HIERARCHICAL BAYESIAN TOPIC MODELING WITH SENTIMENT AND AUTHOR EXTENSION

by

MING YANG

B.E., Nanjing University of Posts and Telecommunications, China, 2009

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2016

Abstract

While the Hierarchical Dirichlet Process (HDP) has recently been widely applied to topic modeling tasks, most current hybrid models for concurrent inference of topics and other factors are not based on HDP.

In this dissertation, we present two new models that extend an HDP topic modeling framework to incorporate other learning factors. One model injects Latent Dirichlet Allocation (LDA) based sentiment learning into HDP. This model preserves the benefits of nonparametric Bayesian models for topic learning, while learning latent sentiment aspects simultaneously. It automatically learns different word distributions for each single sentiment polarity within each topic generated.

The other model combines an existing HDP framework for learning topics from free text with latent authorship learning within a generative model using author list information. This model adds one more layer into the current hierarchy of HDPs to represent topic groups shared by authors, and the document topic distribution is represented as a mixture of topic distribution of its authors. This model automatically learns author contribution partitions for documents in addition to topics.

HIERARCHICAL BAYESIAN TOPIC MODELING WITH SENTIMENT AND AUTHOR EXTENSIONS

by

MING YANG

B.E., Nanjing University of Posts and Telecommunications, China, 2009

A DISSERTATION

submitted in partial fulfillment of the
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computing and Information Sciences
College of Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

2016

Approved by:

Major Professor
William H. Hsu

Copyright

Abstract

While the Hierarchical Dirichlet Process (HDP) has recently been widely applied to topic modeling tasks, most current hybrid models for concurrent inference of topics and other factors are not based on HDP.

In this dissertation, we present two new models that extend an HDP topic modeling framework to incorporate other learning factors. One model injects Latent Dirichlet Allocation (LDA) based sentiment learning into HDP. This model preserves the benefits of nonparametric Bayesian models for topic learning, while learning latent sentiment aspects simultaneously. It automatically learns different word distributions for each single sentiment polarity within each topic generated.

The other model combines an existing HDP framework for learning topics from free text with latent authorship learning within a generative model using author list information. This model adds one more layer into the current hierarchy of HDPs to represent topic groups shared by authors, and the document topic distribution is represented as a mixture of topic distribution of its authors. This model automatically learns author contribution partitions for documents in addition to topics.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank all the people who have helped me during my Ph.D study here in Kansas State University for all these years.

First and Foremost, I would like to thank my advisor William H. Hsu. He is very helpful, knowledgeable, patient and enthusiastic. Dr. Hsu is very supportive, and has many brilliant ideas, I have learned a lot from him throughout my study under his supervision.

I also want to express my gratitude to all my committee members, Dr. Torben Amtoft, Dr. Xinming Ou, and Dr. Shing I Chang, for their suggestions and advice, which helped me a lot with this dissertation.

I also appreciate all the help from my colleagues in KDD (Laboratory for Knowledge Discovery in Databases) group, especially Wesam Elshamy, Surya Kallumadi, Joshua Weese, and many undergraduate students. I want to thank Qiaozhu Mei, Chong Wang, and Hongning Wang, who helped me in various ways.

Last but not least, I want to thank all my family members for taking care of me and supporting me all the time.

Dedication

To my family and my friends.

Chapter 1

Introduction

Nonparametric Bayesian topic model frameworks[1,2], such as the Hierarchical Dirichlet Process (HDP)[3], have been proven to work successfully and more accurately than other extant approaches such as latent semantic analysis (LSA)[4], and its probabilistic analogue[5]. HDPs have also been used directly and solely in many real-world applications. However, as a fundamental text analysis framework, extensions to HDP have not garnered much attention within the area of natural language processing.

In the real-world applications alluded to above, the topic extraction problem is always accompanied by other learning needs, such as sentiment analysis[6], author identification[7], community detection[8,9], and so on. To make full use of the benefits and advantages of the HDP topic inference framework, and in particular to learn a better hidden structure of documents, the synthesis of HDP with learning models from other text analysis studies deserves exploration.

Based on a deep investigation of topic modeling and the theoretical foundations of the HDP framework, this dissertation aims to extend HDP topic modeling framework to incorporate sentiment analysis/author identification learning needs, to form hybrid text analysis models. These hybrid models can solve topic modeling and sentiment analysis/author identification problems in the meantime.

The primary novel contribution of this work is the systematic and principled extension of HDP to incorporate sentiment and co-authorship as independent properties of document corpora, which we accomplish by synthesizing basic HDPs with generative formulations of sentiment and author components. We treat sentiment as a separate parameter to be paired with topic parameters, so that the full pair (dyad) of sentiment and topic conditionally vary based on hyperparameters governing the disposition of a document author. This new approach allows us to capture sentiment-topic parameters within a holistic nonparametric Bayesian framework. Independently of this, we treat authors as participating entities represented within the traditional HDP mixture model, which we extend to capture authors as DP mixtures of global topics in which they have inferable expertise, and documents in corpora as finite mixture of its authors, in whose creation they have participated. This is the first sentiment-topic model we know of that incorporates sentiment as an orthogonal component of any such HDP-based hybrid topic model, and similarly the first HDP-based author-topic model.

The central thesis of this work is that extending the HDP using Latent Dirichlet Allocation (LDA), and similar nonparametric Bayesian formulations of sentiment and author components, allows straightforward extensions to accurately capture and infer meaningful sentiment-topic combinations, as well as useful author-topic distributions for imputation of author expertise. This can be empirically evaluated in our applications by looking at our prediction result for predefined categorical rating values from inferred topic-level sentiment result from our `HDPsent` model in domains such as product and service reviews, using fully unsupervised learning. Furthermore, we are also able to validate the posterior distribution of authors and attributed topics learned from our `HDPauthor` model in academic publication corpora by our performance on some retrieval tasks.

1.1 Topic Modeling

Since the rise of text-driven data mining and decision support in a wide variety of application domains such as recommender systems and personalized decision support, text analytics systems have been well-studied and developed. Topic modeling, as one major branch in this field, has been used in many domains, such as discovering and generating topics in global corpora, identifying and differentiating language patterns for different topics, and associating topics with documents. Topic models are also helpful in many natural language processing (NLP) subareas, including document summarization, generation, classification and organization, and in particular text-based information retrieval (IR) and information extraction (IE).

The major milestones in topic modeling are based on building probabilistic generative models[10][11]. This includes Probabilistic Latent Semantic Indexing (pLSI)[5], Latent Dirichlet Allocation (LDA)[12], and nonparametric Bayesian hierarchical model - Hierarchical Dirichlet Process (HDP)[3].

These topic models have been proved to be powerful and robust for learning topics from corpus. Instead of classifying or clustering documents to separate categories, these models capture the underlying latent probabilistic mixing proportions of multiple categories for each document. For example, one document on bioinformatics may admit different proportions of topics such as "biology", "data mining" and "statistics". Meanwhile, another document on social network analysis may represent a mixture of identifiable topics such as "graph theory", "data mining" and "statistics". Global topics may be represented in multiple documents. This statistical mixture model does not only helps to identify topics for documents more accurately, but also improves the word distribution gathering for different topics.

1.1.1 Problem Definition

There are many ways of defining and solving the topic modeling problem. In this dissertation, however, we focus only on probabilistic methods of constructing statistical mixture models to simulate a generative process of text for documents.

From this point of view, in topic modeling, we generally define and use word sequences in text collection as data to analyze. Therefore, in the text collection, we only use *words* as the basic unit of the data set, representing its granularity. We ignore the punctuation in documents, the sentence structure of words, as well as the part-of-speech (POS) tagging of words.

Here we define the following terms:

1. Each distinct word is treated as one distinct variable in data set, denoted as w. The set of all distinct words in whole text collections is denoted as vocabulary W with size V. For simplicity, we index each word in vocabulary beforehand as $W = \{1, ..., V\}$, and then represent each word by its index id.

2. Each document in collection is considered to be represented by an array of N words, regardless of punctuation and non-word characters. It is denoted as $d_j = \{x_{j1}, x_{j2}, ..., x_{jN}\}$. Variable x_{ji} represents the ith word token in jth document, whose value should be one $w \in \{1, ..., V\}$. Although we refer to each variable x_{ji} by its position, we here assume that each token is generated independently from all other tokens in this document, given the generative model. Therefore, the order of word tokens in a document does not matter. And we also assume that each document is generated independently from all other documents, so that the order of documents in text collection does not matter, too. This exchangeability feature allows us to treat each document as a *bag of words (BOW)*, which means that the positions of words in same document are interchangeable, and the positions of documents are also interchangeable.

3. The whole data set consists of a collection of documents, hereafter referred to as

corpus, which represents the set $D = \{d_1, d_2, ..., d_m\}$.

1.1.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) was introduced by Blei[12] and is a widely used generative statistical model of text collection. Instead of directly producing multinomial distributions of words in topics, and multinomial distributions of topics in each document, LDA brings in the Dirichlet distribution as a conjugate prior for these multinomial distributions.

This model defines a hierarchical Bayesian model for generative process for word tokens in text. Here we represent a graphical plate model of LDA generative process in figure 1.1:

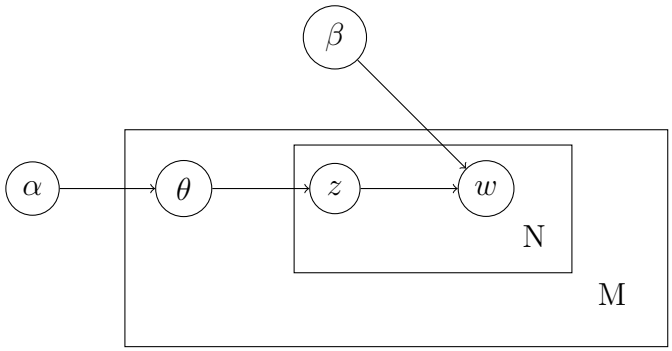

Figure 1.1: Graphical plate model of LDA

This model predefines K topics in a corpus, and then associates each word with one latent topic variable z, where $z \in \{1, ...K\}$. Therefore document j that is originally denoted as $d_j = \{x_{j1}, x_{j2}, ..., x_{jN}\}$ can also be represented by this sequence of latent topic variables $d_j = \{z_{j1}, z_{j2}, ..., z_{jN}\}$, which is sampled according to the multinomial probability distribution over topic categories for this document, denoted $\pi_j = \{\pi_{j1}, ..., \pi_{jK}\}$. We also assume that each topic k is associated with a multinomial probability distribution over the whole vocabulary W with word size V, denoted as $\phi_k = \{\phi_{k1}, ..., \phi_{kV}\}$.

Since multinomial distributions can have Dirichlet distributions as prior parameters, In this model we makes use of this feature, and assume that the topic distributions for documents $\{\pi_1, ..., \pi_m\}$ all have Dirichlet distribution $Dir(\alpha)$ as their conjugate prior. And

the word distributions for topics $\{\phi_1, ..., \phi_K\}$ have Dirichlet distribution $Dir(\beta)$ as their conjugate prior.

The generative process of LDA for word tokens can be represented as follows:

1. For each topic k, we sample $\phi_k \sim Dir(\beta)$.
2. For each document d_j, we sample $\pi_j \sim Dir(\alpha)$.
3. For each token x_{ji} in document d_j at position i:
   (a) We sample a latent topic label $z_{ji} \sim Multinomial(\pi_j)$.
   (b) We sample a word $w \sim Multinomial(\phi_{z_{ji}})$.

The inference part of the LDA model is complex, since it involves posterior distribution calculation of latent variables θ and z generated by LDA model for documents, given the observed data w and prior hyper parameters α and β:

$$p(\theta, z|w, \alpha, \beta) = \frac{p(\theta, z, w|\alpha, \beta)}{p(w|\alpha, \beta)} \tag{1.1}$$

This posterior distribution is unable to compute directly, so that the exact inference of LDA model is intractable. There are two major algorithms applied widely for approximate inference of LDA, *Variational Inference*[13] and *Gibbs Sampling*[14].

Here we introduce the inference process using Gibbs sampling algorithm. Gibbs sampling does not require to infer latent parameters θ and ϕ explicitly. These parameters can be integrated out through the assignment of z.

According to the definition of Gibbs sampling, we do not need to sample all latent variables in whole data set $\{z_{11}, z_{12}, ..., z_{mN-1}, z_{mN}\}$ together, whose joint probability is actually intractable. We can sequentially sample each z based on values of all other z.

Thus, following Griffiths[14], the conditional posterior distribution of z_{ji} given values of all other variables is:

$$p(z_{ji} = k|\mathbf{z}^{-ji}, \mathbf{w}, \alpha, \beta) \propto p(w_{ji}|z_{ji} = k, \mathbf{z}^{-ji}, \mathbf{w}^{-ji}, \beta)p(z_{ji} = k|\mathbf{z}^{-ji}, \alpha) \tag{1.2}$$

where $\mathbf{z}^{-ji} = \{z_{j'i'}|j'i' \neq ji\}$ and $\mathbf{w}^{-ji} = \{w_{j'i'}|j'i' \neq ji\}$.

In this equation, however, $p(z_{ji} = k|\mathbf{z}_{-ji}, \alpha)$ can be treated as a predictive new sample z_{ji} from multinomial distribution θ_j with $Dir(\alpha)$ as its conjugate prior, and $\mathbf{z}^{-ji}$ as its observed data set. To calculate this predictive posterior distribution of variable z_{ji}, we can infer that:

$$p(z_{ji} = k|\mathbf{z}^{-ji}, \alpha) = \frac{n_{jk}^{-ji} + \alpha}{n_{j\cdot}^{-ji} + K\alpha} \tag{1.3}$$

Similarly, $p(w_{ji}|z_{ji} = k, \mathbf{z}^{-ji}, \mathbf{w}^{-ji}, \beta)$ can also be deemed as a predictive new sample of w_{ji} from multinomial distribution ϕ_k with $Dir(\beta)$ as its conjugate prior, $\mathbf{z}^{-ji}$ and $\mathbf{w}^{-ji}$ as its observed data set. We can similarly infer that:

$$p(w_{ji}|z_{ji} = k, \mathbf{z}^{-ji}, \mathbf{w}^{-ji}, \beta) = \frac{n_{kw}^{-ji} + \beta}{n_{k\cdot}^{-ji} + W\beta} \tag{1.4}$$

Putting equations 1.3 and 1.4 together, we can easily get the conditional sampling probability $p(z_{ji} = k|\mathbf{z}^{-ji}, \mathbf{w}, \alpha, \beta)$. Then we can directly use Gibbs sampling schema to sample each z sequentially until the Markov chain converges and reaches a stable state.

1.1.3 Hierarchical Dirichlet Process

The hierarchical Dirichlet process (HDP) is a widely used generative model for topic learning. HDPs were introduced by Teh[3] and are a type of nonparametric hierarchical Bayesian model. One of its most favorable features is that the number of topics that a user has to set up beforehand is not directly bounded, but only regulated by a prior probability of generating a new topic.

The graphical plate model corresponding to HDP mixture model is shown in figure 1.2:

In this model, H can be treated as a prior distribution over topics. It defines a global

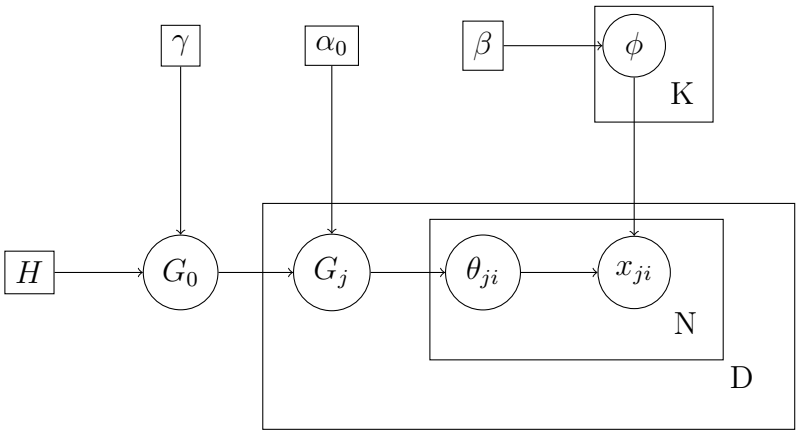

Figure 1.2: Graphical plate model of HDP

measure G_0 for the whole corpora as a Dirichlet Process with H as base measure, and γ as concentration parameter. For each document d_j in this corpora, it generates its own probability distribution G_j over topics as Dirichlet Process with G_0 as base measure, and α_0 as concentration parameter. Then the topic label θ_{ji} is sampled from G_j, word token x_{ji} then is generated similarly to LDA according to its topic label.

The two-level hierarchical Dirichlet process mixture model can be represented as:

$$\begin{aligned} &G_0|\gamma, H \sim DP(\gamma, H) && \\ &G_j|\alpha_0, G_0 \sim DP(\alpha_0, G_0) && \text{for each } j, \\ &\theta_{ji}|G_j \sim G_j && \text{for each } j \text{ and } i, \\ &x_{ji}|\theta_{ji} \sim F(\theta_{ji}) && \text{for each } j \text{ and } i, \end{aligned} \tag{1.5}$$

Since exact inference over HDPs is also intractable, this model also contains two widely used approximate inference techniques, *Variational Inference*[15][16] and *Gibbs Sampling*[3] as a special form of *Markov chain Monte Carlo* (MCMC) algorithm. HDP uses *Chinese restaurant franchise* as a representation framework for building posterior distribution of latent variables for Gibbs Sampling. Although Gibbs sampling is not as computationally efficient

or easy to be scaled as variational inference, it is one more accurate and unbiased way for parameter estimation, and it is also widely used in many applications.

According to Chen[17], with the representation framework of *Chinese restaurant franchise*[3], the generative process of HDP for word tokens can be represented as follows:

1. Draw an infinite number of topics $\phi_k \sim Dir(\beta)$ for $k = \{1, 2, 3...\}$.
2. Draw stick-breaking topic proportions as $\nu_k \sim Beta(1, \gamma)$ for $k = \{1, 2, 3...\}$.
3. For each document d_j:
   (a) we sample document-level topic atoms $k_{jt} \sim Multinomial(\sigma(\boldsymbol{\nu}))$ for each table $t = \{1, 2, 3...\}$.
   (b) we then sample document-level stick-breaking proportions as $\pi_{jt} \sim Beta(1, \alpha)$ for each table $t = \{1, 2, 3...\}$.
   (c) For each token x_{ji} in document d_j at position i:
      i. We sample a latent topic label $z_{ji} \sim Multinomial(\sigma(\boldsymbol{\pi_j}))$.
      ii. We sample a word $w \sim Multinomial(\phi_{z_{ji}})$.

Here $\sigma(\boldsymbol{\nu})$ and $\sigma(\boldsymbol{\pi_j})$ are distributions constructed by stick-breaking algorithm[18][19] with proportions of $\boldsymbol{\nu} = \{\nu_k | k = 1, 2, 3, ...\}$ and $\boldsymbol{\pi_j} = \{\pi_{jt} | t = 1, 2, 3, ...\}$ as:

$$\begin{aligned} \sigma_k(\boldsymbol{\nu}) &= \nu_k \prod_{i=1}^{k-1}(1 - \nu_i) \\ \sigma_t(\boldsymbol{\pi_j}) &= \pi_{jt} \prod_{i=1}^{t-1}(1 - \pi_{ji}) \end{aligned} \tag{1.6}$$

Thus the Chinese Restaurant Franchise Process[20] could be represented in Figure 1.3:

1.1.4 Markov Chain Monte Carlo and Gibbs Sampling

Since the inference algorithm for the statistical mixture model that I am going to introduce is basically Gibbs Sampling, which is one specific algorithm developed from Markov Chain

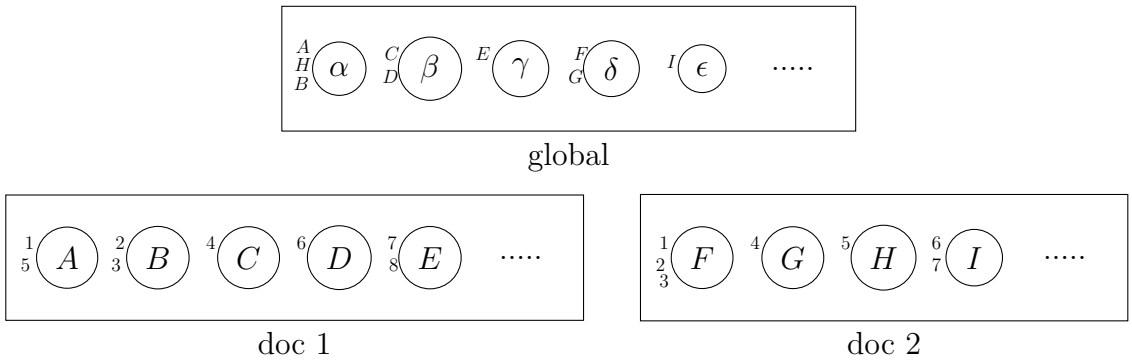

Figure 1.3: HDP: Chinese Restaurant Franchise Representation

Monte Carlo (MCMC) framework[21][22], it is also worth writing about the basic theories of this approximate inference technique.

