ABSTRACT

Title of dissertation: Models, Inference, and Implementation
for Scalable Probabilistic Models of Text

Ke Zhai, Doctor of Philosophy, 2014

Dissertation directed by: Professor Jordan Boyd-Graber
iSchool, UMIACS

Unsupervised probabilistic Bayesian models are powerful tools for statistical analysis, especially in the area of information retrieval, document analysis and text processing. Despite their success, unsupervised probabilistic Bayesian models are often slow in inference due to inter-entangled mutually dependent latent variables. In addition, the parameter space of these models is usually very large. As the data from various different media sources—for example, internet, electronic books, digital films, etc—become widely accessible, lack of scalability for these unsupervised probabilistic Bayesian models becomes a critical bottleneck.

The primary focus of this dissertation is to speed up the inference process in unsupervised probabilistic Bayesian models. There are two common solutions to scale the algorithm up to large data: parallelization or streaming. The former achieves scalability by distributing the data and the computation to multiple machines. The latter assumes data come in a stream and updates the model gradually after seeing each data observation. It is able to scale to larger datasets because it usually takes

only one pass over the entire data.

In this dissertation, we examine both approaches. We first demonstrate the effectiveness of the parallelization approach on a class of unsupervised Bayesian models—topic models, which are exemplified by *latent Dirichlet allocation* (LDA). We propose a fast parallel implementation using variational inference on the MapReduce framework, referred to as Mr. LDA. We show that parallelization enables topic models to handle significantly larger datasets. We further show that our implementation—unlike highly tuned and specialized implementations—is easily extensible. We demonstrate two extensions possible with this scalable framework: 1) informed priors to guide topic discovery and 2) extracting topics from a multilingual corpus.

We propose polylingual tree-based topic models to infer topics in multilingual corpora. We then propose three different inference methods to infer the latent variables. We examine the effectiveness of different inference methods on the task of machine translation in which we use the proposed model to extract domain knowledge that considers both source and target languages. We apply it on a large collection of aligned Chinese-English sentences and show that our model yields significant improvement on BLEU score over strong baselines.

Other than parallelization, another approach to deal with scalability is to learn parameters in an online streaming setting. Although many online algorithms have been proposed for LDA, they all overlook a fundamental but challenging problem—the vocabulary is constantly evolving over time. To address this problem, we propose an online LDA with infinite vocabulary—infvoc LDA. We derive online hybrid inference for our model and propose heuristics to dynamically order, expand, and contract the set of words in our vocabulary. We show that our algorithm is able to discover better topics by incorporating new words into the vocabulary and constantly refining the topics over time.

In addition to LDA, we also show generality of the online hybrid inference framework by applying it to adaptor grammars, which are a broader class of models subsuming LDA. With proper grammar rules, it simplifies to the exact LDA model, however, it provides more flexibility to alter or extend LDA with different grammar rules. We develop online hybrid inference for adaptor grammar, and show that our method discovers high-quality structure more quickly than both MCMC and variational inference methods.

Models, Inference, and Implementation for Scalable Probabilistic Models of Text

by

Ke Zhai

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2014

Preliminary Examination Committee:
Professor Jordan Boyd-Graber (Advisor)
Professor Hal Daumé III (Committee Member)
Professor Ramani Duraiswami (Committee Member)
Professor Jimmy Lin (Committee Member)
Professor Philip Resnik (Department's Representative)

Dedication

This thesis is lovingly dedicated to my father Fushan, my mother Rongxian and my wife Yuening. Their support, encouragement, and constant love have sustained me throughout my life.

For all the laughter and tears.

Acknowledgments

I feel very lucky to spend five years in the Department of Computer Science, University of Maryland. I owe my gratitude to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost I would like to thank my advisor, Professor Jordan Boyd-Graber for his excellent advices and thorough help during my study. He gives me an invaluable opportunity to work on challenging and extremely interesting projects over the past four years. He has always made himself available for help and advice and there has never been an occasion when I have knocked on his door and he has not given me time. He works closely with me for all deadlines, regardless of weekends or midnight. He is very patient on helping me preparing slides and talks. I am indeed grateful for his mentoring on trustworthy attitude in research and faithful devotion in science. It has been a great pleasure to work with and learn from such an extraordinary individual.

I would also like to thank my co-advisor, Professor Jimmy Lin. Without his expertise in large scale distributed computing and advance knowledge in cutting-edge technologies, this thesis would have been a distant dream. Thanks are due to Professor Philip Resnik, Professor Hal Daumé III, and Professor Ramani Duraiswami for agreeing to serve on my thesis committee and for sparing their invaluable time reviewing this manuscript.

My colleagues at the *Computational Linguistics and Information Processing* (CLIP) lab have also enriched my graduate life in many ways and deserve a special mention. I am sincerely grateful for their open-mind, creativity, collaboration and willing to help. Ke Wu, an extremely smart and very responsible close friend, helped me a lot on the adaptor grammar and machine translation projects. He also collaborated with me on several course works, and shared many interesting ideas on good coding skills. Dr. Nima Asadi and Mohamad Alkhouja, wonderful collaborators, helped me on implementing informed prior extension and running some baseline experiments during the Mr. LDA project. Dr. Vladimir Eidelman, a patient senior Ph.D. student, provided help by setting up the experiment pipeline during the machine translation project.

I would also like to thank the faculty members associated with CLIP lab, including Professor Louiqa Raschid, Professor Naomi Feldman and Professor Douglas W. Oard for their thorough comments and helpful suggestions. Thank Joe Webster and fellow technicians in *University of Maryland Institute for Advanced Computer Studies* (UMIACS) for the prompt supports and comprehensive service. In addition, my interactions with other fellow CLIP lab mates (in alphabetical order of last name and then first name): Snigdha Chaturvedi, Leonardo Claudino, Dr. Amit Goyal, Alvin Grissom II, He He, Hua He, Mohit Iyyer, Dr. Jagadeesh Jagarlamudi, Dr. Jiarong Jiang, Dr. Junhui Li, Dr. Taesun Moon, John Morgan, Thang Nguyen, Viet-An Nguyen, Greg Sanders, Dr. Ferhan Ture have been very fruitful and enjoyable. I am indeed fortunate to be part of the CLIP and UMIACS.

This thesis cannot be done without the help from outside of Maryland. I also want to take this chance to thank my mentor Dr. Jason D. Williams in Microsoft Research at Redmond, Professor Sinead Williamson in University of Texas at Austin, Professor Shay B. Cohen in University of Edinburgh, my mentor Dr. Prasad Chalasani in Yahoo! labs at New York. I would also like to thank my fellow colleagues and friends during internship (in alphabetical order of last name and then first name): Xi Chen, Chang Liu, Dr. Huan Wang.

I would also like to acknowledge help and support from some staff members in the department: Jennifer Story, Fatima Bangura and Arlene E. Schenk.

My dearest friends—who are like family members to me—have been a crucial factor in finishing my graduate study smoothly. I would like to express my gratitude to (in alphabetical order of last name and then first name) Xiqun Chen, Xiaojie Cong, Xiang He, Chen Huang, Mingyang Ji, Dandan Li, Taotao Liu, Yangwen Liu, Teng Long, Yijing Lu, Bo Sun, Yong Sun, Li Wan, Zheng Zhang, Qian Zhou, Kangyuan Zhu for their friendship and support. Thank them all for their care and for always being there for me.

I owe my deepest thanks to my parents, who have always stood by me and guided me through my career, and have pulled me through against impossible odds at times. In addition, I want to thank my lovely wife Yuening, who is truly my soul mate, both for life and work. I feel extremely lucky to meet and fall in love with her during my graduate life. I also want to thank all my other relatives, who have been very supportive during my study. Words cannot express the gratitude I owe to them all.

Finally, it is impossible to remember all, and I apologize to those I have inadvertently left out.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

One central goal of computer science is to manage a large and growing collection of data, and to extract meaningful information out of it. As the data from internet, magazines, books, digital films, and all other media sources become widely accessible, they require more effective methods to discover useful patterns and valuable information from them. To understand these data cannot solely depend on human annotation. One popular technique for navigating these large unannotated data is to use statistical modeling.

In this work, we primarily focus on the problem of modeling text corpora. Namely, we want to find a set of patterns of short descriptions (namely, the *topic*) among a collection of *documents*. Many statistical approaches have been proposed to address this problem, i.e., *latent semantic indexing* (Deerwester et al., 1990; Papadimitriou et al., 1998, LSI) and *probabilistic latent semantic indexing* (Hofmann, 1999, pLSI). Among all of these, one of the most widely used statistical frameworks for navigating large unannotated document collections is topic models, which are exemplified by *latent Dirichlet allocation* Blei et al. (2003, LDA), as a generative model for document-centric corpora.

LDA is a powerful tool for statistical analysis of document collections and other discrete data. It assumes that the words of each document arise from a mixture of topics, each of which is a multinomial distribution over the vocabulary. LDA is completely unsupervised, i.e. requires no human annotation, and discovers the thematic trends in a corpus. In addition to capturing which topics exist in a corpus, LDA also associates documents with these topics.

The problem of estimating the latent parameters in probabilistic Bayesian models is referred to *inference*. Although several inference algorithms have been proposed, many of them are slow. Inference speed is certainly a critical bottleneck of many probabilistic Bayesian models, in particular, the LDA and its variants. This certainly does not meet the needs of industry. For example, up till Jun 2014, Google processes about 100PB (petabytes) and stores additional 15EB (exabytes) data per day.[1] In such "large data" settings, slow inference prevents probabilistic Bayesian models from being used in industrial engineering applications as well as academia research projects.

There are two natural ways to deal with this problem. One is to parallelize the algorithm by dividing the computations across multiple machines. This reduces the total running time of the algorithm. Another solution is to stream the algorithm in an online setting and gradually update the parameters over time, so that the algorithm would only take one or few passes over the entire dataset.

Parallelization is a common technique that is often used to speed up an algorithm. It splits the computational and memory requirements of an algorithm onto multiple machines. By investing more computational power and utilize more hardware resources, one shall expect to speed-up the inference process of Bayesian

[1] `http://followthedata.wordpress.com/2014/06/24/data-size-estimates/`

models significantly. This method is particularly appealing to industrial applications, especially with well-designed frameworks and well-deployed computational resources. For example, distributed file systems like Google file system (Ghemawat et al., 2003) or Hadoop (White, 2010) are used to store and backup data. Large-scale computation frameworks like MapReduce (Dean & Ghemawat, 2004), GPU (Kirk & Hwu, 2010) or Pregel (Malewicz et al., 2010) provide generic mechanisms to speed up algorithms. In addition, distributed databases and processing unit like Hive (Thusoo et al., 2009) are used for fast queries and efficient data management.

Another approach is to stream Bayesian statistical models online. Instead of batch learning, which loads the entire dataset and updates the parameters, online learning processes data piece by piece and update the model incrementally after every iteration. After processing a significant size of the data, one shall expect the model to converge. The online approach usually takes only one pass over the entire dataset and assumes past data are no longer available. This approach fits well into industrial applications as new data constantly arrive. Another advantage of doing this, as useful add-ons to the model, one can likely to extract additional information throughout time, for example, the topic shift of the corpus, the evolution of vocabularies, etc. This is particularly useful in time-dependent or chronological-ordered data, for example, finding research topic momentum in scientific document collections from 1950's till now.

1.1 Overview and Organization

In this section, we describe the general organization, and list down our contributions throughout this dissertation. Our main contributions in this thesis lay across three difference aspects—model, inference and implementation.

- Contribution to **model** refers to the proposal of a new probabilistic Bayesian model for the data.
- Contribution to **inference** refers to the development and derivation of new inference methods for an either new or existing model.
- Contribution to **implementation** refers to the development and release of new scalable and fast implementation about an inference technique or model.

We structure the remainder of this dissertation as follows and **highlight our contributions in bold**.

Chapter 2 briefly reviews latent Dirichlet allocation topic models and explains variational inference technique in detail. We show full derivations of the updates for variational expectation maximization algorithm.

In Chapter 3, we **improve the implementation** of existing models and inference techniques using MapReduce—Mr. LDA. As opposed to other techniques which use Gibbs sampling, our proposed framework is based on variational inference, which easily fits into a distributed environment. We compare the scalability of Mr. LDA against Mahout (Foundation et al., 2010), an existing large scale topic modeling package. We show that Mr. LDA out-performs Mahout both in running time and held-out likelihood. More importantly, our variational implementation—unlike highly tuned and specialized implementations based on Gibbs sampling—is easily extensible. We also demonstrate two extensions of our framework possible with this scalable framework: informed priors to guide topic discovery and extracting topics from a multilingual corpus.

In Chapter 4, we first review MCMC inference (Griffiths & Steyvers, 2004). We also discuss the hybrid inference mode (Mimno et al., 2012), which interleaves MCMC sampling *inside* variational inference that creates sparse sufficient statistics to reduce the memory and time requirement.

In Chapter 5, we 1) **propose novel polylingual tree-based topic models** to infer topics in multilingual environment, 2) **derive three different inference schemes** to infer latent variables in the model, and 3) **implement our model using MapReduce** to scale up to large datasets and evaluate it on a downstream task of statistical machine translation. Previous work uses only the source language and completely ignores the target language, which can disambiguate domains. Our proposed polylingual tree-based topic models consider both source and target languages. We show that our proposed model is able to infer better translation domains and improve the translation quality. We evaluate our model on a Chinese to English translation task and yield significant improvement over strong baselines.

One other approach to scale up an algorithm is to stream the data and update the parameters online.

In Chapter 6, we review the online variational inference for topic models (Hoffman et al., 2010). We then discuss the Dirichlet process (Ferguson, 1973) and its generalization—Pitman-Yor process (Pitman & Yor, 1997). In addition, we review the truncation free updates (Wang & Blei, 2012) for Bayesian nonparametric distribution.

