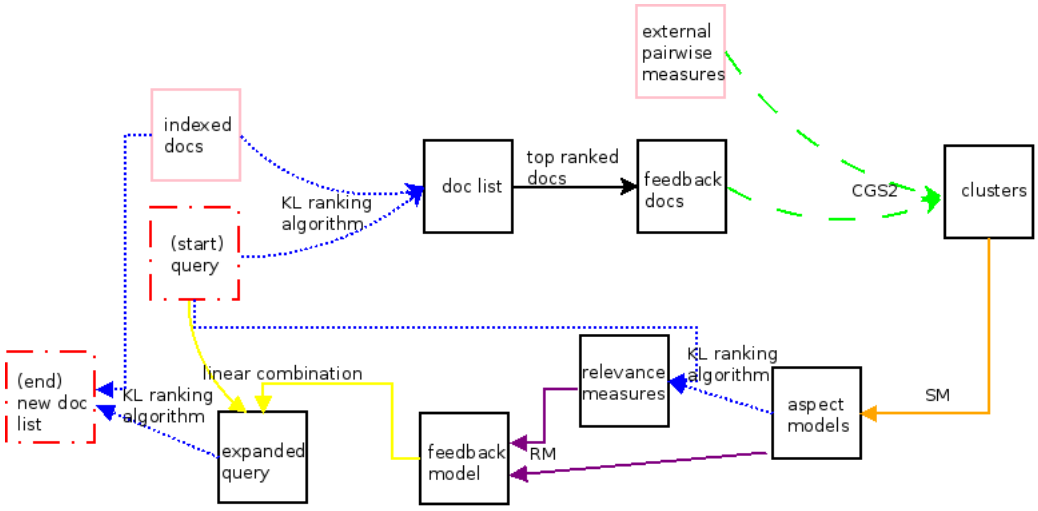


Figure 2.1: Framework for the feedback task

2.2 Finite Models

2.2.1 K-means

First, let us review K-means where K is fix and known. Given a set of n documents, the objective of K-means is to group them into K clusters by minimizing the following function.

$$\sum_{i=1}^{n}\sum_{j=1}^{K} z_{ij}||\overrightarrow{w_i}-\overrightarrow{\mu_j}||^2, z_{ij}=\begin{cases}1 & \text{if doc } i \text{ is assigned to cluster } j\\ 0 & \text{otherwise}\end{cases} \quad (2.1)$$

where z_{ij} is a cluster indicator variable to be estimated, $\overrightarrow{\mu_j}$ is the mean of cluster j to be estimated, and $\overrightarrow{w_i}$ is a known vector representation of the word sequence of document i.

An iterative procedure is used to estimate parameters by randomly initializing $\overrightarrow{\mu_j}$ for all j and then coordinately estimating z_{ij} and $\overrightarrow{\mu_j}$ for all j and i until convergence. Note that each document in this model is assigned to one and only one cluster in a deterministic way, which is called the hard assignment.

2.2.2 K-Mixture Models

A K-mixture of Gaussian model with known co-variance matrix τI, where τ is a positive scalar and I is an identity matrix, has been shown to be a probabilistic extension of K-means [11]. In this model, the hard assignment is replaced with the soft assignment by allowing each document to be allocated to K clusters in a probabilistic way. Let us review a K-mixture model without specifying the modeling distribution. A K-mixture model assumes that documents are generated as:

1. For each document i (there are n documents)
   (a) independently generate a latent model indicator, $z_i \sim \text{Mult}(z|\overrightarrow{\pi})$
   (b) independently generate a word sequence, $\overrightarrow{w_i} \sim d(\overrightarrow{w}|\theta_{z_i})$

where "Mult" denotes a Multinomial distribution (Mult), d is a family of modeling distributions (say, the Gaussian family in a K-mixture of Gaussian model), $\overrightarrow{\pi}$ is a K-dim mixing coefficient (which implies $\sum_{k=1}^{K} \pi_k = 1$ and $\pi_k > 0$ for all k), θ is parameters of a distribution in the family, which is referred to as a base model, and $\Theta = (\theta_1, \ldots, \theta_K)$ is a set of the K base models to be estimated.

According to the generative procedure, the data likelihood in a K-mixture model is:

$$\Pr(W) = \prod_i (\sum_{j=1}^{K} \pi_j \times d(\overrightarrow{w_i}|\theta_j)) = \prod_i g(\overrightarrow{w_i}) \tag{2.2}$$

where $g(\cdot) = \sum_{j=1}^{K} \pi_j \times d(\cdot|\theta_j)$ is a K-mixture of distributions.

The formal proof that a K-mixture of Gaussian with known co-variance matrix τI is the probabilistic extension of K-means can be found in [11]. Informally, readers can observe that when the expectation-maximization algorithm is used, the estimation procedure is

similar to the iterative procedure for K-means. What is more, the cluster indicator variable in K-means corresponds to the latent model indicator. Lastly, when $\tau \to 0$, a K-mixture of Gaussian model has been proven to be equivalent to a model of K-means [11].

2.2.3 Bayesian K-Mixture Models

A finite Bayesian extension of K-mixture models can be formulated by putting a prior distribution on mixing coefficient $\overrightarrow{\pi}$ and a prior distribution on base models $\Theta = (\theta_1, \ldots, \theta_K)$. One extension given below is to choose a Dirichlet distribution for the mixing coefficient and a prior distribution G_0 for the base models. The finite Bayesian mixture model generates data by:

1. Generate a K-dim mixing coefficient $\overrightarrow{\pi} \sim \text{Dir}(\overrightarrow{\pi}|\alpha \times \overrightarrow{1}/K)$, where α is a scalar hyper-parameter
2. For each cluster c (there are K clusters)
   (a) independently generate a base model, $\theta_c \sim G_0$, where G_0 is a prior distribution of base models
3. For each document i (there are n documents)
   (a) independently generate a latent model indicator, $z_i \sim \text{Mult}(z|\overrightarrow{\pi})$
   (b) independently generate a word sequence, $\overrightarrow{w_i} \sim d(\overrightarrow{w}|\theta_{z_i})$

where "Dir" denotes a Dirichlet distribution (Dir).

Note that a Dirichlet distribution is a conjugate prior of multinomial distribution. If the mixing coefficient, $\overrightarrow{\pi}$, is marginalized out, the generative procedure becomes:

1. For each cluster c (there are K clusters)
   (a) independently generate a base model, $\theta_c \sim G_0$

2. For each document i (there are n documents)

   (a) generate a latent model indicator, $z_i \sim \text{DirMult}(z|\alpha \times \overrightarrow{1}/K)$

   (b) independently generate a word sequence, $\overrightarrow{w_i} \sim d(\overrightarrow{w}|\theta_{z_i})$

where "DirMult" denotes a Dirichlet-Multinomial distribution (DirMult).

Note that document-specific model indicators, $\overrightarrow{z} = (z_1, \ldots, z_n)$, are dependent. The probability of generating $\overrightarrow{z}$ under the Dirichlet-Multinomial distribution, $\text{DirMult}(\alpha \times \overrightarrow{1}/K)$, is:

$$\Pr(\overrightarrow{z}|\overrightarrow{\alpha}) = \frac{\Gamma(\alpha)}{\Gamma(n+\alpha)} \prod_{k=1}^{K} \frac{\Gamma(n_k + \frac{\alpha}{K})}{\Gamma(\frac{\alpha}{K})} \tag{2.3}$$

where $\overrightarrow{\alpha} = \alpha \times \overrightarrow{1}/K$, n_k is the number of count of base model k, $\sum_{k=1}^{K} n_k = n$, $\Gamma(\cdot)$ denotes the gamma function, $\overrightarrow{z} = (z_1, \ldots, z_n)$, z_i is a model indicator of the i-th document and $z_i \in \{1, \ldots, K\}$.

2.3 Chinese Restaurant Process Prior Distributions

If we replace the Dirichlet-Multinomial distribution used in the Bayesian mixture model with a Chinese restaurant process, the adapted mixture model becomes a Chinese restaurant process mixture model.The generative procedure of the model is:

1. For each document i

   (a) generate a latent model indicator, $z_i \sim \text{CRP}(z|\alpha)$

   (b) If the latent model indicator z_i is

      i. new, independently generate a new base model $\theta_{z_i} \sim G_0$

      ii. old, do nothing (reuse the existing base model θ_{z_i})

   (c) independently generate a word sequence, $\overrightarrow{w_i} \sim d(\overrightarrow{w}|\theta_{z_i})$

where "CRP" denotes a Chinese Restaurant Process (CRP) and α is a scalar hyperparameter of the process.

	notions	
Context	document	base model
Chinese restaurant process	customer	table
Distance-dependent Chinese restaurant process	node	weakly connected subgraph
Document clustering	document	cluster

Table 2.1: Equivalent notions in different contexts

Since a Chinese restaurant process are essential, we first review two kinds of the process and discuss corresponding mixture models. Our review is inspired from the work [6]. A Chinese restaurant process is a combinatorial stochastic process. Table 2.1 summarizes equivalent notions mentioned in this chapter.

2.3.1 Standard Chinese Restaurant Processes

A standard Chinese restaurant process uses an analogy that customers sit at tables in a Chinese restaurant to describe the generative procedure. Typically, the process generates data by:

1. when $i = 1$, the first customer always sits at a new table with an unique label
2. when $i > 1$, the i-th customer sits at
   (a) a new table, with probability in proportion to α and assigned a distinct label to the table
   (b) any one of occupied tables $z_i = j$, with probability in proportion to the count of customers at table j

where α is a scalar hyper-parameter of the process.

The process is closely related to the Dirichlet-Multinomial distribution. Let us show the relationship. For the standard Chinese restaurant process, we assume that there are

$n-1$ customers (documents) sitting at K tables (base models) in the restaurant, which implies $n > K$. When the n-th customer (document) comes in the restaurant, according to the standard Chinese restaurant process, the probability of the n-th customer (document) sitting at an exiting table j (base model j) is:

$$\Pr(z_n = j|\overrightarrow{z_{-n}}, \alpha) = \frac{n_j}{n-1+\alpha} \tag{2.4}$$

where n_j is the number of the $n-1$ customers (documents) at table j, $\overrightarrow{z_{-n}} = (z_1, \ldots, z_{n-1})$ represents a vector of labels (model indicators) assigned to these n customers (documents) excluding the n-th label (model indicator) and $\sum_{k=1}^{K} n_k = n-1$.

For the Dirichlet-Multinomial distribution, according to Equation 2.3, the conditional probability of the model indicator of the n-th document assigned to base model j in a K-dim Dirichlet-Multinomial distribution with hyper-parameter $\frac{\alpha}{K}$ is

$$\Pr(z_n = j|\overrightarrow{z_{-n}}, \tfrac{\alpha}{K}) = \frac{\Pr(\overrightarrow{z_{-n}} \cup \{z_n=j\}|\frac{\alpha}{K})}{\Pr(\overrightarrow{z_{-n}}|\frac{\alpha}{K})} = \frac{n_j+\frac{\alpha}{K}}{n-1+\alpha} \tag{2.5}$$

When $K \to \infty$, Equation 2.5 becomes Equation 2.4. Therefore, a standard Chinese restaurant process is informally viewed as an infinite extension of a Dirichlet-Multinomial distribution.

A Chinese restaurant process mixture model using a standard Chinese restaurant process is called a standard Chinese restaurant process mixture model. In this model, a document and a document-specific model indicator respectively represent a customer and a table label in the context of the process. The description of the generative procedure of the mixture model is given below. In the model, documents are generated as:

1. For each document i,
   (a) generate a latent model indicator z_i
      i. if $i = 1$, $z_i = c_{new}$, and independently generate a new base model, $\theta_{new} \sim G_0$, with a distinct new label c_{new}
      ii. else
         A. $z_i = c_{new}$, with probability in proportion to α, and independently generate a new base model, $\theta_{new} \sim G_0$, with a distinct new label c_{new}

B. $z_i = c_j$, with probability in proportion to $\sum_{k=1}^{i-1} I(z_k = c_j)$, where $I(\cdot)$ is an indicator function and c_j is an existing label

(b) independently generate a word sequence, $\overrightarrow{w_i} \sim d(\overrightarrow{w}|\theta_{z_i})$

2.3.2 Distance-dependent Chinese Restaurant Processes

Let us recall the apple example. If two Web pages are about information techniques, they should be similar. On the other hand, one page about food and another page about information technique should be less similar. Given these additional pairwise information, it is relatively easy to identify relevant aspects of the query. Intuitively, external pairwise information can play an important role. Distance-dependent Chinese restaurant processes [1] have been proposed to utilize external pairwise information. In a distance-dependent Chinese restaurant process, a node and a weakly connected sub-graph respectively represent a customer and a table in a standard Chinese restaurant process. Given pairwise similarity-like measures, a distance-dependent Chinese restaurant process is:

1. when i=1, the first node is added in an empty graph and a directed edge from node i to itself is added, which implies that a new weakly connected sub-graph is created
2. when $i > 1$, the i-th node is added into the graph and a directed edge from the i-th node to the j-th node is added, where j is:
   (a) i, with probability in proportion to α
   (b) any exiting node k ($k < i$), with probability in proportion to h_{ik}, where h_{ik} is a non-negative similarity-like measure between node i and node k

where α is a scalar hyper-parameter of the process.