Monte Carlo Integration[23] makes use of *Law of large numbers*[24]. It approximates the integral of a complex function by a sample mean. We assume that X is a random variable that draws from a probability distribution $\pi(\cdot)$. If we want to calculate the expectation of function $f(x)$ given probability distribution of x as $\pi(\cdot)$, then we can get:

$$\begin{aligned} E[f(X)] &= \frac{\int f(x)\pi(x)\mathrm{d}x}{\int \pi(x)\mathrm{d}x} \\ &\approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) \qquad (x_i \sim \pi(\cdot)) \end{aligned} \tag{1.7}$$

However, in some cases, it is difficult or impossible to draw samples directly and independently from a complex probability distribution. One way to solve this problem is to construct a Markov chain[25] whose stationary distribution is $\pi(\cdot)$ and then sample a sequence of random variables $\{x_1, x_2, ..., x_N\}$ through this Markov chain. Each state in Markov chain is a variable value, which is sampled from last sample value using transition function.

Gibbs Sampling is one algorithm developed from MCMC method. It is a special case of Metropolis-Hastings algorithm[26] from MCMC. It is always used for solving the sampling problem of a multivariate probability distribution, while the joint probability of the set of variables in intractable.

Assume that the random variable X we want to sample is a k-dimensional multivariate variable as $X = \{X_1, X_2, ..., X_k\}$ while its joint probability $p(X) = p(X_1, X_2, ..., X_k)$ is infeasible to compute directly. Instead, we can sample component variable i of X in jth sample x_{ji} from its conditional probability on all the other variable as

$$p(x_{ji}|x_{j1}, x_{j2}, ..., x_{ji-1}, x_{j-1i+1}, ..., x_{j-1k}) \tag{1.8}$$

Thus, in Gibbs Sampling, each multivariate variable X is sampled by sequentially sampling each of its component variables, conditionally on current values of all other variables.

1.2 Sentiment-Topic Model: `HDPsent`

One research area closely related to topic modeling is sentiment analysis, which refers to the uses of text for learning the underlying polarity (positive or negative tone) and subjective attitude of author (or authors) of documents. Early approaches towards using machine learning to detect the overall sentiment polarity of text documents used basic supervised inductive learning for classification. Hypothesis languages and learning algorithms underlying such techniques include Naive Bayes, Maximum Entropy, and Support Vector Machines, as applied by Pang[27 6] and Liu[28].

Compared to overall sentiment polarity learning, however, detailed sentiment polarity learning combined with topics is more favorable. Topic learning embedded into sentiment analysis provides users and researchers with a hybrid model for simultaneous topic distribution and sentiment polarity analysis of documents. Moreover, it also helps to enhance the ability for isolating sentiment polarities from different topics in same document, and provides with the ability to infer separate aspect-level sentiment clusters with different word distributions.

There are some benefits and advantages we can gain from a hybrid topic sentiment model. By modeling sentiment analysis along with topic learning under HDP framework, we are not

restricted by predefined topic size. We can not only discover new topics representing different data groups, but also form sentiment word clusters under each of the topics generated.

Furthermore, we can identify different word distributions with same sentiment polarity under different topics, as well as differentiate same word with different sentiment polarities on different topics. This flexibility improves our ability of learning topic and sentiment combination clusters across the whole corpora more precisely, also improves our ability of identifying the aspect-level sentiment polarities on different topics in one document.

1.3 Author-Topic Model: `HDPauthor`

Another extension of topic modeling is to incorporate author identification information within documents into the learning process.

This research problem consists of several key technical objectives, one of which is to identify topic interests for each author according to the documents that one participates in. For documents finished by cooperation of a set of authors, we also want to learn the contribution for each author involved in this document. Moreover, author identification information itself can be very helpful as a supporting learning resource for topic learning of documents. By constructing global topic interests of authors across corpora, knowledge of authorship can help us to learn topics for documents better. Finally, by computing the topic distribution of all documents that same author participates in, the topic interests of this author can be more accurately inferred as well.

Besides topic learning for documents and authors, our `HDPauthor` model also achieves learning of mixing proportions for authors of each document. The learning result can be used directly for estimation of author contribution.

Examples of applications of topic and author mixture learning model include author identity disambiguation problem. In scientific publications, distinct authors frequently have the same name. There are also some authors who show up in different papers with dif-

ferent names due to variations in abbreviation. Incorporating the feature of topic interest distributions for authors would help us to alleviate this disambiguation problem.

While author searching, grouping, ranking and recommendation are useful tools in many document/author retrieval applications, the topic interests of authors learned from this model also provide features for direct similarity comparison between different authors, using other advanced machine learning techniques.

1.4 Road Map

This dissertation aims to cover two hybrid inference model as extension of HDP topic learning framework. The chapters are organized as follows:

In Chapter 2, we present one novel hybrid learning approach based on the existing HDP topic learning framework, which combines topic modeling with sentiment analysis within one generative inference process. This model preserves the benefits of nonparametric Bayesian models for topic learning, while learns latent aspect-level sentiment features for each topic generated simultaneously.

In Chapter 3, we introduce one novel model that extends the current HDP topic model to incorporate author cooperation information. This model infers topic interests of each author involved in a corpus first, and then establishes the topic distribution of each document in the corpus as a finite mixture of the topic interest of all its authors. This model not only manages to learn topics for documents, and topic interests for each author, but also is able to learn author contributions for each document.

In Chapter 4, we describe in detail the data sets we gathered from real-world text for experiments on our models. We also introduce the criteria we use for evaluation of these experiments. We then describe our experiments and document results that we collected from them.

In Chapter 5, we present conclusions regarding the derivation and use of the model, and

review remaining open problems and some research directions regarding these models that we propose to explore in future work.

Chapter 2

Sentiment-Topic Model: `HDPsent`

With the growing need for analyses of free text that extract both feature information and sentiment polarity, hybrid probabilistic models that support concurrent topic and sentiment analysis have also increased in relevance and significance. Many models treat topic modeling and sentiment analysis as separate and independent processes, which lacks the ability for isolating sentiment polarity from different topics. We would like to infer the topics of documents, but also want to infer the sentiment information for these topics.

There are some algorithms which already attempt to build a hybrid inference model for topic and sentiment learning[29 30], but these models do not fully make use of the current state of the field in nonparametric Bayesian HDP models as a representational framework.

For example, when we analyze product or service reviews, it is crucial that we have separate sentiment polarity information for each feature aspects, which helps us to differentiate opinion words for different aspects from one review text. This, in turn, extends our ability for feature-specific sentiment polarity analysis.

In this chapter we present a technique for simultaneously inferring sentiment and topic from free text, extending existing HDP models, called `HDPsent`. Our model is the first to extend the existing HDP model by adding a sentiment label l along with a topic label k to each token in a document. This approach uses Gibbs sampling for inference, as do

implementations of the Chinese restaurant franchise process (CRFP) for the generative HDP model.

2.1 Related Work

Some significant work in the past decade has begun to combine topic modeling and sentiment analysis in a single model. In applications of the Topic Sentiment Mixture Model (TSM)[30], a Probabilistic Latent Semantic Analysis (PLSA) model is used to represent the generative process. Furthermore, even it assigns topic label for each word (excluding background words), that word itself is sampled from either general positive, negative model, or that specific topic model. This generative process generalizes sentiment polarity model and has limited ability to make different sentiment polarity word distributions for different topics. However, our intuition is that different topics might treat same words with different sentiment strength, or even different polarity. For example, the word "small" might be a positive word when it is describing the size of a MP3 player, but might be a negative word when it is describing the storage capacity of that MP3 player. One approach to handling this problem is word sense induction[31], which is beyond the scope of this work.

Our model is mainly inspired by and builds upon the Joint Topic/Sentiment Model[29], which uses a Latent Dirichlet Allocation (LDA) model in topic modeling to incorporate sentiment analysis. In this model it is assumed that each word is labeled using both a topic label k and a sentiment label l, and that each word is sampled from a word distribution given both k and l. However, this inherits several basic limitations from LDA which the overall model incurs. It predefines and limits the number of topics K initially, which is impractical for large corpora. For example, for a large corpus with various service/product reviews (such as *Yelp* review data[32]), it is hard for users to regulate the number of topics in advance. Furthermore, it is also inappropriate for users to predefine this parameter, since the number of total features would be extremely large but each review document would

only occupy a few of them. The nonparametric Bayesian features of HDP can help us to alleviate this problem.

Other hybrid approaches include multi-grain topic models[33], which have some flexibility with respect to local (aspect-level) topics, but are predominantly LDA-based and tied to fixed, preset numbers of topics. Yet another approach is constrained LDA[34], which uses clustering approaches to discovery synonymy (synonym sets) of words taken as feature terms. Both of these techniques are aimed at incorporating sentiment into LDA as a monolithic topic model and thus have limited ability to evolve a topic hierarchy, account for dynamic topic drift, and incorporate models of topics in relation to authors.

2.2 Model Introduction

In our `HDPsent` model, we assume that each token in documents does not only carry latent topic information, but also represents sentiment attitude of writer. Therefore, while HDP only assigns a topic label k to each word, we add a sentiment label l to each word, along with its topic label k. We assume that for each topic component existing in each document, there is a sentiment distribution for it. Thus, each word is sampled from a word distribution specifically for the combination of its topic and sentiment label. The number of sentiment polarity values is always small and well-defined in advance. In our model, we therefore fix the number of sentiment labels in advance, which follows convention in sentiment analysis research area. We set $L = \{positive,\ negative,\ neutral\ \}$, which denote positive words, negative words, and descriptive words separately. However, this model makes no restriction on the number of sentiment labels as long as it is predefined and fixed. Sentiment labels as $L = \{strongly\ positive,\ weakly\ positive,\ neutral,\ weakly\ negative,\ strongly\ negative\ \}$ is also a desirable sentiment range segmentation. Because of the simplicity and non-hierarchical (flat) nature of this independent semantic component, we use a Latent Dirichlet Allocation (LDA) model for latent sentiment label allocation, while using a nonparametric Bayesian

HDP model.

Here in Figure 2.1 we show an example about how word distributions of different sentiment polarities vary for different topics.

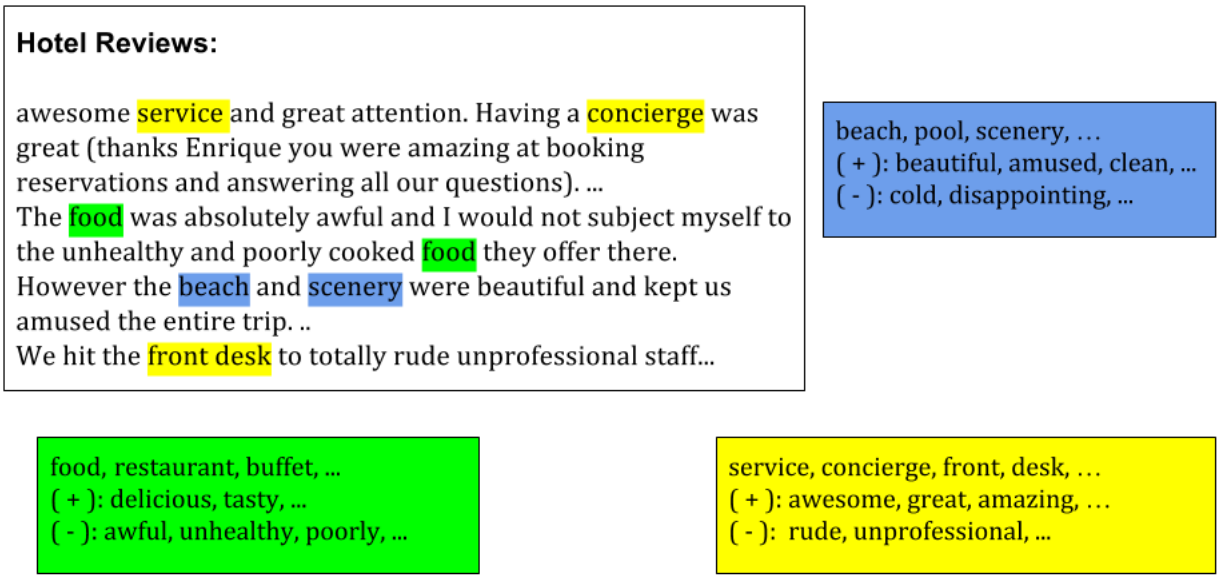

Figure 2.1: Example of topics and sentiment polarities in hotel reviews

There are several other advantages of our model. First and foremost is that it enables us to infer different word distributions for the same topic, with different sentiment polarities. Thus, from different word distributions for different sentiment polarities, we can isolate descriptive words, positive words, and negative words from the same topic.

Another advantage is that our model makes it possible to infer sentiment distributions for each topic mentioned in the document. This will allow researchers and users to develop a deeper and more detailed sentiment analysis for not only the whole document, but also each different aspect in the document. This would potentially aid them in differentiating the distinct views of an author towards the topic aspects that are reflected within a document.

2.3 Model Definition

As with the model representation that we described in Chapter 2, we define $D = \{d_1, d_2, ..., d_m\}$ to be the corpus that we want to analyze, and $x_j = \{x_{j1}, x_{j2}, ...\}$ to be the word array in document d_j. We then assume that each word x_{ji} is associated with a latent dyadic topic-sentiment combination label, denoted $< \theta, l >$, where θ is the factor corresponding to the observation variable x_{ji}, which is associated with one global topic k, and l is one latent sentiment label from one predefined sentiment label set L.

We extend the existing generative model for HDP framework to accommodate sentiment label l for word x_{ji} generation as shown in figure 2.2:

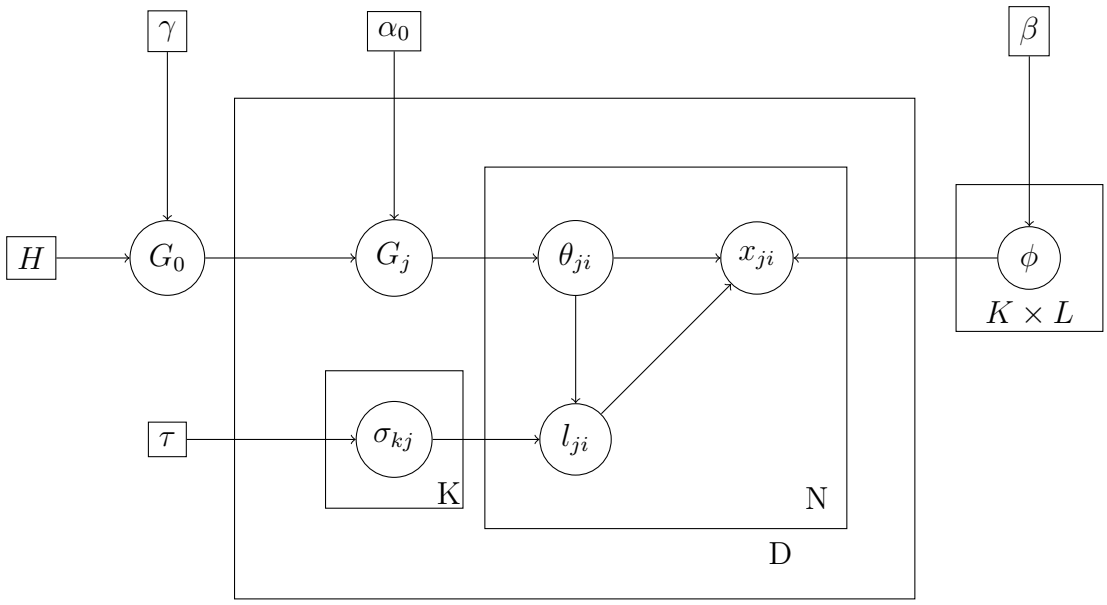

Figure 2.2: Plate model for HDP model with sentiment labels

In this model, the global probability measure G_0 represents a global topic distribution, drawn from a Dirichlet process with two generative hyperparameters: a base measure H and a concentration parameter γ. Each document j then generates its own probability measure for local topic distribution G_j from a Dirichlet process with G_0 as its base measure and α_0 as a concentration parameter:

$$
\begin{aligned}
&G_0|\gamma, H \sim DP(\gamma, H) \\
&G_j|\alpha_0, G_0 \sim DP(\alpha_0, G_0) \quad \text{for each } j,
\end{aligned}
\tag{2.1}
$$

Each observation x_{ji} in document j position i has two parameters, θ_{ji} and l_{ji}. θ_{ji} is independently and identically distributed (i.i.d.), drawn from G_j. Because each θ_{ji} is associated with an observation ψ_{jt}, which in turn has a corresponding factor k_{jt} sampled from G_0, we can denote $\theta_{ji} = \psi_{jt}$, $\psi_{jt} = \phi_k$ where $k_{jt} = k$. So that each θ_{ji} is actually associated with one global topic group k. In our model, for each distinct k emerged in document j, we assume that there is a particular sentiment distribution for k denoting the author's subjective attitude towards this topic. Therefore we generate a Dirichlet distribution σ_{jk} over the sentiment label set L, denoting this sentiment distribution for topic k in document j, with $Dir(\tau)$ as its conjugate prior. Then the sentiment label l_{ji} for observation x_{ji} is drawn from this distribution, given its topic label k. This is given by:

$$
\begin{aligned}
&\sigma_{jk} \sim Dir(\tau) && \text{for each existing } k \text{ in each } j, \\
&\theta_{ji}|G_j \sim G_j && \text{for each } j \text{ and } i, \\
&l_{ji} \sim Mult(\sigma_{jk_{\theta_{ji}}}) && \text{for each } j \text{ and } i,
\end{aligned}
\tag{2.2}
$$

We want to not only discover the differences of word distributions between same sentiment polarities in different topics, but also differentiate the word distributions for same topic with different sentiment polarities. Therefore, we assume that each distinct $< k, l >$ combination should form a distinct word distribution. Here we denote that $F(k, l)$ is a Dirichlet distribution over the whole vocabulary for specific $< k, l >$ combination, which uses $Dir(\beta)$ as its conjugate prior. Then each observation x_{ji} is drawn from this distribution with the latent $< \theta_{ji}, l_{ji} >$ generated through the generative model:

$$
\begin{aligned}
F(k,l) &\sim Dir(\beta) \\
x_{ji}|\theta_{ji}, l_{ji} &\sim F(k,l) \quad \text{for each } j \text{ and } i,
\end{aligned}
\tag{2.3}
$$

To illustrate the generative process of our `HDPsent` model with sentiment and topic generation, we can extend the generative process of *Chinese restaurant franchise* framework for traditional HDP model presented in[17] as:

1. Draw an infinite number of topics with predefined set of sentiment polarities: $\phi_{kl} \sim Dir(\beta)$ for $k = \{1, 2, 3...\}$, $L = \{1, 2, ..., l\}$.
2. Draw stick-breaking topic proportions as $\nu_k \sim Beta(1, \gamma)$ for $k = \{1, 2, 3...\}$.
3. For each document d_j:
   (a) we sample document-level topic atoms $k_{jt} \sim Multinomial(\sigma(\boldsymbol{\nu}))$ for each table $t = \{1, 2, 3...\}$.
   (b) we then sample document-level stick-breaking proportions as $\pi_{jt} \sim Beta(1, \alpha)$ for each table $t = \{1, 2, 3...\}$.
   (c) For each distinct k, we sample the sentiment distribution $\sigma_{jk} \sim Dir(\tau)$
   (d) For each token x_{ji} in document d_j at position i:
      i. We sample a latent topic label $\theta_{ji} \sim Multinomial(\sigma(\boldsymbol{\pi_j}))$.
      ii. We sample a latent sentiment label $l_{ji} \sim Multinomial(\sigma_{jk_{\theta_{ji}}})$
      iii. We sample a word $w \sim F(\theta_{ji}, l_{ji})$.