In Chapter 7, we focus on online streaming updates for topic models. We 1) **propose a novel online topic model with infinite vocabulary**, 2) **derive online hybrid inference** for our proposed model, and 3) demonstrate our **implementation** is able to incorporate new words into vocabulary and refine topics over time. Unlike all past online approaches, our model addresses a challenge, but often overlooked problem—the vocabulary is constantly changing and evolving throughout time. Vanilla LDA assumes a topic is drawn from a finite Dirichlet distribution, i.e., the vocabulary is fixed. This assumption precludes words being added or dropped over time. Particularly in online cases, this is neither reasonable nor appealing. There are many reasons immutable vocabularies do not make sense. For example, new words ("crowdsource") are invented or words cross languages ("Gangnam" from Korean to English) are introduced, or words ("whan that Aprill"[2]) are outdated. To be flexible, online topic models must be able to capture the invention and deletion of a word in the vocabulary. Essentially, a better alternative is to draw the topic—distribution over vocabulary—from a Dirichlet process Ferguson (1973), which is a nonparametric extension of Dirichlet distribution and has supports over possibly infinite number of atoms. In Chapter 7, we use the Dirichlet process as the topic prior and present *online topic models with infinite vocabulary*, which is a Bayesian nonparametric extension of online LDA.

[2] from the book *Tales of Caunterbury*

An alternative approach to address the above problem is adaptor grammars (Johnson et al., 2007). Adaptor grammars are appealing because they are very flexible in prototyping different probabilistic Bayesian models, for example, Johnson (2010) use the adaptor grammar to implement topic models. Adaptor grammars break the strong independence assumptions of typical grammar models—particularly *probabilistic context free grammar* (Stolcke, 1995, PCFG). The model can be viewed as a nonparametric extension to PCFG. However, the weaker statistical independence assumptions that adaptor grammars make come at the cost of expensive inference.

In Chapter 8, we show the generality of our online hybrid inference framework to adaptor grammars. We 1) **develop online hybrid inference** for an existing Bayesian nonparametric model—adaptor grammars, and 2) demonstrate our **implementation** is able to scale to significant larger datasets than past approaches. Our online hybrid inference algorithm processes examples in small batches in a streaming manner. As more data are observed, our approach is able to expand, adjust and prune the sets of adapted grammar rules over time, which obviates the need for expensive preprocessing required by previous approaches. This also makes our algorithm appealing to much larger datasets.

Finally, in Chapter 9, we summarize our contributions, conclude this dissertation and discuss several possible future research directions.

	Chapter 3	Chapter 5	Chapter 7	Chapter 8
Task	Topic Models	Topic Models	Topic Models	Adaptor Grammar
Approach	Parallelization	Parallelization	Online Streaming	Online Streaming
Model	Blei et al. (2003)	√	√	Johnson et al. (2007)
Inference		√	√	√
Implementation	√	√	√	√

Table 1.1: Main contributions in this dissertation.

1.2 Contributions

In this section, we summarize our main contributions in this thesis as illustrated in Table 1.1. Our contributions in this dissertation are:

- We implement a large-scale distributed topic modeling package—Mr. LDA—in MapReduce on the existing LDA model using previously proposed variational inference method. We show our implementation is able to scale up to significant larger datasets, and yields better performance than Mahout. We demonstrate the flexibility of our implementation using two extensions: 1) informed priors to incorporate human prior knowledge into topic discovery and 2) polylingual LDA for modeling topics in multilingual environment.
- We propose a novel polylingual tree-based topic model to infer topics on multilingual corpora. We further derive three different inference schemes to infer latent variables of our proposed model. We implement our model using MapReduce and scale it up to large datasets. We evaluate the performance of our model on a downstream task of statistical machine translation.
- We propose a novel online topic model which supports possibly infinite vocabulary. We derive online hybrid inference for our proposed model. We demonstrate our implementation is able to incorporate new words into vocabulary and refine

topics over time more effectively than many past approaches.

- We derive the online hybrid inference for an existing Bayesian nonparametric model—adaptor grammars. We show that our implementation is able to scale up to larger datasets than past approaches.

Chapter 2

Background: Topic Models and Variational Inference

Compared to past approaches (Deerwester et al., 1990; Hofmann, 1999), *latent Dirichlet allocation* (Blei et al., 2003, LDA), discovers a set of topics, which are distributions over all words. These semantic coherent topics describe the main themes of a corpus. Besides, it also discovers the topic proportion—a distribution over all topics—for each document, from which we know the main themes of a document. In the rest of this chapter, we first review some domains and applications that use LDA in Section 2.1. We then describe the generative story of LDA in Section 2.2. Finally, in Section 2.3, we overview two popular inference techniques on LDA.

2.1 Applications and Domains

LDA has been successfully applied to many applications in modeling textuary datasets. For example, in the field of computational linguistics, Griffiths et al. (2005) explore a composite framework based on *hidden Markov model* (HMM) and LDA to jointly model *part-of-speech* (POS) tags and topics. Boyd-Graber & Resnik (2010) relax the bag-of-words assumption and incorporate syntax structure into LDA. Toutanova & Johnson (2008) propose a Bayesian LDA-based model for the task of POS tagging in a semi-supervised environment.

In field of information retrieval, Bhattacharya & Getoor (2006) propose a

LDA-based model for unsupervised entity resolution. Shu et al. (2009) extend the LDA model to discover named entities in scientific journal automatically. Kataria et al. (2011) add external domain knowledge extracted from wikipedia into LDA and propose a hierarchical semi-supervised model for name entity resolution. Wei & Croft (2006) use the topics extracted from LDA as document features on the task of text retrieval. Zhang et al. (2007) formulate a social network into textuary corpus using connected arcs as vocabulary and apply LDA to discover community structures.

In addition, LDA is popular in humanity research, literature study, politics and cognition science, for example, understand scientific ideas (Hall et al., 2008), discover political perspectives (Paul & Girju, 2010) and understand the connection between Bayesian models and cognition (Landauer et al., 2006; Griffiths et al., 2007), etc. It is useful in many application, such as document classification, or revealing latent structures and hidden relationship between documents and trends.

Although our primary focus in this thesis is on text corpora and collection of discrete data, LDA is widely used in other domains in computer science as well, such as microarray experiments (Perina et al., 2010), genome clustering (Falush et al., 2003) and discover patterns in population genetics (Shringarpure & Xing, 2008) in the field of computational biology.

In the field of computer vision, Sivic et al. (2005) extract local *scale-invariant features* (Lowe, 1999, SIFT), which are referred as the "visual words", and apply LDA to discover object categories in image datasets. Blei & Jordan (2003) extend the LDA framework and incorporate correlation between images and human annotated tags. Li Fei-Fei & Perona (2005) use LDA to cluster the features of images (referred as "codewords") and categorize natural scenes in image datasets. Wang et al. (2009a) further extend these two works and jointly model the images, class labels and human annotations.

Despite the successes of topic models, they suffer from slow inference, especially in the case of big data. In this thesis, we are going to discuss different approaches to scale up topic models to large datasets.

2.2 Latent Dirichlet Allocation

We review the generative process of vanilla LDA with K topics of V words (Blei et al. (2003) and Griffiths & Steyvers (2004) offer more thorough reviews). Let us refer the hyperparameters of the Dirichlet distributions for documents and topics to α and β respectively.

Latent Dirichlet allocation (Blei et al., 2003, LDA) follows a simple generative process. It assumes K topics, each of which is drawn from a Dirichlet distribution prior, $\boldsymbol{\beta_k} \sim \mathrm{Dir}(\eta), k = \{1, \ldots, K\}$. Given the topics, LDA subsequently generates a document collection as following:

```
for each document $d$ in a corpus $D$ do
    Choose distribution $\theta_d$ over topics from a Dirichlet distribution $\boldsymbol{\theta}_d \sim \mathrm{Dir}(\alpha^\theta)$.
    for each of the $n = 1, \ldots, N_d$ word indexes do
        Choose a topic $z_n$ from the distribution over topics of current document $z_n \sim \mathrm{Mult}(\boldsymbol{\theta}_d)$.
        Choose a word $w_n$ from the appropriate topic's distribution over words $p(w_n|\boldsymbol{\beta}_{z_n})$.
    end for
end for
```

2.3 Inference

Given observed documents, posterior inference discovers the latent variables that best explain an observed corpus. Several inference schemes have been well developed for LDA and its variants. The two most commonly used inference methods for probabilistic Bayesian models are the *Markov chain Monte Carlo* (MCMC) approximation (Neal, 1993; Robert & Casella, 2004) and *variational Bayesian inference* (VB) or *variational inference* for short (Jordan et al., 1999). The former relies on drawing random samples from a Markov chain whose stationary distribution is the posterior of interest. The model is guaranteed to converge—under some additional assumptions on the Markov chain—after significant number of samples. One special case, where the Markov chain is defined by the conditional distribution of each latent variable, is Gibbs sampling (Geman & Geman, 1990). It is widely applied in many Bayesian statistical models (Teh, 2006; Griffiths & Steyvers, 2004; Finkel et al., 2007; Griffiths & Ghahramani, 2005).

The latter, variational inference, approximates a posterior distribution with a simplified *variational distribution.* Typically, the variational distribution is usually from a more manageable family of distributions by assuming independence structure that may not be present in the true posterior. This leads to a distribution that is easier to factorize and/or estimate. The goal is trying to find the “best” fit distribution or settings inside this family by minimizing the *Kullback-Leibler* (KL) divergence between the true posterior and variational distribution. This gives us a lower bound on the model likelihood. We are able to learn the model by maximizing this lower bound. Like MCMC, this approach has also found widespread use in Bayesian statistical models (Blei et al., 2003; Blei & Jordan, 2005; Blei & Lafferty, 2005; Wang & Blei, 2009).

Variational methods enjoy clear convergence criteria, tend to be faster than MCMC in high-dimensional problems and provide particular advantages over MCMC sampling when latent variable pairs are not conjugate. Gibbs sampling requires conjugacy, and other forms of sampling that can handle non-conjugacy, such as Metropolis-Hastings, are much slower than variational methods. In the next section, we are going to discuss about the variational inference updates for LDA.

2.3.1 Variational Inference

Inference in probabilistic models uncovers the latent variables that best explain observed data. Variational methods, based on techniques from statistical physics, use optimization to find a distribution over the latent variables that is close to the posterior log likelihood of interest (Jordan et al., 1999; Wainwright & Jordan, 2008). Variational methods provide effective approximations in topic models and nonparametric Bayesian models (Blei & Jordan, 2005; Teh et al., 2006; Kurihara et al., 2007). In this chapter, we first review the a broad class of variational inference technique in general form. Then, we use LDA as an example to illustrate how to estimate the latent variables using variational inference.

General Variational Inference Variational inference methods cast the inference on a graphical model as an optimization problem. Let us refer all latent variables in the true graphical model as $\boldsymbol{Z}$, parametrized by $\boldsymbol{\Theta}$. With variational methods, we begin by positing a family of distributions $q \in \boldsymbol{Q}$ over the same latent variables $\boldsymbol{Z}$ with a simpler dependency pattern than p. This simpler distribution is called the variational distribution and is parametrized by $\boldsymbol{\Omega}$, a set of variational parameters. Variational methods then minimize the *Kullback-Leibler* (KL) divergence between the variational distribution q and the true posterior. This is equivalent to optimizing the parameters in q and hence find the member in q that is closest to the true posterior distribution.

Evidence Lower Bound Variational inference minimizes the KL divergence between the variational distribution and the posterior distribution. The objective function to optimize in such case is often referred to as the *evidence lower bound*

(ELBO), which is a lower bound on the observed data $\boldsymbol{X}$,

$$\begin{aligned}
\log p(\boldsymbol{X}) &= \log \int p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Theta}) \delta \boldsymbol{Z} \delta \boldsymbol{\Theta} \\
&= \log \int p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Theta}) \frac{q(\boldsymbol{Z}, \boldsymbol{\Omega})}{q(\boldsymbol{Z}, \boldsymbol{\Omega})} \delta \boldsymbol{Z} \delta \boldsymbol{\Theta} \\
&= \log \mathbf{E}_q [\frac{p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Theta})}{q(\boldsymbol{Z}, \boldsymbol{\Omega})}] \\
&\geq \mathbb{E}_q \left[\log \left(p(\boldsymbol{X}|\boldsymbol{Z}) p(\boldsymbol{Z}|\boldsymbol{\Theta})\right)\right] - \mathbb{E}_q \left[\log q(\boldsymbol{Z}|\boldsymbol{\Omega})\right] \\
&= \mathcal{L}. \qquad (2.1)
\end{aligned}$$

The ELBO $\mathcal{L}$ contains two terms. The first term is the expected joint data likelihood under the variational distribution q. The second term is the entropy of the variational distribution q. Both of these two terms are tractable and can be computed easily.

Variational inference fits the variational parameters Ω to tighten this lower bound and thus minimizes the KL divergence between the variational distribution and the true posterior (Jordan et al., 1999; Wainwright & Jordan, 2008).

$$\begin{aligned}
\mathrm{KL}[q(\boldsymbol{Z}, \boldsymbol{\Omega}) || p(\boldsymbol{Z}, \boldsymbol{\Theta}|\boldsymbol{X})] &= \mathbb{E}_q \left[\log q(\boldsymbol{Z}, \boldsymbol{\Omega})\right] - \mathbb{E}_q \left[\log p(\boldsymbol{Z}, \boldsymbol{\Theta}|\boldsymbol{X})\right] \\
&= \mathbb{E}_q \left[\log q(\boldsymbol{Z}, \boldsymbol{\Omega})\right] - \mathbb{E}_q \left[\log p(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Theta})\right] - \log p(\boldsymbol{X}) \\
&= -\mathcal{L} + \text{const}. \qquad (2.2)
\end{aligned}$$

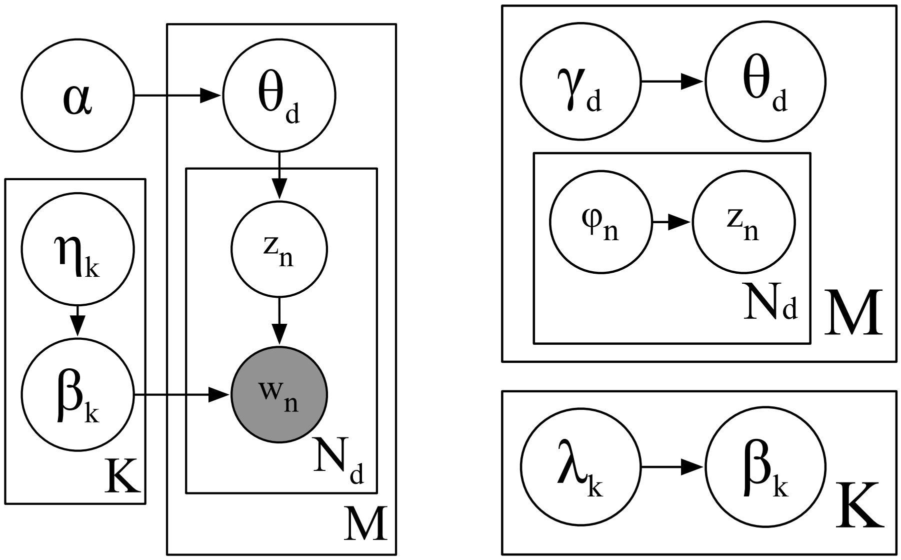
The variational distribution is typically chosen by removing probabilistic dependencies from the true distribution. This makes inference tractable and also induces independence in the variational distribution between latent variables. This independence can be engineered to allow parallelization of independent components across multiple computers.