A distance-dependent Chinese restaurant process generalizes a standard Chinese restaurant process by using an un-directed connected sub-graph to represent a table in the stan-

dard Chinese restaurant process. We show that given special similarity-like pairwise measures, a distance-dependent Chinese restaurant process is equivalent to a standard Chinese restaurant process. Recall that in a standard Chinese restaurant process, the probability of the i-th customer (document) at table k (base model k) is:

$$\Pr(z_i = k|\overrightarrow{z_{-i}}, \alpha) \propto \begin{cases} n_k & \text{for } k \leq K \\ \alpha & \text{for } k = K+1 \end{cases} \tag{2.6}$$

where $\overrightarrow{z_{-i}} = (z_1, \ldots, z_{i-1})$, K is the number of occupied tables by i customers (documents) , and n_k is the number of $i-1$ customers (documents) at table k (base model k).

Definition 33. *Order-specific measure s_{ij} is a non-negative similarity-like pairwise measure between the i-th node and the j-th node. s_{ij} is defined as:*

$$s_{ij} = \begin{cases} h_{ij} & \textit{for } j < i \\ 0 & \textit{otherwise} \end{cases} \tag{2.7}$$

where h_{ij} is a non-negative similarity-like pairwise measure between node i and node j.

Note that the partial order "$<$" defined in s_{ij} depends on the order of nodes added in a graph. What is more, s_{ij} and h_{ij} are different. Consider the following example to illustrate the difference. Suppose the similarity-like pairwise measure between document a and document c is 0.5, which means $h_{ac} = h_{ca} = 0.5$. If document a and document c are the first node and the third node respectively added in a graph, we know that $s_{13} = 0$ and $s_{31} = 0.5$. On the other hand, if document a and document c are the second node and the first node respectively added in a graph, we know that $s_{12} = 0$ and $s_{21} = 0.5$.

According to the distance-dependent Chinese restaurant process, the probability of adding an edge from the i-th node (document) to the j-th node (document) is:

$$\Pr(e_i = j|\overrightarrow{e_{-i}}, \alpha) \propto \begin{cases} \alpha & \text{for } j = i \\ s_{ij} & \text{otherwise} \end{cases} \tag{2.8}$$

where $\overrightarrow{e_{-i}} = (e_1, \ldots, e_{i-1})$, s_{ij} is the order-specific measure between node i and node j, and $e_i = j$ denotes a directed edge from the i-th node to the j-th node.

Now, we show that given the following similarity-like measures, the distance-dependent Chinese restaurant process is equivalent to the standard Chinese restaurant process.

$$h_{ij} \equiv 1 \quad \text{for any } i, j \text{ , which implies } \quad s_{ij} = 1 \quad \text{for } j \in \{1, \ldots, i-1\} \tag{2.9}$$

Claim: The distance-dependent Chinese restaurant process is equivalent to the standard Chinese restaurant process in this setting.

Proof. when i=1, it is obvious that Equation 2.6 and Equation 2.8 are equivalent since a new table is open in the context of the standard Chinese restaurant process and a weakly connected sub-graph is created in the context of the distance-dependent Chinese restaurant process.

We assume that when $i = n$ the claim holds. The assumption implies that each weakly connected sub-graph in the context of the distance-dependent Chinese restaurant process corresponds to one table in the context of the standard Chinese restaurant process.

When $i = n + 1$,

Case one: A new table and a new weakly connected sub-graph are created with the same probability.

For the standard Chinese restaurant process, according to Equation 2.6, the probability of the i-th customer sits in a new table is

$$\Pr(z_i = (K+1)|\overrightarrow{z_{-i}}, \alpha) = \frac{\alpha}{n+\alpha}$$

where $\sum_{k=1}^{K} n_k = n$

For the distance-dependent Chinese restaurant process, according to Equation 2.8, the probability of adding a directed edge from the i-th node to itself is

$$\Pr(e_i = i|\overrightarrow{e_{-i}}, \alpha) = \frac{\alpha}{n+\alpha}$$

where $\sum_{j=1}^{i-1} s_{ij} = n$

Case two: A customer joins in an existing table and a node is added into a corresponding existing weakly connected sub-graph with the same probability.

For the standard Chinese restaurant process, according to Equation 2.6, the probability of the i-th customer sits in an exiting table t is

$$\Pr(z_i = t|\overrightarrow{z_{-i}}, \alpha) = \frac{n_t}{n+\alpha}$$

where $\sum_{k=1}^{K} n_k = n$ and $t \in \{1, \ldots, K\}$

For the distance-dependent Chinese restaurant process, by the induction assumption, before adding the i-th node, there exists a weakly connected sub-graph, g_t, corresponding to table t. The assumption also implies that the sub-graph has n_t nodes. According to

Equation 2.8, the probability of creating a directed edge from the i-th node to any node k $(k < i)$ in the sub-graph is

$$\Pr(e_i \in N(g_t)|\overrightarrow{e_{-i}}, \alpha) = \frac{\sum_{j \in N(g_t)} s_{ij}}{\sum_{j=1}^{i-1} s_{ij} + \alpha} = \frac{n_t}{n + \alpha}$$

where $\sum_{j=1}^{i-1} s_{ij} = n$, and $N(g_t)$ is a set of nodes in sub-graph g_t □

A Chinese restaurant process mixture model using a distance-dependent Chinese restaurant process is called a distance-dependent Chinese restaurant process mixture model. The generative procedure of the model is:

1. For each document i in a given document order,
   (a) generate a directed edge from document i to document j, where j is
      i. document i, with probability in proportion to α (which implies that a new weakly connected sub-graph with a distinct label c_{new} is created)
         A. independently generate a corresponding new base model, $\theta_{new} \sim G_0$
         B. let latent model indicator $z_i = c_{new}$
      ii. document k $(k < i)$, with probability in proportion to order-specific measure s_{ik} (which implies the document i is weakly connected to an existing sub-graph with label $c_{existing}$)
         A. let latent model indicator $z_i = c_{existing}$
   (b) independently generate a word sequence, $\overrightarrow{w_i} \sim d(\overrightarrow{w}|\theta_{z_i})$

2.4 Model Estimation

The Gibbs sampling scheme is used in model estimation. Before we discuss the sampling scheme, we first introduce an important property.

2.4.1 Exchangeability

Definition 34. *A membership set c_i is a nonempty set of customers (nodes/documents) at table (weakly-connected subgraph/base model) i.*

Definition 35. *A configuration is a nonempty set of membership sets formed by a Chinese restaurant process.*

Definition 36. *Given a configuration generated by a Chinese restaurant process, if the probability of generating the configuration does not depend on the order of customers (nodes/documents), the process is exchangeable.*

A standard Chinese restaurant process is exchangeable. Let us consider a toy example to illustrate this property. Let us suppose that three customers (documents) named a,b,c, respectively, enter the restaurant. What is more, customer a (document a) and customer c (document c) are at the same table (base model) while customer b (document b) is at another table (base model). Membership sets are $\{a, c\}$ and $\{b\}$ respectively. The configuration is $C = \{\{a, c\}, \{b\}\}$.

Case one: If these customers (documents) enter the restaurant in the order $\overrightarrow{z_1} = (a, b, c)$, according to the standard Chinese restaurant process, the probability of generating configuration C is

$$\Pr(C|\overrightarrow{z_1}, \alpha) = \Pr(D_1) \times \Pr(E_2|D_1) \times \Pr(F_1|D_1 \cup E_2) = 1 \times \frac{\alpha}{1+\alpha} \times \frac{1}{2+\alpha}$$

where D_1, E_2, and F_1 denote customer a (document a) at table 1 (base model 1), customer b (document b) at table 2 (base model 2), and customer c (document c) at table 1 (base model 1) respectively. Figures 2.2, 2.3, 2.4 illustrate the generative procedure given the order (a, b, c).

Case two: If these customers (documents) enter the restaurant in the order $\overrightarrow{z_2} = (b, c, a)$, according to the standard Chinese restaurant process, the probability of generating configuration C is

$$\Pr(C|\overrightarrow{z_2}, \alpha) = \Pr(E_1) \times \Pr(F_2|E_1) \times \Pr(D_2|E_1 \cup F_2) = 1 \times \frac{\alpha}{1+\alpha} \times \frac{1}{2+\alpha}$$

where D_2, E_1, and F_2 denote customer a (document a) at table 2 (base model 2), customer b (document b) at table 1 (base model 1) , and customer c (document c) at table 2 (base model 2) respectively. Figures 2.5, 2.6, 2.7 illustrate the generative procedure given the order (b, c, a).

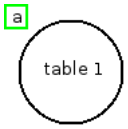


Figure 2.2: Customer a comes in the order (a, b, c)

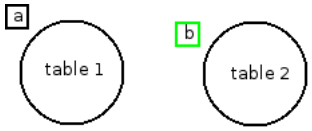


Figure 2.3: Customer b comes in the order (a, b, c)

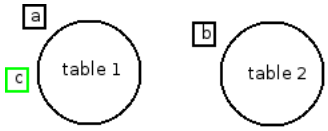


Figure 2.4: Customer c comes in the order (a, b, c)

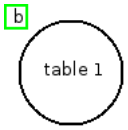


Figure 2.5: Customer b comes in the order (b, c, a)

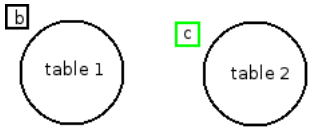


Figure 2.6: Customer c comes in the order (b, c, a)

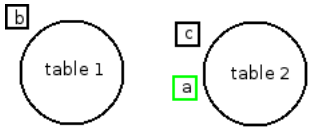


Figure 2.7: Customer a comes in the order (b, c, a)

We can observe that the probability of generating the configuration for these two cases are identical. In fact, given any order of customers entering the restaurant, the probability remains the same. This fact can be used to prove that a standard Chinese restaurant process is exchangeable. Note that in general exchangeability does not hold in a distance-dependent Chinese restaurant process. Let us consider an example in the context of the distance-dependent Chinese restaurant process to demonstrate this difference. Let us recall that $e_i = j$ denotes a directed edge from node i to node j. Suppose there are three nodes (documents) named a,b,c, respectively, pairwise similarity-like measures are given below, and H is a set of these measures.

$$h_{ij} = \begin{cases} 2.0 & \text{for } (i,j) = (a,c) \text{ and } (i,j) = (c,a) \\ 1.0 & \text{otherwise} \end{cases} \quad , \text{ for } i \ , \ j \in \{a, b, c\} \tag{2.10}$$

Case one: If these nodes (documents) are added in the order (a, b, c), then the order-specific pairwise measures are:

$$s_{ij} = \begin{cases} h_{ac} = 2.0 & \text{for } (i,j) = (3,1) \\ 1.0 & \text{other cases when } j < i \end{cases} \text{ , for } i \text{ , } j \in \{1,2,3\} \tag{2.11}$$

Figure 2.8 illustrates the order-specific measures of the distance-dependent Chinese restaurant process in the order (a, b, c). Note that black edges and red edges in figures correspond to edge candidates to be selected in the process. Weights of the black edges are the pairwise measures.

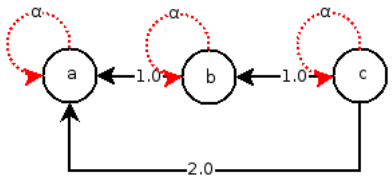


Figure 2.8: Edge candidates in order (a, b, c)

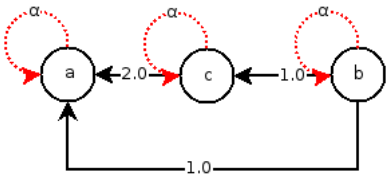


Figure 2.9: Edge candidates in order (a, c, b)

According to the distance-dependent Chinese restaurant process, the probability that node a (the first document) and node c (the third document) are weakly connected (in the same base model) given the order (a, b, c), is

$$\begin{aligned} &\Pr(\text{node } a \text{ and node } c \text{ are weakly connected} | \alpha, H, (a,b,c)) \\ &= \Pr(e_1 = 1, e_2 = 1, e_3 = 1) + \Pr(e_1 = 1, e_2 = 2, e_3 = 1) + \Pr(e_1 = 1, e_2 = 1, e_3 = 2) \\ &= \frac{1.0}{1.0+\alpha}\frac{2.0}{1.0+2.0+\alpha} + \frac{\alpha}{1.0+\alpha}\frac{2.0}{1.0+2.0+\alpha} + \frac{1.0}{1.0+\alpha}\frac{1.0}{1.0+2.0+\alpha} \\ &= \frac{3+2\alpha}{(1+\alpha)(3+\alpha)} \end{aligned} \tag{2.12}$$

Figures 2.10, 2.11, and 2.12 demonstrate all possible cases, when node a (document a) and node c (document c) are weakly connected (in the same base model) in the order (a, b, c). Note that $e_i = j$ denotes an edge from i to j and blue edges represent edges are selected.

Case two: If these nodes (documents) are added in the order (a, c, b), then the order-specific pairwise measures are:

$$s_{ij} = \begin{cases} h_{ac} = 2.0 & \text{for } (i,j) = (2,1) \\ 1.0 & \text{other cases when } j < i \end{cases} \text{ , for } i \text{ , } j \in \{1,2,3\} \tag{2.13}$$

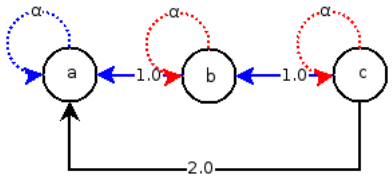


Figure 2.10: $\Pr(e_1 = 1, e_2 = 1, e_3 = 2)$

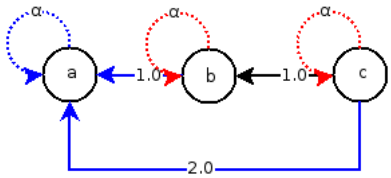


Figure 2.11: $\Pr(e_1 = 1, e_2 = 1, e_3 = 1)$

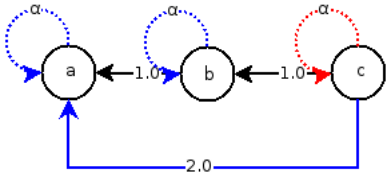


Figure 2.12: $\Pr(e_1 = 1, e_2 = 2, e_3 = 1)$

Figure 2.9 illustrates the order-specific measures of this distance-dependent Chinese restaurant process in the order (a, c, b).