Here $\sigma(\boldsymbol{\nu})$ and $\sigma(\boldsymbol{\pi_j})$ are distributions constructed by stick-breaking algorithm with proportions of $\boldsymbol{\nu} = \{\nu_k | k = 1, 2, 3, ...\}$ and $\boldsymbol{\pi_j} = \{\pi_{jt} | t = 1, 2, 3, ...\}$ as:

$$\begin{aligned}\sigma_k(\boldsymbol{\nu}) &= \nu_k \prod_{i=1}^{k-1}(1-\nu_i) \\ \sigma_t(\boldsymbol{\pi_j}) &= \pi_{jt} \prod_{i=1}^{t-1}(1-\pi_{ji})\end{aligned} \tag{2.4}$$

2.4 Inference

In this section, we want to use the extended *Chinese restaurant franchise process* (CRFP) generative model that we described above to infer the Gibbs sampling schema for `HDPsent` model.

Here we define $\boldsymbol{\theta}^{-ji}$ and $\boldsymbol{l}^{-ji}$ are latent labels of all data items except observation x_{ji}:

$$\begin{aligned}\boldsymbol{\theta}^{-ji} &:= \{\theta_{j'i'} | j'i' \neq ji\} \\ \boldsymbol{l}^{-ji} &:= \{l_{j'i'} | j'i' \neq ji\}\end{aligned} \tag{2.5}$$

We assume in this model that each word is drawn from $F(< \theta_{ji}, l_{ji} >) = \phi_{kl}$, which is dependent on the combination of θ_{ji} and l_{ji}. We also assume that the latent sentiment label l_{ji} is drawn from a Dirichlet sentiment distribution for the specific topic parameter factor θ_{ji} in document d_j. So that we can obtain the posterior conditional of $< \theta_{ji}, l_{ji} >$ as:

$$\begin{aligned}&p(\theta_{ji}, l_{ji} | x_{ji}, \boldsymbol{\theta}^{-ji}, \boldsymbol{l}^{-ji}) \\ &\qquad \propto p(x_{ji} | \theta_{ji}, l_{ji}) \cdot p(l_{ji} | \boldsymbol{l}^{-ji}, \boldsymbol{\theta}^{-ji}, \theta_{ji}) \cdot p(\theta_{ji} | \boldsymbol{\theta}^{-ji})\end{aligned} \tag{2.6}$$

Here $p(\theta_{ji} | \boldsymbol{\theta}^{-ji})$ denotes the conditional distribution of topic factor θ_{ji} given all other data points.

We assume that the topic distribution for observations should follow HDP model; thus, to integrate out G_0 and G_j, the conditional distribution calculation for θ_{ji} in each G_j and

ψ_{jt} for global G_0 should be similar to[3] equations (24) and (25). These can in turn be represented as follows:

$$\theta_{ji}|\theta_{j1},...,\theta_{ji-1},\alpha_0,G_0 \sim \sum_{t=1}^{m_{j\cdot}} \frac{n_{jt\cdot}}{i-1+\alpha_0}\delta_{\psi_{jt}} + \frac{\alpha_0}{i-1+\alpha_0}G_0 \tag{2.7}$$

and

$$\psi_{jt}|\psi_{11},...,\psi_{jt-1},\gamma_0,H \sim \sum_{k=1}^{K} \frac{m_{\cdot k}}{m_{\cdot\cdot}+\gamma}\delta_{\phi_k} + \frac{\gamma}{m_{\cdot\cdot}+\gamma}H \tag{2.8}$$

Now, we designate $\boldsymbol{\tau_k} = \{\tau_{k1},...,\tau_{kL}\}$ to represent the probability distribution of sentiment label set L for topic k. Since the size of L is predefined, this is a simple multinomial distribution across the document; therefore, we can simply choose a Dirichlet distribution as its conjugate prior:

$$\boldsymbol{\tau_k} \sim Dir(\boldsymbol{\sigma}) \tag{2.9}$$

We assume that each topic existing in one document has its own sentiment distribution. Therefore, the sentiment label for one word in document is independent from other words in this document on different topics. This also follows our intuition in writing a document, our sentiment attitude in different topics would be quite different even in the same document.

Thus, the posterior sentiment distribution of topic k only takes into consideration the counts of word tokens with sentiment labels for the same topic:

$$p(\boldsymbol{\tau_k}|\boldsymbol{\sigma},\boldsymbol{l},\boldsymbol{k}) \sim Dir(\sigma_1 + N_{dkl_1},...,\sigma_L + N_{dkl_L}) \tag{2.10}$$

Therefore, the conditional probability of sentiment label l_{ji} for each data point x_{ji} can easily obtained by integrating $\boldsymbol{\tau_k}$ out of equation (2.9), given the topic factor $\theta_{ji} = \phi_k$, eliminating x_{ji}:

$$
\begin{aligned}
&P(l_{ji}|\boldsymbol{l}^{-ji}, \boldsymbol{k}^{-ji}, \boldsymbol{\sigma}, k_{x_{ji}} = k) \\
&= \int \tau_l Dir(\boldsymbol{\tau}|\sigma_1 + N_{dkl_1}^{-ji}, ..., \sigma_L + N_{dkl_L}^{-ji}) d\boldsymbol{\tau} \\
&= \frac{\sigma_l + N_{dkl}^{-ji}}{\sum \sigma + N_{dk\cdot}^{-ji}}
\end{aligned} \tag{2.11}
$$

Finally, with the sampled latent variable combination $< \theta_{ji}, l_{ji} >$ associated with data x_{ji}, we can obtain the topic label for table t associated with θ_{ji} by $k_{jt} = k$. The word token of x_{ji} should be drawn from word distribution denoted as $F(k, l)$.

For each word distribution for different topic-sentiment combination, we assume that it is derived from a Dirichlet distribution, with conjugate prior H. Here we can simply use ϕ_{kl} to denote this word distribution. Therefore, the conditional density of x_{ji} under $< k, l >$ can be calculated depending on all data points in the component k possessing the same sentiment label l, leaving x_{ji} out; Then we can just directly use[3] equation(30) to calculate the conditional probability of word token variable x_{ji} as:

$$
f_{kl}^{-x_{ji}}(x_{ji}) = p(x_{ji}|k, l) = \frac{\int f(x_{ji}|\phi_{kl}) \prod_{\substack{j'i' \neq ji, \\ \theta_{j'i'}=k, \\ l_{j'i'}=l}} f(x_{j'i'}|\phi_{kl}) h(\phi_{kl}) d\phi_{kl}}{\int \prod_{\substack{j'i' \neq ji, \\ \theta_{j'i'}=k, \\ l_{j'i'}=l}} f(x_{j'i'}|\phi_{kl}) h(\phi_{kl}) d\phi_{kl}} \tag{2.12}
$$

And if the data item x_{ji} being assigned to a combination with new topic as $< k^{new}, l >$, it means that it is assigned to a new ϕ with no prior data items. So the posterior probability is only dependent on conjugate prior H, which can be represented as:

$$
f_{k^{new}l}^{-x_{ji}}(x_{ji}) = p(x_{ji}) = \int f(x_{ji}|\phi_{kl}) h(\phi_{kl}) d\phi_{kl} \tag{2.13}
$$

2.5 Sampling schema

Using all these probabilities that we derived above, we can now work out the Gibbs sampling schema for posterior sampling of each data item x_{ji} using this extended *Chinese restaurant franchise process* framework (CRFP) for our `HDPsent` model.

Sampling t

We denote local index variable for each θ_{ji} as t_{ji}, and sample this index variable directly using Gibbs Sampling and the marginal represented in equation 2.7:

$$p(t_{ji} = t, l_{ji} = l | \boldsymbol{t}^{-ji}, \boldsymbol{l}^{-ji} k) \propto \begin{cases} n_{jt}^{-ji} \cdot p(l_{x_{ji}} | k, \boldsymbol{l}^{-ji}, \boldsymbol{k}^{-ji}) \cdot f_{k_{jt} l}^{-x_{ji}}(x_{ji}) & \text{if } t \text{ previously used,} \\ \alpha_0 \cdot p(x_{ji} | \boldsymbol{t}^{-ji}, \boldsymbol{l}^{-ji}, \boldsymbol{k}, t_{ji} = t^{new}) & \text{if } t \text{ is new.} \end{cases} \tag{2.14}$$

For the new table sampled, we can similarly derive the probability as:

$$p(k_{jt^{new}} = k, l_{ji} = l | \boldsymbol{t}, \boldsymbol{l}^{-ji}, \boldsymbol{k}^{-jt^{new}}) \propto \begin{cases} m_{\cdot k} \cdot p(l_{x_{ji}} | k, \boldsymbol{l}^{-ji}, \boldsymbol{k}^{-ji}) \cdot f_{kl}^{-x_{ji}}(x_{ji}) & \text{if } k \text{ previously used,} \\ \gamma \cdot p(l_{x_{ji}} | k^{new}, \boldsymbol{l}^{-ji}, \boldsymbol{k}^{-ji}) \cdot f_{k^{new} l}^{-x_{ji}}(x_{ji}) & \text{if } k \text{ is new.} \end{cases} \tag{2.15}$$

Here $f_{kl}^{-x_{ji}}(x_{ji})$ and $f_{k^{new} l}^{-x_{ji}}(x_{ji})$ are conditional densities of data x_{ji} given all other data items that can be calculated by equation 2.12 and 2.13.

Sampling k

Similarly, we denote global topic index variable for each ψ_{jt} as k_{jt}, and sample this index variable directly.

Sampling k for each table is a little different from the HDP process. This is because we only assume that the topic distribution for data items follow HDP framework. Therefore,

it is possible that the data points being assigned to same table share the same topic label k, but admit different sentiment labels l.

As a consequence, the data points in the same table may belong to different $F(k,l)$ components. Here we assume that when we sample global topic k for each table, we do not change the sentiment labels of word tokens in this table. So that the probability of one table belongs to a specific k is a combination of probabilities of independent groups of tokens from different $F(k,l)$ components for all existing l in this table. This probability can be written as:

$$f_k^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt}) = \prod_{\substack{l \in L \\ \boldsymbol{x}_{jlt}=\{x_{ji}|x_{ji}\in t, l_{ji}=l\}}} p(l|k,d) f_{kl}^{-\boldsymbol{x}_{jlt}}(\boldsymbol{x}_{jlt}) \tag{2.16}$$

where $P(l|k,d)$ can be calculated using the posterior probability of the Dirichlet sentiment distribution that we illustrated in equation 2.11.

And also the probability of one table belongs to a new topic k^{new} should also be calculated as a combination of probabilities of these tokens from separate $F(k^{new},l)$ components for all existing l in this table. Similarly, this probability can be written as:

$$f_{k^{new}}^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt}) = \prod_{\substack{l \in L \\ \boldsymbol{x}_{jlt}=\{x_{ji}|x_{ji}\in t, l_{ji}=l\}}} p(l|d) f_{k^{new}l}^{-\boldsymbol{x}_{jlt}}(\boldsymbol{x}_{jlt}) \tag{2.17}$$

Here $p(l|d)$ represents the overall sentiment distribution across the document.

Since we have figured out the calculation of $f_k^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt})$ and $f_{k^{new}}^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt})$, the probability of table t is assigned to each k follows the traditional sampling schema according to 2.8 as:

$$p(k_{jt}=k|\boldsymbol{t},\boldsymbol{l}^{-ji},\boldsymbol{k}^{-jt}) \propto \begin{cases} m_{\cdot k} \cdot f_k^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt}) & \text{if } k \text{ previously used,} \\ \gamma \cdot f_{k^{new}}^{-\boldsymbol{x}_{jt}}(\boldsymbol{x}_{jt}) & \text{if } k \text{ is new.} \end{cases} \tag{2.18}$$

Thus the pseudo-code of our sampling inference algorithm is depicted in Algorithm 1:

Algorithm 1 `HDPsent` algorithm

procedure GIBBS–HDPSENT
  for each document $d_j \in D$ **do**
    for each word token $x_{ji} \in d_j$ **do**
      Incrementally sample $< \theta_{ji}, l_{ji} >$ for x_{ji}
      Change l_{ji} to its predefined inital value l_w given word $w = x_{ji}$
      Increase statistical counts for $< \theta_{ji}, l_w >$
    end for
  end for
  while not converged **do**
    for each document $d_j \in D$ **do**
      for each word token $x_{ji} \in d_j$ **do**
        Decrease statistical counts for old $< \theta_{ji}, l_{ji} >$
        Sample $< \theta, l >$ for x_{ji}
        Increase statistical counts for new $< \theta_{ji}, l_{ji} >$
      end for
      for each table $\psi_{jt} \in d_j$ **do**
        Decrease statistical counts for old k_{jt}
        Sample k for ψ_{jt}
        Increase statistical counts for new k_{jt}
      end for
    end for
  end while
end procedure

2.6 Model Prior

Traditional HDP model rarely introduce asymmetric priors for both documents and topics. However, our model imports aspect-level sentiment layer into traditional HDP model, which requires certain degree of structured asymmetric priors for sentiment modeling.

2.6.1 Sentiment Prior

In our model, the sentiment distribution is dependent only on the data in same topic. This does not cause problems in LDA models, but **does** cause problems in HDP models, because HDP model spawns new topics at certain probabilities:

$$p(\boldsymbol{\tau}|\boldsymbol{\sigma}, \boldsymbol{l}^{-ji}, \boldsymbol{k}^{-ji}, k^{new}) \sim Dir(\sigma_1 + 0, ..., \sigma_L + 0) = Dir(\boldsymbol{\sigma}) \tag{2.19}$$

Without any prior knowledge for sentiment labels for tokens assigned to new topic (or newly emerged topic with only few tokens assigned to within this document), the sentiment label for this token, is solely (or largely) dependent on its conjugate prior $Dir(\boldsymbol{\sigma})$. This is still acceptable if we assume that sentiment distributions of different topics are totally independent from each other in the same document. However, most of the time, we intend to have similar sentiment attitude across most topics we write about in the same document. So that we can set up document-specific priors for sentiment distribution, and topic can have its own sentiment distributions drawn from this prior.

Here we introduce different $\boldsymbol{\sigma}$ for different documents as its own conjugate prior. Using the LDA prior schema from[35] for sentiment distributions, we use σ' as a concentration parameter for σ, and obtain:

$$\sigma_{dl} = \sum_l \sigma_l \cdot \frac{N_{d \cdot l} + \sigma'_l}{N_{d \cdot \cdot} + \sum_l \sigma'_l} \tag{2.20}$$

This allows equation (2.11) to be rewritten as:

$$P(l_{x_{ji}}|\boldsymbol{l}^{-ji}, \boldsymbol{k}^{-ji}, \boldsymbol{\sigma}, k_{x_{ji}} = k) = \begin{cases} \frac{\sigma_{dl}+N_{dkl}^{-ji}}{\sum_l \sigma_{dl}+N_{dk\cdot}^{-ji}} & \text{if } k \text{ previously used,} \\ \frac{\sigma_{dl}}{\sum_l \sigma_{dl}} & \text{if } k \text{ is new.} \end{cases} \tag{2.21}$$

2.6.2 Word Prior

Since our word distribution $F(k, l)$ has only the global conjugate prior $Dir(\beta)$, as shown in figure 2.2, any new $< k, l >$ combination has the same symmetric prior. In pure topic models, this is acceptable since we do not have and may not set up any prior knowledge for word distribution in the new topic at all. However, on the one hand, we already have strong prior bias for sentiment polarity of many words in English vocabulary, according to their semantic meanings. On the other hand, the sentiment polarity of same word across different topics although is not fixed, but has less tendency to be changed. For example, even though we do not have a prior preference for a word such as "good" in a new topic k^{new}, we shall have some prior preference for "good" in a new combination $< k^{new}, positive >$, versus a new combination $< k^{new}, negative >$.

This prior also helps us to adjust the probability for sampling word for sentiment labels. Without this prior, the sentiment assignment for words in the same topic can easily be reversed from their usual meaning, with positive words assigned to the predefined negative category, and negative ones to the positive category.

Using the same prior schema, and defining β' to be the concentration parameter for β, we directly obtain:

$$\beta_{lw} = \sum_w \beta_w \cdot \frac{N_{\cdot lw} + \beta'_w}{N_{\cdot l\cdot} + \sum_w \beta'_w} \tag{2.22}$$

Thus, parameters in equation (2.12) can easily be integrated out, resulting in:

$$f_{kl}^{-x_{ji}}(x_{ji}) = \begin{cases} \frac{\beta_{lw}+N_{klw}^{-x_{ji}}}{\sum_w \beta_{lw}+N_{kl\cdot}^{-x_{ji}}} & \text{if } k \text{ previously used,} \\ \frac{\beta_{lw}}{\sum_w \beta_{lw}} = \frac{N_{\cdot lw}+\beta'_w}{N_{\cdot l\cdot}+\sum_w \beta'_{lw}} & \text{if } k \text{ is new.} \end{cases} \tag{2.23}$$

Chapter 3

Author-Topic Model: `HDPauthor`

While the characterization of topic modeling as estimating the topic distribution of documents was developed many years and has been used since, there is also a growing need for topic interest learning of **authors**. Moreover, the contribution of different authors to a single document is also a learning problem that needs to be studied. Our objective as discussed in this chapter to develop a generative mixture model extending current topic models, which is capable of simultaneously learning and identifying the topic interests of authors, topic distribution across documents, and author contributions to documents.

Currently there are already many significant works on Bayesian methods for author mixture models[36][37] *without* topic modeling. There is also some work in the literature on LDA-based author-topic learning frameworks[38][39]. However, because these models are variation and extension based on LDA, using Dirichlet multinomial mixture models, all of them admit predefined limits on the number of topics.

In real-world applications, the number global topics across whole corpora may not be fixed or boundable. However, each author usually only works on and is good at a small set of topics, and each document written by a group of authors is also usually written on a small set of topics. Therefore, the nonparametric Bayesian feature of hierarchical Dirichlet process for topic modeling can help us to solve the problem, and infer a better learning

algorithm compared to existing LDA-based author-topic learning models.

In this chapter, we present a statistical generative mixture model called `HDPauthor`, for scientific articles with authors; this model extends our existing HDP model to incorporate authorship information. It uses nonparametric hierarchical Bayesian modeling to learn the topic interests of each author across the documents in which that author participates. It treats the topic distribution for local multi-author document as a finite mixture of distributions of the authors. It benefits from traditional HDP model features that the global number of topics is unbounded. Each author from text collection also shares unbounded number of topics from global topic pool. This model also enables researchers and users to infer contribution proportions of different authors for one document.

3.1 Related Work

There are many works that have already incorporated co-authorship into topic modeling. One significant model is the Author-Topic model[38 40]. This model extends the LDA model to include authorship information. It makes it possible to simultaneously learn both the relevance of different global topics in document, and the interests of topics for authors. It associates not only a mixture of topics with each document but also a mixture of topics with each author, which makes it able to sample words from probability distributions generated usng a combination of these two factors. In similar fashion to the LDA model, the total number of topics for the whole corpus must be predetermined in advance, with no flexibility over the number of topics generated. This model also lacks the ability to share only a small subset of topics across different documents, as well as across different authors. Therefore, it learns distribution of each topic in this large group of topics for each document and each author.