Mean Field Variational Inference for LDA One simplest variational distribution family is the mean field variational distribution, where all hidden variables are mutually independent and governed by distinct parameters. Such a variational distribution naturally leads to coordinate ascent algorithm, where hidden variables can be updated in turn during the optimization step. In addition, the entropy term in the ELBO factorizes into independent components, and hence is easy to compute.

Let us use LDA as an example to illustrate mean field variational inference. We illustrate the graphical model of LDA in Figure 2.1(a). The observed data $\boldsymbol{X}$ are in the form of M documents, each of which contains N_d words $\{w_1, w_2, \ldots, w_{N_d}\}$. The latent variables $\boldsymbol{Z}$ are corpus-level distributions over vocabularies per topic $\boldsymbol{\beta}$, document-level distributions over topics per document $\boldsymbol{\theta}$ and topic assignment per word $\boldsymbol{z}$.

The mean field variational distribution for LDA is shown in Figure 2.1(b). It assumes each latent variable follows its unique distribution and governed by its own variational parameters. The distribution over vocabulary for topic $\boldsymbol{\beta}_k$ is drawn from a variational Dirichlet distribution with parameter $\boldsymbol{\lambda}$. The distribution over topics for document $\boldsymbol{\theta}_d$ is drawn from a variational Dirichlet distribution governed by $\boldsymbol{\gamma}$ and topic assignment z_{dn} is drawn from a variational multinomial distribution with parameter $\boldsymbol{\phi}$.

The mean field variational distribution q for LDA breaks the connection



(a) LDA

(b) Variational

Figure 2.1: Graphical model of LDA and the mean field variational distribution. Each latent variable, observed datum, and parameter is a node. Lines between represent possible statistical dependence. Shaded nodes are observations; rectangular plates denote replication; and numbers in the bottom right of a plate show how many times plates' contents repeat. In the variational distribution (right), the latent variables θ, β, and z are explained by a simpler, fully factorized distribution with variational parameters γ, λ, and ϕ.

between words and documents

$$q(\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\beta}) = \prod_k \mathrm{Dir}(\boldsymbol{\beta}_k \,|\, \lambda_k) \prod_d \mathrm{Dir}(\boldsymbol{\theta}_d \,|\, \gamma_d) \prod_n \mathrm{Mult}(z_{d,n} \,|\, \phi_{d,n}). \tag{2.3}$$

where Dir(•) represents a Dirichlet distribution, and Mult(•) is a multinomial distribution.

We can then write down the fully factorized evidence lower bound $\mathcal{L}$ according to Eqn. (2.1) as

$$
\begin{aligned}
\mathcal{L} =& \mathbb{E}_q \left[\log \big(p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\beta}) p(\boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{\alpha}, \boldsymbol{\eta}) \big) \right] - \mathbb{E}_q \left[\log q(\boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\beta}|\boldsymbol{\phi}, \boldsymbol{\gamma}, \boldsymbol{\lambda}) \right] \\
=& \mathbb{E}_q \left[\log \big(p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\beta}) p(\boldsymbol{z}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{\alpha}) p(\boldsymbol{\beta}|\boldsymbol{\eta}) \big) \right] - \mathbb{E}_q \left[\log \big(q(\boldsymbol{z}|\boldsymbol{\phi}) q(\boldsymbol{\theta}|\boldsymbol{\gamma}) q(\boldsymbol{\beta}|\boldsymbol{\lambda}) \big) \right] \\
=& \mathbb{E}_q \left[\log p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\beta}) \right] + \mathbb{E}_q \left[\log p(\boldsymbol{z}|\boldsymbol{\theta}) \right] + \mathbb{E}_q \left[\log p(\boldsymbol{\theta}|\boldsymbol{\alpha}) \right] + \mathbb{E}_q \left[\log p(\boldsymbol{\beta}|\boldsymbol{\eta}) \right] \\
& - \mathbb{E}_q \left[\log q(\boldsymbol{z}|\boldsymbol{\phi}) \right] - \mathbb{E}_q \left[\log q(\boldsymbol{\theta}|\boldsymbol{\gamma}) \right] - \mathbb{E}_q \left[\log q(\boldsymbol{\beta}|\boldsymbol{\lambda}) \right] \\
=& \mathbb{E}_q \left[\log \prod_d \prod_n p(w_{dn}|z_{dn}, \boldsymbol{\beta}) \right] + \mathbb{E}_q \left[\log \prod_d \prod_n p(z_{dn}|\boldsymbol{\theta}_d) \right] \\
& + \mathbb{E}_q \left[\log \prod_d p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \right] + \mathbb{E}_q \left[\log \prod_k p(\boldsymbol{\beta}_k|\boldsymbol{\eta}) \right] \\
& - \mathbb{E}_q \left[\log \prod_d q(\boldsymbol{z}_d|\boldsymbol{\phi}_d) \right] - \mathbb{E}_q \left[\log \prod_d q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d) \right] - \mathbb{E}_q \left[\log \prod_k q(\boldsymbol{\beta}_k|\boldsymbol{\lambda}_k) \right] \\
=& \sum_d \sum_n \mathbb{E}_q \left[\log p(w_{dn}|z_{dn}, \boldsymbol{\beta}) \right] + \sum_d \sum_n \mathbb{E}_q \left[\log p(z_{dn}|\boldsymbol{\theta}_d) \right] \\
& + \sum_d \mathbb{E}_q \left[\log p(\boldsymbol{\theta}_d|\boldsymbol{\alpha}) \right] + \sum_k \mathbb{E}_q \left[\log p(\boldsymbol{\beta}_k|\boldsymbol{\eta}) \right] \\
& - \sum_d \mathbb{E}_q \left[\log q(\boldsymbol{z}_d|\boldsymbol{\phi}_d) \right] - \sum_d \mathbb{E}_q \left[\log q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d) \right] - \sum_k \mathbb{E}_q \left[\log q(\boldsymbol{\beta}_k|\boldsymbol{\lambda}_k) \right]. \qquad (2.4)
\end{aligned}
$$

Variational inference then updates all the variational parameters in turn using coordinate ascent. The overall variational inference framework resembles a standard *expectation maximization* (EM) algorithm,[1] and alternates between updating the expectations of the variational distribution q and maximizing the probability of the parameters given the "observed" expected counts.

[1] This is sometimes referred as variational EM algorithm, because it optimizes an objective function described in the space of variational distribution q. It reduces to classical EM algorithm if $p \equiv q$.

2.3.2 The Need for Large-Scale Topic Models

However, both MCMC and VB approaches take significant time to converge. MCMC, in particular Gibbs sampling, constantly samples the value of a latent variable conditioned on all others. While every single sampling iteration in MCMC is fast, it requires a long burn-in period and takes hundreds or even thousands of iterations to reach a relatively stable distribution. Meanwhile, the MCMC approach lacks a clear criterion on the convergence of the sampler. Variational inference, on the other hand, clearly defines the objective function for optimization with a lower bound on the KL divergence, and typically takes dozens of iterations to converge. Nevertheless, every iteration in a variational inference approach usually requires significantly longer running time than MCMC.

The slowness in learning speed certainly become a large bottleneck of many Bayesian statistical models, in particularly, the LDA and its variants. In this thesis, we address this limitation via two different approaches: a distributed approach and an online streaming approach. In Chapter 3, we first discuss a distributed approach by using MapReduce framework to scale up LDA. We then review hybrid inference mode which interleaves MCMC sampling inside the variational inference framework (Chapter 4). In Chapter 5, we focus on a large scale distributed topic models for mutlilingual corpus with hybrid inference by incorporating word and document correlations.

We then change gear to another approach to scale up topic models—online updates—by first reviewing the truncation-free updates in Chapter 6. In Chapter 7, we propose a online topic model which expands the size of vocabulary along inference, using hybrid inference and truncation-free updates. We further apply the online hybrid inference framework to adaptor grammars—an existing flexible nonparametric Bayesian model that generalizes topic models—and demonstrate how to use it to quickly prototyping new nonparametric Bayesian models in Chapter 8.

Chapter 3

Mr. LDA: A Flexible Large Scale Topic Modeling Package

The MapReduce framework for large-scale data processing (Dean & Ghemawat, 2004) is simple to learn but flexible enough to be broadly applicable to many different algorithms. Designed at Google and open-sourced by Yahoo!, Hadoop MapReduce is one of the mainstays of industrial data processing and has also been gaining traction for problems of interest to the academic community such as machine translation (Dyer et al., 2008), language modeling (Brants et al., 2007), and grammar induction (Cohen & Smith, 2009).

In this chapter, we scale up a simple existing topic model—LDA—by parallelizing the algorithm with the MapReduce programming framework (Mr. LDA).[1] Mr. LDA relies on variational inference Blei et al. (2003), as opposed to the prevailing trend of using Gibbs sampling. We argue for using variational inference over MCMC sampling approach in Section 3.1. Section 3.2 describes how variational inference naturally fits into the MapReduce framework. In Section 3.3, we discuss two specific extensions of LDA to demonstrate the flexibility of the proposed framework. These are an informed prior to guide topic discovery and a new inference technique for inferring topics in multilingual corpora (Mimno et al., 2009). Next, we evaluate Mr. LDA's ability to scale in Section 3.4 before summarize this chapter in Section 3.5.

[1] Code is available for download at `http://mrlda.cc`.

3.1 Scaling out LDA

In practice, probabilistic models work by maximizing the log-likelihood of observed data given the structure of an assumed probabilistic model. Less technically, generative models tell a story of how your data came to be with some pieces of the story missing; inference fills in the missing pieces with the best explanation of the missing variables. Because exact inference is often intractable (as it is for LDA), complex models require approximate inference.

3.1.1 Why not Gibbs Sampling?

One of the most widely used approximate inference techniques for such models is *Markov chain Monte Carlo* (MCMC) sampling, where one samples from a Markov chain whose stationary distribution is the posterior of interest (Neal, 1993; Robert & Casella, 2004). Gibbs sampling, where the Markov chain is defined by the conditional distribution of each latent variable, has found widespread use in Bayesian models (Neal, 1993; Teh, 2006; Griffiths & Steyvers, 2004; Finkel et al., 2007). MCMC is a powerful methodology, but it has drawbacks. Convergence of the sampler to its stationary distribution is difficult to diagnose, and sampling algorithms can be slow to converge in high dimensional models (Robert & Casella, 2004).

Blei et al. (2003) present the first approximate inference technique for LDA based on variational methods, but the collapsed Gibbs sampler by Griffiths & Steyvers (2004) has been more popular in the community because it is easier to implement. However, such methods inevitably have intrinsic problems that lead to difficulties in moving to web-scale: shared state, randomness, too many short iterations, and lack of flexibility.

Shared State Unless the probabilistic model allows for discrete segments to be statistically independent of each other, it is difficult to conduct inference in parallel. However, we want models that allow specialization to be shared across many different corpora and documents when necessary, so we typically cannot assume this independence.

At the risk of oversimplifying, collapsed Gibbs sampling for LDA is essentially multiplying the number of occurrences of a topic in a document by the number of times a word type appears in a topic across all documents. The former is a document-specific count, but the latter is shared across the entire corpus. For techniques that scale out collapsed Gibbs sampling for LDA, the major challenge is keeping these second counts for collapsed Gibbs sampling consistent when there is not a shared memory environment.

Newman et al. (2008) consider a variety of methods to achieve consistent counts: creating hierarchical models to view each slice as independent or simply syncing counts in a batch update. Yan et al. (2009) first cleverly partition the data using integer programming (an NP-Hard problem). Wang et al. (2009b) use message passing to ensure that different slices maintain consistent counts. Smola & Narayanamurthy (2010) use a distributed memory system to achieve consistent counts in LDA, and Ahmed et al. (2012) extend the approach more generally to latent variable models. Li et al. (2014) propose to use alias sampling techniques to further reduce the computation complexity.

Gibbs sampling approaches to scaling thus face a difficult dilemma: completely synchronize counts, which can compromise scaling, or allow for inconsistent counts, which could negatively impact the quality of inference. In contrast to engineering work-arounds, variational inference provides a *mathematical* solution of how to scale inference for LDA. By assuming a variational distribution that treats documents as independent, we can parallelize inference without a need for synchronizing counts (as required in collapsed Gibbs sampling).

Randomness By definition, Monte Carlo algorithms depend on randomness. However, MapReduce implementations assume that every step of computation will be the same, no matter where or when it is run. This allows MapReduce to have greater fault-tolerance, running multiple copies of computation subcomponents in case one fails or takes too long. This is, of course, easily fixed (e.g. by seeding a random number generator in a shard-dependent way), but it adds another layer of complication to the algorithm. Variational inference, given an initialization, is deterministic, which is more in line with MapReduce's system for ensuring fault tolerance.