The probability that node a (the first document) and node c (the second document) are weakly connected (in the same base model) given the order (a, c, b) is:

$$\begin{aligned} &\Pr(\text{node } a \text{ and node } c \text{ are weakly connected}|\alpha, H, (a, c, b)) \\ &= \textstyle\sum_j \Pr(e_1 = 1, e_2 = 1, e_3 = j) \\ &= \Pr(e_1 = 1, e_2 = 1) \\ &= \tfrac{2}{2+\alpha} \end{aligned} \tag{2.14}$$

Figures 2.13, 2.14, and 2.15 demonstrate all possible cases, when node a (the first document) and node c (the second document) are weakly connected (in the same base model) in the order (a, c, b).

We can observe that in general Equation 2.12 and Equation 2.14 are different, which means in general the order of nodes matters in a distance-dependent Chinese restaurant process. On the other hand, if h_{ac} and h_{ca} are 1.0 instead of 2.0, these order-specific measures independent on the order of nodes. This is the case when the distance-dependent Chinese restaurant process is equivalent to the corresponding standard Chinese restaurant process. Recall that these nodes (documents) are added in the order (a, b, c), then the order-specific pairwise measures are:

$$s_{ij} = \begin{cases} h_{ac} = 1.0 & \text{for } (i, j) = (3, 1) \\ 1.0 & \text{other cases when } j < i \end{cases}, \text{ for } i \text{ , } j \in \{1, 2, 3\} \tag{2.15}$$

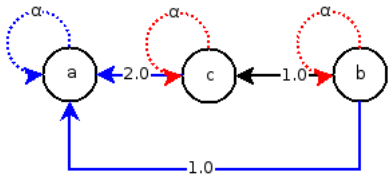


Figure 2.13: $\Pr(e_1 = 1, e_2 = 1, e_3 = 1)$

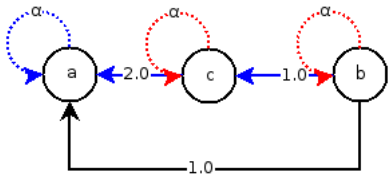


Figure 2.14: $\Pr(e_1 = 1, e_2 = 1, e_3 = 2)$

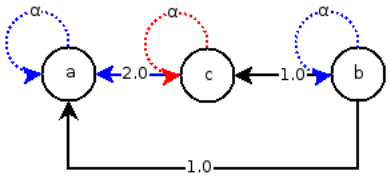


Figure 2.15: $\Pr(e_1 = 1, e_2 = 1, e_3 = 3)$

If these nodes (documents) are added in the order (a, c, b), then the order-specific pairwise measures are:

$$s_{ij} = \begin{cases} h_{ac} = 1.0 & \text{for } (i,j) = (2,1) \\ 1.0 & \text{other cases when } j < i \end{cases} \quad , \text{ for } i \ , j \in \{1,2,3\} \tag{2.16}$$

2.4.2 Collapsed Gibbs Samplers for CRPMs

Recall that in Chapter 1, a multinomial distribution over words in a vocabulary is used to model a document. We use a multinomial family of distributions as modeling distributions for documents ($d(\cdot) = \text{Mult}(\cdot)$) and a Dirichlet distribution as a conjugate prior distribution of these modeling distributions. ($G_0 = Dir(\gamma \times \overrightarrow{1})$, where γ is a positive scalar). Collapsed Gibbs samplers are used in model estimation since the conjugate prior is used. Unlike a standard Gibbs sampler, a collapsed Gibbs sampler does not sample variables which are modeled by the conjugate prior, since these variables can be integrated out. In a Chinese restaurant process mixture model, base models $\{\theta_1, \dots\}$ can be marginalized, since G_0 is a conjugate prior of these base models. According to the Rao-Blackwell theorem, a collapsed Gibbs sampler in general is better than a standard Gibbs sampler in terms of small variance [21].

Collapsed Gibbs Sampler for s-CRPM

A Collapsed Gibbs Sampler for standard Chinese restaurant process mixture models (CGS1) was proposed in [17], where only base model indicators, $\overrightarrow{z}$, are sampled. The sampler is

given in Algorithm 1, where c_{new} is a distinct new label and $Unique(\cdot)$ is a function which constructs a set containing elements of an input vector. The following example illustrates the definition of the function. Given $\overrightarrow{z} = (2, 1, 2, 3)$, $Unique(\overrightarrow{z})$ is $\{1, 2, 3\}$

Algorithm 1: Collapsed Gibbs Sampler for s-CRPM(α,G_0)

input : word sequences of n documents, $D = (\overrightarrow{d_1}, \ldots, \overrightarrow{d_n})$ and maximum iteration $t_{\max}$

output: model indicators for these documents, $\overrightarrow{z} = (z_1, \ldots, z_n)$

initialize $\overrightarrow{z}^{(0)}$, and $t \leftarrow 0$;
while *iteration* $t < t_{max}$ **do**
  $\overrightarrow{z}^{(t+1)} \leftarrow \overrightarrow{z}^{(t)}$;
  for *document* $i \leftarrow 1$ **to** n **do**
    $\overrightarrow{z_{-i}}^{(t+1)} \leftarrow \overrightarrow{z}^{(t+1)} \setminus \{z_i^{(t)}\}$;
    for *model indicator* $c \in Unique(\overrightarrow{z_{-i}}^{(t+1)}) \cup \{c_{new}\}$ **do**
      compute $\Pr(z_i = c|\overrightarrow{d_i}, D_{-i}, \overrightarrow{z_{-i}}^{(t+1)}, G_0, \alpha)$ by Equation 2.17;
    end
    generate $z_i^{(t+1)} \sim \Pr(z_i|\overrightarrow{d_i}, D_{-i}, \overrightarrow{z_{-i}}^{(t+1)}, G_0)$;
    $\overrightarrow{z}^{(t+1)} \leftarrow \overrightarrow{z_{-i}}^{(t+1)} \cup \{z_i^{(t+1)}\}$;
  end
  $t \leftarrow t + 1$;
end
$\overrightarrow{z} \leftarrow \overrightarrow{z}^{(t_{\max})}$;

According to the generative procedure of the standard Chinese restaurant mixture model, for document i, base model indicator z_i is drawn by:

$$\begin{aligned}\Pr(z_i = c|\overrightarrow{d_i}, D_{-i}, \overrightarrow{z_{-i}}, G_0, \alpha) \propto \; & \Pr(z_i = c, \overrightarrow{d_i}|D_{-i}, \overrightarrow{z_{-i}}, G_0, \alpha) \\ & = \Pr(z_i = c|\overrightarrow{z_{-i}}, \alpha) \Pr(\overrightarrow{d_i}|z_i = c, \overrightarrow{z_{-i}}, D_{-i}, G_0)\end{aligned} \tag{2.17}$$

where c is either an existing model label or a new distinct model label, z_i is the model indicator for document i, $\overrightarrow{d_i}$ is the word sequence for document i, $\overrightarrow{z_{-i}}$ denotes observed model indicators excluding z_i, D_{-i} denotes the set of word sequences excluding d_i, and G_0 is a conjugate prior for base models.

Since a standard Chinese restaurant process is exchangeable, the following holds:

$$\begin{aligned}\Pr(z_i = c|\overrightarrow{z_{-i}}, \alpha) \propto\ & \Pr(\overrightarrow{z_{-i}} \cup \{z_i = c\}|\alpha) \\ & = \Pr((z_1, \ldots, z_{i-1}, c, z_{i+1}, \ldots, z_n)|\alpha) \\ & = \Pr((z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n, c)|\alpha) \\ & \propto \begin{cases} \alpha & \text{if c is a label of a new base model} \\ n_{-i,c} & \text{if c is a label of an existing base model} \end{cases}\end{aligned} \tag{2.18}$$

where $n_{-i,c}$ is the number of these documents at table c (base model c) excluding document i in the order $(z_1, \ldots z_{i-1}, z_{i+1}, \ldots, z_n, c)$.

Since G_0 is a conjugate prior of base models, the following equations are closed-form expressions.

$$\begin{aligned}&\Pr(\overrightarrow{d_i}|z_i = c, \overrightarrow{z_{-i}}, D_{-i}, G_0) \\ &= \begin{cases} \int_{\theta_{\text{new}}} \Pr(\overrightarrow{d_i}|\theta_{\text{new}}) \Pr(\theta_{\text{new}}|G_0) d\theta_{\text{new}} & \text{if c is a label of a new base model} \\ \int_{\theta_c} \Pr(\overrightarrow{d_i}|\theta_c) \Pr(\theta_c|D_{-i,c}, G_0) d\theta_c & \text{if c a label of an existing base model} \end{cases}\end{aligned} \tag{2.19}$$

and

$$\begin{aligned}\Pr(\theta_c|D_{-i,c}, G_0) &= \frac{\Pr(\theta_c, D_{-i,c}|G_0)}{\Pr(D_{-i,c}|G_0)} \\ &= \frac{\Pr(\theta_c|G_0) \Pr(D_{-i,c}|\theta_c)}{\int_{\theta_c} \Pr(D_{-i,c}|\theta_c) \Pr(\theta_c|G_0) d\theta_c}\end{aligned} \tag{2.20}$$

where $D_{-i,c}$ denotes documents at table c (base model θ_c) excluding document i given the order $\overrightarrow{z_{-i}} = (z_1, \ldots, z_{i-1}, z_{i+1}, \ldots, z_n)$.

Let us consider a toy example to illustrate Equation 2.19. Suppose there are five nodes (documents) named a, b, c, d, e and they enter the restaurant in the order (a, b, c, d, e). At iteration $t+1$, the sampler is going to sample table label (model indicator), z_3, for the third document (document c). Let us consider the following case, $(z_1^{(t+1)}, z_2^{(t+1)}, z_3^{(t)}, z_4^{(t)}, z_5^{(t)}) = (1, 2, 3, 2, 2)$. Figure 2.16 illustrates the restaurant derived from table assignments $(1, 2, 3, 2, 2)$.

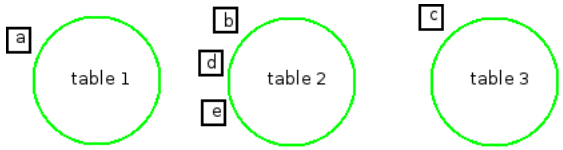


Figure 2.16: $(z_1^{(t+1)}, z_2^{(t+1)}, z_3^{(t)}, z_4^{(t)}, z_5^{(t)}) = (1, 2, 3, 2, 2)$

Let $\overrightarrow{z_{-3}} = (z_1^{(t+1)}, z_2^{(t+1)}, \_, z_4^{(t)}, z_5^{(t)})$. Figure 2.17 demonstrates the restaurant derived from table assignments $\overrightarrow{z_{-3}}$. $Unique(\overrightarrow{z_{-3}}) = \{1, 2\}$, which means there are table 1 and table 2 in the restaurant.

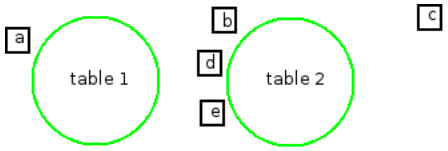


Figure 2.17: $(z_1^{(t+1)}, z_2^{(t+1)}, \_, z_4^{(t)}, z_5^{(t)}) = (1, 2, \_, 2, 2)$

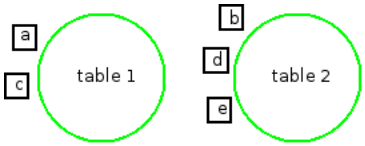


Figure 2.18: $(z_1^{(t+1)}, z_2^{(t+1)}, z_3^{(t+1)}, z_4^{(t)}, z_5^{(t)}) = (1, 2, 1, 2, 2)$

When $z_3^{(t+1)} = 1$, it implies that document c joins in table 1 and table assignments become $\overrightarrow{z_{-3}} \cup \{1\} = (z_1^{(t+1)}, z_2^{(t+1)}, z_3^{(t+1)}, z_4^{(t)}, z_5^{(t)}) = (1, 2, 1, 2, 2)$. Figure 2.18 demonstrates the restaurant derived from table assignments $(1, 2, 1, 2, 2)$. We know that the following holds:

$$\begin{aligned}
\Pr(\overrightarrow{d_3}|z_3^{(t+1)} = 1, \overrightarrow{z_{-3}}, D_{-3}, G_0) &= \frac{\Pr(\overrightarrow{d_3}, D_{-3}|z_3^{(t+1)}=1, \overrightarrow{z_{-3}}, G_0)}{\Pr(D_{-3}|z_3^{(t+1)}=1, \overrightarrow{z_{-3}}, G_0)} \\
&= \frac{\Pr(\overrightarrow{d_3}, D_{-3}|z_3^{(t+1)}=1, \overrightarrow{z_{-3}}, G_0)}{\Pr(D_{-3}|\overrightarrow{z_{-3}}, G_0)} \\
&= \frac{\Pr(D_{\{a,c\}}|G_0)\Pr(D_{\{b,d,e\}}|G_0)}{\Pr(D_{\{a\}}|G_0)\Pr(D_{\{b,d,e\}}|G_0)} \\
&= \frac{\Pr(D_{\{a,c\}}|G_0)}{\Pr(D_{\{a\}}|G_0)} \\
&= \Pr(D_{\{c\}}|D_{\{a\}}, G_0) \\
&= \Pr(\overrightarrow{d_c}|D_{\{a\}}, G_0) \\
&= \int_{\theta_1} \Pr(\overrightarrow{d_c}|\theta_1)\Pr(\theta_1|D_{-c,1}, G_0)d\theta_1
\end{aligned} \tag{2.21}$$

where $\overrightarrow{d_c} = \overrightarrow{d_3}$ since document c in the third customer (document) and $D_{-c,1} = D_{\{a\}}$ is a set of documents in table 1 excluding document c.