Models proposed by Dai[41 42] are based on nonparametric HDP model for topic-author problem. This approach combines author identities with associated topics as a group. This

group defines a Dirichlet process (DP) over author entities and topics, which in turn is then drawn from a global author and topic DP. This model is mainly geared towards disambiguation of author entities. However, this model combines authors and topics in the same DP, which fails to decouple topics from authors. Therefore, it lacks the ability to share the same topics between different authors, and also makes it difficult to infer author contributions to these documents.

3.2 Model Introduction

Our `HDPauthor` model is a nonparamatric hierarchical Bayesian model for author-topic generation. This model assumes that topic distribution of each document is a finite mixture of distribution components of the authors of this document. We can then infer that each token in the document is written by one and only one of the authors in the author list of this document, associated with the topic distribution of this author. This assumption enables us to set up latent author label for each word token along with its topic label. This latent author label helps us to infer both topic interests of authors and mixture parameters in documents for each author.

By using an HDP framework, we also assume that each author is associated with a topic distribution which is drawn based on the same underlying base measure as global topic distribution in whole corpora, with different variability. The global topic atoms are shared by all authors, but each author only occupies a small subset of these global topic components, with different stick-breaking weights. This local probability measure of each author represents the topic interests of this author. Different authors share different topic interests.

The topic distribution of each author is learned using the mixture generative model of all documents that the author participates in. The topic distribution of each document is not drawn from the global topic distribution directly, but represented by this mixture model

of all its authors indirectly. Since we already assume that each token is written by one and only one of the authors with the particular topic distribution of this author, then the latent topic labels combined with latent author labels helps us to infer the topic distribution of documents. Therefore, each document is represented by a union of all topics contributed by each of its authors.

Here in Figure 3.1 we illustrate an example of document produced through the cooperation of several authors. The content of the document is the abstract of one paper[43] on machine learning for gene expression data. Author *Yoseph Barash* mainly works on biology and bioinformatics, who contributes more on biology related topics, while author *Nir Friedman* is an expert in Bayesian inference and machine learning, which results in his having higher probability of machine learning-related topics.

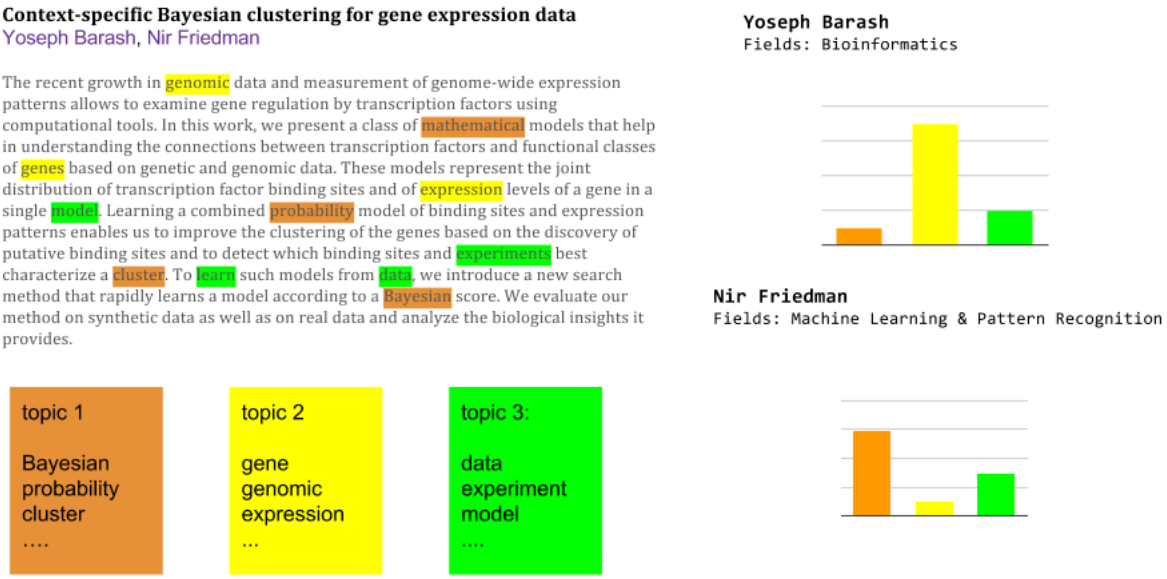

Figure 3.1: Example of topic modeling with author cooperation

3.3 Model Definition

The document representation in our model also follows our definition stated in Chapter 2. We assume $D = \{d_1, d_2, ...\}$ is a collection of scientific articles, composed of a series of words from vocabulary V as $x_j = \{x_{j1}, x_{j2}, ...\}$. Furthermore, in our `HDPauthor` model, we have extra co-authorship information. We assume that each document has a set of authors $a_j = \{a_{j1}, a_{j2}, ...\}$ who cooperated in writing this document d_j.

Previously we have assumed that each token in a document is written by one of the authors for this whole document. Therefore, here we associate one latent author label q from the author set $\boldsymbol{a}_j$ for each token in document d_j along with original latent topic label k.

This latent author label a not only helps us to directly calculate the contribution of each author for the document, but also enables the aggregation of topic distribution for each author across the whole corpus.

We generate G_0 as the corpus-level set of topics as a Dirichlet Process with H as base measure and γ as its concentration parameter. A topic component is denoted ϕ_g. Each author a that exists in the entire corpus corresponds to a Dirichlet Process G_a that shares the same global base distribution of topics G_0, with concentration parameter η. As with the HDP model, the author-level G_a only shares a small subset of corpus-level topics.

$$\begin{aligned} G_0|\gamma, H &\sim DP(\gamma, H) \\ G_a|\eta, G_0 &\sim DP(\eta, G_0) \end{aligned} \tag{3.1}$$

Unlike in the traditional HDP model, we do not draw a Dirichlet process G_j of each document d_j from the global G_0 as $G_j \sim DP(\alpha_0, G_0)$. Instead, we set up a mixture of components from probability measures of all authors of this document. We then denote the mixing proportion vector as $\boldsymbol{\pi_j} =< \pi_{j1}, ..., \pi_{j|\boldsymbol{a}_j|} >$. Therefore, all of its elements must

be positive and sum to one. Since each document is written by a fixed group of authors, we can here simply assume that $\boldsymbol{\pi_j}$ is drawn from a symmetric Dirichlet distribution with concentration parameter ϵ.

$$\boldsymbol{\pi_j} \sim Dir(\epsilon) \tag{3.2}$$

For a mixing proportion vector π_j, there are two ways of drawing G_j from a Dirichlet process for the mixture of the probability measures of all its authors, designated $\{G_a | a \in \boldsymbol{a}_j\}$. The first method is to combine the probability measures G_a of authors as a new base measure first, then draw a DP with this base measure combination for document d_j; this DP can be formulated as follows:

$$G_j \sim DP(\alpha_0, \sum_{a \in a_j} \pi_{ja} \cdot G_a) \tag{3.3}$$

Another method is to first draw separate DPs from each of the authors of the document d_j with the author's own probability measure G_a as the base measure, and then calculate the probability measure of d_j as a mixture of these DPs. The mathematical formula we derive for this method is:

$$G_j \sim \sum_{a \in \boldsymbol{a}_j} \pi_{ja} \cdot DP(\alpha_0, G_a) \tag{3.4}$$

Each observation x_{ji} in document d_j is associated with a combination of two parameters $< a_{ji}, \theta_{ji} >$ sampled from this mixture G_j. In this combination, a_{ji} is author label $a \in \boldsymbol{a}_j$, which indicates the "class" label of this author mixture model. θ_{ji} is the parameter specifying the one of the author's topic component for x_{ji}, which is sampled from the probability measure G_a of the author a selected. Therefore, this θ_{ji} is associated with table t_{ji}, which is an instance of mixture component ω_{ak} from author $a = a_{ji}$; ω_{ak} is then associated with one global topic component g. Given global topic component g, the token x_{ji} arises from a Dirichlet distribution over the whole vocabulary based on this topic label g, which is the

component factor assigned to k_{jt} in its associated parameter θ_{ji}, denoted as $F(g)$:

$$\begin{aligned} < a_{ji}, \theta_{ji} > | G_j &\sim G_j \\ x_{ji} | \theta_{ji} &\sim F(\theta_{ji}) \end{aligned} \tag{3.5}$$

As we explained above, the factor θ_{ji} for each observation x_{ji} is associated with global topic mixture component g. Here we can simply use ϕ_g to denote this distribution. Therefore, the conditional density of each observation x_{ji} under this particular ϕ_g given all other observations can be derived similarly to[3] equation(30):

$$f_g^{-xji}(x_{ji}) = \frac{\int f(x_{ji}|\phi_g) \prod_{\substack{j'i' \neq ji, \\ \theta_{j'i'}=g}} f(x_{j'i'}|\phi_g) h(\phi_g) d\phi_g}{\int \prod_{\substack{j'i' \neq ji, \\ \theta_{j'i'}=g}} f(x_{j'i'}|\phi_g) h(\phi_g) d\phi_g} \tag{3.6}$$

And the conditional probability of data item x_{ji} being assigned to a new topic g^{new} is also only dependent on the conjugate prior H. This can be represented as:

$$f_{g^{new}}^{-xji}(x_{ji}) = \int f(x_{ji}|\phi_g) h(\phi_g) d\phi_g \tag{3.7}$$

Here in figure 3.2 we illustrate the graphical plate model for our `HDPauthor` model with one more layer of author probability measures injected into the original HDP model:

To present the generative process of our `HDPauthor` model within an author layer, we can extend the generative process of *Chinese restaurant franchise* framework for the traditional HDP model presented in[17] as:

1. Draw an infinite number of topics $\phi_g \sim Dir(\beta)$ for $g = \{1, 2, 3...\}$.
2. Draw stick-breaking topic proportions as $\nu_g \sim Beta(1, \gamma)$ for $g = \{1, 2, 3...\}$.
3. For each author a:

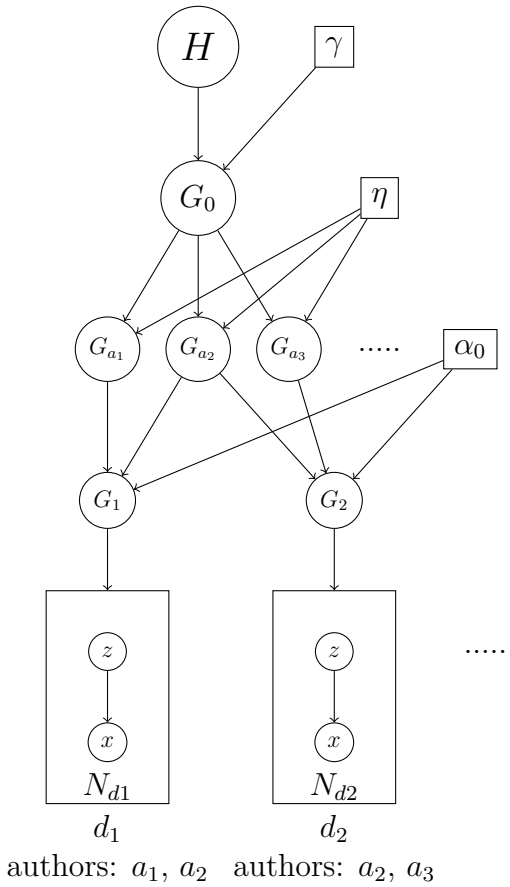

Figure 3.2: Plate Model for HDP model with authors

(a) we sample author-level topic atoms $g_{ak} \sim Multinomial(\sigma(\boldsymbol{\nu}))$ for each author component $k_a = \{1, 2, 3...\}$.

(b) we then sample author-level stick-breaking proportions as $\mu_{ak} \sim Beta(1, \eta)$ for each author component $k_a = \{1, 2, 3...\}$.

4. For each document d_j:

(a) We sample the author mixing proportions for authors of this document as $\boldsymbol{\pi_j} \sim Dir(\epsilon)$

(b) we sample document-level author component atoms k_{jt} from the author mixture model for each table $t = \{1, 2, 3...\}$.

(c) We then sample document-level stick-breaking proportions as $\delta_{jt} \sim Beta(1, \alpha)$ for each table $t = \{1, 2, 3...\}$.

(d) For each token x_{ji} in document d_j at position i:

i. We sample a latent topic label $\theta_{ji} \sim Multinomial(\sigma(\boldsymbol{\delta_j}))$.

ii. We sample a word $w \sim Multinomial(\phi_{\theta_{ji}})$.

Here $\sigma(\boldsymbol{\nu})$ and $\sigma(\boldsymbol{\delta_j})$ are distributions constructed by stick-breaking algorithm with proportions of $\boldsymbol{\nu} = \{\nu_k | k = 1, 2, 3, ...\}$ and $\boldsymbol{\delta_j} = \{\delta_{jt} | t = 1, 2, 3, ...\}$ as:

$$\begin{aligned} \sigma_k(\boldsymbol{\nu}) &= \nu_k \prod_{i=1}^{k-1}(1 - \nu_i) \\ \sigma_t(\boldsymbol{\delta_j}) &= \delta_{jt} \prod_{i=1}^{t-1}(1 - \delta_{ji}) \end{aligned} \tag{3.8}$$

3.4 Inference

The primary inferential mechanism for our model is based on a Gibbs sampling-based implementation of the Chinese restaurant franchise process (CRFP) model. We should extend this representation framework to inject an author layer, and calculate all posterior distributions for latent variables.

Inference for model (3.3)

Here we compute the marginal of G_j under this author mixture Dirichlet process model with G_0 and G_a are integrated out. We want to compute the conditional distribution of θ_{ji} given all other variables; we thus extend[3] equation (24) to fit our model for model 3.3, to obtain:

$$\theta_{ji} | \theta_{j1}, ..., \theta_{ji-1}, \alpha_0, G_j, G_{a0}, G_{a1}, ... \sim \sum_{t=1}^{m_{j\cdot}} \frac{n_{jt}}{n_{j\cdot}^{-ji} + \alpha_0} \delta_{\psi_{jt}} + \frac{\alpha_0}{n_{j\cdot}^{-ji} + \alpha_0} \sum_{a \in \boldsymbol{a}_j} \pi_{ja} \cdot G_a \tag{3.9}$$

Here, ψ_{jt} represents the table-specific indicator that indicates the component choice k_{jt} from author a_{jt}'s probability measure. A drawing from this mixture model can be divided into two parts. If the former summation is chosen, then x_{ji} is assigned to an existing ψ_{jt}, and we can denote $\theta_{ji} = \psi_{jt}$. If the latter summation is chosen, we have to create a new document-specific table t^{new}, and assign it to one of the authors according to mixing proportion vector of authors for document d_j, where each $\pi_{ja} \in \boldsymbol{\pi_j}$ represents the probability that table t^{new} belongs to author a. Then we can draw one new $\psi_{jt^{new}}$ from the probability measure of author a represented as G_a.

G_a for each author a in the corpus appears in all documents in which this author participates. It should be integrated out through all ψ_{jt} that $a_{jt} = a$. We use m_{ak} to indicate the total number of tables t such that $k_{jt} = k$ and $a_{jt} = a$. To integrate out each G_a, we can get:

$$\psi_{jt}|\psi_{11}, ..., \psi_{jt-1}, \eta, G_0 \sim \sum_{k=1}^{l_{a\cdot\cdot}} \frac{m_{ak}}{m_{a\cdot\cdot} + \eta}\delta_{\omega_{ak}} + \frac{\eta}{m_{a\cdot\cdot} + \eta}G_0 \tag{3.10}$$

This mixture is also divided into two parts. If we draw sample ψ_{jt} from the former part, then we assign it to an existing component k from author a, we can denote it as $\psi_{jt} = \omega_{ak}$. If the latter part is chosen, we will create one new component k^{new} for author a. and we draw this new $\omega_{ak^{new}}$ from global topic probability measure G_0.

Finally we can integrate out this global probability measure G_0 by all cluster components ω_{ak} from all existing authors in whole corpora. Here we use l_g to indicate the total number of ω_{ak} such that $g_{ak} = g$. The integral can then be represented similarly to[3] equation (25):

$$\omega_{ak}|\omega_{11}, ..., \omega_{ak-1}, \gamma, H \sim \sum_{g=1}^{G} \frac{l_{g\cdot}}{l_{\cdot\cdot} + \gamma}\delta_{\phi_g} + \frac{\gamma}{l_{\cdot\cdot} + \gamma}H \tag{3.11}$$

Similarly, if the former is chosen, we assign the existing topic component ϕ_g to ω_{ak}; if the latter is chosen, we create a new topic g^{new} sampled from base measure H.

Inference for model (3.4)

For mixing model 3.4, each document's probability measure is divided into $|\boldsymbol{a}_j|$ independent components, where the probability of each component $a \in \boldsymbol{a}_j$ to be chosen is determined by $\pi_{ja} \in \boldsymbol{\pi}_j$ from this document-specific mixing proportion vector $\boldsymbol{\pi}_j$. Once a specific author a is chosen, the probability distribution of θ_{ji} follows the Dirichlet process $DP(\alpha_0, G_a)$ where $a \in \boldsymbol{a}_j$, using the probability measure of author a denoted as G_a to be its base measure. Therefore, with G_0 and G_a integrated out, we can obtain the distribution of θ_{ji} given all other variables, as:

$$\theta_{ji}|\theta_{j1}, ..., \theta_{ji-1}, \alpha_0, G_j, G_{a1}, G_{a2}, ... \sim \sum_{a \in \boldsymbol{a}_j} \pi_{ja} \cdot \Big(\sum_{t=1}^{m_{ja\cdot}} \frac{n_{jt}}{n_{ja\cdot}^{-ji} + \alpha_0} \delta_{\psi_{jt}} + \frac{\alpha_0}{n_{ja\cdot}^{-ji} + \alpha_0} G_a \Big) \quad (3.12)$$

These two models differ only in the construction of the mixture of authors with each author's own probability measure, drawn from shared global infinite topic mixture model in one document. The constructions of each author's probability measure and global topic measure are same. Therefore, the posterior conditional calculation of ψ_{jt} and ω_{ak} for model (3.4) are same as presented in equation 3.10 and 3.10.

3.5 Sampling schema

According to this series of marginals that we integrated out above, we can now go on to calculate the posterior sampling schema for our Gibbs sampling inference process.

Since we have two mixture models for combining author topic components into one document, as stated in mixture model (3.3) and model (3.4), the integrals that we inferred in equation 3.9 and equation 3.12 will result in two different ways of calculating the posterior conditional distributions of a_{ji} and θ_{ji} accordingly.