Many Short Iterations A single iteration of Gibbs sampling for LDA with K topics is very quick. For each word, the algorithm performs a simple multiplication to build a sampling distribution of length K, samples from that distribution, and updates an integer vector. In contrast, each iteration of variational inference is difficult; it requires the evaluation of complicated functions that are not simple arithmetic operations directly implemented in an ALU (these are described in Section 3.2).

This does not mean that variational inference is slower, however. Variational inference typically requires dozens of iterations to converge, while Gibbs sampling requires thousands (determining convergence is often more difficult for Gibbs sampling). Moreover, the requirement of Gibbs sampling to keep a consistent state means that there are many more synchronizations required to complete inference, increasing the complexity of the implementation and the communication overhead. In contrast, variational inference requires synchronization only once per iteration (dozens of times for a typical corpus); in a naïve Gibbs sampling implementation, inference requires synchronization after every word in every iteration (potentially billions of times for a moderately-sized corpus).

Mimno et al. (2012) propose a hybrid stochastic inference algorithm for LDA, which benefits from both words. On the local document level, the method uses MCMC sampling to obtain sparse samples for topic distribution per document; and on the corpus level update, it updates the word distribution per topic using variational inference. They also show that this hybrid MCMC-variational inference algorithm yields better performance and significant speed-ups than vanilla variational inference. In Chapter 4, we will discuss this method in detail.

Extension and Flexibility Compared to Mr. LDA, many Gibbs samplers are highly tuned specifically for LDA, which restricts extensions and enhancements, one of the key benefits of the statistical approach. The techniques to improve inference for collapsed Gibbs samplers (Yao et al., 2009) typically reduce flexibility; the factorization of the conditional distribution is limited to LDA's explicit formulation. Adapting such tricks beyond LDA requires repeating the analysis to refactorize the conditional distribution. In Section 3.3.1 we add an informed prior to topics' word distribution, which guides the topics discovered by the framework to psychologically plausible concepts. In Section 3.3.2, we adapt Mr. LDA to learn multilingual topics.

3.1.2 Related Work

Nallapati et al. (2007) extended variational inference for LDA to a parallelized setting. Their implementation uses a master-slave paradigm in a distributed environment, where all the slaves are responsible for the E-step and the master node gathers all the intermediate outputs from the slaves and performs the M-step. While this approach parallelizes the process to a small-scale distributed environment, the final aggregation/merging showed an I/O bottleneck that prevented scaling beyond a handful of slaves because the master has to explicitly read all intermediate results from slaves.

Chu et al. (2007) develop a general and exact technique for parallel programming of a large class of machine learning algorithms for multicore processors. They adapt MapReduce (Dean & Ghemawat, 2004) paradigm—on multiple processors instead of machines—to demonstrate the efficiency of parallelization approach on a variety of learning algorithms, including *logistic regression* (LR), *na ive Bayes* (NB), *support vector machine* (SVM) etc. Although they do not explicitly apply their parallelization methods on LDA, they demonstrate the effectiveness on the general *expectation maximization* (EM) algorithm, which is—as discussed in Chapter 2—the general framework for variational inference. Their experimental results show approximately linear speedup with an increasing number of processors.

Mr. LDA addresses these problems by parallelizing the work done by a single master (a reducer is only responsible for a single topic) and relying on the MapReduce framework, which can efficiently marshal communication between compute nodes. Building on the MapReduce framework also provides advantages for reliability and monitoring not available in an *ad hoc* parallelization framework.

The MapReduce (Dean & Ghemawat, 2004) framework was originally inspired from the map and reduce functions commonly used in functional programming. It adopts a divide-and-conquer approach. Each *mapper* processes a small subset of data and passes the intermediate results as key value pairs to *reducers*. The reducers receive these inputs in sorted order, aggregate them, and produce the final result. In addition to mappers and reducers, the MapReduce framework allows for the definition of *combiners* and *partitioners*. Combiners perform local aggregation on the key value pairs after map function. Combiners help reduce the size of intermediate data transferred and are widely used to optimize a MapReduce process. Partitioners control how messages are routed to reducers.

Mahout (Foundation et al., 2010), an open-source machine learning package, provides a MapReduce implementation of variational inference LDA, but it lacks

	Framework	Inference	Likelihood	Asymmetric α	Hyperparameter	Informed β	Multilingual
Mallet (McCallum, 2002)	Multi-thread	Gibbs	√	√	√	×	√
GPU-LDA (Yan et al., 2009)	GPU	Gibbs & V.B.	√	×	×	×	×
Async-LDA (Asuncion et al., 2008)	Multi-thread	Gibbs	√	×	√	×	×
N.C.L. (Nallapati et al., 2007)	Master-Slave	V.B.	∼	×	×	×	×
pLDA (Wang et al., 2009b)	MPI & MapReduce	Gibbs	∼	×	×	×	×
Y!LDA (Smola & Narayanamurthy, 2010)	Hadoop	Gibbs	√	√	√	×	×
Mahout (Foundation et al., 2010)	MapReduce	V.B.	√	×	×	×	×
Mr. LDA	**MapReduce**	**V.B.**	√	√	√	√	√

Table 3.1: Comparison among different approaches. Mr. LDA supports all of these features, as compared to existing distributed or multi-threaded implementations. (∼ - not available from available documentation.)

features required by mature LDA implementations such as supplying per-document topic distributions and optimizing hyperparameters. Wallach et al. (2009) explain how this is essential for model quality. Without per-document topic distributions, many of the downstream applications of LDA (e.g., document clustering) become more difficult.

Table 3.1 provides a general overview and comparison of features among different approaches for scaling LDA. Mr. LDA is the only implementation which supports all listed capabilities in a distributed environment.

3.2 Mr. LDA

Variational EM alternates between updating the expectations of the variational distribution q and maximizing the probability of the parameters given the "observed"

Algorithm 1 Mapper

```
Input:
KEY - document ID $d \in [1, C]$, where $C = |\mathcal{C}|$.
VALUE - document content.
Output:
KEY - key pair $\langle p_l, p_r \rangle$.
VALUE - value $\sigma'$.
Configure
Load in $\alpha$'s, $\lambda$'s and $\gamma$'s from distributed cache.
Normalize $\lambda$'s for every topic.
Map
Initialize a zero $V \times K$-dimensional matrix $\phi$.
Initialize a zero $K$-dimensional row vector $\sigma$.
Read in document content $\|w_1, w_2, \ldots, w_V\|$
repeat
    for all $v \in [1, V]$ do
        for all $k \in [1, K]$ do
            Update $\phi_{v,k} = \frac{\lambda_{v,k}}{\sum_v \lambda_{v,k}} \cdot \exp \Psi(\gamma_{d,k})$.
        end for
        Normalize $\phi_v$, set $\sigma = \sigma + w_v \phi_{v,*}$
    end for
    Update row vector $\gamma_{d,*} = \alpha + \sigma$.
until convergence
for all $k \in [1, K]$ do
    for all $v \in [1, V]$ do
        Emit $\langle k, v \rangle : w_v \phi_{v,k}$.
    end for
    Emit $\langle \triangle, k \rangle : \left( \Psi(\gamma_{d,k}) - \Psi\left( \sum_{l=1}^{K} \gamma_{d,l} \right) \right)$. {Section 3.2.4}
    Emit $\langle k, d \rangle : \gamma_{d,k}$ to file.
end for
Aggregate $\mathcal{L}$ to global counter. {ELBO, Section 3.2.5}
```

expected counts. The remainder of this chapter focuses on adapting the parameter updates into the MapReduce framework and challenges of working at a large scale. We focus on the primary components of a MapReduce algorithm: the mapper, which processes a single unit of data (in this case, a document); the reducer, which processes a single view of globally shared data (in this case, a topic parameter); the partitioner, which distributes the workload to reducers; and the driver, which controls the overall algorithm. The interconnections between the components of Mr. LDA are depicted in Figure 3.1.

3.2.1 Mapper: Update ϕ and γ

Each document has associated variational parameters γ and ϕ. The mapper computes the updates for these variational parameters and uses them to create the sufficient statistics needed to update the global parameters. In this section, we describe the computation of these variational updates and how they are transmitted to the reducers.

Given a document, the updates for ϕ and γ are

$$\phi_{v,k} \propto \mathbb{E}_q \left[\beta_{v,k}\right] \cdot e^{\Psi(\gamma_k)}, \qquad \gamma_k = \alpha_k + \sum_{v=1}^{V} \phi_{v,k},$$

where $v \in [1, V]$ is the term index and $k \in [1, K]$ is the topic index. In this case, V is the size of the vocabulary $\mathcal{V}$ and K denotes the total number of topics. The expectation of β under q gives an estimate of how compatible a word is with a topic; words highly compatible with a topic will have a larger expected β and thus higher values of ϕ for that topic.

Algorithm 1 illustrates the detailed procedure of the Map function. In the first iteration, mappers initialize variables, e.g. seed λ with the counts of a single document. For the sake of brevity, we omit that step here; in later iterations, global parameters are stored in *distributed cache* – a synchronized read-only memory that is shared among all mappers (White, 2010) – and retrieved prior to mapper execution in a configuration step.

A document is represented as a term frequency sequence $\vec{w} = \|w_1, w_2, \ldots, w_V\|$, where w_i is the corresponding **term frequency in document** d. For ease of notation, we assume the input term frequency vector $\vec{w}$ is associated with all the terms in the vocabulary, i.e., if term t_i does not appear at all in document d, $w_i = 0$.

Because the document variational parameter γ and the word variational parameter ϕ are tightly coupled, we impose a local convergence requirement on γ in the Map function. This means that the mapper alternates between updating γ and ϕ until γ stops changing.

3.2.2 Partitioner: Evenly Distribute Workloads

The Map function in Algorithm 1 emits sufficient statistics for updating the topic variational distribution λ. These sufficient statistics are keyed by a composite key set $\langle p_{\mathsf{left}}, p_{\mathsf{right}} \rangle$. These keys can take two forms: tuple of topic and word identifier or, when the value represents the sufficient statistics for α updating, a unique value $\triangle$ and a topic identifier.

A partitioner is required to ensure that messages from the mappers are sent to the appropriate reducers. Each reducer is responsible for updating the per-topic variational parameter associated with a single topic indexed by k. This is accomplished by ensuring the partitioner sorts on topic only. A consequence of this is that any reducers beyond the number of topics is superfluous. Given that the vast majority of the work is in the mappers, this is typically not an issue for LDA. Algorithm 2 illustrates the pseudo-code of the partitioner.

Algorithm 2 Partitioner

```
Input:
KEY - key pair $\langle p_l, p_r \rangle$.
Partitioner
1: Partition data according to $p_l$ only, ignore $p_r$.
```

Algorithm 3 Reducer

```
Input:
KEY - key pair $\langle p_{\mathsf{left}}, p_{\mathsf{right}} \rangle$.
VALUE - an iterator $\mathcal{I}$ over sequence of values.
Output:
KEY - topic index $k \in [1, K]$.
VALUE - $V$-dimensional column vector $\phi_{*,k}$.
Reduce
1: Compute the sum $\sigma$ over all values in the sequence $\mathcal{I}$. $\sigma$ is un-normalized $\lambda$ if $p_{\mathsf{left}} \neq \triangle$ and $\alpha$ sufficient statistics (refer to Section 3.2.4 for more details) otherwise.
2: Emit $\langle p_{\mathsf{left}}, p_{\mathsf{right}} \rangle : \sigma$.
```

3.2.3 Reducer: Update λ

The Reduce function updates the variational parameter λ for distribution over vocabulary per topic. It requires aggregation over all intermediate ϕ vectors

$$\lambda_{v,k} = \eta_{v,k} + \sum_{d=1}^{C} \left(w_v^{(d)} \phi_{v,k}^{(d)} \right),$$

where $d \in [1, C]$ is the document index and $w_v^{(d)}$ denotes the number of appearances of term v in document d. Similarly, C is the number of documents. Although the variational update for λ does not include a normalization, the expectation $\mathbb{E}_q[\beta]$ requires the λ normalizer. In Mr. LDA, the $\lambda_{v,k}$ parameters are distributed to all mappers, and the normalization is taken care of by the mappers in a configuration step prior to every iteration.

To improve performance, we use combiners to facilitate the aggregation of sufficient statistics in mappers before they are transferred to reducers. This decreases bandwidth and saves the reducer computation.

3.2.4 Driver: Update α

Effective inference of topic models depends on learning not just the latent variables β, θ, and z but also estimating the hyperparameters, particularly α. The α parameter controls the sparsity of topics in the document distribution and is the primary mechanism that differentiates LDA from previous models like pLSI and LSI; not optimizing α risks learning suboptimal topics (Wallach et al., 2009).

Updating hyperparameters is also important from the perspective of equalizing differences between inference techniques; as long as hyperparameters are optimized, there is little difference between the *output* of inference techniques (Asuncion et al., 2009).

Maximizing the global parameters in MapReduce can be handled in a manner analogous to EM (Wolfe et al., 2008); the expected counts (of the variational distribution) generated in many parallel jobs are efficiently aggregated and used to recompute the top-level parameters.

The driver program marshals the entire inference process. On the first iteration, the driver is responsible for initializing all the model parameters (K, V, C, η, α); the number of topics K is user specified; C and V, the number of documents and types, is determined by the data; the initial value of α is specified by the user; and λ is randomly initialized or otherwise seeded.

The driver updates α after each MapReduce iteration. We use a Newton-Raphson method which requires the Hessian matrix and the gradient,

$$\alpha_{\mathsf{new}} = \alpha_{\mathsf{old}} - \mathcal{H}^{-1}(\alpha_{\mathsf{old}}) \cdot g(\alpha_{\mathsf{old}}), \tag{3.1}$$

where the Hessian matrix $\mathcal{H}$ and α gradient are, respectively, as

$$\mathcal{H}(k,l) = \delta(k,l) C \Psi'(\alpha_k) - C\Psi'\left(\textstyle\sum_{l=1}^{K} \alpha_l\right), \tag{3.2}$$

$$g(k) = \underbrace{C\left(\Psi\left(\sum_{l=1}^{K}\alpha_l\right) - \Psi(\alpha_k)\right)}_{\text{computed in driver}} + \underbrace{\sum_{d=1}^{C}\underbrace{\Psi(\gamma_{d,k}) - \Psi\left(\sum_{l=1}^{K}\gamma_{d,l}\right)}_{\text{computed in mapper}}}_{\text{computed in reducer}}. \tag{3.3}$$

The Hessian matrix $\mathcal{H}$ depends entirely on the vector α, which changes during updating α. The gradient g, on the other hand, can be decomposed into two terms: the α-*tokens* (i.e., $\Psi(\sum_{l=1}^{K}\alpha_l) - \Psi(\alpha_k)$) and the γ-tokens (i.e., $\sum_{d=1}^{C}\Psi(\gamma_{d,k}) - \Psi(\sum_{l=1}^{K}\gamma_{d,l})$). We can remove the dependence on the number of documents in the gradient computation by computing the γ-tokens in mappers. This observation allows us to optimize α in the MapReduce environment.