Similarly, from Figure 2.19 we can show that when $z_3^{(t+1)} = 2$ and the following holds:

$$\begin{aligned}
\Pr(\overrightarrow{d_3}|z_3^{(t+1)} = 2, \overrightarrow{z_{-3}}, D_{-3}, G_0) &= \frac{\Pr(\overrightarrow{d_3}, D_{-3}|z_3^{(t+1)}=2, \overrightarrow{z_{-3}}, G_0)}{\Pr(D_{-3}|\overrightarrow{z_{-3}}, G_0)} \\
&= \int_{\theta_2} \Pr(\overrightarrow{d_c}|\theta_2)\Pr(\theta_2|D_{-c,2}, G_0)d\theta_2
\end{aligned} \tag{2.22}$$

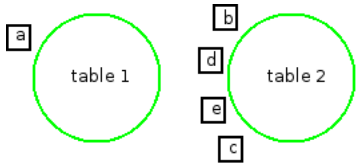


Figure 2.19: $(z_1^{(t+1)}, z_2^{(t+1)}, z_3^{(t+1)}, z_4^{(t)}, z_5^{(t)}) = (1, 2, 2, 2, 2)$

where $\overrightarrow{d_c} = \overrightarrow{d_3}$ and $D_{-c,2} = D_{\{b,d,e\}}$ is a set of documents in table 2 excluding document c.

When $z_3^{(t+1)} = c_{\text{new}}$, it implies that document c open a new table with distinct label $c_{\text{new}} = 4$ and table assignments become $\overrightarrow{z_{-3}} \cup \{4\} = (z_1^{(t+1)}, z_2^{(t+1)}, z_3^{(t+1)}, z_4^{(t)}, z_5^{(t)}) = (1, 2, 4, 2, 2)$. Figure 2.20 demonstrates the restaurant derived from table assignments $(1, 2, 4, 2, 2)$. We know that the following holds:

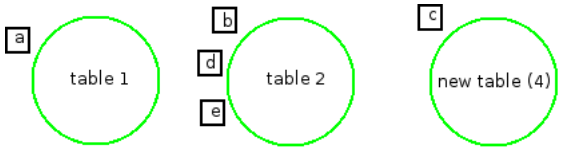


Figure 2.20: $(z_1^{(t+1)}, z_2^{(t+1)}, z_3^{(t+1)}, z_4^{(t)}, z_5^{(t)}) = (1, 2, 4, 2, 2)$

$$
\begin{aligned}
\Pr(\overrightarrow{d_3}|z_3^{(t+1)} = 4, \overrightarrow{z_{-3}}, D_{-3}, G_0) &= \frac{\Pr(\overrightarrow{d_3}, D_{-3}|z_3^{(t+1)}=4, \overrightarrow{z_{-3}}, G_0)}{\Pr(D_{-3}|z_3^{(t+1)}=4, \overrightarrow{z_{-3}}, G_0)} \\
&= \frac{\Pr(\overrightarrow{d_3}, D_{-3}|z_3^{(t+1)}=4, \overrightarrow{z_{-3}}, G_0)}{\Pr(D_{-3}|\overrightarrow{z_{-3}}, G_0)} \\
&= \frac{\Pr(D_{\{a\}}|G_0)\Pr(D_{\{b,d,e\}}|G_0)\Pr(D_{\{c\}}|G_0)}{\Pr(D_{\{a\}}|G_0)\Pr(D_{\{b,d,e\}}|G_0)} \\
&= \Pr(D_{\{c\}}|G_0) \\
&= \Pr(\overrightarrow{d_c}|G_0) \\
&= \int_{\theta_{\text{new}}} \Pr(\overrightarrow{d_c}|\theta_{\text{new}})\Pr(\theta_{\text{new}}|G_0)d\theta_{\text{new}}
\end{aligned}
\tag{2.23}
$$

where $\overrightarrow{d_c} = \overrightarrow{d_3}$ and $\theta_{\text{new}} = \theta_4$.

In the application of document clustering, $\Pr(\overrightarrow{d_i}|\theta_c)$ is a multinomial distribution and $\Pr(\theta_c|G_0)$ is a Dirichlet prior distribution. In this case, $\Pr(\theta_c|D_{-i,c}, G_0)$ is a Dirichlet posterior distribution and $\int_{\theta_c} \Pr(D_{-i,c}|\theta_c)\Pr(\theta_c|G_0)d\theta_c$ is closely related to a Dirichlet-Multinomial distribution.

Collapsed Gibbs Sampler for dd-CRPM

We describe a Collapsed Gibbs Sampler for distance dependent Chinese restaurant process mixture models (CGS2) proposed in [1]. We assume that the order of nodes (documents) is given and pairwise similarity-like measures are known. In order words, the order-specific measures are pre-computed. Let S is a set of these order-specific measures. We refer to this sampler as CGS2. The sampler is given in Algorithm 2.

Based on the generative procedure of the distance-dependent Chinese restaurant mixture model, a directed edge for the i-th document is assigned by:

$$\begin{aligned}\Pr(e_i = j|\overrightarrow{d_i}, D_{-i}, \overrightarrow{e_{-i}}, G_0, \alpha, S) \propto \; & \Pr(e_i = j, \overrightarrow{d_i}|D_{-i}, \overrightarrow{e_{-i}}, G_0, \alpha, S) \\ & = \Pr(e_i = j|\overrightarrow{e_{-i}}, \alpha, S) \Pr(\overrightarrow{d_i}|e_i = j, \overrightarrow{e_{-i}}, D_{-i}, G_0)\end{aligned} \tag{2.24}$$

where $\overrightarrow{d_i}$ is the word sequence for the i-th document, G_0 is the conjugate prior for base models, when $e_i = j$, it denotes a directed edge from the i-th document to the j-th document, D_{-i} is an order list of documents excluding $\overrightarrow{d_i}$ and $\overrightarrow{e_{-i}}$ is an order list of edges excluding e_i.

According to the distance-dependent Chinese restaurant process, the following holds:

$$p(e_i = j|\overrightarrow{e_{-i}}, \alpha, S) \propto \begin{cases} \alpha & \text{for } i = j \\ s_{ij} & \text{otherwise} \end{cases} \tag{2.25}$$

where s_{ij} is the pre-computed order-specific measure between the i-th document to the j-th document.

Let U is a set of nodes (documents) in a weakly connected sub-graph containing the i-th node (document) in the graph derived from edge assignments $\overrightarrow{e_{-i}}$. Let V_j is a set of nodes (documents) in a connected sub-graph containing the i-th node (document) in the graph derived from edge assignments $\overrightarrow{e_{-i}} \cup \{e_i = j\}$. Note that $U \subseteq V_j$. According to the generative procedure of the mixture model, the following holds:

$$\Pr(\overrightarrow{d_i}|e_i = j, \overrightarrow{e_{-i}}, D_{-i}, G_0) = \begin{cases} 1 & \text{if } (V_j - U) = \emptyset \\ \frac{\Pr(D_{(V)}|G_0)}{\Pr(D_{(U)}|G_0) \Pr(D_{(V-U)}|G_0)} & \text{otherwise} \end{cases} \tag{2.26}$$

where $D_{(U)}$ and $D_{(V_j)}$ are word sequences of documents in set U and set V_j respectively and

$$\Pr(D_{(\cdot)}|G_0) = \textstyle\int_\theta \Pr(D_{(\cdot)}|\theta) \Pr(\theta|G_0) d\theta \tag{2.27}$$

Note that if $(V_j - U) \neq \emptyset$, U and $(V_j - U)$ are two distinct weakly connected sub-graphs in the graph derived from edge assignments $\overrightarrow{e_{-i}}$ and Equation 2.27 is a closed-form expression due to the conjugate prior G_0. In the application of document clustering, Equation 2.27 is closely related to a Dirichlet-Multinomial distribution.

The following example illustrates Equation 2.26. Suppose there are five nodes (documents) named a, b, c, d, e and they are added in a graph in the order (a, b, c, d, e). At iteration $t+1$, the sampler is going to sample an edge starting from the third node (document c) to node $e_3^{(t+1)}$. Note that e_i denotes not only a directed edge but also the end node of the edge. Recall that when $e_i = j$, it denotes a directed edge from the i-th document to the j-th document. Let us consider the following case, $(e_1^{(t+1)}, e_2^{(t+1)}, e_3^{(t)}, e_4^{(t)}, e_5^{(t)}) = (1, 2, 2, 3, 3)$. Figure 2.21 illustrates the graph derived from edge assignments $(1, 2, 2, 3, 3)$.

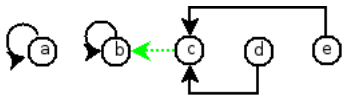


Figure 2.21: $(e_1^{(t+1)}, e_2^{(t+1)}, e_3^{(t)}, e_4^{(t)}, e_5^{(t)}) = (1, 2, 2, 3, 3)$

Let $\overrightarrow{e_{-3}} = (e_1^{(t+1)}, e_2^{(t+1)}, \_, e_4^{(t)}, e_5^{(t)})$. Figure 2.22 demonstrates the graph derived from edge assignments $\overrightarrow{e_{-3}}$. We can know that set $U = \{c, d, e\}$

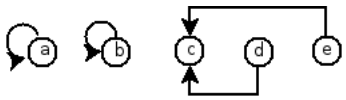


Figure 2.22: $(e_1^{(t+1)}, e_2^{(t+1)}, \_, e_4^{(t)}, e_5^{(t)}) = (1, 2, \_, 3, 3)$

Case one: When $e_3^{(t+1)} = 1$, it implies that a directed edge from document c to document a is added since document a and document c are the first node and the third node respectively. Therefore, $\overrightarrow{e_{-3}} \cup \{e_3^{(t+1)} = 1\} = (e_1^{(t+1)}, e_2^{(t+1)}, e_3^{(t+1)}, e_4^{(t)}, e_5^{(t)}) = (1, 2, 1, 3, 3)$. Figure 2.23 demonstrates the graph derived from edge assignments $(1, 2, 1, 3, 3)$. In this case, set $V_1 = \{a, c, d, e\}$.

For the graph derived from edge assignments $\overrightarrow{e_{-3}}$, we observe that documents a, c, d, e fall in one weakly connected sub-graph in Figure 2.22. Note that these documents fall in two weakly connected sub-graphs $\{a\}$ and $\{c, d, e\}$ according to Figure 2.22. The following

holds:

$$\begin{aligned}
\Pr(\overrightarrow{d_3}|e_3^{(t+1)}=1,\overrightarrow{e_{-3}},D_{-3},G_0) &= \frac{\Pr(\overrightarrow{d_3},D_{-3}|e_3^{(t+1)}=1,\overrightarrow{e_{-3}},G_0)}{\Pr(D_{-3}|e_3^{(t+1)}=1,\overrightarrow{e_{-3}}G_0)} \\
&= \frac{\Pr(\overrightarrow{d_3},D_{-3}|e_3^{(t+1)}=1,\overrightarrow{e_{-3}},G_0)}{\Pr(D_{-3}|\overrightarrow{e_{-3}},G_0)} \\
&= \frac{\Pr(D_{\{a,c,d,e\}}|G_0)\Pr(D_{\{b\}}|G_0)}{\Pr(D_{\{a\}}|G_0)\Pr(D_{\{b\}}|G_0)\Pr(D_{\{c,d,e\}}|G_0)} \\
&= \frac{\Pr(D_{\{a,c,d,e\}}|G_0)}{\Pr(D_{\{a\}}|G_0)\Pr(D_{\{c,d,e\}}|G_0)} \\
&= \frac{\Pr(D_{V_1}|G_0)}{\Pr(D_{(V_1-U)}|G_0)\Pr(D_U|G_0)}
\end{aligned} \tag{2.28}$$

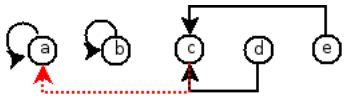


Figure 2.23: $(e_1^{(t+1)}, e_2^{(t+1)}, e_3^{(t+1)}, e_4^{(t)}, e_5^{(t)}) = (1,2,1,3,3)$

Case two: When $e_3^{(t+1)} = 2$, it implies that a directed edge from document c to document b is added since document b and document c are the second node and the third node respectively. Therefore, $\overrightarrow{e_{-3}} \cup \{e_3^{(t+1)} = 2\} = (e_1^{(t+1)}, e_2^{(t+1)}, e_3^{(t+1)}, e_4^{(t)}, e_5^{(t)}) = (1,2,2,3,3)$. Figure 2.24 demonstrates the graph derived from edge assignments $(1,2,1,3,3)$. In this case, set $V_2 = \{b,c,d,e\}$. Similarly, we can show that:

$$\begin{aligned}
\Pr(\overrightarrow{d_3}|e_3^{(t+1)}=2,\overrightarrow{e_{-3}},D_{-3},G_0) &= \frac{\Pr(\overrightarrow{d_3},D_{-3}|e_3^{(t+1)}=2,\overrightarrow{e_{-3}},G_0)}{\Pr(D_{-3}|\overrightarrow{e_{-3}}G_0)} \\
&= \frac{\Pr(D_{\{a\}}|G_0)\Pr(D_{\{b,c,d,e\}}|G_0)}{\Pr(D_{\{a\}}|G_0)\Pr(D_{\{b\}}|G_0)\Pr(D_{\{c,d,e\}}|G_0)} \\
&= \frac{\Pr(D_{\{b,c,d,e\}}|G_0)}{\Pr(D_{\{b\}}|G_0)\Pr(D_{\{c,d,e\}}|G_0)} \\
&= \frac{\Pr(D_{V_2}|G_0)}{\Pr(D_{(V_2-U)}|G_0)\Pr(D_U|G_0)}
\end{aligned} \tag{2.29}$$