3.5.1 Sampling schema for author mixture model (3.3)

Sampling t

Using the integral 3.9 inferred for author mixture model (3.3), the probability that t_{ji} takes a particular existing t should be proportional to the number of tokens in this t as n_{jt}^{-ji}, regardless of the author label a_{jt} for this table t, and the probability that this x_{ji} will be assigned to a new value t is proportional to α_0.

$$p(t_{ji} = t|\boldsymbol{t}^{-ji}, \boldsymbol{a}, \boldsymbol{k}, \boldsymbol{g}) \propto \begin{cases} \frac{n_{jt}^{-ji}}{n_{j\cdot}^{-ji}+\alpha_0} \cdot f_{g_{ak_{jt}}}^{-x_{ji}}(x_{ji}) & \text{if } t \text{ previously used,} \\ \frac{\alpha_0}{n_{j\cdot}^{-ji}+\alpha_0} \cdot p(x_{ji}|t_{ji} = t^{new}, \boldsymbol{a}, \boldsymbol{k}, \boldsymbol{g}) & \text{if } t \text{ is new.} \end{cases} \tag{3.13}$$

If the sampled t_{ji} is new t, we should then sample the author label a_{jt} for this table t from the Dirichlet-based finite author mixture model, and then sample k from the probability measure of author a, given $a_{jt} = a$:

$$\begin{aligned} &p(k_{jt^{new}} = k, a_{jt^{new}} = a|\boldsymbol{t}^{-ji}, \boldsymbol{a}^{-jt^{new}}, \boldsymbol{k}^{-jt^{new}}, \boldsymbol{g}) \\ &= p(a_{jt^{new}} = a|\boldsymbol{a}^{-jt^{new}}) \cdot p(k_{jt^{new}} = k|a_{jt^{new}} = a, \boldsymbol{t}^{-ji}, \boldsymbol{k}^{-jt^{new}}, \boldsymbol{g}) \end{aligned} \tag{3.14}$$

We already denote the mixing proportion vector of authors for document d_j by $\boldsymbol{\pi}_j$. We also assume that this vector follows a Dirichlet distribution with ϵ as its conjugate prior. However, since in this model, we use table t as the base granularity for author-mixing representation, we should use the number of tables m rather than the number of tokens n for this finite author mixing proportion calculation. Here we use m_{ja} to represent the number of tables assigned to author a in document d_j. Thus, we can use the standard Dirichlet integral to calculate posterior probability of author label a_{jt} for this document-specific table t given all other observations, as:

$$p(a_{jt} = a|\boldsymbol{a}^{-jt}, \epsilon) = \frac{m_{ja}^{-jt} + \epsilon}{m_{j\cdot}^{-jt} + |a_j| \cdot \epsilon} \tag{3.15}$$

With the author label $a_{jt^{new}} = a$ selected, we already decide that this table t^{new} is assigned to (and assumed to be written by) author a. This is exactly the extra layer we added to traditional HDP topic models. We should obtain the topic component index of table t_{jt}, not from the global topic distribution, but from the topic distribution of author a. Therefore, we now should obtain the value of $k_{jt^{new}} = k$ be sampled from the probability measure of author G_a as:

$$\begin{aligned} &p(k_{jt^{new}} = k|a_{jt^{new}} = a, \boldsymbol{t}^{-ji}, \boldsymbol{k}^{-jt^{new}}, \boldsymbol{g}) \\ &\propto \begin{cases} m_{ak\cdot}^{-ji} \cdot f_{g_{ak}}^{-x_{ji}}(x_{ji}) & \text{if } k \text{ previously used for } a, \\ \eta \cdot p(x_{ji}|a_{jt^{new}} = a, k_{jt} = k^{new}, \boldsymbol{g}) & \text{if } k = k^{new} \text{ for author } a. \end{cases} \end{aligned} \tag{3.16}$$

Here we use k_{jt} to denote the local k component index for author a_{jt} in doc d_j, table t. If the sampled k_{jt} is new to author a, this means that it creates a new component k for this author a, and this new component k should be then sampled from higher global mixture component g. Similarly to[3] equation (33), we can infer that:

$$p(g_{ak^{new}} = g|\boldsymbol{t}, \boldsymbol{g}^{-ak^{new}}) \propto \begin{cases} l_{g\cdot} \cdot f_g^{-x_{ji}}(x_{ji}) & \text{if } g \text{ previously used,} \\ \gamma \cdot f_{g^{new}}^{-x_{ji}}(x_{ji}) & \text{if } g = g^{new} \text{ is new.} \end{cases} \tag{3.17}$$

Sampling a, k

For author mixture model (3.3), sampling k for each table t is a little different from traditional HDP sampling schema. Specifically, in this model, we add one more author layer above local document-specific topic distribution, so that each t is associated not with one global topic component g directly, but with an author label a and one of the author's

own topic component k. We have to sample t from the mixture model including all cluster components k from all authors $a \in \boldsymbol{a}_j$, with the author mixing proportion vector $\boldsymbol{\pi}_j$.

$$p(a_{jt} = a, k_{jt} = k | \boldsymbol{t}^{-ji}, \boldsymbol{a}^{-jt}, \boldsymbol{k}^{-jt} \boldsymbol{g}) \propto \begin{cases} p(a_{jt} = a | \boldsymbol{a}^{-jt}, \epsilon) \cdot \frac{m_{ak\cdot}^{-jt}}{m_{a\cdot\cdot}^{-jt} + \eta} \cdot f_{g_{ak}}^{-x_{jt}}(x_{jt}) \\ \text{if } k \text{ previously used for } a, \\ p(a_{jt} = a | \boldsymbol{a}^{-jt}, \epsilon) \cdot \frac{\eta}{m_{a\cdot\cdot}^{-jt} + \eta} \cdot p(x_{jt} | a_{jt} = a, k_{jt} = k^{new}, \boldsymbol{g}) \\ \text{if } k \text{ is new for } a. \end{cases} \tag{3.18}$$

Similarly, when $k_{jt} = k^{new}$, we have to obtain a new sample from the global topic probability measure:

$$p(g_{ak^{new}} = g | \boldsymbol{t}, \boldsymbol{g}^{-ak^{new}}) \propto \begin{cases} l_{g\cdot} \cdot f_g^{-x_{jt}}(x_{jt}) & \text{if } g \text{ previously used,} \\ \gamma \cdot f_{g^{new}}^{-x_{jt}}(x_{jt}) & \text{if } g = g^{new} \text{ is new.} \end{cases} \tag{3.19}$$

Sampling g

Finally, we present the sampling schema for global topic distribution $\boldsymbol{g}$, which is sampled from all components k of all existing authors a in corpora. However, each component k for author a contains all tables assigned to author a with its own component index k from documents across the whole corpora that this author participates in. Changing g_{ak} involves the topic membership of a set of word tokens $\boldsymbol{x}_{ak}$ that are assigned to all these tables. We then can denote this set of variables as $\boldsymbol{x}_{ak} = \{x_{ji} | t_{ji} = t, a_{jt} = a, k_{jt} = k, a \in \boldsymbol{a}_j\}$. Then the sampling schema can be presented as:

$$p(g_{ak} = g | \boldsymbol{t}, \boldsymbol{g}^{-ak}) \propto \begin{cases} l_{g\cdot} \cdot f_g^{-\boldsymbol{x}_{ak}}(\boldsymbol{x}_{ak}) & \text{if } g \text{ previously used,} \\ \gamma \cdot f_{g^{new}}^{-\boldsymbol{x}_{ak}}(\boldsymbol{x}_{ak}) & \text{if } g = g^{new} \text{ is new.} \end{cases} \tag{3.20}$$

3.5.2 Sampling schema for author mixture model (3.4)

Sampling t

Using the integral 3.12 inferred for author mixture model (3.4), we discover that the probability that x_{ji} is assigned to a particular author $a \in \boldsymbol{a_j}$ should be calculated first, which is proportional to the document-specific mixing proportion vector $\boldsymbol{\pi_j}$. Thus, the conditional posterior probability that x_{ji} is assigned to a particular table t_{ji} is calculated according to the conditional prior distribution for t_{ji} with all data items in document d_j only associated with author $a = a_{ji}$.

$$\begin{aligned} & p(t_{ji} = t|\boldsymbol{t}^{-ji}, \boldsymbol{a}, \boldsymbol{k}, \boldsymbol{g}) \\ & \propto p(a_{ji} = a|\boldsymbol{a}^{-ji}) \cdot p(t_{ji} = t|a_{ji} = a, \boldsymbol{t}^{-ji}, \boldsymbol{a}, \boldsymbol{k}, \boldsymbol{g}) \end{aligned} \tag{3.21}$$

In this model, since the base granularity for author choice is word token in author-mixing representation, we should use the number of tokens n in the conditional calculation of $\boldsymbol{\pi}_j$. Here we use n_{ja} for indicating the number of tokens assigned to author a, we can get:

$$p(a_{ji} = a|\boldsymbol{a}^{-ji}, \epsilon) = \frac{n_{ja}^{-ji} + \epsilon}{n_{j\cdot}^{-ji} + |a_j| \cdot \epsilon} \tag{3.22}$$

Given author label $a_{ji} = a$ selected, the sample value t_{ji} is calculated by integrating out all possible t_{ji} given all data items with latent author label a. Therefore, the probability that t_{ji} takes an existing t from author a in this document d_j should be proportional to the number of tokens n_{jt}^{-ji} in this t, and the probability that this x_{ji} will be assigned to a new value t is proportional to α_0, following the probability measure of this particular author G_a. We thus get:

$$p(t_{ji} = t|a_{ji} = a, \boldsymbol{t}^{-ji}, \boldsymbol{a}, \boldsymbol{k}, \boldsymbol{g}) \propto \begin{cases} \dfrac{n_{jt}^{-ji}}{n_{ja\cdot}^{-ji} + \alpha_0} \cdot f_{g_{ak_{jt}}}^{-x_{ji}}(x_{ji}) \\ \qquad\qquad \text{if } t \text{ previously used,} \\ \dfrac{\alpha_0}{n_{ja\cdot}^{-ji} + \alpha_0} \cdot p(x_{ji}|t_{ji} = t^{new}, a_{jt^{new}} = a, \boldsymbol{a}, \boldsymbol{k}, \boldsymbol{g}) \\ \qquad\qquad \text{if } t \text{ is new.} \end{cases} \tag{3.23}$$

For simplicity, in this mixture model, we assume $\epsilon = \alpha_0$. Thus numerator $(n_{ja} + \alpha_0)$ in equation 3.23 and denominator $(n_{ja} + \epsilon)$ in equation 3.22 can be canceled. Therefore for all authors in document as $\{a|a \in \boldsymbol{a}_j\}$, we can rewrite equation 3.21 as:

$$p(t_{ji} = t|\boldsymbol{t}^{-ji}, \boldsymbol{a}, \boldsymbol{k}, \boldsymbol{g}) \propto \begin{cases} \dfrac{n_{jt}^{-ji}}{n_{j\cdot}^{-ji} + |a_j| \cdot \epsilon} \cdot f_{g_{ak_{jt}}}^{-x_{ji}}(x_{ji}) \\ \qquad\qquad \text{if } t \text{ previously used,} \\ \dfrac{\alpha_0}{n_{j\cdot}^{-ji} + |a_j| \cdot \epsilon} \cdot p(x_{ji}|t_{ji} = t^{new}, a_{jt^{new}} = a, \boldsymbol{a}, \boldsymbol{k}, \boldsymbol{g}) \\ \qquad\qquad \text{if } t \text{ is new.} \end{cases} \tag{3.24}$$

According to the integrals calculated, the sampling schema for t in model (3.4) differs from model (3.3) only when we sample t_{ji} and a_{ji} for observation x_{ji}. The following sampling schema referring to G_a and G_0 remains the same. Therefore, if a new table $t_{ji} = t^{new}$ is sampled, and the author label $a_{jt} = a$ for this table is also sampled, the calculation of $p(k_{jt^{new}} = k|a_{jt^{new}} = a, \boldsymbol{t}^{-ji}, \boldsymbol{k}^{-jt^{new}}, \boldsymbol{g})$ and $p(g_{ak^{new}} = g|\boldsymbol{t}, \boldsymbol{g}^{-ak^{new}})$ for model (3.4) is exactly as same as equation 3.16 and 3.17.

Sampling a, k

For an author mixture model (3.4), we noticed that if we set $\alpha_0 = \epsilon$, then the probability that a new table t^{new} drawn from the author mixture model is proportional to $\alpha_0 \cdot p(x_{ji}|t_{ji} =$

$t^{new}, a_{jt^{new}} = a, \boldsymbol{a}, \boldsymbol{k}, \boldsymbol{g}$), for all existing authors a in document d_j. Thus we can easily get:

$$p(k_{jt} = k, a_{jt} = a | \boldsymbol{t}^{-ji}, \boldsymbol{a}^{-jt}, \boldsymbol{k}^{-jt} \boldsymbol{g}) \propto \begin{cases} m_{ak\cdot}^{-jt} \cdot f_{g_{ak}}^{-x_{jt}}(x_{jt}) & \text{if } k \text{ previously used for } a, \\ \eta \cdot p(x_{jt} | a_{jt} = a, k_{jt} = k^{new}, \boldsymbol{g}) & \text{if } k \text{ is new for } a. \end{cases} \tag{3.25}$$

Sampling g

Since the global topic distribution $\boldsymbol{g}$ involves only all components k of all existing authors a in corpora, regardless of the author mixture method in local documents. Thus, integration of global topic distribution G_0 is the same for both models (3.3) and model (3.4), as stated in equation 3.20.

3.5.3 Summary of Sampling Schema

The resulting pseudo-code for the general process of our gibbs sampling based inference algorithm is depicted in Algorithm 2:

The graphical representation of this extended *Chinese Restaurant Franchise* inference process for the generative process of our `HDPauthor` model is displayed in Figure 3.3:

Algorithm 2 `HDPauthor` algorithm

procedure GIBBS–HDPAUTHOR
  for each document $d_j \in D$ **do**
    for each word token $x_{ji} \in d_j$ **do**
      Incrementally sample t_{ji} for x_{ji}
      Update statistic values for t_{ji}
    end for
  end for
  while not converged **do**
    for each document $d_j \in D$ **do**
      for each word token $x_{ji} \in d_j$ **do**
        Remove statistic value for old t_{ji}
        Sample t_{ji} for x_{ji}
        Update statistic values for new t_{ji}
      end for
      for each table $\psi_{jt} \in d_j$ **do**
        Remove statistic value for old $< a_{jt}, k_{jt} >$
        Sample $< a, k >$ for ψ_{jt}
        Update statistic values for new $< a_{jt}, k_{jt} >$
      end for
    end for
    for each author $a \in$ author set **do**
      for each component $\omega_{ak} \in a$ **do**
        Remove statistic value for old g_{ak}
        Sample g for ω_{ak}
        Update statistic values for new g_{ak}
      end for
    end for
  end while
end procedure

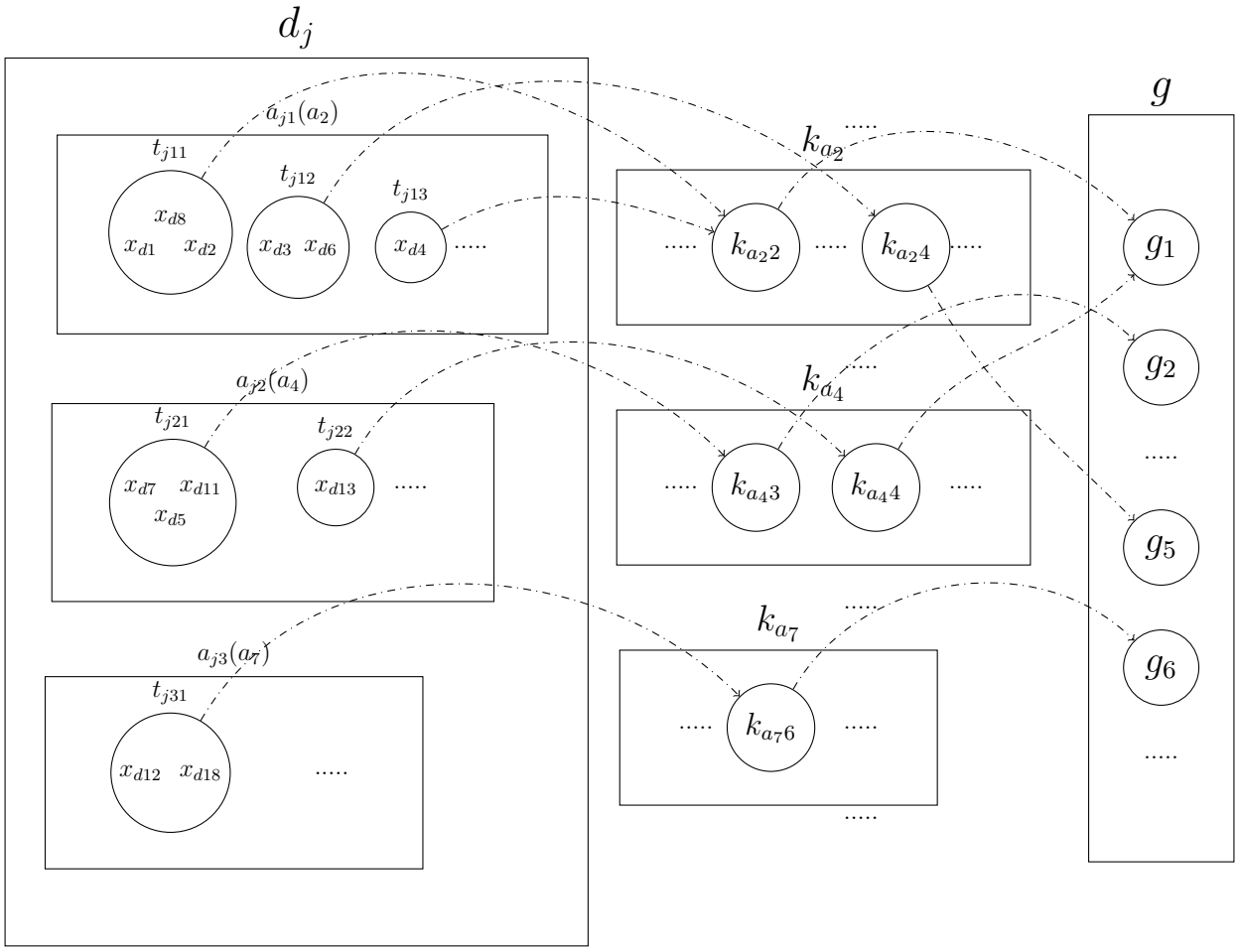

Figure 3.3: Inference process for HDPauthor model

Chapter 4

Experiment

In this chapter, we will show how we use real-world data sets retrieved from different sources for experiments using our models. We will discuss the evaluation criteria that we use for performance analysis of learning results. We will also illustrate the experimental results and performance of our system on the experiments that we conducted.

4.1 `HDPsent` Model Experiments

4.1.1 Test Bed

We chose two data sets for conducting experiments on our model, both of which are product/service review data sets. There are two advantages of conducting experiments for `HDPsent` model on product/service reviews. First of all, customers tend to express strongly subjective evaluations in review text. Reviewers write not only descriptions of products and their personal experiences, but also sentiments towards the product/service that are often strong. Secondly, reviewers typically focus on different aspects of same type of product/service. It is beneficial for us to fulfill aspect-level sentiment modeling in our `HDPsent` model.

***TripAdvisor* data set**

The first data set is the *TripAdvisor* hotel review data set provided by Wang, Lu, and Zhai[44]. This data set consists of a set of hotel review items retrieved from *www.tripadvisor.com*. Each review item contains not only a snippet of the reviewer's free text content of this review, plus the overall rating score values for the hotel in each review range in {-1.0 (data missing), 1.0, 2.0, 3.0, 4.0, 5.0}, but also separate rating values on eight different aspects: {*Business Service, Check in / front desk, Cleanliness, Value, Service, Location, Rooms, Sleep Quality*}, with same value range as overall rating score.

***Yelp* data set**

The other data set we are going to use for our experiment consists of Yelp reviews from Yelp's academic data set [1]. The Yelp review corpus contains customer reviews with high variety among kinds of businesses, such as restaurants, bars, beauty and spas, although restaurants occupy the majority. Each review entry in the Yelp review data set consists of text review content and overall rating score made by reviewer. The rating score for each business also ranges in {1.0, 2.0, 3.0, 4.0, 5.0}. Because of the variety of business categories on Yelp, the total intrinsic number of topics in this review data set is hard to estimate; thus, the categorical features of numerous number of minor businesses are difficult to capture using other models. This characteristic of Yelp reviews is amenable to our nonparametric approach to developing topic and sentiment modeling algorithms.

4.1.2 Evaluation Criteria

Aspect-level review score prediction

It is hard to evaluate the topic distribution, the sentiment distribution, and word distribution that we learned from our model, because we do not have observable "ground-truth" for these distributions[45]. However, the aspect-level rating values on different categories such as {*Business Service, Check in / front desk, Cleanliness, Value, Service, Location, Rooms,*

[1]This data set is available at https://www.yelp.com/academic_dataset

Sleep Quality} in *TripAdvisor* review data set, can be deemed as ground-truth value for sentiment polarity on these predefined topic categories.