Because LDA is a dimensionality reduction algorithm, there are typically a small number of topics K even for a large document collection. As a result, we can safely assume the dimensionality of α, $\mathcal{H}$, and g are reasonably low, and additional gains come from the diagonal structure of the Hessian (Minka, 2000). Hence, the updating of α is efficient and will not create a bottleneck in the driver.

3.2.5 Likelihood Computation

The driver monitors the ELBO and terminate the inference once it is converged. If not, it restarts the process with another round of mappers and reducers. Computing the ELBO gives us

$$\begin{aligned}\mathcal{L}(\gamma, \phi, \lambda; \alpha, \eta) = & \underbrace{\sum_{d=1}^{C} \Phi(\alpha)}_{\text{driver}} + \underbrace{\sum_{d=1}^{C} (\underbrace{\mathcal{L}_d(\gamma, \phi) + \mathcal{L}_d(\phi) - \Phi(\gamma)}_{\text{computed in mapper}})}_{\text{computed in reducer}} \\ & + \underbrace{\sum_{k=1}^{K} \Phi(\eta_{*,k})}_{\text{driver / constant}} - \underbrace{\sum_{k=1}^{K} \underbrace{\Phi(\lambda_{*,k})}_{\text{reducer}}}_{\text{driver}}\end{aligned} \tag{3.4}$$

where

$$\Phi(\mu) = \log \Gamma \left(\textstyle\sum_{i=1} \mu_i \right) - \sum_{i=1} \log \Gamma \left(\mu_i \right) \tag{3.5}$$

$$+ \sum_{i} \left(\mu_i - 1 \right) \left(\Psi \left(\mu_i \right) - \Psi \left(\textstyle\sum_j \mu_j \right) \right). \tag{3.6}$$

$$\mathcal{L}_d(\gamma, \phi) = \sum_{k=1}^{K} \sum_{v=1}^{V} \phi_{v,k} w_v \left[\Psi \left(\gamma_k \right) - \Psi \left(\textstyle\sum_{i=1}^{K} \gamma_i \right) \right], \tag{3.7}$$

$$\mathcal{L}_d(\phi) = \sum_{v=1}^{V} \sum_{k=1}^{K} \phi_{v,k} \left(\sum_{i=1}^{V} w_i \log \frac{\lambda_{i,k}}{\sum_j \lambda_{j,k}} - \log \phi_{v,k} \right), \tag{3.8}$$

Almost all of the terms that appear in the likelihood term can be computed in mappers; the only term that cannot are the terms that depend on α, which is updated in the driver, and the variational parameter λ, which is shared among all documents. All terms that depend on α can be easily computed in the driver, while the terms that depend on λ can be computed in each reducer.

Thus, computing the total likelihood proceeds as follows: each mapper computes its contribution to the likelihood bound $\mathcal{L}$, and emits a special key that is unique to likelihood bound terms and then aggregated in the reducer; the reducers add topic-specific terms to the likelihood; these final values are then combined with the contribution from α in the driver to compute a final likelihood bound.

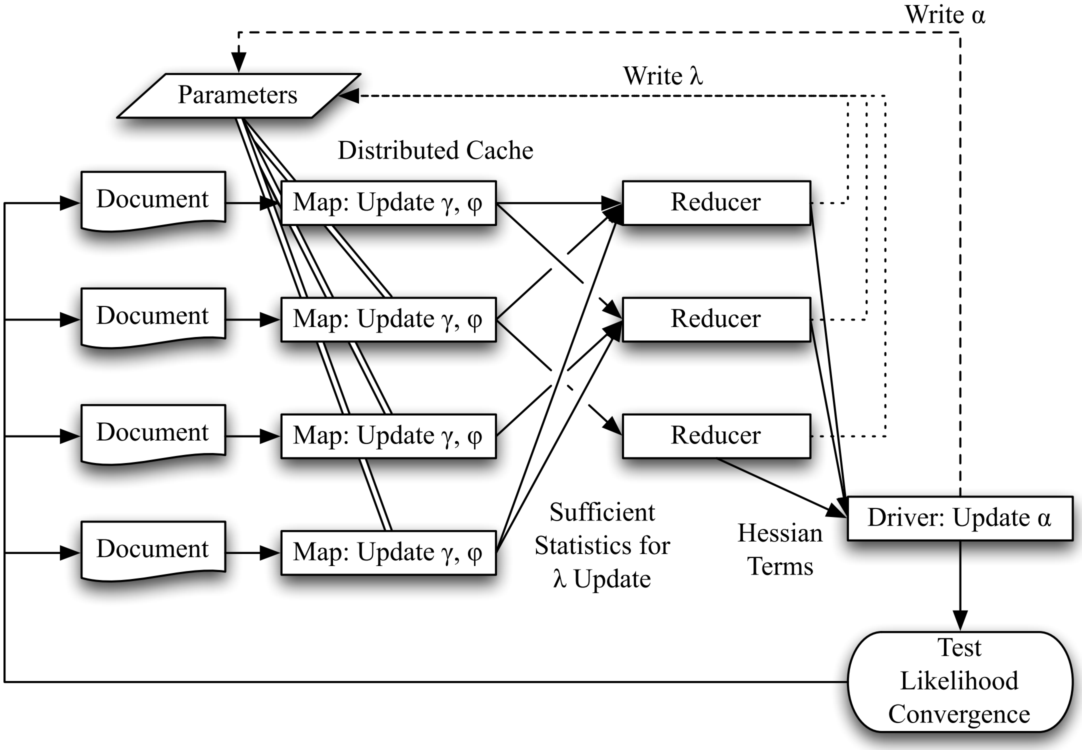


Figure 3.1: Workflow of Mr. LDA. Each iteration is broken into three stages: computing document-specific variational parameters in parallel mappers, computing topic-specific parameters in parallel reducers, and then updating global parameters in the driver, which also monitors convergence of the algorithm. Data flow is managed by the MapReduce framework: sufficient statistics from the mappers are directed to appropriate reducers, and new parameters computed in reducers are distributed to other computation units via the distributed cache.

3.2.6 Structural Optimization

In examining Mr. LDA's performance, the two largest performance limitations were the large number of intermediate values being generated by the mappers and the time it takes for mappers to read in the current variational parameters during during the mapper configuration phase.

Reducer Caching Recall that reducers sum over ϕ contributions and emit the λ variational parameters, but mappers require a normalized form to compute the expectation with of the topic with respect to the variational distribution. To improve the normalization step, we compute the sum of the λ variational parameters in the reducer (Lin & Dyer, 2010; Lin & He, 2009), and then emit this sum before we emit the other λ terms.

Although this requires $O(V)$ additional memory, it is strictly less than the memory required by mappers, so it in practice improves performance by allowing mappers to more quickly begin processing data.

File Merge Loading files in the distributed cache and configuring every mapper and reducer is another bottleneck for this framework. This is especially true if we launch a large number of reducers every iteration — this will result in a large number of small outputs, since Mr. LDA is designed to distribute workload equally. These partial results would waste space if they are significantly smaller than HDFS block size. Moreover, they cause a overhead in file transfer through distributed cache. To alleviate this problem, we merge all relevant output before sending them to the distributed cache for the next iteration.

3.3 Flexibility of Mr. LDA

In this section, we highlight the flexibility of Mr. LDA to accommodate extensions to LDA. These extensions are possible because of the modular nature of Mr. LDA's design.

3.3.1 Informed Prior

The standard practice in topic modeling is to use a same symmetric prior (i.e., $\eta_{v,k}$ is the same for all topics k and words v). However, the model and inference presented in Section 3.2 allows for topics to have different priors. Thus, users can incorporate prior information into the model.

For example, suppose we wanted to discover how different psychological states were expressed in blogs or newspapers. If this were our goal, we might create priors that captured psychological categories to discover how they were expressed in a corpus. The *Linguistic Inquiry and Word Count* (LIWC) dictionary (Pennebaker & Francis, 1999) defines 68 categories encompassing psychological constructs and personal concerns. For example, the *anger* LIWC category includes the words "abuse", "jerk", and "jealous"; the *anxiety* category includes "afraid", "alarm", and "avoid"; and the *negative emotions* category includes "abandon", "maddening", and “sob”. Using this dictionary, we built a prior $\boldsymbol{\eta}$ as follows:

$$\eta_{v,k} = \begin{cases} 10, \text{ if } v \in \text{LIWC category}_k \\ \\ 0.01, \text{ otherwise} \end{cases},$$

where $\eta_{v,k}$ is the informed prior for word v of topic k. This is accomplished via a slight modification of the reducer (i.e., to make it aware of the values of η) and leaving the rest of the system unchanged.

3.3.2 Polylingual LDA

In this section, we demonstrate the flexibility of Mr. LDA by showing how its modular design allows for extending LDA beyond a single language. PolyLDA (Mimno et al., 2009) assumes a document-aligned multilingual corpus. For example, articles in Wikipedia have links to the version of the article in other languages; while the linked documents are ostensibly on the same subject, they are usually not direct translations, and are often written with a culture-specific focus.

PolyLDA assumes that a single document has words in multiple languages, but each document has a common, language agnostic per-document distribution θ (Figure 3.2). Each topic also has different facets for language; these topics end up being consistent because of the links across language encoded in the consistent themes present in documents.

Because of the modular way in which we implemented inference, we can perform multilingual inference by embellishing each data unit with a language identifier l and

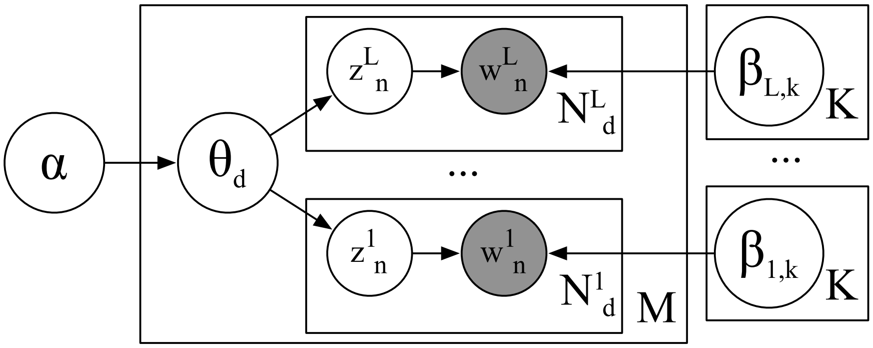


Figure 3.2: Graphical model for polylingual LDA (Mimno et al., 2009). Each document has words in multiple languages. Inference learns the topics across languages that have cooccurring words in the corpus. The modular inference of Mr. LDA allows for inference for this model to be accomplished by the same framework created for monolingual LDA.

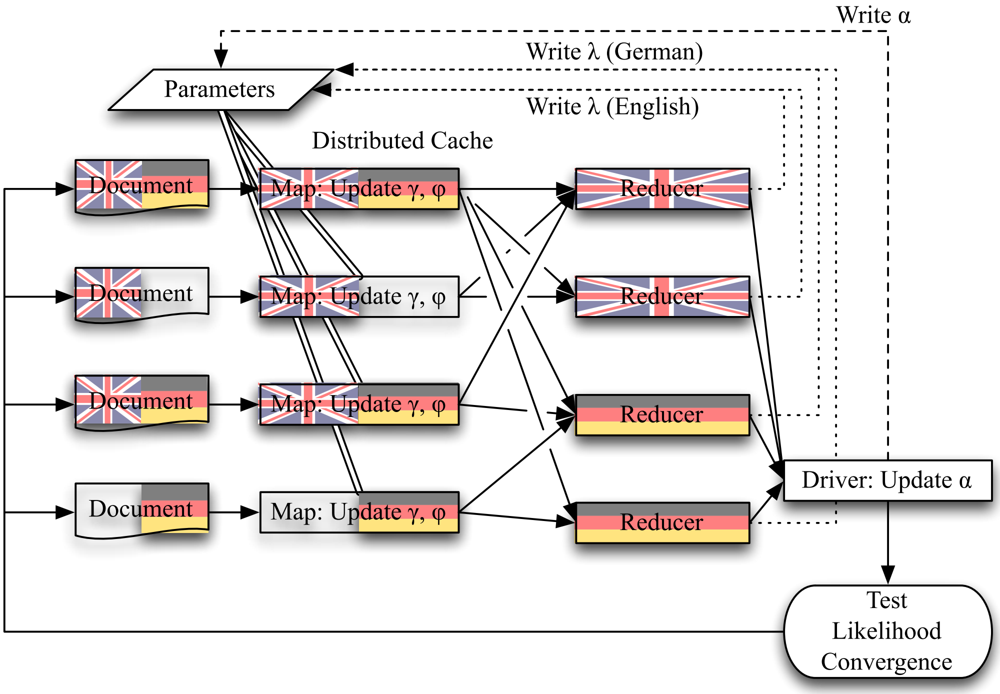


Figure 3.3: Workflow of polylingual LDA. Each iteration is broken into three stages: updating λ happens l times, once for each language, updating ϕ happens using only the relevant language for a word and updating γ happens as usual, combining the contributions of all languages relevant for a document.

change inference as follows:

- Updating λ happens l times, once for each language. The updates for a particular language ignores expected counts of all other languages.
- Updating ϕ happens using only the relevant language for a word.
- Updating γ happens as usual, combining the contributions of all languages relevant for a document.

This is also illustrated in Figure 3.3.