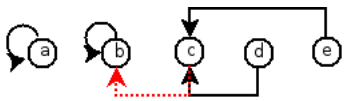


Figure 2.24: $(e_1^{(t+1)}, e_2^{(t+1)}, e_3^{(t+1)}, e_4^{(t)}, e_5^{(t)}) = (1,2,2,3,3)$

Case three: When $e_3^{(t+1)} = 3$, it implies that a directed edge from document c to itself is added and $\overrightarrow{e_{-3}} \cup \{e_3^{(t+1)} = 3\} = (e_1^{(t+1)}, e_2^{(t+1)}, e_3^{(t+1)}, e_4^{(t)}, e_5^{(t)}) = (1,2,3,3,3)$. Figure 2.25 demonstrates the graph derived from edge assignments $(1,2,1,3,3)$. In this case, set $V_3 = \{c,d,e\} = U$. Similarly, we can know that:

$$\begin{aligned}\Pr(\overrightarrow{d_3}|e_3^{(t+1)}=3,\overrightarrow{e_{-3}},D_{-3},G_0) &= \frac{\Pr(\overrightarrow{d_3},D_{-3}|e_3^{(t+1)}=3,\overrightarrow{e_{-3}},G_0)}{\Pr(D_{-3}|\overrightarrow{e_{-3}}G_0)} \\ &= \frac{\Pr(D_{\{a\}}|G_0)\Pr(D_{\{b\}}|G_0)\Pr(D_{\{c,d,e\}}|G_0)}{\Pr(D_{\{a\}}|G_0)\Pr(D_{\{b\}}|G_0)\Pr(D_{\{c,d,e\}}|G_0)} \\ &= 1\end{aligned} \tag{2.30}$$

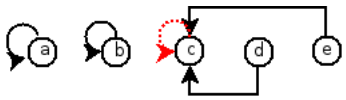


Figure 2.25: $(e_1^{(t+1)}, e_2^{(t+1)}, e_3^{(t+1)}, e_4^{(t)}, e_5^{(t)}) = (1, 2, 3, 3, 3)$

2.5 Hierarchical K-Mixture Model

We discuss a hierarchical K-mixture model which is used to compute pairwise similarity-like measures for document clustering.

2.5.1 Latent Dirichlet Allocation

Recall the Bayesian K-mixture model mentioned in this chapter. We can observe that mixing coefficient $\overrightarrow{\pi}$ is collection-specific and latent model indicator z is document-specific. A hierarchical K-mixture model extends the Bayesian mixture model so that the mixing coefficient is document-specific and the latent model indicator is word-specific. Let us explain this difference in the application of document clustering. The Bayesian mixture model assumes that documents in a collection are generated from the same mixing of base models in expectation. This assumption may be useful to model a collection of feedback documents with respect to a query. In the case of the pseudo-relevant feedback task, if most of feedback documents are relevant, the model is still reasonable. However, if we want to model a collection of heterogeneous documents (say, the whole collection of archived Web pages), a hierarchical mixture model may be a better choice. In a hierarchical K-mixture model, documents are generated from fixed base models but the mixing coefficient of these base models is document-specific. Latent Dirichlet Allocation model (LDA) model [3] is a hierarchical finite mixture model (as known as a mixed membership model) in text mining domain. The hierarchical model assumes that mixing coefficient $\overrightarrow{\pi}$ is document-specific. Without specifying base models, the generative procedure of this hierarchical mixture model is:

1. For each cluster (topic) c (there are K clusters)
   (a) independently generate a base model, $\theta_c \sim G_0$
2. For each document i
   (a) independently generate a document-specific mixing coefficient, $\overrightarrow{\pi_i} \sim \mathrm{Dir}(\overrightarrow{\pi}|\alpha \times \overrightarrow{1}/K)$, where α is a scalar hyper-parameter. Let $\overrightarrow{\alpha} = \alpha \times \overrightarrow{1}/K$
   (b) For the j-th word in document i
      i. independently generate a latent model indicator, $z_{ji} \sim \mathrm{Mult}(z|\overrightarrow{\pi_i})$
      ii. independently generate a word, $w_{ji} \sim d(w|\theta_{z_{ji}})$

The hierarchical K-mixture model can be extended to a hierarchical Chinese restaurant process mixture model (as known as the Chinese restaurant franchise process mixture model and the hierarchical Dirichlet process mixture model). In latent Dirichlet allocation model, modeling distribution $d(\cdot)$ is a multinomial distribution over words in a vocabulary and $G_0 = Dir(\overrightarrow{\beta})$ is a conjugate prior distribution of base models.

2.5.2 Collapsed Gibbs Sampler for LDA

We use a Collapsed Gibbs Sampler for latent Dirichlet allocation models (CGS3) discussed in [9] to estimate latent Dirichlet allocation model. The sampler uses two kinds of conjugacy in this model— $Dir(\overrightarrow{\alpha})$ is a conjugate prior of mixing coefficients and $Dir(\overrightarrow{\beta})$ is a conjugate prior of base models. The Gibbs sampler is given in Algorithm 3.

Suppose there are n documents and word sequences are $D = \{\overrightarrow{d_1}, \ldots, \overrightarrow{d_n}\}$. Let $Z = \{\overrightarrow{z_1}, \ldots, \overrightarrow{z_n}\}$ is a set of document-specific model indicators; $\Theta = \{\theta_1, \ldots, \theta_K\}$ is a set of base models; $\Pi = \{\overrightarrow{\pi_1}, \ldots, \overrightarrow{\pi_n}\}$ is a set of document-specific mixing coefficients.

According to the generative procedure of latent Dirichlet allocation model, the posterior is

$$\begin{aligned} &\Pr(D, Z, \Theta, \Pi|\overrightarrow{\alpha}, G_0) \\ &= \Pr(D, Z, \Theta, \Pi|\overrightarrow{\alpha}, \overrightarrow{\beta}) \\ &= \left(\prod_{t=1}^{K} \Pr(\theta_t|\overrightarrow{\beta})\right) \left[\prod_{i=1}^{n} \Pr(\overrightarrow{\pi_i}|\overrightarrow{\alpha}) \prod_{j=1}^{|\overrightarrow{d_i}|} \Pr(z_{i,j}|\overrightarrow{\pi_i}) \prod_{k=1}^{K} \Pr(d_{i,j}|\theta_t)^{I(z_{i,j}=k)}\right] \end{aligned} \tag{2.31}$$

Equation 2.31 can be written as:

$$\Pr(\Theta|\overrightarrow{\beta})\Pr(\Pi|\overrightarrow{\alpha})\Pr(Z|\Pi)\Pr(D|Z,\Theta) \tag{2.32}$$

Due to the conjugacy, if we integrate out Θ and Π, the marginalized posterior becomes:

$$\Pr(Z, D|\overrightarrow{\alpha}, \overrightarrow{\beta}) = \Pr(Z|\overrightarrow{\alpha})\Pr(D|Z, \overrightarrow{\beta}) \tag{2.33}$$

Note that $Z = (\overrightarrow{z_1}, \ldots, \overrightarrow{z_n})$ are dependent since Π is marginalized. Let $l = (i, j)$. Therefore, d_l denotes the j-th word in document i and z_l denotes the model indicator of d_l. What is more, Z_{-l} denotes a set of sequences of model indicators excluding z_l and D_{-l} denotes a set of word sequences excluding d_l. The following holds:

$$\begin{aligned}
&p(Z_{-l}, D|\overrightarrow{\alpha}, \overrightarrow{\beta}) \\
&= \textstyle\sum_{z_l} p(Z, D|\overrightarrow{\alpha}, \overrightarrow{\beta}) \\
&= \textstyle\sum_{z_l} \Pr(Z|\overrightarrow{\alpha})\Pr(D|Z, \overrightarrow{\beta}) \\
&= \textstyle\sum_{z_l} \Pr(Z_{-l}|\overrightarrow{\alpha})\Pr(z_l|Z_{-l}, \overrightarrow{\alpha})\Pr(D_{-l}|Z, \overrightarrow{\beta})\Pr(d_l|D_{-l}, Z, \overrightarrow{\beta}) \\
&= \textstyle\sum_{z_l} \Pr(Z_{-l}|\overrightarrow{\alpha})\Pr(z_l|Z_{-l}, \overrightarrow{\alpha})\Pr(D_{-l}|Z, \overrightarrow{\beta})\Pr(d_l|z_l, \overrightarrow{\beta}) \\
&= \Pr(Z_{-l}|\overrightarrow{\alpha})\Pr(D_{-l}|Z, \overrightarrow{\beta})\textstyle\sum_{z_l} \Pr(z_l|Z_{-l}, \overrightarrow{\alpha})\Pr(d_l|z_l, \overrightarrow{\beta}) \\
&= \Pr(Z_{-l}|\overrightarrow{\alpha})\Pr(D_{-l}|Z_{-l}, \overrightarrow{\beta})\Pr(d_l|Z_{-l}, \overrightarrow{\alpha}, \overrightarrow{\beta})
\end{aligned} \tag{2.34}$$

where

$$Z = Z_{-l} \cup \{z_l\} \tag{2.35}$$

$$D = D_{-l} \cup \{d_l\} \tag{2.36}$$

Note that $\Pr(d_l|D_{-l}, Z, \overrightarrow{\beta}) = \Pr(d_l|z_l, \overrightarrow{\beta})$ in Equation 2.34 makes use of conditional independence in latent Dirichlet allocation model. Given z_l and $\overrightarrow{\beta}$, not only d_l and D_{-l} but also d_l and Z_{-l} are conditional independent.

Let $Z = Z_{-l} \cup \{z_l = c\}$. The collapsed Gibbs sampler generates model indicator z_l for the j-word in document i by:

$$\Pr(z_l = c|Z_{-l}, D, \overrightarrow{\alpha}, \overrightarrow{\beta}) = \frac{\Pr(Z, D|\overrightarrow{\alpha}, \overrightarrow{\beta})/\Pr(D|\overrightarrow{\alpha}, \overrightarrow{\beta})}{\Pr(Z_{-l}, D|\overrightarrow{\alpha}, \overrightarrow{\beta})/\Pr(D|\overrightarrow{\alpha}, \overrightarrow{\beta})} \tag{2.37}$$

According to Equation 2.33, Equation 2.37 becomes

$$\Pr(z_l = c|Z_{-l}, D, \overrightarrow{\alpha}, \overrightarrow{\beta}) = \frac{\Pr(Z|\overrightarrow{\alpha})\Pr(D|Z, \overrightarrow{\beta})}{\Pr(Z_{-l}, D|\overrightarrow{\alpha}, \overrightarrow{\beta})} \tag{2.38}$$

According to Equation 2.34, Equation 2.38 becomes

$$\begin{aligned}\Pr(z_l = c|Z_{-l}, D, \overrightarrow{\alpha}, \overrightarrow{\beta}) &= \frac{\Pr(Z|\overrightarrow{\alpha})\Pr(D|Z,\overrightarrow{\beta})}{\Pr(Z_{-l}|\overrightarrow{\alpha})p(D_{-l}|Z_{-l},\overrightarrow{\beta})\Pr(d_l|Z_{-l},\overrightarrow{\alpha},\overrightarrow{\beta})} \\ &\propto \frac{\Pr(Z|\overrightarrow{\alpha})}{\Pr(Z_{-l}|\overrightarrow{\alpha})} \times \frac{\Pr(D|Z,\overrightarrow{\beta})}{\Pr(D_{-l}|Z_{-l},\overrightarrow{\beta})}\end{aligned} \tag{2.39}$$

where $\Pr(Z|\overrightarrow{\alpha})$, $\Pr(Z_{-l}|\overrightarrow{\alpha})$, $\Pr(D|Z, \overrightarrow{\beta})$, and $\Pr(D_{-l}|Z_{-l}, \overrightarrow{\beta})$ are Dirichlet-Multinomial distributions in latent Dirichlet allocation model [9] .