However, our model is an unsupervised learning method for topic generation, and it has no direct control on the number of topics generated, nor on the semantic direction of each topic to be generated. Thus, our `HDPsent` model is not able to produce direct predictions on reviewer scores for predefined categories in this data set. For evaluation and performance comparison, we instead use a simple multivariate linear regression algorithm to model the prediction of aspect-level review score on learned results of our model, and evaluate our model by conducting evaluation measure on these predictions, and compare our results with others.

For categorized aspect-level rating value prediction, we use similar evaluation measures as introduced in[44] and[46], such as:

1. Mean square error (MSE) on aspect rating prediction (Δ^2_{aspect})
2. Aspect correlation inside reviews (ρ_{aspect})
3. Aspect correlation across reviews (ρ_{review})
4. Mean Average Precision (MAP)

Here we illustrate how we use multivariate linear regression for aspect-level review score prediction. We use the number of tokens labeled as positive/negative for each learned topic as a feature vector for each review, denoted $x^{(i)}_{pos}$ and $x^{(i)}_{neg}$. Next, we set the ground-truth rating value vector for six aspects, with the overall rating as the target value for machine learning, denoted $y^{(i)} = < y_{overall}, y_{cleanliness}, y_{value}, y_{service}, y_{location}, y_{rooms}, y_{sleep} >$. We then set matrix θ_{pos} and θ_{neg} as for each $x^{(i)}_{pos}$, predicted $\hat{y}^{(i)}_{pos} = x^{(i)}_{pos} \cdot \theta_{pos}$, and for each $x^{(i)}_{neg}$, predicted $\hat{y}^{(i)}_{neg} = x^{(i)}_{neg} \cdot \theta_{neg}$. Finally, we use gradient descent to learn θ_{pos} and θ_{neg} with minimal squared error.

MSE: We use the following definition of mean squared error (MSE) to measure the overall

rating prediction error.

$$MSE = \frac{\sum_{i=1}^{D} \sum_{a=1}^{A} (\hat{y}_a^{(i)} - y_a^{(i)})^2}{D \times A} \tag{4.1}$$

ρ_{aspect}: measures the accuracy for relative ranking order of aspects being learned within review:

$$\rho_{aspect} = \frac{\sum_{i=1}^{D} \rho(\hat{y}^{(i)}, y^{(i)})}{D} \tag{4.2}$$

where $\rho(\hat{y}^{(i)}, y^{(i)})$ is the Pearson correlation coefficient between the predicted rating vector for review i and the corresponding ground-truth rating vector.

ρ_{review}: measures the accuracy for relative ranking order of reviews being learned for each aspect:

$$\rho_{review} = \frac{\sum_{a=1}^{A} \rho(\hat{y}_a, y_a)}{A} \tag{4.3}$$

where $\rho(\hat{y}_a, y_a)$ is the Pearson correlation coefficient between the predicted rating vector for aspect a across all reviews and the corresponding ground-truth rating vector.

MAP: Because the ground-truth rating values are discrete numbers as {*1.0, 2.0, 3.0, 4.0, 5.0*}, it is impractical to predefine the number of top hotels as a constant, or as a fixed percentage, in our evaluation. Therefore, we define MAP in this experiment as the accuracy of ranking the top N hotels as top, where N is assigned dynamically as the total number of hotels in data set whose rating value is the highest value 5.0 as:

$$\begin{aligned} R_a &= \{i | y_a^{(i)} = 5.0\} \\ \hat{R}_a &= \{top\ |R_a|\ reviews\ predicted\} \\ MAP &= \frac{\hat{R}_a \cap R_a}{|R_a|} \end{aligned} \tag{4.4}$$

We also estimate the percentage of top 50 reviews that we ranked, whose ground-truth review value is 5.0 for each aspect. We denote this value as MAP@50.

Perplexity

We also use perplexity to test the convergence of this Markov chain and the performance of our model. The *perplexity* of our model is calculated as:

$$\begin{aligned} \text{perplexity}(\boldsymbol{w}_d|d) &= \exp\Big[-\frac{\sum_d \ln p(\boldsymbol{w}_d|d)}{\sum_d N_d}\Big] \\ p(\boldsymbol{w}_d|d) &= \prod_{x=1}^{N_d}[\sum_{k,l} p(w|k,l)p(l|k,d)p(k|d)] \end{aligned} \tag{4.5}$$

However since we use Gibbs sampling for inference, the expected $p(k|d)$, should be estimated according to our HDP sampling schema as:

$$\begin{aligned} p(k|d) &= \frac{\sum_{k_{jt}=k} n_{jt}}{n_d + \alpha_0} + \frac{\alpha_0}{n_d + \alpha_0} \cdot \frac{m_k}{m_\cdot + \gamma} \\ p(k^{new}|d) &= \frac{\alpha_0}{n_d + \alpha_0} \cdot \frac{\gamma}{m_\cdot + \gamma} \end{aligned} \tag{4.6}$$

And the estimation of $p(l|k,d)$ can be calculated according to Equation 2.11, and $p(w|k,l)$ can be calculated according to Equation 2.12 and 2.13.

4.1.3 TripAdvisor Experiment

We first cleaned our text collections. We used the Stanford *CoreNLP* tool[47] to lemmatize the tokens in the review text. All stop words were also removed. We also removed some review items from data set, if any review value of six aspects was missing, or if the review text was too short. Finally, we filtered out 563 reviews from original data set to construct the data set for our experiments.

We used the sentiment word list extract from MPQA Subjectivity Lexicon[48] to build lists of positive and negative words as prior knowledge for sentiment label initialization. Since we ignore the Part-of-Speech (POS) tags[49] of tokens in text, we preserve only those words whose sentiment polarity is same across all possible POS tags. When we run our model, we first initialize the sentiment label of each word token as positive/negative; if it is

present in the positive/negative word list that we generated above; we label all other data tokens as neutral. Then the following learning process will choose to preserve or change the initial sentiment labels based on the updates by sampling from the posterior probability of sentiment labels. And according to the feature of Markov chain, the sentiment allocation will come to a stable stage when it converges, regardless of the initial values.

We ran a set of experiments for our `HDPsent` model with different initial concentration parameters of α_0, β and γ. Different parameters indicate different degree of variability, which will result in generating different number of topics. In Table 4.1, we present a comparison of four different topics learned from this data set with top neutral, positive, and negative words, with 181 topics learned from this data set.

Topic 5		
Neutral	Positive	Negative
drink	good	hard
food	perfect	extremely
restaurant	nice	bad
service	fresh	cold
staff	outstanding	roll
wine	excellent	slightly
waiter	delicious	spot
time	clean	hassle
bar	top	noisy

Topic 19		
Neutral	Positive	Negative
room	clean	smoke
bed	light	dirty
smell	sound	wipe
door	top	fall
floor	reason	tired
towel	expect	back
day	open	garbage
shower	girl	exhaust
wall	happy	cheap

Topic 27		
Neutral	Positive	Negative
beach	great	spot
water	real	hard
lot	nice	part
chair	beautiful	low
day	warm	empty
swim	helpful	dark
walk	white	slow
rain	spacious	bad
sand	hot	dress

Topic 70		
Neutral	Positive	Negative
room	safe	back
leave	open	rude
arrive	clean	lose
move	tour	problem
check	thankfully	complain
make	valuable	miss
key	nice	month
towel	settle	spot
luggage	good	sad

Table 4.1: Table for four different topics from *TripAdvisor* reviews

Although *TripAdvisor* data set consists of reviews on hotels, the variability is constrained

than reviews on *Yelp.com*, or *Amazon.com*, our model is still able to differentiate reviews on restaurant and dining place, room quality, experience on beach and custom service.

Table 4.2 lists the resulting evaluation measures with different number of topics generated. In this table (+) means that we only use the number of positive tokens we learned for each topic in each document as feature vector, and (-) means that we only use the number of negative tokens as feature vector. We compared our results with LARA model and Support Vector Regression (SVR) model from[44].

Number of topics (sentiment polarity)	Δ^2_{aspect}	ρ_{aspect}	ρ_{review}	MAP	MAP@50
36(+)	0.792	0.350	0.627	0.691	0.854
36(-)	0.792	0.357	0.626	0.455	
137(+)	0.494	0.501	0.789	0.776	0.949
137(-)	0.427	0.518	0.816	0.730	
181(+)	0.388	0.555	0.836	0.808	0.951
181(-)	0.371	0.584	0.847	0.712	
LARA	1.190	0.180	0.425	0.657	0.703
SVR-A	1.012	-0.081	0.804	0.796	0.95
SVR-O	0.855	-0.007	0.579	0.714	0.79

Table 4.2: Evaluation measures for the *TripAdvisor* experiment compared to LARA and baseline models

We can observe that the greater the amount of variability we set for our `HDPsent` model, the more topics generated from our `HDPsent` model, allowing us to get a better prediction on review scores for each aspect. Even with only 36 topics generated, however, we can obtain an outstanding prediction performance compared to that of other prediction methods.

We here represent the perplexity of our model in figure 4.1:

From this perplexity figure, we can discover that in all cases, the perplexity of our model reaches to a stable phrase quickly. This shows that our Markov chain begins to converge early in our learning process. On the other hand, the more variability that we give the system, the lower perplexity it can attain. Our model is able to extract and differentiate minor topics if we give a enough probability for new topic generation.

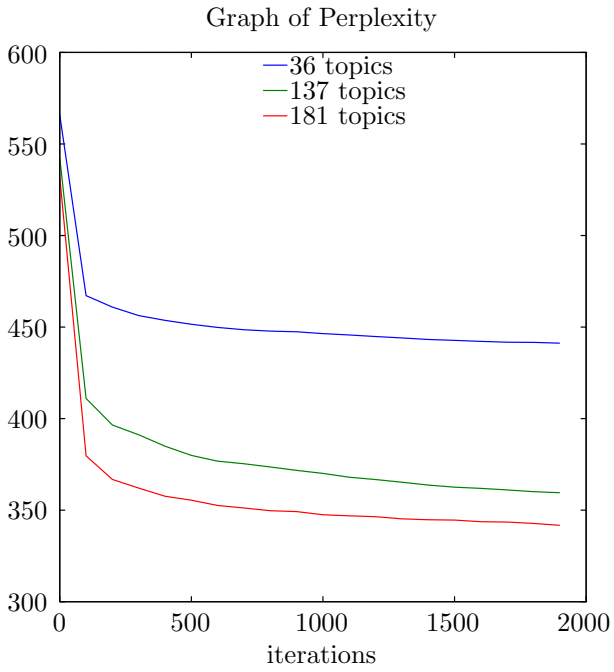

Figure 4.1: Perplexity evolution for *TripAdvisor* experiments

4.1.4 Yelp Experiment

We performed an additional experiment using a subset of the Yelp review corpus. We extracted review text content from this data set, and applied same strategy for data preparation, including word token lemmatization, and sentiment label initialization for this experiment as that we used for the *TripAdvisor* experiment.

We ran our `HDPsent` model in the same way on a data set of 582 reviews from Yelp. Similarly, different parameters will result in different number of topics generated. Here in Table 4.3, we present our learning result for one experiment with 72 topics generated. We illustrate a comparison of four different topics learned from this data set with top neutral, positive, and negative words. For example, we can see that the most frequent neutral words about wedding ceremonies (Topic 3) and restaurants (Topic 8) are quite different. Also, even some generally positive words as "great", "love", "touch" occur in both topics, some words as "fresh", "delicious", "tender" only show up in restaurant-related topics, and "marry", "wonderful" only show up in wedding ceremony-related topics. And in house

and apartment rent related topic (Topic 31), "deal" is presented as top positive word, and "problem", "break", "smoke" are frequently mentioned as negative words. Therefore, our `HDPsent` model can successfully form different sentiment word distributions under different topics, dig out the most commonly appraised, as well as complained aspects in each topics.

Another interesting phenomena is that negation words appear very frequently in both negative lists. It is also understandable, since users always use negation words to describe unpleasant experience, and express negative feelings.

Topic 3		
Neutral	Positive	Negative
wedding	choose	flower
guest	great	didnt
day	marry	handle
estancia	top	yell
venue	special	dont
event	amazing	odd
reception	wonderful	stress
package	touch	bad
ceremony	love	scream

Topic 8		
Neutral	Positive	Negative
taste	fresh	side
flavor	nice	wasnt
dish	delicious	bland
sauce	tender	miss
bit	top	finish
food	enjoy	didnt
order	great	strong
sweet	love	lack
bite	touch	ill

Topic 31		
Neutral	Positive	Negative
apartment	live	complex
year	deal	problem
move	nice	window
place	security	break
time	pretty	dont
month	complaint	wasnt
rent	special	dog
parking	star	open
building	replace	smoke

Topic 39		
Neutral	Positive	Negative
place	pretty	dont
thing	small	didnt
good	great	bad
ive	worth	long
price	nice	reason
time	fair	wouldnt
lot	general	couldnt
people	live	decent
make	friend	expensive

Table 4.3: Table for four different topics from Yelp Reviews

4.2 HDPauthor Experiments

4.2.1 Test Bed

For our HDPauthor models for author-topic learning, we mainly focus on experiments for academic publications. There are several advantages to choose academic publications: first, the cooperation between different authors is frequent. Academic papers are always written by not only one author, but several authors, which helps us to learn the author mixing vectors for each document. Second, each author usually works on only one research area, or a few direction on different research areas that closely related each other. This is also an advantage for us in modeling the topic distribution for each author that exists in a whole corpus. Third, most authors publish several papers, therefore the modeling of author topics can be learned from multiple sources of local documents, rather than a single source.

Hence, here we choose two data sets for conducting experiments on our HDPauthor model, both of which are text collections of academic papers. We chose the *NIPS* data set, which consists of the full text content of papers published in *NIPS* conferences. We also chose the *DBLP* data set, which consists of abstracts of papers published in a high variety of conferences, but in related research areas.

***NIPS* data set**

The data set we are going to use for this model is *NIPS Conference Papers*[2] Volume 0-12, provided by Sam Roweis [3]. NIPS data set contains a collection of OCR processed text of papers published in the Neural Information Processing (NIPS) Conference from 1987 to 1999, which is mainly focus on researches in artificial intelligence, machine learning and computational neuroscience. It contains 1,740 papers in total, each paper consists of full content in text format and an author list of it. And it involves a total of 2,037 authors. This data set is suitable for our model since it is a set of papers in one general research area with papers with different research topics as a combination of slightly different specific research

[2] http://papers.nips.cc/

[3] This data set is available at http://www.cs.nyu.edu/~roweis/data.html

directions. Authors in the neural network related research area always cooperate with each other and publish papers in this conference during this period of 10 years. Therefore, the co-authorship information in this data set can also help us to infer the topic interests mixture for each author.

***DBLP* abstract data set**

We here use another citation network data set [4], extracted from the Digital Bibliography and Library Project (DBLP), ACM Digital Library, and other sources, and provided by Arnetminer[50]. Although this data set is mainly for research on citation analysis, co-author networks and other academic heterogeneous information network analysis, we noticed that this data set contains the metadata of title, author, conference, and abstract (used as document content) for each academic publication entry, which is enough for us to conduct experiments on our `HDPauthor` model. This data set contains 1,572,277 papers in total, from all kinds of fields ranging from math and physics to health informatics.

Since there are too many scientific publications from conferences or journals across almost all research fields in this data set, the research topic range is too comprehensive, and too sparse for our model. The size of this data set is also too huge for us to conduct an efficient learning experiment. To better observe the results of our experiment, we selected only publications from conference in five areas in the computer science category, namely: {*Machine Learning (ML), Information Retrieval (IR) , Artificial Intelligence (AI), Natural Language & Speech (NLP), Data Mining (DM)*}. These are active research fields on different topics but which are mutually related to each other. We then focused only on publications from top ranked conferences from each of the area. In Table 4.4 we list the top conferences we take in our filter that we retrieved from Microsoft Academic Search [5]:

[4] This data set is available at https://aminer.org/billboard/citation

[5] http://academic.research.microsoft.com/

Research Area	conferences
Machine Learning	NIPS, ICML, UAI, IROS, ICPR, ISNN, COLT, ECML, ICDAR, ICANN
Data Mining	KDD, ICDE, CIKM, ICDM, SDM, PKDD, PAKDD, RIAO, DMKD, DASFAA
Natural Language & Speech	NIPS, ACL, ICASSP, COLING, NAACL, EACL, ANLP, HLT, LREC, EMNLP, ASRU
Information Retrieval	SIGIR, TREC, CIKM, DL, JCDL, ECDL, RIAO, ECIR, CLEF, SPIRE
Artificial Intelligence	AAAI, IJCAI, ICML, ICRA, ICGA, AAMAS, UAI, KR, IROS, CEC, ECAI

Table 4.4: Table for top conferences in computer science research areas

4.2.2 Evaluation Criteria

Comparison of topic models with associated authors is also difficult, since we do not have concrete ground truth for evaluating the results of learning. We compare our model to others by conducting an information retrieval (IR) task and evaluating our system's performance on the overall task based on measurable performance on this IR task. Although this is an indirect method for model comparison, finding similar documents, or documents in same research area, is an widely-used application for topic models.

Comparison to other models

For our DBLP experiment, we used publications from top conferences listed in 4.4 from five major research areas in computer science: {*ML, IR, AI, NLP, DM*}. These five major research area headings can be used as category labels for each publication in our data set, according to the category of conference in which they were published. Publications with same category label are assumed to be relevant in our retrieval evaluation.

Some conferences, however, are presented as top conferences in multiple search areas. For example, NIPS (Neural Information Processing Systems) is ranked the top 1 in ML, as well as top 1 in NLP, while ICML (International Conference on Machine Learning) is ranked number 2 in ML and number 3 in AI. In these cases, we allow for multiple labels for papers

published in these conferences. Each paper is associated with a set of category labels, if they are published in such conferences. Since document retrieval tasks only predict retrieved documents as relevant, or non-relevant, we here assume that two documents are relevant if there is at least one category label that matches from the label sets of both sides. For example, papers published in NIPS are relevant to papers published in conferences either in the ML conference list or in the NLP conference list.

We then obtained 100 papers other than the training data set, 20 papers in each category, and used these as the query set for our experiment. For simplicity, we avoided papers from conferences in multiple areas, so that each paper is only associated with exactly one label. We built query word tokens from each query paper using several different methods, and we treated each query consisting of list of word tokens as also as a bag-of-words. We then used information retrieval methods to calculate the relevance of query to each document in corpus. We then ranked the document according to the degree of relevance that we calculated. We compared the relevance ranking result of our model with three other models: Okapi BM25 algorithm[51] for the term frequency - inverse document frequency (TFIDF) retrieval metric, traditional HDP model for pure topic learning, and Author-Topic model[38].

Okapi BM25 algorithm is one variation of the TFIDF-based method, which ranks documents d_j for a given query q by score calculated as:

$$score(d_j, q) = \sum_{w \in q} IDF(w) \cdot \frac{N_{jw} \cdot (k_1 + 1)}{N_{jw} + k_1 \cdot (1 - b + b \cdot \frac{N_{j\cdot}}{\hat{N}})} \tag{4.7}$$

Here $\hat{N}$ is the averaged document length for all document in corpus. We ues D to denote the number of total documents in corpus, and D_w to be the number of documents in corpus that contains word w, as $D_w = |\{d_j | w \in d_j\}|$. And then $IDF(v)$ is calculated as follows:

$$IDF(w) = \log \frac{D - D_w + 0.5}{D_w + 0.5} \tag{4.8}$$

For traditional HDP topic models, we calculate $P(q|d)$ for document ranking, which is

the probability of the sequence of words in a query q be produced by a certain document d. This probability can be calculated as:

$$p(q|d) = \prod_{w \in q} \sum_{k=1}^{K} p(w|k)p(k|d) \tag{4.9}$$

Here $p(w|k)$ and $p(k|d)$ are estimated word distribution for each topic, and topic distribution for each document that learned from HDP model.