From an implementation perspective, PolyLDA is a collection of monolingual Mr. LDA computations sequenced appropriately. Mr. LDA's approach of taking relatively simple computation units, allowing them to scale, and preserving simple communication between computation units stands in contrast to the design choices made by approaches using Gibbs sampling.

For example, Smola & Narayanamurthy (2010) interleave the topic and document counts during the computation of the conditional distribution using the "binning" approach (Yao et al., 2009). While this improves performance, changing any of the modeling assumptions would potentially break this optimization.

In contrast, Mr. LDA's philosophy allows for easier development of extensions of LDA. While we only discuss two extensions here, other extensions are possible. In Chapter 5, we will demonstrate how to apply polylingual LDA as domain knowledge to statistical machine translation. For example, implementing supervised LDA (Blei & McAuliffe, 2007) only requires changing the computation of ϕ and a regression; the rest of the model is unchanged. Implementing syntactic topic models (Boyd-Graber & Blei, 2008) requires changing the mapper to incorporate syntactic dependencies.

	Affective Processes	Negative Emotions	Positive Emotions	Anxiety	Anger	Sadness	Cognitive Process	Insight	Causation	Discrepancy	Tentative	Certainty
Output from TREC	book	fire	film	al	polic	stock	coalit	un	technolog	pound	hotel	art
	life	hospit	music	arab	drug	cent	elect	bosnia	comput	share	travel	italian
	love	medic	play	israel	arrest	share	polit	serb	research	profit	fish	itali
	like	damag	entertain	palestinian	kill	index	conflict	bosnian	system	dividend	island	artist
	stori	patient	show	isra	prison	rose	anc	herzegovina	electron	group	wine	museum
	man	accid	tv	india	investig	close	think	croatian	scienc	uk	garden	paint
	write	death	calendar	peac	crime	fell	parliament	greek	test	pre	design	exhibit
	read	doctor	movie	islam	attack	profit	poland	yugoslavia	equip	trust	boat	opera
Output from Blog	easili	sorri	lord	bird	iraq	level		god	system	sa	pretty	film
	dare	crappi	prayer	diseas	american	weight		christian	http	ko	davida	actor
	truli	bullshit	pray	shi	countri	disord		church	develop	ang	croydon	robert
	lol	goddamn	merci	infect	militari	moder		jesus	program	pa	crossword	william
	needi	messi	etern	blood	nation	miseri		christ	www	ako	chrono	truli
	jealousi	shitti	truli	snake	unit	lbs		religion	web	en	jigsaw	director
	friendship	bitchi	humbl	anxieti	america	loneli		faith	file	lang	40th	charact
	betray	angri	god	creatur	force	pain		cathol	servic	el	surrey	richard

Table 3.2: 12 topics discovered from TREC (top) and BlogAuthorship (bottom) collection with LIWC-derived informed prior. The model associates TREC documents containing words like “arab”, “israel”, “palestinian” and “peace” with *Anxiety*. In the blog corpus, however, the model associates words like “iraq”,“america*”, “militari”, “unit”, and “force” with the *Anger* category.

3.4 Experiments

We implemented Mr. LDA using Java with Hadoop 0.20.1 and ran all experiments on a cluster containing 16 physical nodes; each node has 16 2.4GHz cores, and has been configured to run a maximum of 6 map and 3 reduce tasks simultaneously. The cluster is usually under a heavy, heterogeneous load. In this section, we document the speed and likelihood comparison of Mr. LDA against Mahout LDA, another large scale topic modeling implementation based on variational inference. We report results on three datasets:

- TREC document collection (disks 4 and 5 (NIST, 1994)), newswire documents from the *Financial Times* and *LA Times*. It contains more than 300k distinct types over half a million documents. We remove types appearing fewer than 20 times, reducing the vocabulary size to approximately 60k.
- The BlogAuthorship corpus (Koppel et al., 2006), which contains about 10 million blog posts from American users. In contrast to the newswire-heavy TREC corpus, the BlogAuthorship corpus is more personal and informal. Again, terms in fewer than 20 documents are excluded, resulting in 53k distinct types.
- Paired English and German Wikipedia articles (more than half a million in each language). As before, we ignore terms appearing in fewer than 20 documents, resulting in 170k English word types and 210k German word types. While each pair of linked documents shares a common subject (e.g. "George Washington"), they are usually *not* direct translations. The document pair mappings were established from Wikipedia's interlingual links.

English	game	opera	greek	league	said	italian	soviet	french	japanese	album	york	professor
	games	musical	turkish	cup	family	church	political	france	japan	song	canada	berlin
	player	composer	region	club	could	pope	military	paris	australia	released	governor	lied
	players	orchestra	hugarian	played	childern	italy	union	russian	australian	songs	washington	germany
	released	piano	wine	football	death	catholic	russian	la	flag	single	president	von
	comics	works	hungary	games	father	bishop	power	le	zealand	hit	canadian	worked
	characters	symphony	greece	career	wrote	roman	israel	des	korea	top	john	studied
	character	instruments	turkey	game	mother	rome	empire	russia	kong	singer	served	published
	version	composers	ottoman	championship	never	st	republic	moscow	hong	love	house	received
	play	performed	romania	player	day	ii	country	du	korean	chart	county	member
	video	instrument	romanian	match	wife	di	forces	louis	tokyo	albums	north	vienna
	commic	dance	empire	win	died	saint	army	jean	sydney	singles	virginia	august
	original	concert	bulgarian	final	left	king	communist	belgium	china	uk	senate	academy
	manga	performance	bulgaria	teams	home	archbishop	led	belgian	red	records	carolina	1933
	ball	conductor	wines	scored	took	diocese	peace	les	arms	pop	congress	institute
German	spiel	musik	ungarn	saison	frau	papst	regierung	paris	japan	album	new	berlin
	spieler	komponist	turkei	gewann	the	rom	republik	franzosischen	japanischen	the	staaten	universitat
	serie	oper	turkischen	spielte	familie	ii	sowjetunion	frankreich	australien	platz	usa	deutschen
	the	komponisten	griechenland	karriere	mutter	kirche	kam	la	japanische	song	vereinigten	professor
	erschien	werke	rumanien	fc	vater	di	krieg	franzosische	flagge	single	york	studierte
	gibt	orchester	ungarischen	spielen	leben	bishof	land	le	jap	lied	washington	leben
	commics	wiener	griechischen	wechselte	starb	italien	bevolkerung	franzosischer	australischen	titel	national	deutscher
	veroffentlic	komposition	istanbul	mannschaft	tod	italienischen	ende	russischen	neuseeland	erreichte	river	wien
	2	klavier	serbien	olympischen	kinder	konig	reich	moskau	tokio	erschien	county	arbeitete
	konnen	wien	osmanischen	platz	tochter	kloster	politischen	jean	sydney	a	gouverneur	erhielt
	spiele	komponierte	jahrhundert	verein	kam	i	russland	pariser	japanischer	songs	john	august
	dabei	kompositionen	bulgarien	league	sei	kaiser	staaten	pierre	china	erfolg	university	1933
	spielen	dirigent	budapest	2008	alter	maria	staat	et	wappen	you	amerikanischen	munchen
	spiels	konservatorium	slowakei	kam	geboren	san	politische	les	australische	to	state	mitglied
	ball	musiker	turkische	liga	wegen	erzbishof	israel	petersburg	japans	veroffentlicht	north	april

Table 3.3: Extracted polylingual topics from the Wikipedia corpus. While topics are generally equivalent (e.g. on "computer games" or "music"), some regional differences are expressed. For example, the "music" topic in German has two words referring to "Vienna" ("wiener" and "wien"), while the corresponding concept in English does not appear until the 15th position.

3.4.1 Informed Priors

In this experiment, we build the informed priors from LIWC (Pennebaker & Francis, 1999) introduced in Section 3.3.1. We feed the same informed prior to both the TREC dataset and BlogAuthorship corpus. Throughout the experiments, we set the number of topics to 100, with a subset guided by the informed prior.

Table 3.2 shows topics for both TREC and BlogAuthorship. The prior acts as a seed, causing words used in similar contexts to become part of the topic. This is important for computational social scientists who want to discover how an abstract idea (represented by a set of words) is *actually* expressed in a corpus. For example, public news media (i.e. news articles like TREC) connect positive emotions to entertainment, such as music, film and TV, whereas social media (i.e. blog posts) connect it to religion. The *Anxiety* topic in news relates to middle east, but in blogs it focuses on illness, e.g. bird flu. In both corpora, *Causation* was linked to science and technology.

Using informed priors can discover radically different words. While LIWC is designed for relatively formal writing, it can also discover Internet slang such as "lol" ("laugh out loud") in *Affective Process* category. As a result, an informed prior might be helpful in aligning existing lexical resources with corpora with sparse and/or out-of-dictionary vocabularies, e.g., Twitter data.

On the other hand, some discovered topics do not have a clear relationship with the initial LIWC categories, such as the abbreviations and acronyms in *Discrepancy* category. In other cases, the LIWC categories were different enough from the dataset that model chose not to use topics with ill-fitting priors, e.g. the *Cognitive Process* category.

3.4.2 Polylingual LDA

As discussed in Section 3.3.2, Mr. LDA's modular design allows us to consider models beyond vanilla LDA. To the best of our knowledge, we believe this is the first framework for variational inference for polylingual LDA (Mimno et al., 2009), scalable or otherwise. In this experiment, we fit 50 topics to paired English and German Wikipedia articles. We let the program run for 33 iterations with 100 mappers and 50 reducers. Table 3.3 lists down some words from a set of selected topics.

The results listed indicates a similar topics for both English and German. For example, the topic about Europe ("french", "paris", "russian" and "moscow") in English is matched with the topic in German ("frankreich", "paris", "russischen" and "moskau"). Similar behavior was observed for other topics.

The topics discovered by polylingual LDA are not exact matches, however. For example, the second to last column in Table 3.3 is about North America, but the English words focus on Canada, while the corresponding German topic focuses on the United States. Similarly, the forth last column in English contains keywords like "hong", "kong" and "korean", which did not appear in the top 10 words in German. Since this corpus is not a direct translation, these discrepancies might due to a different perspectives, different editorial styles, or different cultural norms.

3.4.3 Scalability

To measure the scalability and accuracy of Mr. LDA, we compare Mr. LDA with Mahout (Foundation et al., 2010), another large scale topic modeling package based on variational inference. We use Mahout-0.4 as our baseline measure. In this set of experiments, we use 90% of the entire TREC corpus as training data and the rest as test data. We ensure that both packages have identical inputs (i.e. identical preprocessing to remove stopwords and selecting vocabulary). We monitor the held-out likelihood under the settings of 50 and 100 topics.

In all experiments, we set the memory limit for every mapper and reducer instance to 2.0-GB. For the hyper-parameter α, Mahout uses a default setting of $\frac{50}{K}$ (recall that K is the number of topic). In order for the results to be comparable, for Mr. LDA, we start the hyper-parameter α from same setting as in Mahout. Mr. LDA continuously updates vector α in the driver program, whereas Mahout does not. All experiments are carried out with 100 mapper instances and 20 reducer instances. We then plot the held-out log-likelihood of test data against the (cumulative) training time. Our empirical results show that, with identical data and hardware, Mr. LDA out-performs Mahout LDA in both the speed and likelihood.

We let both models run for 40 iterations. The held-out likelihood was computed using the variational distribution obtained after every iteration. Figure 3.4(a) shows the result for 50 topics. Mr. LDA runs faster than Mahout. In addition, Mr. LDA yields a better held-out log-likelihood than Mahout, probably as a consequence of hyper-parameter updating—a critical step for variational inference that Mahout

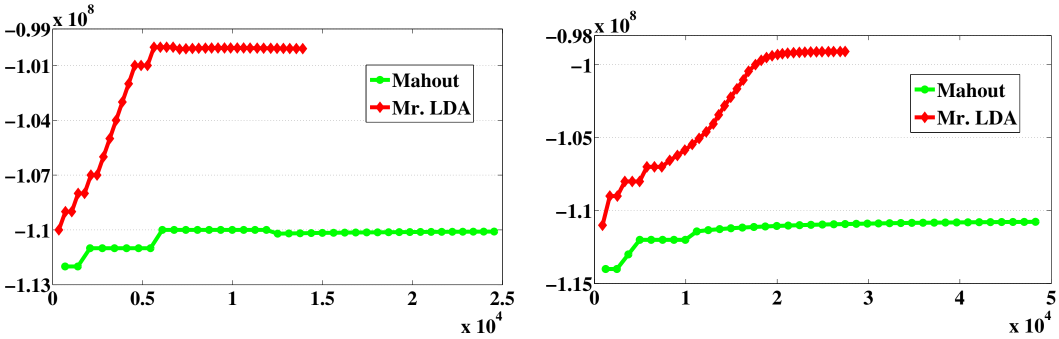


Figure 3.4: Training time vs. held-out log-likelihood on 50 (left) and 100 (right) topics. This figure shows the accumulated training time of the model against the held-out log-likelihood over Mr. LDA and Mahout measured over 40 iterations on 50 topics. Markers indicate the finishing point of a iteration. Mr. LDA out-performs Mahout both in speed and likelihood.

does not support.

When we double the total number of topics to 100, the difference in processing time is magnified. Mr. LDA converges faster than Mahout, again due to the hyperparameter updating. Comparing to the previous diagram of 50 topics, we observe that the training time of Mr. LDA is approximately doubled, which suggesting Mr. LDA scales out effectively.

In-mapper-combiner *In-mapper-combiner* (iMC) provides an efficient way to speed up the intermediate shuffling and sorting. Every mapper instance effectively caches up key-value pairs, aggregates the values and flush them all upon closing or memory reaches a limit. Therefore, it significantly reduces the total number of intermediate key-value pairs. In this experiment, every mapper instance caches the top 10000 frequent words and measure job status. The total number of topics is 1000. Table 3.4 records down the job status averaging over 20 iterations. In-mapper-combiner does reduce the size of intermediate key-value pairs by almost a order of

	w/o IMC	w IMC
Running Time ($\times 10^3$s)	10.635	4.015
Combine Input Records ($\times 10^{10}$)	1.612	14.328
Combine Output Records ($\times 10^9$)	9.786	71.063
Map Output Bytes ($\times 10^{11}$)	1.108	11.651
Map Output Records ($\times 10^9$)	6.931	72.815

Table 3.4: Job status averaging over 20 iterations for 1000 topics. In-mapper-combiner caches top 10000 frequent tokens. In-mapper-combiner significantly reduces the total number of intermediate key-value pairs. Hence, reduces the overall running time.

magnitude, hence speed up the entire learning process significantly.