Now, we estimate a document-specific mixing coefficient $\overrightarrow{\pi_i}$ for document i given Z obtained from the Gibbs sampler.

$$\begin{aligned}\Pr(\overrightarrow{\pi_i}|Z, D, \overrightarrow{\alpha}, \overrightarrow{\beta}) &\propto \Pr(\overrightarrow{\pi_i}, Z|D, \overrightarrow{\alpha}, \overrightarrow{\beta}) \\ &= \Pr(Z|\overrightarrow{\pi_i}, D, \overrightarrow{\alpha}, \overrightarrow{\beta})\Pr(\overrightarrow{\pi_i}|D, \overrightarrow{\alpha}, \overrightarrow{\beta}) \\ &= \Pr(Z|\overrightarrow{\pi_i}, D, \overrightarrow{\alpha}, \overrightarrow{\beta})\Pr(\overrightarrow{\pi_i}|D, \overrightarrow{\alpha}, \overrightarrow{\beta}) \\ &= \Pr(Z_{-i}|D, \overrightarrow{\alpha}, \overrightarrow{\beta})\Pr(\overrightarrow{z_i}|\overrightarrow{\pi_i}, D, \overrightarrow{\alpha}, \overrightarrow{\beta})\Pr(\overrightarrow{\pi_i}|D, \overrightarrow{\alpha}, \overrightarrow{\beta}) \\ &\propto \Pr(\overrightarrow{z_i}|\overrightarrow{\pi_i})\Pr(\overrightarrow{\pi_i}|\overrightarrow{\alpha})\end{aligned} \tag{2.40}$$

Note that we make use of conditional independence of the latent Dirichlet allocation model in Equation 2.40. $\Pr(\overrightarrow{\pi_i}|Z, D, \overrightarrow{\alpha}, \overrightarrow{\beta})$ is a Dirichlet distribution since $\Pr(\overrightarrow{z_i}|\overrightarrow{\pi_i})$ is a Multinomial distribution and $\Pr(\overrightarrow{\pi_i}|\overrightarrow{\alpha})$ is a Dirichlet distribution. In this thesis we use the following maximum a posteriori estimator to estimate $\overrightarrow{\pi_i}$:

$$\Pr(\pi_{i,j}) = \frac{\sum_{t=1}^{|\overrightarrow{z_i}|} \mathrm{I}(z_{i,t}=j) + \frac{\alpha}{K}}{|\overrightarrow{z_i}| + \alpha} \tag{2.41}$$

where $\mathrm{I}(\cdot)$ is an indicator function. The estimator is different from the Dirichlet smoothing estimator in Chapter 1. Let us recall that the Dirichlet smoothing estimator for a document is to estimate a multinomial distribution over words in a vocabulary. However, this estimator is to estimate a multinomial distribution over K clusters (topics).

2.6 Pairwise Similarity-like Measures

In this section, we propose two pairwise measures for distance-dependent Chinese restaurant process mixture models used in the pseudo feedback task.

2.6.1 Global Latent Measure

Definition 37. *The Jensen-Shannon divergence between θ_i and θ_j is defined as:*

$$JS(\theta_i||\theta_j) = \frac{(KL(\theta_i||\theta_m)+KL(\theta_j||\theta_m))}{2}, \theta_m = \frac{(\theta_j+\theta_i)}{2} \tag{2.42}$$

where "KL" denotes the Kullback-Leibler divergence in Definition 22.

Given a trained latent Dirichlet allocation model on the entire collection of archived documents, the pairwise similarity-like measure between document i and document j is based on the Jensen-Shannon (JS) divergence of document-specific mixing coefficients learned from the latent Dirichlet allocation model. The intuition of this measure is that two documents are similar if they have similar mixing coefficients of clusters.

The pairwise similarity-like measure between document i and document j is:

$$h_{ij} = h_{ji} = \exp(-v \times \mathrm{JS}(\theta_i||\theta_j)) \tag{2.43}$$

where v is a scaling parameter, $\theta_{(\cdot)} = \mathrm{Mult}(\overrightarrow{\pi_{(\cdot)}})$ is a document-specific Multinomial distribution over K clusters in the latent Dirichlet allocation model.

2.6.2 Oracle Measure

Definition 38. *An information need model θ_{info} is a multinomial distribution over words in a vocabulary. The model is used to represent an user's information need.*

Definition 39. *A merged document model $\theta_{i\cup j}$ for document i and document j is a multinomial distribution over words in a vocabulary. The maximum likelihood estimator is used for this model, which is defined as:*

$$\Pr_{\theta_{i\cup j}}(w) = \frac{tf_w^{(i)}+tf_w^{(j)}}{L} \tag{2.44}$$

where $tf_w^{(i)}$ is term frequency of word w in document i, and $L = \sum_w tf_w^{(i)} + tf_w^{(j)}$, is the sum of length of document i and document j.

The idea of this measure is that document i and document j should be similar if the merged document model is close to an information need model and far away from a background noise model.

The measure is defined as:

$$h_{ij} = h_{ij} = \exp(-v \times [\mathrm{KL}(\theta_{i\cup j}||\theta_{\mathrm{info}}) - \mathrm{KL}(\theta_{i\cup j}||\theta_{\mathrm{back}})]) \tag{2.45}$$

where v is a scaling parameter, $\theta_{i\cup j}$, θ_{info} and θ_{back} represent the merged document model, an information need model and a background noise model respectively.

Since the information need model, θ_{info}, is estimated by using relevance judgments, this measure is referred to as the oracle measure.

Algorithm 2: Collapsed Gibbs Sampler for dd-CRPM(α,S,G_0)

input : word sequences of n documents, $D = (\overrightarrow{d_1}, \ldots, \overrightarrow{d_n})$ and maximum iteration $t_{\max}$

output: model indicators for these documents, $\overrightarrow{z} = (z_1, \ldots, z_n)$

initialize edge indicators $\overrightarrow{e}^{(0)}$, and $t \leftarrow 0$;
while *iteration* $t < t_{max}$ **do**
    $\overrightarrow{e}^{(t+1)} \leftarrow \overrightarrow{e}^{(t)}$;
    for *document* $i \leftarrow 1$ **to** n **do**
        $\overrightarrow{e_{-i}}^{(t+1)} \leftarrow \overrightarrow{e}^{(t+1)} \setminus \{e_i^{(t)}\}$;
        for *end node* $j \leftarrow 1$ **to** n **do**
            compute $\Pr(e_i = j | \overrightarrow{d_i}, D_{-i}, \overrightarrow{e_{-i}}^{(t+1)}, G_0, \alpha, S)$ by Equation 2.24;
        end
        generate $e_i^{(t+1)} \sim \Pr(e_i | \overrightarrow{d_i}, D_{-i}, \overrightarrow{e_{-i}}^{(t+1)}, G_0, \alpha, S)$;
        $\overrightarrow{e}^{(t+1)} \leftarrow \overrightarrow{e_{-i}}^{(t+1)} \cup \{e_i^{(t+1)}\}$;
    end
    $t \leftarrow t + 1$;
end
initialize model indicators $\overrightarrow{z}$;
for *document* $i \leftarrow 1$ **to** n **do**
    for *weakly sub-graph* g_c *formed by* $\overrightarrow{e}^{(t_{max})}$ **do**
        if $i \in g_c$ **then**
            `// each document in one and only one sub-graph`
            $\overrightarrow{z_i} \leftarrow c$, where c is the label of sub-graph g_c;
            break;
        end
    end
end

Algorithm 3: Collapsed Gibbs Sampler for LDA(α,G_0)

`// Note that in LDA,` $G_0 =$ `Dir(`$\overrightarrow{\beta}$`)`

input : word sequences of n documents, $D = (\overrightarrow{d_1}, \ldots, \overrightarrow{d_n})$ and maximum iteration $t_{\max}$

output: model indicators for these documents, $Z = (\overrightarrow{z_1}, \ldots, \overrightarrow{z_n})$

initialize $Z^{(0)}$, and $t \leftarrow 0$;
while *iteration* $t < t_{max}$ **do**
  $Z^{(t+1)} \leftarrow Z^{(t)}$;
  for *document* $i \leftarrow 1$ **to** n **do**
    `//` Z_i `denotes word-specific model indicators,` $\overrightarrow{z_i}$`, for doc` i
    $\overrightarrow{z_i}^{(t+1)} \leftarrow Z_i^{(t+1)}$;
    for *the* j*-th word in word sequence* $\overrightarrow{d_i}$ **do**
      for *model indicator* $c \leftarrow 1$ **to** K **do**
        `//` `Let` $l = (i, j)$ `denotes word` j `in doc` i
        `//` `Let` $\overrightarrow{\alpha} = \alpha \times \overrightarrow{1}/K$
        `//` z_l `denotes the model indicator for word` j `in doc` i
        `//` $\overrightarrow{z_{-l}}$ `denotes model indicators,` $\overrightarrow{z_i}$`, in doc` i `excluding` z_l
        compute $\Pr(z_l = c|\overrightarrow{z_{-l}}^{(t+1)}, D, \overrightarrow{\alpha}, G_0)$ by Equation 2.37;
      end
    end
    generate $z_l^{(t+1)} \sim \Pr(z_l|\overrightarrow{z_{-l}}^{(t+1)}, D, \overrightarrow{\alpha}, G_0)$;
    $\overrightarrow{z_i}^{(t+1)} \leftarrow \overrightarrow{z_{-l}}^{(t+1)} \cup \{z_l^{(t+1)}\}$;
    $Z_i^{(t+1)} \leftarrow \overrightarrow{z_i}^{(t+1)}$;
  end
  $t \leftarrow t + 1$;
end
$Z \leftarrow Z^{(t_{\max})}$;

Chapter 3

Experiments

3.1 Experimental Setup

Recall the research question in Chapter 2. In order to answer this question, we investigated the following sub-questions in detail through experiments.

1. How much performance is gained in the pseudo-relevance feedback task by using the Chinese restaurant process mixture models (standard Chinese restaurant process mixture models and distance-dependence Chinese restaurant process mixture models) to cluster top-ranked documents compared against the simple mixture method and the relevance method?

2. How does using document-to-document pairwise measures in distance-dependence Chinese restaurant process mixture models affect performance?

3. How sensitive is the performance to hyper-parameters of these mixture models?

Throughout our experiments, we use a background noise model θ_{back}, where the background noise model is estimated by:

$$\Pr_{\theta_{\text{back}}}(t) = \frac{tf_t}{L} \tag{3.1}$$

where tf_t is the term frequency of term t across documents in a collection, and $L = \sum_w tf_w$ is the total number of terms in the collection.

Variables	Usage	Value
u in Equation 1.6	The Dirichlet smoothing parameter	1500 (Default value in the Lemur toolkit)
$\pi^{(SM)}$ in Subsection 1.2.4	The mixing coefficient for SM	0.9 (Suggested by [14])
$w_d^{(RM)}$ in Equation 1.10	A relevance measure of a feedback document model for RM	$\exp(\text{score}(\theta_{\text{d}}^{(RM)}\|\theta_{\text{query}}))$ (Suggested by [13])
w_a in Equation 3.4	A relevant measure of an aspect model for RM	$\exp(\text{score}(\theta_{\text{aspect}}\|\theta_{\text{query}}))$
$\pi^{(q)}$ in Equation 3.2	Smoothing weight for expanded query model	0.5 (Suggested by [15])
$\overrightarrow{\beta}$ in Subsection 2.3.1, 2.3.2, and 2.5.1	The parameter of $G_0 = \text{Dir}(\overrightarrow{\beta})$ in LDA, s-CRPM and dd-CRPM	$\overrightarrow{0.1}$
$\overrightarrow{\alpha}$ in Subsection 2.5.1	The hyper-parameter of mixing coefficient in LDA	$\frac{\overrightarrow{50}}{K}$, where K is # of latent base models (Suggested by [7])
$t_{max}^{(LDA)}$ in Algorithm 3	# of iterations of CGS3 for LDA	2000
$t_{\text{burn-in}}^{(LDA)}$ in Algorithm 3	# iterations of the burn-in period of CGS3 for LDA	1500
t_{max} in Algorithm 2	# of iterations of CGS2 for s-CRPM and dd-CRPM	1500
$t_{\text{burn-in}}$ in Algorithm 2	# of iterations of the burn-in period of CGS2 for s-CRPM and dd-CRPM	1000
K in Subsection 2.5.1	# of latent base models in LDA	200
W_{top} in Equation 3.2	# of top informative terms for a truncated model	100
$D_{feedback}$ in Subsection 3.1	# of top retrieved documents used as feedback documents	100
D_{ret} in Subsection 3.1	# of retrieved documents in a ranked doc list	1000

Table 3.1: Fixed parameters used in experiments

Definition 40. *A truncated feedback model, $\theta^{(\cdot)}_{truncated}$, is a W_{top}-dim multinomial distribution over words, and contains only the top W_{top} words from a feedback model, $\theta^{(\cdot)}_{feedbck}$, in terms of probability.*

All experiments followed the same setup as below. Table 3.1 summarizes the fixed parameters used in the experiments. The overall experimental procedure follows the framework mentioned in Subsection 2.1. The framework is summarized below:

In the first retrieval phrase, a query model θ_{query} is constructed from a query, and given the query model, documents are retrieved. Top $D_{feedback}$ ranked documents are considered as feedback documents.

Secondly, in the query expansion phrase, feedback models, $\theta^{(\cdot)}_{feedback}$, are respectively estimated by the simple mixture method, the relevance method, the standard Chinese restaurant process mixture model and distance-dependence Chinese restaurant process mixture models. Given a feedback model, an expanded query model θ_{expand} is constructed based on Definition 30. In practice, a truncated feedback model is used to construct an expanded query model.

$$\theta_{expand} = \pi^{(q)} \times \theta_{query} + (1 - \pi^{(q)}) \times \theta^{(\cdot)}_{truncated} \tag{3.2}$$

where $\theta^{(\cdot)}_{truncated}$ is a truncated feedback model (for example, $\theta^{(SM)}_{truncated}$ is a truncated feedback model for the simple mixture method), $\pi^{(q)}$ is a smoothing weight for the expanded query model, and θ_{query} is the query model.

Finally, in the second retrieval phrase, given the expanded query model, a ranked list which contains D_{ret} documents is returned as a search result.

3.1.1 Dataset and Text Pre-processing

We used a data set from the Robust track of Text REtrieval Conference (TREC) 2004, which has 528,155 news articles and 646,707 unique terms. The average length of terms in a document is 487. We used three query sets for this collection, each of which contains 50 TREC queries and relevance judgments. Each TREC query contains a title field, a description field and a narrative field. For a TREC query, the description and narrative fields are used to describe the information need behind the query, and the title field is

Field	Content
TREC query ID	301
Title	International Organized Crime
Description	Identify organizations that participate in international criminal activity, the activity, and, if possible, collaborating organizations and the countries involved.
Narrative	A relevant document must as a minimum identify the organization and the type of illegal activity (e.g., Columbian cartel exporting cocaine). Vague references to international drug trade without identification of the organization(s) involved would not be relevant.