We also implemented the Author-Topic (AT) model, to compare our `HDPauthor` model to an LDA-based author-topic mutual learning model. This model is an extension of LDA topic modeling, which assumes that the topic distribution for each author is drawn from a Dirichlet distribution, and the word distribution for each topic is also drawn from a Dirichlet distribution. The generative AT model assumes that the author label for each token is sampled uniformly from the author list of document, and then the topic label for each token is sampled according to the topic distribution for this author. We then used the query likelihood calculation that Rosen presents in equation (11) of[38] as:

$$\begin{aligned} p(q|d_j) &= \prod_{w \in q} [\frac{1}{|a_j|} \sum_{a \in a_j} \sum_{k]1}^{K} p(w|k)p(k|a)] \\ &\propto \prod_{w \in q} [\frac{1}{|a_j|} \sum_{a \in a_j} \sum_{k=1}^{K} \frac{N_{kw} + \beta}{N_{k\cdot} + V\beta} \cdot \frac{N_{ak} + \alpha}{N_{a\cdot} + K\alpha}] \end{aligned} \tag{4.10}$$

In our `HDPauthor` model, we also calculate $p(q|d)$ for document relevance ranking. Since we assume that each document is a finite mixture of authors in this document, and each author is associated with a topic distribution, the query likelihood calculation for `HDPsent` model can be presented as:

$$p(q|d_j) = \prod_{w \in q} [\sum_{k=1}^{K} p(w|g)p(g|d_j)] \tag{4.11}$$

In this equation, $p(g|d_j)$ is the posterior approximation of a topic distribution for doc-

ument d_j, which is represented as a mixture of Dirichlet processes for all its authors with mixing proportion vector $\boldsymbol{\pi}_j$ that we set in this model, which can be directly inferred from our learning result.

Thus, for mixing model (3.3), we can get the estimated $p(g|d)$ as:

$$\begin{aligned} p(g|d) &= \frac{\sum_{g_{jt}=g} n_{jt}}{n_d + \alpha_0} + \frac{\alpha_0}{n_d + \alpha_0} \cdot \big[\sum_{a \in \boldsymbol{a}_j} \pi_{ja} \cdot p(g|a) \big] \\ p(g^{new}|d) &= \frac{\alpha_0}{n_d + \alpha_0} \cdot \big[\sum_{a \in \boldsymbol{a}_j} \pi_{ja} \cdot p(g^{new}|a) \big] \end{aligned} \tag{4.12}$$

For mixing model (3.4), since in our experiment we already set $\epsilon = \alpha_0$ to simplify the probability calculation, we can infer the estimated $p(g|d)$ as:

$$\begin{aligned} p(g|d) &= \frac{\sum_{g_{jt}=g} n_{jt}}{n_d + |\boldsymbol{a}_j| \cdot \alpha_0} + \frac{\alpha_0}{n_d + |\boldsymbol{a}_j| \cdot \alpha_0} \cdot \big[\sum_{a \in \boldsymbol{a}_j} p(g|a) \big] \\ p(g^{new}|d) &= \frac{\alpha_0}{n_d + |\boldsymbol{a}_j| \cdot \alpha_0} \cdot \big[\sum_{a \in \boldsymbol{a}_j} p(g^{new}|a) \big] \end{aligned} \tag{4.13}$$

Here $p(g|a)$ and $p(g^{new}|a)$ are Dirichlet process-based topic distributions for each author existing in the corpus. This can be computed approximately from our learning result. We have:

$$\begin{aligned} p(g|a) &= \frac{\sum_{g_{ak}=g} m_{ak}}{m_{a\cdot} + \eta} + \frac{\eta}{m_{a\cdot} + \eta} \cdot p(g) \\ p(g^{new}|a) &= \frac{\eta}{m_{a\cdot} + \eta} \cdot p(g^{new}) \end{aligned} \tag{4.14}$$

Here $p(g)$ and $p(g^{new})$ are global topic distributions, that:

$$\begin{aligned} p(g) &= \frac{l_g}{\sum_g l_g + \gamma} \\ p(g^{new}) &= \frac{\gamma}{\sum_g l_g + \gamma} \end{aligned} \tag{4.15}$$

Finally, $p(w|g)$ is the word distribution for each topic generated, and can be estimated

using Equation 3.6 and 3.7.

Evaluation of ranked retrieval results

Because a traditional precision-recall curve only depicts performance on a single query, and generally always appears as a jagged curve, it is difficult to make quantified comparison between different queries, or to represent performance on a set of queries. Instead, we use 11-point interpolated average precision[52] to represent average performance overfor the set of queries, and to directly compare results from different models.

11-point interpolated precision sets fixed recall values $r = \{0.0, 0.1, 0.2, ..., 1.0\}$ which are 11 equidistant points on the scale from 0.0 to 1.0. The interpolated precision value at each recall level r_i is then defined as the highest precision value afterwards, which can be represented:

$$p(r) = \max_{r' \geq r} \text{Precision}(r') \tag{4.16}$$

Finally, we average $p = \{p(r = 0.0), p(r = 0.1), ..., p(r = 1.0)\}$ for all queries in a query set, so that we can plot our performance on query set as a single averaged 11-point interpolated precision-recall curve, and make a direct comparison between performance of different models.

Perplexity

Perplexity is an evaluation method widely used in topic modeling. This measurement helps us to quantitatively evaluate how well our model predicts new documents, when our data set is unlabeled. With author mixture injected in our `HDPauthor` model, we can establish perplexity as follows:

$$\begin{aligned} \text{perplexity}(\boldsymbol{w}_d | d, \boldsymbol{a}_d) &= \exp\Big[- \frac{\ln p(\boldsymbol{w}_d | d, \boldsymbol{P}_d)}{N_d} \Big] \\ p(\boldsymbol{w}_d | d, \boldsymbol{a}_d) &= \prod_{x=1}^{N_d} \Big[\sum_g p(w|g) p(g|d) \Big] \end{aligned} \tag{4.17}$$

Here $p(g|d)$ and $p(w|g)$ for each possible topic (including new topic) can be calculated in a similar fashion to our calculation for Equation 4.11.

4.2.3 NIPS Experiment

To better assess learning of cooperation between authors who publish papers in a single conference, we extracted a subset of papers with denser connections between authors in the Neural Information Processing Systems (NIPS) conference, which emphasizes neural and probabilistic models. We finally obtained a data set with 873 papers, written by 850 authors in total.

Here in Table 4.5 we demonstrate an example of 4 selected frequent topics with its 10 most likely words and 10 most likely authors listed in a descending order.

We can observe from Table 4.5 that our model is able to successfully differentiate specific research areas and directions among papers in the NIPS conference. Topic 1 and Topics 2 are general topics commonly exists in almost all the documents across the whole data set, and shared by almost all authors. We can easily obtain that Topic 1 is a general topic for machine learning and computational neuroscience which is the overall subject for NIPS. Topic 2 is a general topic representing research and experiment methods in computer science area. Therefore, nearly every paper published in this conference will carry these two topics. The top authors listed in these two topics are also active authors that that have many publications in the NIPS conference.

However, our `HDPauthor` model is able to discover a variety of more specific research areas in neuroscience, including developments in algorithms, applications of neural netowrks, etc. We can easily spot specific research subjects as "speech recognition", "visual system", "artificial intelligence", and "Bayesian learning" are clearly represented by the top words from these topics. Our `HDPauthor` model is also good at identifying most contributed authors from each of these learned topics. And we can observe that some well-known authors, such as Christopher M. Bishop (Bishop_C), Christof Koch (Koch_C), and Satinder Singh (Singh_S)

are ranked high in the subjects related to their research areas.

Topic 1			
Word	Prob	Author	Prob
network	0.107	Sejnowski_T	0.056
input	0.045	Mozer_M	0.035
neural	0.028	Hinton_G	0.022
learning	0.028	Bengio_Y	0.022
unit	0.027	Jordan_M	0.020
output	0.027	Chen_H	0.016
weight	0.023	Moody_J	0.016
training	0.019	Stork_D	0.016
time	0.014	Munro_P	0.014
system	0.013	Sun_G	0.013

Topic 2			
Word	Prob	Author	Prob
set	0.015	Sejnowski_T	0.032
result	0.015	Jordan_M	0.025
figure	0.014	Hinton_G	0.022
number	0.013	Koch_C	0.020
data	0.011	Dayan_P	0.019
function	0.010	Moody_J	0.015
based	0.008	Mozer_M	0.014
model	0.008	Tishby_N	0.014
method	0.008	Barto_A	0.013
case	0.008	Viola_P	0.013

Topic 109			
Word	Prob	Author	Prob
gaussian	0.036	Bishop_C	0.222
process	0.021	Williams_C	0.173
function	0.020	Schottky_B	0.146
distribution	0.019	Winther_O	0.092
bayesian	0.019	MacKay_D	0.085
prior	0.018	Vivarelli_F	0.078
posterior	0.017	Marion_G	0.073
evidence	0.015	Ferrari-T_G	0.048
covariance	0.015	Sollich_P	0.033
error	0.011	Beal_M	0.026

Topic 98			
Word	Prob	Author	Prob
image	0.049	Koch_C	0.119
visual	0.028	Horiuchi_T	0.106
field	0.023	Ruderman_D	0.088
system	0.020	Bialek_W	0.068
pixel	0.017	Dimitrov_A	0.05
filter	0.015	Bair_W	0.038
signal	0.013	Indiveri_G	0.035
object	0.013	Viola_P	0.030
center	0.012	Zee_A	0.030
local	0.011	Miyake_S	0.027

Topic 72			
Word	Prob	Author	Prob
policy	0.040	Singh_S	0.630
state	0.035	Duff_M	0.098
algorithm	0.034	Mansour_Y	0.069
learning	0.031	Crites_R	0.053
method	0.015	Sutton_R	0.041
probability	0.014	Munos_R	0.031
function	0.012	Gullapalli_V	0.022
reward	0.012	Barto_A	0.015
optimal	0.011	Thrun_S	0.011
problem	0.011	Neuneier_R	0.006

Topic 110			
Word	Prob	Author	Prob
word	0.053	Tebelskis_J	0.107
speech	0.042	Franco_H	0.089
recognition	0.037	Bourlard_H	0.086
training	0.025	De-Mori_R	0.084
frame	0.020	Rahim_M	0.069
system	0.017	Waibel_A	0.055
error	0.014	Hild_H	0.043
hmm	0.013	Chang_E	0.038
level	0.012	Singer_E	0.036
output	0.012	Bengio_Y	0.035

Table 4.5: Example of top topics learned from *NIPS* experiment

Table 4.6 presents famous authors whom we selected, and lists the topics for each of them. Since Topic 1 and Topic 2 are common topics for almost all authors, we omitted these two topics, and only listed the three most likely topics besides Topic 1 and Topic 2:

Hinton_G (Geoffrey Hinton)		
Topic 154	Topic 132	Topic 98
model	expert	image
image	task	visual
unit	mixture	field
hidden	network	system
hinton	architecture	pixel
code	gating	filter
digit	weight	signal
vector	nowlan	object
energy	soft	center
space	competitive	local

Bengio_Y (Yoshua Bengio)		
Topic 90	Topic 110	Topic 28
model	word	gate
data	speech	unit
parameter	recognition	input
mixture	training	threshold
distribution	frame	circuit
likelihood	system	polynomial
algorithm	error	output
probability	hmm	layer
density	level	parameter
gaussian	output	machine

LeCun_Y (Yann LeCun)		
Topic 84	Topic 18	Topic 25
feature	tdnn	state
recognition	delay	action
cun	speaker	learning
digit	recognition	time
character	time	reinforcement
output	waibel	policy
layer	architecture	function
denker	window	step
image	network	control
vector	net	optimal

Platt_J (John Platt)		
Topic 94	Topic 83	Topic 115
hand	smo	chip
image	svm	neuron
network	training	circuit
character	algorithm	neural
recognition	kernel	analog
template	set	figure
pixel	problem	system
system	svms	vlsi
frame	vector	output
convolutional	linear	voltage

Table 4.6: Example of top topics for selected authors learned from *NIPS* experiment

Our model is able to associate each author with both general topics and a small subset of specific topics which represent the technical expertise of each author. This representation also matches our intuition regarding the knowledge of experts. An expert typically masters foundational knowledge in one general research area, as well as basic techniques for conducting research, and by definition also has deep and specialized knowledge in a few subareas of this area. Several authors cooperate and utilize their own knowledge, both general and

specialized, to finish a scientific article. While LDA-based author-topic model has to assign each predefined topic with certain probability for each author, our `HDPauthor` model is able to dynamically discover **the** specialized research area for each author, and only impute topics related to these subareas of expertise to authors.

4.2.4 DBLP Experiment

We retrieved publications from all the top conferences listed in Table 4.4. Considering to the fast evolution of subjects in research areas, we only collected papers published during the period from the years 2000 through 2010 (newest time in data set). Also, for better learning of topic distributions and author contributions for each paper in data set, we filtered out papers whose abstracts were too short. To get a denser and closer connected author-cooperation data set, we also filtered out borderline papers if authors did not contribute to many other papers. We then generated a data set for experiment with abstracts from 3,177 papers as documents, and with a total of 2,428 authors involved.

We ran experiments with different parameter settings on both mixing model (3.3) and model (3.4). Different parameter settings would result in different distribution in global topics, topics for each author, and also local topic and author contribution for each document. We represent the perplexity evolution calculated from Equation 4.17 of our Gibbs sampling process in Figure 4.2:

In Figure 4.2 we can observe that the per-word likelihood score estimated from our Gibbs sampling schema for both mixing model (3.3) and model (3.4) converges quickly after a few of iterations at the very beginning, and it reaches a stable stage very soon and maintains this stable perplexity from then on.

We chose one learning result from mixing model (3.3) as an example. This experiment generated 196 topics in total from this learning process, we manually examined those topics with highest probability across the whole corpus, and from them we chose four topics highly related to research areas of { *DM, AI, IR, ML* }, here we illustrate the table of top words

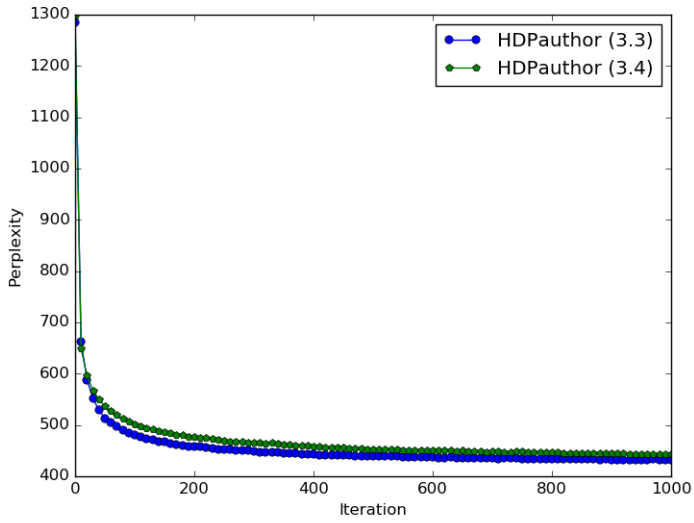

Figure 4.2: Perplexity evolution for *DBLP* experiments

and top authors for these four selected topics as example in Table 4.7.

Our `HDPauthor` model demonstrates its robustness in successfully generating separate topics in different research areas from a relatively small data set in an unsupervised way, even though these areas are highly related. It is able to identify the most frequent words in different research directions, such as "data", "mining" in *DM*; "agent", "strategy" in *AI*; "document", "retrieval" in *IR*; and also "learn", "reinforcement" in ML. It is also able to discover many well-known authors in these research directions, as Charu C. Aggarwal and Philip S. Yu in *DM*; Nicholas R. Jennings in *AI*; ChengXiang Zhai, W. Bruce Croft in *IR*; and Andrew Y. Ng in *ML*.

In Table 4.8 we also illustrate two examples of top topics for two well-known authors, ChengXiang Zhai and JiaWei Han:

We can clearly observe that Topic 1 in the *DBLP* data set is a general research topic. This topic is shared by most authors in this data set. For author ChengXiang Zhai, we can obtain that our HDPauthor model can sucessfully differentiate topics on "Information Retrieval", "Bayesian algorithm" and "Supervised learning". For author JiaWei Han, big name in data mining, we can get his focus research areas include "Classification", "Clustering" and "Web

Topic 3				Topic 11			
Word	Prob	Author	Prob	Word	Prob	Author	Prob
data	0.21	Charu C. Aggarwal	0.070	agent	0.147	Nicholas R. Jennings	0.076
stream	0.072	Jimeng Sun	0.046	mechanism	0.027	Sarit Kraus	0.056
mining	0.037	Philip S. Yu	0.035	system	0.018	Jeffrey S. Rosenschein	0.045
change	0.021	Kenji Yamanishi	0.034	negotiation	0.017	Kagan Tumer	0.036
time	0.020	Hans-Peter Kriegel	0.031	strategy	0.016	Kate Larson	0.036
application	0.012	Wei Wang	0.030	multi	0.014	Michael Wooldridge	0.035
real	0.012	Qiang Yang	0.028	problem	0.014	Moshe Tennenholtz	0.030
online	0.0094	Yong Shi	0.025	show	0.014	Vincent Conitzer	0.029
detect	0.008	Xiang Lian	0.019	multiagent	0.013	Sandip Sen	0.028
detection	0.008	Pedro P. Rodrigues	0.018	design	0.011	Victor R. Lesser	0.025

Topic 24				Topic 39			
Word	Prob	Author	Prob	Word	Prob	Author	Prob
document	0.093	ChengXiang Zhai	0.11	learn	0.093	Matthew E. Taylor	0.090
retrieval	0.066	Iadh Ounis	0.073	learning	0.084	Shimon Whiteson	0.079
query	0.055	Maarten de Rijke	0.020	reinforcement	0.034	Andrew Y. Ng	0.059
term	0.035	W. Bruce Croft	0.020	policy	0.033	Peter Stone	0.054
information	0.027	Laurence A. F. Park	0.020	task	0.032	Bikramjit Banerjee	0.051
model	0.026	James P. Callan	0.019	algorithm	0.029	Sherief Abdallah	0.040
relevance	0.021	Donald Metzler	0.017	transfer	0.019	Sridhar Mahadevan	0.039
feedback	0.020	Guihong Cao	0.017	action	0.019	Michael H. Bowling	0.036
collection	0.019	C. Lee Giles	0.016	function	0.018	Kagan Tumer	0.033
language	0.017	Oren Kurland	0.016	domain	0.016	David Silver	0.022

Table 4.7: Example of top topics learned from *DBLP* experiment

mining".

ChengXiang Zhai				JiaWei Han			
Topic 24	Topic 1	Topic 150	Topic 140	Topic 1	Topic 83	Topic 93	Topic 2
document	base	model	label	base	classification	clustering	web
retrieval	algorithm	distribution	learning	algorithm	classifier	cluster	page
query	approach	topic	data	approach	feature	data	link
term	paper	probabilistic	learn	paper	training	object	text
information	show	bayesian	supervise	show	class	algorithm	content
model	method	modeling	semus	method	data	set	document
relevance	propose	mixture	classification	propose	method	high	information
feedback	problem	data	unlabeled	problem	learning	dataset	category
collection	result	probability	active	result	selection	propose	search
language	set	random	training	set	learn	type	semantic

Table 4.8: Example of top topics of specific authors learned from *DBLP* experiment

We also use the evaluation criteria we introduced in 4.2.2 to compare our `HDPauthor` model to other models as Okapi BM25, HDP modeling, Author-Topic (AT) model, by conducting academic document retrieval tasks for queries constructed from academic documents outside training data set. We retrieved 100 papers from data set, and used four methods to construct list of query word tokens from query paper:

1. We use title of each query paper as query tokens for retrieval.
2. We use title of each query paper, associated with author names as query tokens for retrieval.
3. We use abstract of each query paper as query tokens for retrieval.
4. We use abstract of each query paper, associated with author names as query tokens for retrieval.

Okapi BM25 is a pure information retrieval technique, and HDP model is only for topic modeling. Both of them are not able to be incorporated with author information directly. We then follow the steps from [40], add author names to each document in data set as additional word tokens, and use author names of each query paper as additional query tokens for retrieval.

For AT model and `HDPauthor` model, since we can derive topic distribution for each author directly from learned result, we add topic similarity score as one more measurement in retrieval score calculation.