3.5 Summary

Understanding large text collections such as those generated by social media requires algorithms that are unsupervised and scalable. In this chapter, we present Mr. LDA, which fulfills both of these requirements. Beyond text, LDA is continually being applied to new fields such as music (Hu & Saul, 2009) and source code (Maskeri et al., 2008). All of these domains struggle with the scale of data, and Mr. LDA could help them better cope with large data.

Mr. LDA represents a viable alternative to the existing scalable mechanisms for inference of topic models. Its design easily accommodates other extensions, as we have demonstrated with the addition of informed priors and multilingual topic modeling, and the ability of variational inference to support non-conjugate distributions allows for the development of a broader class of models than could be built with Gibbs samplers alone. In Section 5, we will discuss how can we apply these ideas to discover correlations in mutlilingual corpus, and improve the performance of statistical machine translation system. Mr. LDA, however, would benefit from many of the efficient, scalable data structures that improved other scalable statistical models (Talbot & Osborne, 2007); incorporating these insights would further improve performance and scalability.

Mr. LDA framework is designed to facilitate many other possible topic model extensions or variates based on variational inference approach. As we discussed earlier in Section 3.3, supervised LDA (Blei & McAuliffe, 2007) different from vanilla LDA in the sense of updating the topic word distribution — we need to incorporate label information into the *phi* updating. Syntactic topic models (Boyd-Graber & Blei, 2008) shares the general framework with LDA, but add in syntactic dependencies information during the mapper phase.

While we focused on LDA, the approaches used here are applicable to many other models. Variational inference is an attractive inference technique for the MapReduce framework, as it allows the selection of a variational distribution that breaks dependencies among variables to enforce consistency with the computational constraints of MapReduce. Developing automatic ways to enforce those computational constraints and then automatically derive inference (Winn & Bishop, 2005; Stan Development Team, 2014) would allow for a greater variety of statistical models to be learned efficiently in a parallel computing environment. In Chapter 8, we will focus on the application of adaptor grammar (Johnson et al., 2007), which provides a generic way to infer the latent variable in a hierarchical Bayesian network.

Variational inference is also attractive for its ability to handle online updates Hoffman et al. (2010). Mr. LDA could be extended to more efficiently handle online batches in streaming inference (Hoffman et al., 2010), allowing for even larger document collections to be quickly analyzed and understood. In Chapter 6, we will explain in detail about the online variational updates. It takes only one pass over the entire dataset and update the latent parameters gradually after seeing a subset of data each time.

Chapter 4

Background: Hybrid Variational-MCMC Inference

Recall that in Section 2.3, one common inference method for LDA—other than variational Bayesian inference—is *Markov chain Monte Carlo* sampling (Neal, 1993; Griffiths & Steyvers, 2004, MCMC). MCMC approach has an advantage over variational Bayesian method: each single iteration during inference is short, although it generally requires many more bookkeeping efforts and more iterations to converge (Section 3.1).

Hybrid inference (Mimno et al., 2012) benefits from both MCMC's short iteration and variational Bayesian's parallelize format. It takes advantage of parallelizable variational inference for global variables (Wolfe et al., 2008) while enjoying the sparse, efficient updates for local variables (Neal, 1993).

In the rest of this chapter, we first give a brief review over MCMC sampling approach in Section 4.1, and then discuss the hybrid inference mode in Section 4.2. In Chapter 5, we explore the effectiveness of these three inference techniques—MCMC sampling, variational EM and hybrid inference—on polylingual tree-based topic models, which is targeted to modeling topics in multilingual environment.

4.1 Markov chain Monte Carlo

As discussed in Section 2.3, MCMC relies on drawing random samples from a Markov chain whose stationary distribution is the posterior of interest. Gibbs sampling—a special case of MCMC inference approach—is when the conditional posterior distributions of the target distribution can be sampled exactly, which is the case in LDA due to Dirichlet and multinomial conjugacy. It is widely applied in many Bayesian statistical models (Griffiths & Steyvers, 2004; Griffiths & Ghahramani, 2005; Teh, 2006; Finkel et al., 2007).

Recall that the generative story for LDA in Section 2.2 gives the following complete probability model as shown in Figure 2.1(a),

$$\begin{aligned}
\boldsymbol{\beta}_k &\sim \text{Dirichlet}(\eta), \text{ for } k \in \{1, \ldots, K\}; \\
\boldsymbol{\theta}_d &\sim \text{Dirichlet}(\alpha), \text{ for } d \in \{1, \ldots, D\}; \\
z_{dn}|\boldsymbol{\theta}_d &\sim \text{Discrete}(\boldsymbol{\theta}_d), \text{ for } n \in \{1, \ldots, N_d\}; \\
w_{dn}|z_{dn}, \boldsymbol{\beta}_{z_{dn}} &\sim \text{Discrete}(\boldsymbol{\beta}_{z_{dn}}), \text{ for } n \in \{1, \ldots, N_d\};
\end{aligned}$$

where $\boldsymbol{\beta}_k$ is the distribution over all the vocabulary for topic k and $\boldsymbol{\theta}_d$ is the distribution over all the topics for document d.

Let us denote $\boldsymbol{w}$ as all tokens in the corpus and $\boldsymbol{z}$ as the topic assignments for

$\mathbf{w}$. Given $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, the joint distribution of LDA is

$$p(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{\theta}, \boldsymbol{\beta}; \alpha, \eta) = \prod_k p(\boldsymbol{\beta}_k|\eta) \cdot \prod_d p(\boldsymbol{\theta}_d|\alpha) \prod_n p(z_{dn}|\boldsymbol{\theta}_d) p(w_{dn}|z_{dn}, \boldsymbol{\beta}) \tag{4.1}$$

$$= \prod_d p(\boldsymbol{\theta}_d|\alpha) \prod_n p(z_{dn}|\boldsymbol{\theta}_d) \cdot \prod_k p(\boldsymbol{\beta}_k|\eta) \prod_d \prod_n p(w_{dn}|z_{dn}, \boldsymbol{\beta}). \tag{4.2}$$

Integrating over the hidden variables $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, the joint distribution $p(\boldsymbol{w}, \boldsymbol{z}; \alpha, \eta)$ can be represented as

$$\begin{aligned} p(\boldsymbol{w}, \boldsymbol{z}; \alpha, \eta) &= p(\boldsymbol{z}; \alpha) \cdot p(\boldsymbol{w}|\boldsymbol{z}; \eta) \\ &= \int_{\boldsymbol{\theta}} \prod_d p(\boldsymbol{\theta}_d|\alpha) \prod_n p(z_{dn}|\boldsymbol{\theta}_d) d\boldsymbol{\theta} \cdot \int_{\boldsymbol{\beta}} \prod_k p(\boldsymbol{\beta}_k|\eta) \prod_d \prod_n p(w_{dn}|z_{dn}, \boldsymbol{\beta}) d\boldsymbol{\beta} \\ &= \int_{\boldsymbol{\theta}} p(\boldsymbol{z}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\alpha) d\boldsymbol{\theta} \cdot \int_{\boldsymbol{\beta}} p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\beta}) p(\boldsymbol{\beta}|\eta) d\boldsymbol{\beta} \end{aligned} \tag{4.3}$$

Since the Dirichlet prior is conjugate to the multinomial distribution, the posterior distribution is still a Dirichlet distribution. By integrating out $\boldsymbol{\theta}$ and $\boldsymbol{\beta}$, we have

$$p(\boldsymbol{z}; \alpha) = \int_{\theta} p(\boldsymbol{z}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\alpha) d\boldsymbol{\theta} = \Big(\frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K}\Big)^D \prod_{d=1}^{D} \frac{\prod_k \Gamma(n_{k|d} + \alpha)}{\Gamma(n_{\cdot|d} + K\alpha)}, \tag{4.4}$$

$$p(\boldsymbol{w}|\boldsymbol{z}; \eta) = \int_{\boldsymbol{\beta}} p(\boldsymbol{w}|\boldsymbol{z}, \boldsymbol{\beta}) p(\boldsymbol{\beta}|\eta) d\boldsymbol{\beta} = \Big(\frac{\Gamma(V\eta)}{\Gamma(\eta)^V}\Big)^T \prod_{k=1}^{T} \frac{\prod_w \Gamma(n_{w|k} + \eta)}{\Gamma(n_{\cdot|k} + V\eta)}, \tag{4.5}$$

where $n_{k|d}$ is number of times topic k appears in document d, and $n_{w_{d,n}|k}$ is the number of times type w_{dn} has been assigned to topic k. $n_{\cdot|d}$ and $n_{\cdot|k}$ are the counts aggregated over the corresponding index.

The state of any latent variable is conditioned on the current state of all the other variables. As a result, for the latent topic assignment of any word w_{dn}, we sample from its full conditional distribution $p(z_{dn}|\boldsymbol{z}_{-dn}, \boldsymbol{w}; \alpha, \eta)$

$$
\begin{aligned}
p(z_{dn} = k|\boldsymbol{z}_{-dn}, \boldsymbol{w}; \alpha, \eta) &= \frac{p(z_{dn} = k, \boldsymbol{z}_{-dn}, \boldsymbol{w}; \alpha, \eta)}{p(\boldsymbol{z}_{-dn}, \boldsymbol{w}; \alpha, \eta)} \\
&= \frac{p(z_{dn} = k, \boldsymbol{z}_{-dn}; \alpha)}{p(\boldsymbol{z}_{-dn}; \alpha)} \cdot \frac{p(\boldsymbol{w}|z_{dn} = k, \boldsymbol{z}_{-dn}; \eta)}{p(\boldsymbol{w}|\boldsymbol{z}_{-dn}; \eta)} \\
&= \frac{\frac{\prod_{j \neq k} \Gamma(n_{j|d}+\alpha) \cdot \Gamma(n_{k|d}+\alpha+1)}{\Gamma(n_{\cdot|d}+K\alpha+1)}}{\frac{\prod_{k} \Gamma(n_{k|d}+\alpha)}{\Gamma(n_{\cdot|d}+K\alpha)}} \cdot \frac{\frac{\prod_{w \neq w_{dn}} \Gamma(n_{w|k}+\eta) \cdot \Gamma(n_{w_{dn}|k}+\eta+1)}{\Gamma(n_{\cdot|k}+V\eta+1)}}{\frac{\prod_{w} \Gamma(n_{w|k}+\eta)}{\Gamma(n_{\cdot|k}+V\eta)}} \\
&= \frac{\Gamma(n_{k|d} + 1 + \alpha)}{\Gamma(n_{k|d} + \alpha)} \frac{\Gamma(n_{\cdot|d} + K\alpha)}{\Gamma(n_{\cdot|d} + K\alpha + 1)} \frac{\Gamma(n_{w_{dn}|k} + \eta + 1)}{\Gamma(n_{w_{dn}|k} + \eta)} \frac{\Gamma(n_{\cdot|k} + V\eta)}{\Gamma(n_{\cdot|k} + V\eta + 1)} \\
&= \frac{n_{k|d} + \alpha}{n_{\cdot|d} + T\alpha} \cdot \frac{n_{w_{dn}|k} + \eta}{n_{\cdot|k} + V\eta} \qquad (4.6)
\end{aligned}
$$

4.2 Hybrid Variational-MCMC Inference

Recall that variational Bayesian inference follows an *expectation maximization* approach, where the local variational parameters $\boldsymbol{\phi}_{dn}$ (the variational distribution over all the topics for the n^{th} token in d^{th} document) are updated in the expectation step. Instead of computing this distribution explicitly, Mimno et al. (2012) propose to use MCMC inference to approximate it. The distribution sampled from MCMC will be subsequently sparse—namely, only a few topics will be activated. They further show that such a sparse representation for $\boldsymbol{\phi}_{dn}$ improves performance.

LDA models each of the D documents in a corpus as a mixture of K topics. It can be specified by corpus-level global variables and document-level local variables. The global variables are K topic-word distributions $\{\boldsymbol{\beta}_k\}$ over all the vocabulary.

For a document d with N_d tokens, the local variables are the distribution over all the topics $\boldsymbol{\theta}_d$ and the topic indicator variables $\{z_{dn}\}$ for all N_d words.

Unlike the standard mean-field variational Bayesian inference (Blei et al., 2003) which imposes a variational distribution for $\boldsymbol{\theta}$, the hybrid approach marginalizes this distribution. The variational distribution in this case is

$$q(\boldsymbol{z}, \boldsymbol{\beta}) = \prod_d q(\boldsymbol{z}_d) \prod_k q(\boldsymbol{\beta}_k). \tag{4.7}$$

This factorization differs from the usual mean-field family for topic models. The distribution $q(\boldsymbol{z}_d)$ is a single distribution over the K^{N_d} possible topic configurations, rather than a product of N_d distributions, each over K possible values.

Following Bishop (2006), the optimal variational distribution over topic configurations for a document, holding all other variational distribution fixed, is

$$\begin{aligned} q(\boldsymbol{z}_d) &\propto \exp\left\{\mathbb{E}_{q(\neq \boldsymbol{z}_d)}[\log p(\boldsymbol{z}_d|\alpha) p(\boldsymbol{w}_d|\boldsymbol{z}_d, \boldsymbol{\beta})]\right\} \qquad (4.8) \\ &= \frac{\Gamma(K\alpha)}{\Gamma(K\alpha + N_d)} \prod_k \frac{\Gamma(\alpha + \sum_n \mathbb{I}\left[z_{dn} = k\right])}{\Gamma(\alpha)} \prod_n \exp\left\{\mathbb{E}_q[\log \beta_{z_{dn} w_{dn}}]\right\}, \end{aligned}$$

where $\neg \boldsymbol{z}_d$ denotes the set of all unobserved variables besides $\boldsymbol{z}_d$, and $\mathbb{I}\left[\bullet\right]$ is the indicator function.