Table 3.2: Fields in a TREC query

Statistics	Query set 1	Query set 2	Query set 3
TREC query IDs	301-350	351-400	401-450
# of queries used in experiments	49	48	50
# of judged docs	77109	60454	86830
# of identified relevant docs	4650	4287	4728

Table 3.3: Statistics of query sets

used as a query submitted to an information retrieval system. Table 3.2 shows a TREC query. We discarded three test queries since no relevant document was found among the top-$D_{feedback}$ rated documents for these queries. Some statistics of the query sets used in our experiments can be found at Table 3.3.

The Lemur toolkit was used to index documents in the TREC collection. Our test queries were taken from the title field of the query sets. Documents and test queries were pre-processed by a standard procedure implemented in the toolkit. In a retrieval phrase, given a query model, the toolkit returns a ranked document list based on the KL ranking algorithm.

Latent Dirichlet allocation model was trained via the collapsed Gibbs sampler (CGS3) by sampling $t_{\text{max}}^{(LDA)}$ times. The first $t_{\text{burn-in}}^{(LDA)}$ samples were discarded. Figure 3.1 is a trace plot of the perplexity. Note that the perplexity is closely related to the log-likelihood of data.

Definition 41. *The perplexity of the latent Dirichlet allocation model is defined as:*

$$perplexity(D) = \exp\left(\frac{-\log(\Pr(D|Z,\vec{\alpha},\vec{\beta}))}{|D|}\right) \tag{3.3}$$

where $D = (\overrightarrow{d_1}, \ldots, \overrightarrow{d_n})$ *is a set of word sequences of a collection,* n *is the number of documents in the collection,* $Z = (\overrightarrow{z_1}, \ldots, \overrightarrow{z_n})$ *obtained from the sampler per iteration,* $\log(\Pr(D|Z, \overrightarrow{\alpha}, \overrightarrow{\beta}))$ *is the log-likelihood of data and* $|D| = \sum_{i=1}^{n} |\overrightarrow{d_i}|$.

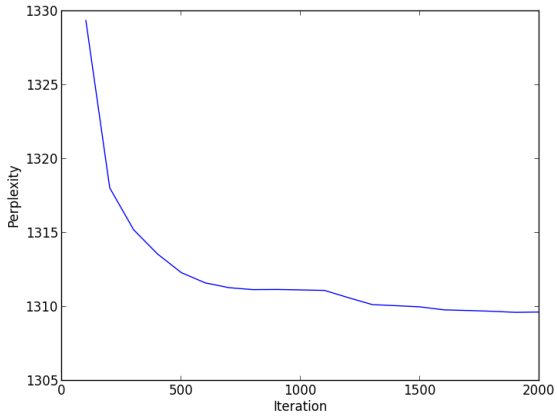


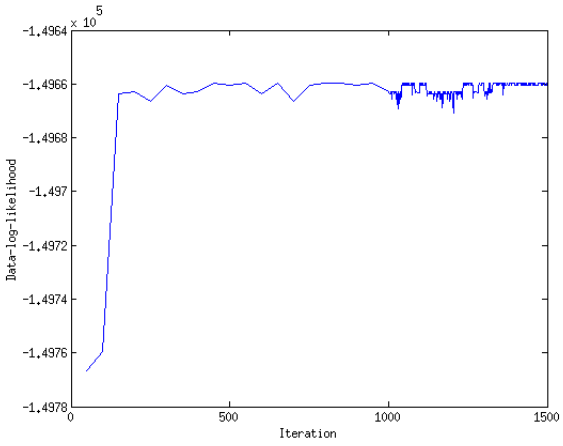
Figure 3.1: LDA of 200 latent base models

3.1.2 Estimation in Query Expansion

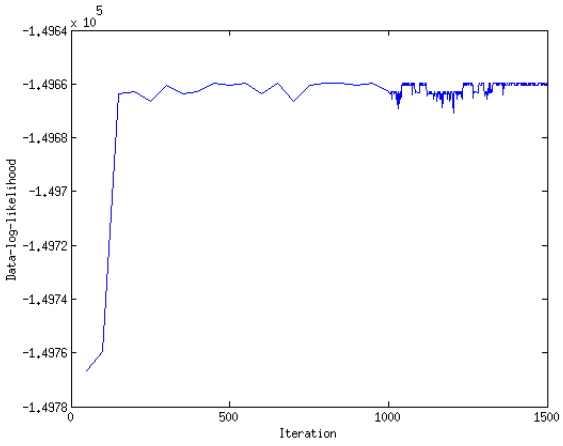
Since standard Chinese restaurant process mixture model is a special case of distance-dependence Chinese restaurant process mixture model, we used only the collapsed Gibbs sampler (CGS2) for the distance-dependence Chinese restaurant process mixture model in our experiments. The sampler sampled t_{max} times and the first $t_{\text{burn-in}}$ samples were discarded. Note that a fixed random number sequence was used in the sampler across all test queries.

Some trace plots of the log-likelihood of these mixture model are shown in Figures 3.2, 3.4, 3.3, which are trace plots for TREC query 310. Note that the log-likelihoods in the figures were recorded at each 100 iterations during the burn-in period. After the period, we recorded the log-likelihoods at each one iteration.

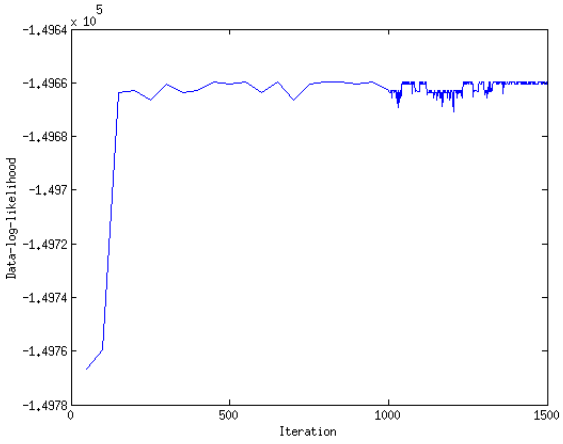
Let us recall Step 4 in Subsection 2.1's framework. Once the Gibbs sampler returns samples, we can estimate an aspect model θ_{aspect} for a cluster configuration derived from these samples. For example, documents in cluster c (base model c) derived from the samples are used to estimate θ_{aspect_c} using the simple mixture method.



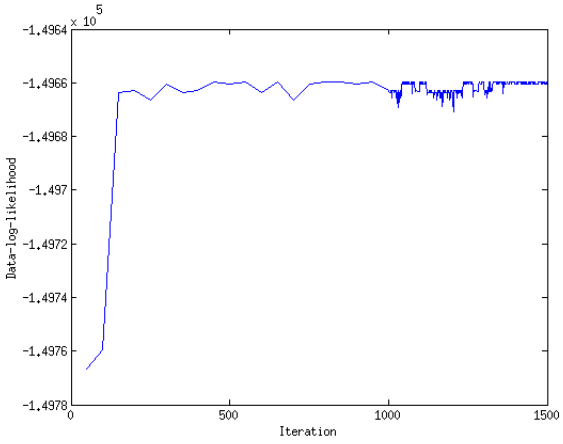
(a) s-CRPM at $\alpha = 0.01$



(b) s-CRPM at $\alpha = 0.1$

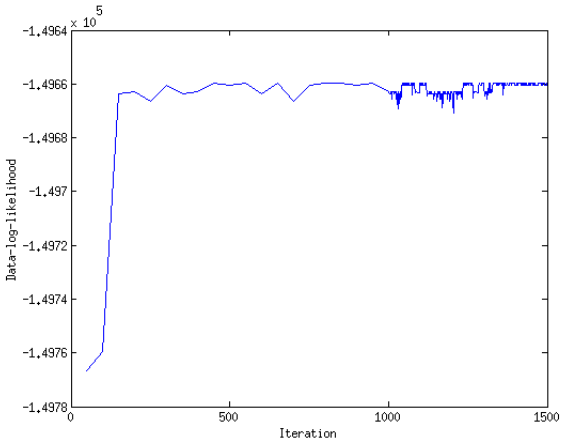


(c) s-CRPM at $\alpha = 1.0$

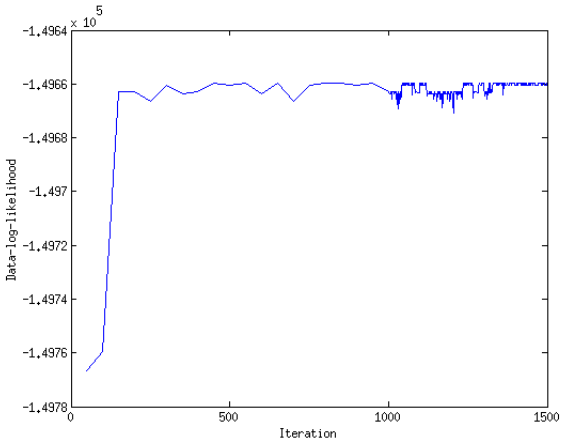


(d) s-CRPM at $\alpha = 10.0$

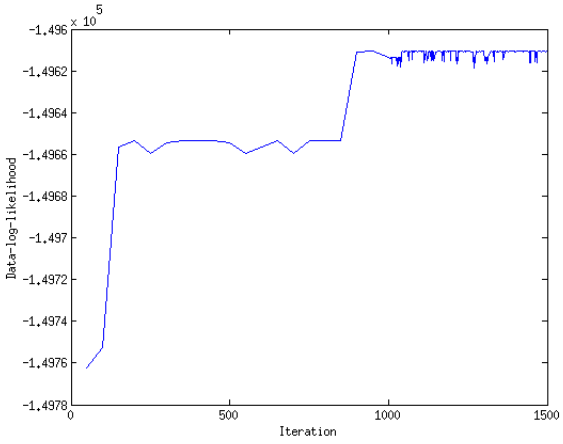
Figure 3.2: For TREC query 310



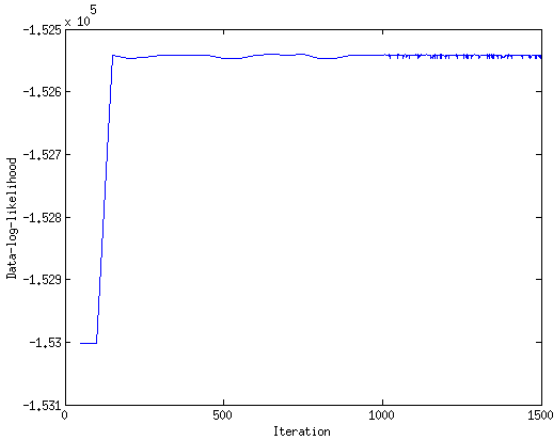
(a) dd-CRPM@ORC where $v = 0.01$



(b) dd-CRPM@ORC where $v = 0.1$

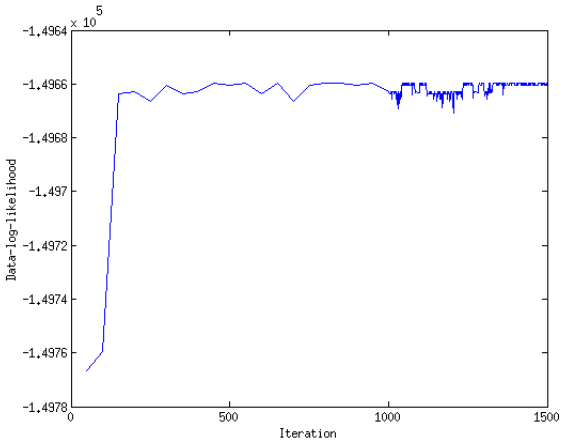


(c) dd-CRPM@ORC where $v = 1.0$

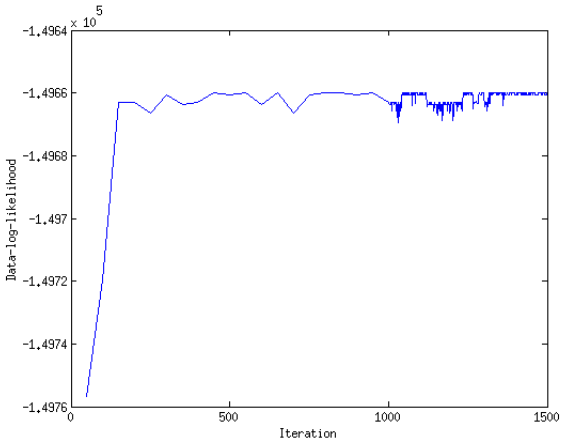


(d) dd-CRPM@ORC where $v = 10.0$

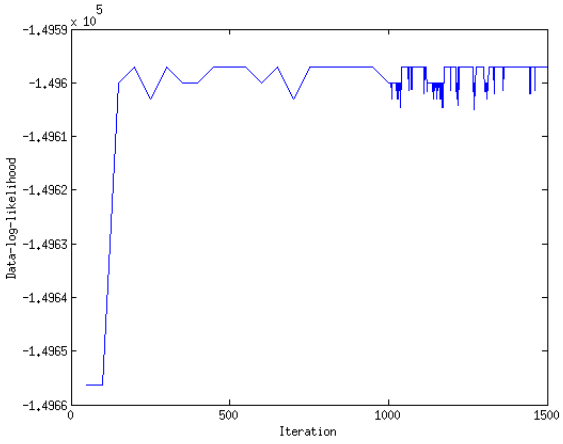
Figure 3.3: For TREC query 310



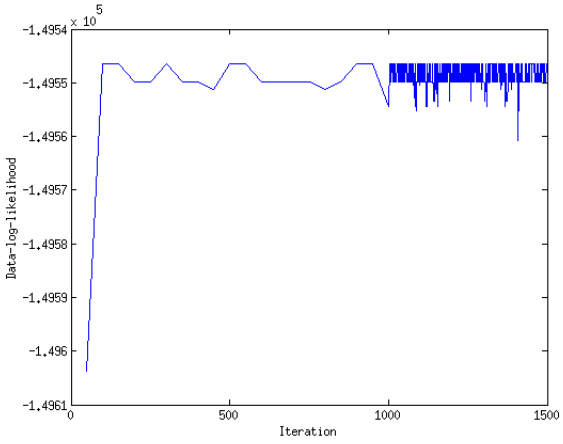
(a) dd-CRPM@LDA where $v = 0.01$



(b) dd-CRPM@LDA where $v = 0.1$



(c) dd-CRPM@LDA where $v = 1.0$



(d) dd-CRPM@LDA where $v = 10.0$

Figure 3.4: For TREC query 310

According to Step 5 in Subsection 2.1's framework, given the aspects models, we estimate a feedback model by adopting the relevance method. The adopted method for the aspect models is:

$$\theta_{\text{feedback}} = \frac{\sum_{\text{aspect a}} w_a \times \theta_{\text{aspect}_a}}{\sum_{\text{aspect a}} w_c} \tag{3.4}$$

where w_a is a relevance measure of aspect a, and θ_{aspect_a} is an aspect model for aspect a.