We here rewrite Equation 4.11 in evaluation criteria for $p(q|d)$ calculation for document relevance ranking as:

$$p(q, \boldsymbol{a}_q|d_j, \boldsymbol{a}_j) = \omega \cdot p(q|d_j) + (1-\omega) \cdot similarity(\boldsymbol{a}_q, \boldsymbol{a}_j) \tag{4.18}$$

Since each author in this model is represented as a vector of topic distribution, we can use cosine similarity[53] to calculate the distance between two vectors represented by topic distribution from authors. For our evaluation purpose, we here simply average the topic distribution of all associated authors for both query document and retrieval document, regardless of the author mixing vector learned from our model. We then calculate cosine similarity as the similarity score for these two averaged topic distribution for authors from two sides:

$$\begin{aligned} similarity(\boldsymbol{a}_q, \boldsymbol{a}_j) &= \cos(\frac{1}{|\boldsymbol{a}_q|}\sum_{a\in\boldsymbol{a}_q} p(\boldsymbol{g}|a), \frac{1}{|\boldsymbol{a}_j|}\sum_{a\in\boldsymbol{a}_j} p(\boldsymbol{g}|a)) \\ \cos(p(\boldsymbol{g}|a_1), p(\boldsymbol{g}|a_2)) &= \frac{\sum_g [p(g|a_1)p(g|a_2)]}{\sqrt{\sum_g p(g|a_1)^2}\sqrt{\sum_g p(g|a_2)^2}} \end{aligned} \tag{4.19}$$

Here in Figure 4.3 e illustrate our performance compared to other models. We set $\omega = 0.5$ for Equation 4.18. We implemented the AT model, and set $K = 200$ for this experiment. We used one Python library called Gensim[54] for HDP topic learning. The learning result generated from mixing model (3.3) contains 196 topics in total, and learning result generated from mixing model (3.4) contains 191 topics.

We can infer from the precision-recall curve comparison that using abstracts as query tokens would give all models a better retrieval result than only using titles as query tokens. Both AT model and `HDPauthor` model perform significantly better than Okapi BM 25 and

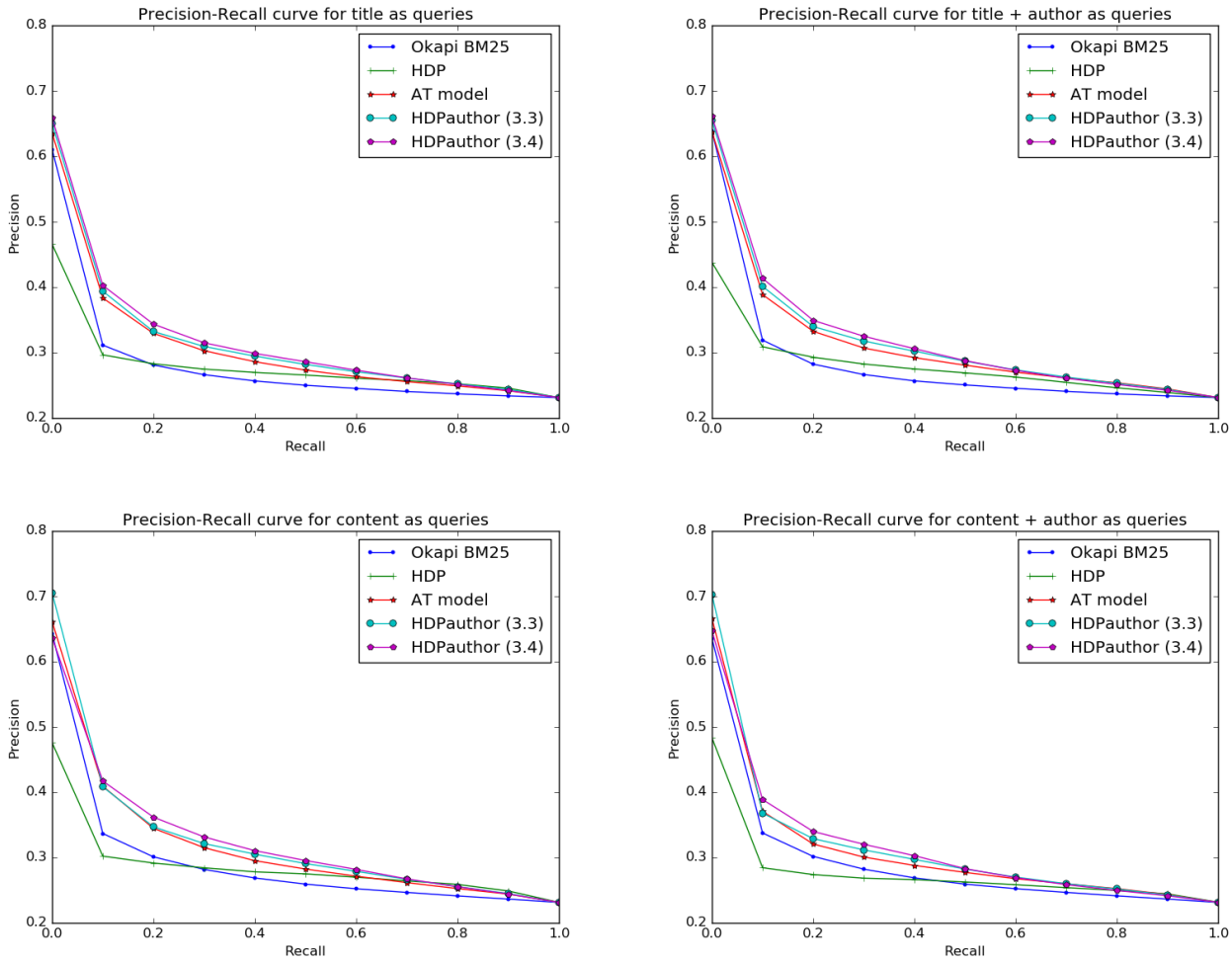

Figure 4.3: Precision-Recall curve for document retrieval for *DBLP* experiment

HDP model, which suggests that incorporation of author information brings improvement to topic modeling, even if we do not include author names in queries, or if we do not explicitly make use of author information in the retrieval task.

Moreover, incorporation of author information into a query improves in retrieval performance across all models. One reason is because the author name represented as a word token is quite rare and unique in data set, which gives a high IDF score for Okapi BM25. HDP does not gain much from author names, however, since infrequent words do not affect topic learning much in traditional HDP model. The author cosine similarity metric also

helps us to identify the similarity between authors for query and authors for documents in data set. Even author similarity alone, without word tokens exist in query, gives us an adequate indication for document retrieval ranking.

Our `HDPauthor` model performs better than AT model under all four situations, although the difference is not quite significant. One main reason is that the topic distributions for authors learned from our `HDPauthor` model is much more skewed than for the LDA-based AT model.

Chapter 5

Conclusion

In this dissertation, we proposed two mixture models that combined HDP nonparametric Bayesian topic models with sentiment analysis and author identification. These two unsupervised learning models can be directly and indirectly applied to practical applications, and solve real-world problems in free text analytics such as inference of overall sentiment and author-centric information retrieval.

5.1 `HDPsent` Model

We have synthesized a Dirichlet process for aspect-level sentiment with the traditional HDP, called `HDPsent`. Unlike other LDA based topic-sentiment hybrid models, this permits the number of topics to be updated based on shared parameters of the generative topic model, rather than restrict them to a predefined, fixed set for a text document collection or to a predefined lexicon for these topics. Furthermore, it allows sentiments associated with these aspects to be inferred concurrently.

A key novel contribution of this topic model is the ability to automatically generate different topics with different word distributions for different sentiment polarities. We learn to assign weights from each topic to a set of aspects that we seek to infer using gradient descent

learning. This permits empirical evaluation by calculating correlation with historical ground truth (on all reviews and ranked reviews) using the experimental test bed (*TripAdvisor*) we developed.

Our model has focused on the design and development of an extended generative model, rather than on inference techniques for this model, for which we chose to use Gibbs sampling for ease of implementation (and parallelization). As with Gibbs sampling-based inference for traditional HDP, the main limitation of our system implementation is its lack of scalability. Our continuing work includes investigating and developing methods for approximation of this model by variational inference.

Broader applications of our inferential model thus include the discovery of new aspects not previously defined for a text corpus such as a collection of reviews. Additionally, the ability to track the evolution of aspect-level sentiments and topics over time is an important area of potential future work.

Our model requires some prior knowledge of sentiment words for initialization. However, this prior knowledge does not need to be very accurate. In the learning process, it can automatically update word tokens to different sentiment label in each topic, and is also robust enough to correct mistakes in prior knowledge.

5.2 `HDPauthor` Model

We also presented a HDP-based hierarchical, nonparametric Bayesian generative model for author-topic hybrid learning, called `HDPauthor`. This model represents each author as a Dirichlet process of global topics, and represents each document as a mixture of these Dirichlet processes of its authors. This model concurrently learns not only the topic interests of authors and the topic distribution of documents as classical topic models, but also the author contributions for documents. It also preserves the benefits of the nonparametric Bayesian hierarchical topic model. Our model uses a purely unsupervised learning

methodology; it requires neither knowledge about documents nor data about authors.

A key novel contribution of our `HDPauthor` model is our ability to represent each document, each author, and global topics as Dirichlet processes, or mixtures of Dirichlet processes. Therefore, none of them suffers from restrictions on the number of topic components that the user should define beforehand for all other LDA-based hybrid models[40]. Thus, the emergence of new topic components and fading out of old topic components can be easily detected and accounted for using our framework.

Our model can be directly applied to document retrieval tasks. Other applications of our model include searching, or grouping of authors, based on topic distribution vectors learned for each author in corpus. The contribution of authors can also be inferred from our model, which can be used for author ranking. Our model can also facilitate to build more sophisticated models for disambiguity of different authors identities with same names, and detection of different author names for same author identity.

5.3 Future Work

In future work, there are several directions that I would like to explore:

1. Numerical sentiment strength learning. While our model treats sentiment label as discrete values from $\{positive, negative, neutral\}$ set, we may consider to add numerical sentiment score for words as indication of strength of sentiment polarity. For example, while *"good"* might be assigned to 2.0 as a mildly positive word, *"fabulous"* might be assigned to 4.0 as a strongly positive word. This strength can be automatically learned from the model, which helps us to quantify the strength of sentiment polarity in text.

2. The development and widespread use of distributed data processing frameworks such as Hadoop[55], Spark[56], etc. gives us several options using which we can develop distributed and parallel Gibbs sampling inference methods for our `HDPsent` and `HDPauthor`

model. There is already some significant work on distributed learning of HDP[57][58][59]. This framework would help us to accelerate our learning process on huge data set. However, this parallel inference method would involve delicate updating of global parameters, fast global combination of new topics from different local working nodes, and some other issues introduced by parallel learning.

3. A variational approximate inference[15][60] approach for our models. Although we use Gibbs sampling as inference technique for model learning, we can study and develop variation inference method for approximate inference also. While the Gibbs sampling method is more straightforward and easier to translate from a mathematical model into a procedural implementation, variational approximate inference for HDP model[61] is more challenging to perform[62], but is more efficient and converges more quickly. By working out a variational inference method for our model, we can more easily apply it to large-scale data.

4. Temporal analysis of topic interest shift for authors, while sentiments shift on same topics. Using timestamps such as the publication dates of papers, we can construct a temporal learning model based on our static document and author topic mixture model to learn the shift of topic interests of authors along a timeline. We can also learn the overall topic shift across the entire research area. With timestamped data such as the text of news comments, or blog articles, we can also be able to observe sentiment change or trends in people's opinions on the same topic along a timeline. This might also help us to make predictions about voting results in politics. Dynamic topic learning can be adapted from both Gibbs sampling-based learning algorithms[63] and variational inference-based learning algorithms[16][64], and in discrete-time[65] or continuous-time[66] formats.

5. Author disambiguation[67][41] is also an interesting topic to explore. In our model, we have no capability to differentiate authors with the same presented name - that is,

the same rendered or recorded name. We are also not able to identify the same author using different name presentations. An author disambiguation algorithm can be developed from our model using the topic similarity matrix learned from our model, along with co-author information.

6. The combination of the `HDPauthor` model with a citation network[68 50] can help us to construct a better model for author and document retrieval model. Our `HDPauthor` model only learns mixing proportion of authors in each document, which can be deemed as the "quantity" of each author's work, while the citation network can help us to analyze the "quality" of authors' work. If we can build a mixture model learning both "quantity" and "quality" of authors and their works, we then should be able to get a better retrieval performance for document and author search tasks.

Bibliography

[1] Nils Lid Hjort, Chris Holmes, Peter Müller, and Stephen G Walker. *Bayesian nonparametrics*, volume 28. Cambridge University Press, 2010.

[2] Peter Müller and Fernando A Quintana. Nonparametric bayesian data analysis. *Statistical science*, pages 95–110, 2004.

[3] 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.

[4] Susan T Dumais. Latent semantic analysis. *Annual review of information science and technology*, 38(1):188–230, 2004.

[5] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

[6] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[7] Mark Steyvers, Padhraic Smyth, Michal Rosen-Zvi, and Thomas Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 306–315. ACM, 2004.

[8] Nan Du, Bin Wu, Xin Pei, Bai Wang, and Liutong Xu. Community detection in large-scale social networks. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 workshop on Web mining and social network analysis*, pages 16–25. ACM, 2007.

[9] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.

[10] Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

[11] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

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

[13] Yee W Teh, David Newman, and Max Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *Advances in neural information processing systems*, pages 1353–1360, 2006.

[14] Tom Griffiths. Gibbs sampling in the generative model of latent dirichlet allocation. 2002.

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

[16] Chong Wang, John W Paisley, and David M Blei. Online variational inference for the hierarchical dirichlet process. In *International conference on artificial intelligence and statistics*, pages 752–760, 2011.

[17] Chong Wang Xi Chen and Alex Smola Eric P Xing. Variance reduction for stochastic variational inference.

[18] Hemant Ishwaran and Lancelot F James. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453), 2001.

[19] John Paisley. A simple proof of the stick-breaking construction of the dirichlet process.

[20] Yee Whye Teh. A tutorial on dirichlet processes and hierarchical dirichlet processes. *Tutorial at Machine Learning Summer School, Tübingen*, 2007.

[21] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[22] Walter R Gilks. *Markov chain monte carlo*. Wiley Online Library, 2005.

[23] Christian Robert and George Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[24] Alberto Leon-Garcia. Probability and random processes. *Addison Wesley, Table*, 3: 126–127, 1989.

[25] Sean P Meyn and Richard L Tweedie. *Markov chains and stochastic stability*. Springer Science & Business Media, 2012.

[26] Siddhartha Chib and Edward Greenberg. Understanding the metropolis-hastings algorithm. *The american statistician*, 49(4):327–335, 1995.

[27] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[28] Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012.

[29] Chenghua Lin and Yulan He. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 375–384. ACM, 2009.

[30] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180. ACM, 2007.

[31] Wesam Elshamy, Doina Caragea, and William Hsu. Ksu kdd: Word sense induction by clustering in topic space. In *Proceedings of the 5th International Workshop on Semantic Evaluation, Uppsala, Sweden, July 2010*, pages 367–370. Association for Computational Linguistics, 2010.

[32] Yelp. Yelp's academic dataset. https://www.yelp.com/academic_dataset, 2012.

[33] Ivan Titov and Ryan McDonald. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International Conference on World Wide Web (WWW 2008)*, pages 111–120. ACM Press, 2008.

[34] Zhongwu Zhai, Bing Liu, Hua Xu, and Peifa Jia. Constrained lda for grouping product features in opinion mining. lecture notes in computer science. In *Lecture Notes in Computer Science*, pages 448–459. Springer, 2011.

[35] Hanna M Wallach, David Minmo, and Andrew McCallum. Rethinking lda: Why priors matter. 2009.

[36] Steve Waterhouse, David MacKay, Tony Robinson, et al. Bayesian methods for mixtures of experts. *Advances in neural information processing systems*, pages 351–357, 1996.

[37] Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.

[38] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pages 487–494. AUAI Press, 2004.

[39] David Mimno and Andrew McCallum. Topic models conditioned on arbitrary features with dirichlet-multinomial regression. *arXiv preprint arXiv:1206.3278*, 2012.

[40] Michal Rosen-Zvi, Chaitanya Chemudugunta, Thomas Griffiths, Padhraic Smyth, and Mark Steyvers. Learning author-topic models from text corpora. *ACM Transactions on Information Systems (TOIS)*, 28(1):4, 2010.

[41] Andrew M Dai and Amos J Storkey. Author disambiguation: a nonparametric topic and co-authorship model. In *NIPS Workshop on Applications for Topic Models Text and Beyond*, pages 1–4, 2009.

[42] Andrew M Dai and Amos J Storkey. The grouped author-topic model for unsupervised entity resolution. In *Artificial Neural Networks and Machine Learning–ICANN 2011*, pages 241–249. Springer, 2011.

[43] Yoseph Barash and Nir Friedman. Context-specific bayesian clustering for gene expression data. *Journal of Computational Biology*, 9(2):169–191, 2002.

[44] Hongning Wang, Yue Lu, and Chengxiang Zhai. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 783–792. ACM, 2010.

[45] Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1105–1112. ACM, 2009.

[46] Hao Wang and Martin Ester. A sentiment-aligned topic model for product aspect rating prediction.

[47] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In

Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages 55–60, 2014.

[48] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.

[49] Kristina Toutanova and Christopher D Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 13*, pages 63–70. Association for Computational Linguistics, 2000.

[50] Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 990–998. ACM, 2008.

[51] K Sparck Jones, Steve Walker, and Stephen E. Robertson. A probabilistic model of information retrieval: development and comparative experiments: Part 2. *Information Processing & Management*, 36(6):809–840, 2000.

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

[53] Amit Singhal. Modern information retrieval: A brief overview. *IEEE Data Eng. Bull.*, 24(4):35–43, 2001.

[54] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP*

Frameworks, pages 45–50, Valletta, Malta, May 2010. ELRA. http://is.muni.cz/publication/884893/en.

[55] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. The hadoop distributed file system. In *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, pages 1–10. IEEE, 2010.

[56] Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin, Scott Shenker, and Ion Stoica. Spark: cluster computing with working sets. In *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*, volume 10, page 10, 2010.

[57] David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed algorithms for topic models. *The Journal of Machine Learning Research*, 10:1801–1828, 2009.

[58] Sinead Williamson, Avinava Dubey, and Eric Xing. Parallel {M} arkov chain {M} onte {C} arlo for nonparametric mixture models. In *Proceedings of the 30th International Conference on Machine Learning*, pages 98–106, 2013.

[59] Alexander Smola and Shravan Narayanamurthy. An architecture for parallel topic models. *Proceedings of the VLDB Endowment*, 3(1-2):703–710, 2010.

[60] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[61] Yee W Teh, Kenichi Kurihara, and Max Welling. Collapsed variational inference for hdp. In *Advances in neural information processing systems*, pages 1481–1488, 2007.

[62] Samuel Gershman, Matt Hoffman, and David Blei. Nonparametric variational inference. *arXiv preprint arXiv:1206.4665*, 2012.

[63] Amr Ahmed and Eric P Xing. Timeline: A dynamic hierarchical dirichlet process

model for recovering birth/death and evolution of topics in text stream. *arXiv preprint arXiv:1203.3463*, 2012.

[64] Chong Wang and David M Blei. Truncation-free online variational inference for bayesian nonparametric models. In *Advances in neural information processing systems*, pages 413–421, 2012.

[65] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006.

[66] Chong Wang, David Blei, and David Heckerman. Continuous time dynamic topic models. *arXiv preprint arXiv:1206.3298*, 2012.

[67] Yang Song, Jian Huang, Isaac G Councill, Jia Li, and C Lee Giles. Efficient topic-based unsupervised name disambiguation. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries*, pages 342–351. ACM, 2007.

[68] Vladimir Batagelj. Efficient algorithms for citation network analysis. *arXiv preprint cs/0309023*, 2003.