For standard Chinese restaurant process mixture models, hyper-parameter α was tuned. We found that when α ranges from 0.001 to 1000, the result remains the same, as can be observed in Figure 3.2. In order to compare with the distance-dependence Chinese restaurant process mixture model, we set α to 1. For the distance-dependence Chinese restaurant process mixture model, hyper-parameter α was fixed to 1 and the scaling parameter used in external measures (2.45 and 2.43), v, was set to 0.1.

For the global latent measures, given the samples obtained from the Gibbs sampler (CGS3) for the latent Dirichlet allocation model, we can compute the measures based on Equations 2.41 and 2.43. For the oracle measures, the information need model was estimated by applying the simple mixture method to relevant documents obtained from relevance judgment.

3.2 Results

For evaluation, we used MAP, MNDCG@1000 and MP@20. Tables 3.4, 3.5, 3.6 summarize the results of our experiments, where "@LDA" denotes the distance-dependence Chinese restaurant process mixture model using the global latent measures, and "@ORC" denotes the mixture model using the oracle measures. "*" and "+" denote that p values in the both significant tests—randomized test and t test—are less than 0.05 compared against the relevance method and the simple mixture method respectively. Similarly, "**" and "++" denote that p values in these significant tests are less than 0.01 compared against the baseline methods respectively.

From these tables, we can observe that the number of clusters used in a query-specific clustering algorithm affects the performance of the feedback task at all evaluation metrics. The Chinese restaurant process mixture models in general perform better than the

	Evaluation Metric		
Method	MAP	MNDCG@1000	MP@20
SM (baseline)	0.2568	0.4890	0.3235
RM (baseline)	0.2621	0.5069	0.3459
s-CRPM	**0.2670**	**0.5121+**	**0.3633**+**
dd-CRPM@LDA	0.2646	0.5113+	0.3622**+
dd-CRPM@ORC	0.2660	0.5117+	0.3612**+

Table 3.4: Results for query set 1, where $\alpha = 1$ and $v = 0.1$

	Evaluation Metric		
Method	MAP	MNDCG@1000	MP@20
SM (baseline)	0.2076	0.4792	0.3583
RM (baseline)	0.2134	0.4789	0.3990
s-CRPM	0.2257**+	0.4983**+	**0.4115+**
dd-CRPM@LDA	0.2258**+	0.4990**+	0.4063+
dd-CRPM@ORC	**0.2265**+**	**0.5002**+**	0.4083+

Table 3.5: Results for query set 2, where $\alpha = 1$ and $v = 0.1$

	Evaluation Metric		
Method	MAP	MNDCG@1000	MP@20
SM (baseline)	**0.2632**	**0.5640**	0.4050
RM (baseline)	0.2466	0.5386	0.4000
s-CRPM	0.2613**	0.5572**	**0.4130**
dd-CRPM@LDA	0.2607*	0.5569**	0.4110
dd-CRPM@ORC	0.2610**	0.5573**	0.4120

Table 3.6: Results for query set 3, where $\alpha = 1$ and $v = 0.1$

baseline methods—the relevance method and the simple mixture method. However, the standard Chinese restaurant process mixture model performs better than the distance-dependence Chinese restaurant process mixture model. We further did sensitivity analysis of the distance-dependence Chinese restaurant process mixture model.

3.3 Sensitivity Analysis

In order to study the sensitivity of the scaling parameter defined in our proposed measures, we set v from low to high. Tables 3.7, 3.8, 3.9, 3.10, 3.11, 3.12 show the performance at different parameter settings.

	Evaluation Metric		
dd-CRPM@LDA	MAP	MNDCG@1000	MP@20
v=0.01	0.2644	0.5113	0.3633
v=0.1	0.2646	0.5113	0.3622
v=1.0	0.2642	0.5097	0.3622
v=10	0.2654	0.5111	0.3643

Table 3.7: Results for query set 1, where $\alpha = 1$

	Evaluation Metric		
dd-CRPM@LDA	MAP	MNDCG@1000	MP@20
v=0.01	0.2258	0.4987	0.4063
v=0.1	0.2258	0.4990	0.4063
v=1.0	0.2239	0.4956	0.4083
v=10	0.2228	0.4938	0.4073

Table 3.8: Results for query set 2, where $\alpha = 1$

	Evaluation Metric		
dd-CRPM@LDA	MAP	MNDCG@1000	MP@20
v=0.01	0.2609	0.5566	0.4130
v=0.1	0.2607	0.5569	0.4110
v=1.0	0.2610	0.5569	0.4100
v=10	0.2588	0.5555	0.4070

Table 3.9: Results for query set 3, where $\alpha = 1$

	Evaluation Metric		
dd-CRPM@ORC	MAP	MNDCG@1000	MP@20
v=0.01	0.2648	0.5105	0.3602
v=0.1	0.2660	0.5117	0.3612
v=1.0	0.2650	0.5096	0.3592
v=10	0.2546	0.4988	0.3429

Table 3.10: Results for query set 1, where $\alpha = 1$

	Evaluation Metric		
dd-CRPM@ORC	MAP	MNDCG@1000	MP@20
v=0.01	0.2259	0.4989	0.4073
v=0.1	0.2265	0.5002	0.4083
v=1.0	0.2247	0.4973	0.4073
v=10	0.2137	0.4862	0.3833

Table 3.11: Results for query set 2, where $\alpha = 1$

	Evaluation Metric		
dd-CRPM@ORC	MAP	MNDCG@1000	MP@20
v=0.01	0.2611	0.5572	0.4130
v=0.1	0.2610	0.5573	0.4120
v=1.0	0.2606	0.5572	0.4120
v=10	0.2549	0.5537	0.4020

Table 3.12: Results for query set 3, where $\alpha = 1$

From these tables, we can observe that a weak prior belief (say $v = 0.1$), tends to perform better than a strong prior belief (say $v = 10$). What is more, a strong prior belief may cause the Gibbs sampler to take too long to converge or it may not converge at all. The performance is not sensitive given a reasonable range of the parameter (say, v from 0.01 to 1.0). However, these proposed measures did not help the distance-dependence Chinese restaurant process mixture model outperform the standard Chinese restaurant process mixture model, possibly because data dominate prior belief. This fact can also be

observed in the trace plots of the log-likelihood in Figures 3.2, 3.4, and 3.3. From these figures, a prior belief has little impact on the log-likelihood if the belief is not too strong.

Chapter 4

Conclusion and Future Work

4.1 Conclusion

Based on our experiments, we now answer the sub-questions in Chapter 3. The Chinese restaurant process mixture models have promising performance compared against the relevance method and the simple mixture method. This fact implies that the number of clusters plays an important role in the effectiveness of the feedback task. The proposed measures have little impact on the clustering result, possibly because data dominate prior beliefs. For the standard Chinese restaurant process mixture model, hyper-parameter α does not change the cluster configuration when α falls within a reasonable range. However, the scaling parameter does affect the cluster configuration. When the parameter is big enough, a prior belief dominates data, thus affecting the cluster configuration.

4.2 Future Work

Several investigation could extend our study. In this thesis, we examined these Chinese restaurant process mixture models only in a homogeneous collection. In future, we plan to study these models in heterogeneous datasets such as collections of Web pages. According to TREC reports, the relevance method is often more robust than the simple mixture method in heterogeneous datasets. On the other hand, the simple mixture method is reported to perform better than the relevance method in homogeneous datasets. We expect that these Chinese restaurant process mixture models will perform well since they can automatically adjust the model based on the nature of a dataset. Studying the hierarchical

extension of Chinese restaurant process mixture models such as the Chinese restaurant franchise process mixture models [19] in the query-specific clustering is another direction. It is interesting to study these Chinese restaurant process mixture models in a classification context. Although in the query-specific clustering context the distance-dependent model cannot outperform the standard model, we expect that the distance-dependent model will outperform the standard model, since usually prior regularization is essential in classification tasks. Lastly, thanks to the promising performance of the standard Chinese restaurant process mixture model, the truncated mode using variational inference [2, 12] in the feedback task is worth to explore. Variational inference is fast and suitable for an online retrieval system, since a Gibbs sampler takes too long to use in practice.

Acronyms

AP Average Precision. 3

Bern Bernoulli distribution. 10

CGS1 Collapsed Gibbs Sampler for standard Chinese restaurant process mixture models. 27

CGS2 Collapsed Gibbs Sampler for distance dependent Chinese restaurant process mixture models. 32

CGS3 Collapsed Gibbs Sampler for latent Dirichlet allocation models. 36

CRP Chinese Restaurant Process. 16

CRPM Chinese Restaurant Process Mixture model. 11

dd-CRPM Distance Dependent Chinese Restaurant Process Mixture model. 11

Dir Dirichlet distribution. 15

DirMult Dirichlet-Multinomial distribution. 16

IR Information Retrieval. 2

JS Jensen-Shannon. 39

KL Kullback-Leibler. 5

LDA Latent Dirichlet Allocation models. 35

MAP Mean Average Precision. 5

MNDCG@N Mean Normalized Discounted Cumulative Gain at position N. 5

MP@N Mean Precision at position N. 5

Mult Multinomial distribution. 14

NDCG@N Normalized Discounted Cumulative Gain at position N. 3

P@N Precision at position N. 3

RM Relevance Method. 10

s-CRPM Standard Chinese Restaurant Process Mixture model. 11

SM Simple Mixture method. 9

References

[1] David M Blei and Peter I Frazier. Distance dependent chinese restaurant processes. *The Journal of Machine Learning Research*, 12:2461–2488, 2011.

[2] David M Blei, Michael I Jordan, et al. Variational inference for dirichlet process mixtures. *Bayesian analysis*, 1(1):121–143, 2006.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[4] Stefan Büttcher, Charles LA Clarke, and Gordon V Cormack. *Information retrieval: Implementing and evaluating search engines*. Mit Press, 2010.

[5] W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010.

[6] Bela A Frigyik, Amol Kapila, and Maya R Gupta. Introduction to the dirichlet distribution and related processes. *Department of Electrical Engineering, University of Washignton, UWEETR-2010-0006*, 2010.

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

[8] Jiyin He, Edgar Meij, and Maarten de Rijke. Result diversification based on query-specific cluster ranking. *Journal of the American Society for Information Science and Technology*, 62(3):550–571, 2011.

[9] Gregor Heinrich. Parameter estimation for text analysis. Technical report, Technical report, 2005.

[10] Katherine A Heller and Zoubin Ghahramani. Bayesian hierarchical clustering. In *Proceedings of the 22nd international conference on Machine learning*, pages 297–304. ACM, 2005.

[11] Brian Kulis and Michael I. Jordan. Revisiting k-means: New algorithms via bayesian nonparametrics. In *ICML*, 2012.

[12] Kenichi Kurihara, Max Welling, and Yee Whye Teh. Collapsed variational dirichlet process mixture models. In *IJCAI*, volume 7, pages 2796–2801, 2007.

[13] Victor Lavrenko and W Bruce Croft. Relevance based language models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127. ACM, 2001.

[14] Yuanhua Lv and ChengXiang Zhai. Adaptive relevance feedback in information retrieval. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 255–264. ACM, 2009.

[15] Yuanhua Lv and ChengXiang Zhai. A comparative study of methods for estimating query language models with pseudo feedback. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1895–1898. ACM, 2009.

[16] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[17] Radford M Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of computational and graphical statistics*, 9(2):249–265, 2000.

[18] J. Pitman. *Combinatorial stochastic processes*, volume 1875 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 2006.

[19] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical dirichlet processes. *Journal of the american statistical association*, 101(476), 2006.

[20] C.J. Van Rijsbergen. *Information retrieval*. Butterworths, 1979.

[21] Deepak Venugopal and Vibhav Gogate. Dynamic blocking and collapsing for gibbs sampling. In *Uncertainty in Artificial Intelligence (UAI)*, 2013.

[22] ChengXiang Zhai. Statistical language models for information retrieval. *Synthesis Lectures on Human Language Technologies*, 1(1):1–141, 2008.

[23] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410. ACM, 2001.