We can not explicitly evaluate this distribution because we would have to consider a combinatorial number of topic configurations. To use stochastic gradient ascent, however, we only need to approximate this distribution. We use MCMC to sample topic configurations from $q(\boldsymbol{z}_d)$, then use the empirical average of these samples to estimate the expectations.

We randomly initialize a topic configuration $\boldsymbol{z}_d$, then iteratively resample the topic indicator for each word from the conditional distribution over that word given all other topic assignment:

$$q(z_{dn} = k | \boldsymbol{z}_{-dn}) \propto (\alpha + \sum_{m \neq n} \mathbb{I}\left[z_{dm} = k\right]) \exp\{\mathbb{E}_q[\log \beta_{k w_{dn}}]\}. \tag{4.9}$$

The distribution over topics for the d^{th} documents can be therefore approximated by collecting samples from this empirical distribution upon convergence to its stationary distribution.

After some burn-in sweeps, we could start collecting samples from the empirical distribution of the topic configuration. The expected sufficient statistics of word v in topic k can be approximated by the average of all these samples over all documents and all tokens:

$$\mathbb{E}_q[n_{kv}] \approx \sum_d \sum_n \mathbb{E}_q[\mathbb{I}\left[z_{dn} = k\right] \mathbb{I}\left[w_{dn} = v\right]]. \tag{4.10}$$

The global parameters $\boldsymbol{\lambda}$—distribution over vocabulary per topic—can be subsequently updated as

$$\lambda_{kv} = \eta + \sum_d \sum_n \mathbb{E}_q[\mathbb{I}\left[z_{dn} = k\right] \mathbb{I}\left[w_{dn} = v\right]], \tag{4.11}$$

where η is the topic Dirichlet prior.

This approach lets us take advantage of sparse computation which can be potentially scalable to larger datasets. In addition, Mimno et al. (2012) show that this approach yields much better performance than the standard variational Bayesian inference method and MCMC sampling approach.

In Chapter 5, we explore the effectiveness of these three inference techniques—MCMC sampling, variational EM and hybrid inference—on polylingual tree-based topic models, which is targeted to modeling topics in multilingual environment. This approach is also one of the corner stone for our online hybrid inference framework for Bayesian nonparametric models, which we will discuss in Chapter 7 and 8.

Chapter 5

Polylingual Tree-Based Topic Models

In Section 3.3.2, we demonstrate one flexible extension of Mr. LDA—polylingual LDA to model topics for multilingual corpus. In this chapter, we continue on this path and focus on modeling topics in multilingual environment.

In Section 5.1, we propose a novel model—polylingual tree-based topic models (ptLDA)—to learn topics from multilingual corpus. Our model significantly differs from all past multilingual topic models in the way that it uses information from *both* external dictionaries and document alignments simultaneously to infer more meaningful and reliable topics. In Section 5.2, we derive three different inference techniques—MCMC sampling, variational EM, and hybrid variational-MCMC inference—for this new model.

In Section 5.3, we evaluate the effectiveness of our polylingual tree-based topic models using a downstream application of *statistical machine translation* (Koehn, 2009, SMT). We parallelize our model using MapReduce and scale it up to large collection of aligned datasets. We use the inferred topics as domain knowledge to improve the performance over baseline SMT systems. We show that ptLDA offers better domain adaptation than other topic models for machine translation.

5.1 Polylingual Tree-based Topic Models

Several topic models have been proposed to bridge the chasm between languages in the past using different information, e.g., document connections (Mimno et al., 2009), dictionaries (Boyd-Graber & Resnik, 2010), word correlations (Hu et al., 2011; Hu & Boyd-Graber, 2012; Hu et al., 2013) and word alignments (Zhao & Xing, 2006).

In this section, we bring existing tree-based topic models (Boyd-Graber et al., 2007, tLDA) and polylingual topic models (Mimno et al., 2009, pLDA) together, and create the polylingual tree-based topic model (ptLDA) that incorporates both word-level correlations and document-level alignment information.

In the rest of this section, we first review tree-based topic models in Section 5.1.1 and polylingual topic models in Section 5.1.2. We propose and describe our new polylingual tree-based topic models in Section 5.1.3.

5.1.1 Word-level Correlations

Tree-based topic models incorporate the correlations between words by encouraging words that appear together in a **concept** to have similar probabilities given a topic. These concepts can come from WordNet (Boyd-Graber & Resnik, 2010), domain experts (Andrzejewski et al., 2009), or user constraints (Hu et al., 2013). When we gather concepts from bilingual resources, these concepts can connect different languages. For example, if a bilingual dictionary defines “电脑” as “computer”, we combine these words in a concept.

We organize the vocabulary in a tree structure based on these concepts (Figure 5.1): words in the same concept share a common parent node, and then that concept becomes one of many children of the root node. Words that are not in any concept—**uncorrelated words**—are directly connected to the root node. We call this structure the **tree prior**.

When this tree serves as a prior for topic models, words in the same concept are correlated in topics. For example, if "电脑" has high probability in a topic, so will "computer", since they share the same parent node. With the tree priors, each topic is no longer a distribution over word types, instead, it is a distribution over paths, and each path is associated with a word type. The same word could appear in multiple paths, and each path represents a unique sense of this word.

5.1.2 Document-level Alignments

Lexical resources connect languages and help guide the topics. However, these resources are sometimes brittle and may not cover the whole vocabulary. Aligned document pairs provide a more corpus-specific, flexible association across languages.

Polylingual topic models (Mimno et al., 2009) assume that the aligned documents in different languages share the same topic distribution and each language has a unique topic distribution over its word types. This level of connection between languages is flexible: instead of requiring the exact matching on words and sentences, only a coarse document alignment is necessary, as long as the documents discuss the same topics.

5.1.3 Combine Words and Documents

We propose polylingual tree-based topic models (PTLDA), which connect information across different languages by incorporating both word correlation (as in TLDA) and document alignment information (as in PLDA). In the context of this dissertation, we assume the tree structure is given a priori. For the detailed information on how to build meaningful prior tree structure, please refer to Hu et al. (2011); Hu (2014).

As in LDA, each word token is associated with a topic. However, tree-based topic models introduce an additional step of selecting a concept in a topic responsible for generating each word token. This is represented by a path $y_{d,n}$ through the topic's tree.

The probability of a path in a topic depends on the transition probabilities in a topic. Each concept i in topic k has a distribution over its children nodes governed by a Dirichlet prior: $\pi_{k,i} \sim \text{Dir}(\beta_i)$. Each path ends in a word (i.e., a leaf node) and the probability of a path is the product of all of the transitions between topics it traverses. Topics have correlations over words because the Dirichlet parameters can encode positive or negative correlations (Andrzejewski et al., 2009).

With these correlated in topics in hand, the generation of documents is very similar to LDA. For every document d, we first sample a distribution over topics θ_d from a Dirichlet prior $\text{Dir}(\alpha)$. For every token in the documents, we first sample a topic z_{dn} from the multinomial distribution θ_d, and then sample a path y_{dn} along the tree according to the transition distributions specified by topic z_{dn}. Because every path y_{dn} leads to a word w_{dn} in language l_{dn}, we append the sampled word w_{dn} to document $d_{l_{dn}}$. Aligned documents have words in both languages; monolingual documents only have words in a single language.

The full generative process is:

```
for topic $k \in 1, \cdots, K$ do
    for each internal node $n_i$ do
        draw a distribution $\pi_{ki} \sim \mathrm{Dir}(\beta_i)$
    end for
end for
for document set $d \in 1, \cdots, D$ do
    draw a distribution $\theta_d \sim \mathrm{Dir}(\alpha)$
    for each word in documents $d$ do
        choose a topic $z_{dn} \sim \mathrm{Mult}(\theta_d)$
        sample a path $y_{dn}$ with probability $\prod_{(i,j) \in y_{dn}} \pi_{z_{dn},i,j}$
        $y_{dn}$ leads to word $w_{dn}$ in language $l_{dn}$
        append token $w_{dn}$ to document $d_{l_{dn}}$
    end for
end for
```

If we use a flat symmetric Dirichlet prior instead of the tree prior, we recover pLDA; and if all documents are monolingual (i.e., with distinct distributions over topics θ), we recover tLDA. ptLDA connects different languages on both the word level (using the word correlations) and the document level (using the docu-

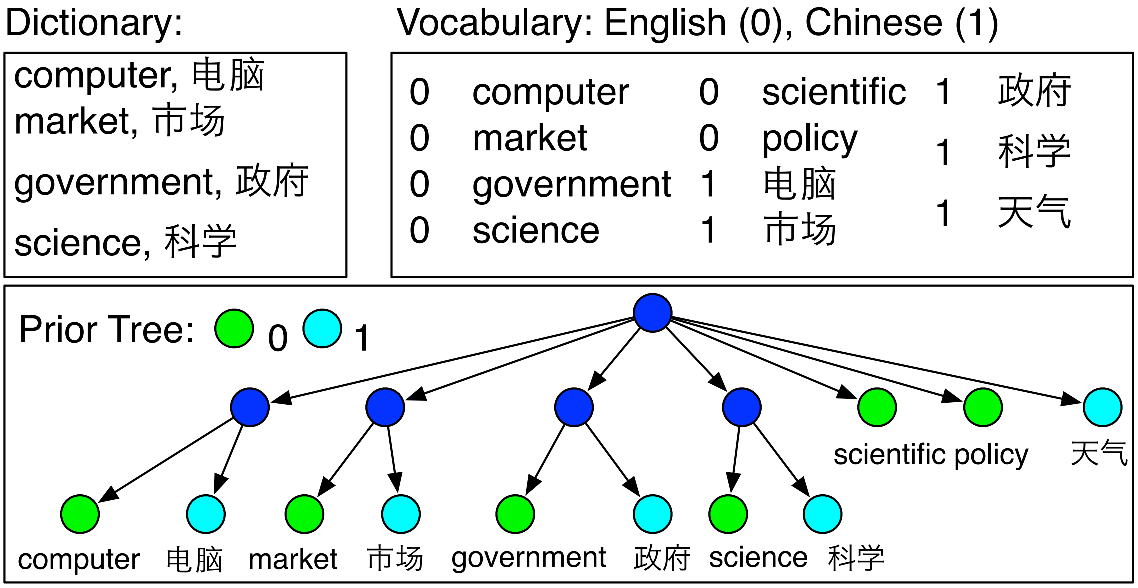


Figure 5.1: An example of constructing a prior tree from a bilingual dictionary: word pairs with the same meaning but in different languages are concepts; we create a common parent node to group words in a concept, and then connect to the root; uncorrelated words are connected to the root directly. Each topic uses this tree structure as a prior.

ment alignments). We compare these models' machine translation performance in Section 5.3.

5.2 Inference

Inference of probabilistic models discovers the posterior distribution over latent variables. For a collection of D documents, each of which contains N_d number of words, the latent variables of ptLDA are: transition distributions $\boldsymbol{\pi}_{ki}$ for every topic k and internal node i in the prior tree structure; multinomial distributions over topics $\boldsymbol{\theta}_d$ for every document d; topic assignments z_{dn} and path y_{dn} for the n^{th} word

w_{dn} in document d. The joint distribution of polylingual tree-based topic models is

$$p(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\pi}; \alpha, \beta) = \prod_k \prod_i p(\boldsymbol{\pi}_{ki}|\beta_i) \cdot \prod_d p(\boldsymbol{\theta}_d|\alpha) \cdot \prod_d \prod_n p(z_{dn}|\boldsymbol{\theta}_d) \cdot \prod_d \prod_n \big(p(y_{dn}|z_{dn}, \boldsymbol{\pi}) p(w_{dn}|y_{dn})\big). \tag{5.1}$$

Exact inference is intractable, so we turn to approximate posterior inference to discover the latent variables that best explain our data. We explore multiple inference schemes, including variational inference (Section 2.3.1), MCMC sampling (Section 4.1) and hybrid approach (Section 4.2). While all of these methods optimize the joint likelihood but they might give different results on a downstream translation task.

5.2.1 Markov Chain Monte Carlo Inference

We use a collapsed Gibbs sampler for tree-based topic models to sample the path y_{dn} and topic assignment z_{dn} for word w_{dn},

$$\begin{aligned} & p(z_{dn} = k, y_{dn} = s|\neg\boldsymbol{z}_{dn}, \neg\boldsymbol{y}_{dn}, \boldsymbol{w}; \alpha, \boldsymbol{\beta}) \\ & \propto \mathbb{I}\left[\Omega(s) = w_{dn}\right] \cdot \frac{N_{k|d} + \alpha}{\sum_{k'}(N_{k'|d} + \alpha)} \cdot \prod_{i \to j \in s} \frac{N_{i \to j|k} + \beta_{i \to j}}{\sum_{j'}(N_{i \to j'|k} + \beta_{i \to j'})}, \end{aligned} \tag{5.2}$$

where $\Omega(s)$ represents the word that path s leads to, $N_{k|d}$ is the number of tokens assigned to topic k in document d and $N_{i \to j|k}$ is the number of times edge $i \to j$ in the tree assigned to topic k, excluding the topic assignment z_{dn} and its path y_{dn} of current token w_{dn}. In practice, we sample the latent variables using efficient sparse updates (Yao et al., 2009; Hu & Boyd-Graber, 2012).

5.2.2 Variational Bayesian Inference

Variational Bayesian inference approximates the posterior distribution with a simplified *variational distribution* q over the latent variables: document topic proportions θ, transition probabilities π, topic assignments z, and path assignments y.

Variational distributions typically assume a mean-field distribution over these latent variables, removing all dependencies between the latent variables. We follow this assumption for the transition probabilities $q(\boldsymbol{\pi} \,|\, \boldsymbol{\lambda})$ and the document topic proportions $q(\boldsymbol{\theta} \,|\, \boldsymbol{\gamma})$; both are variational Dirichlet distributions. However, due to the tight coupling between the path and topic variables, we must model this joint distribution as one multinomial, $q(\boldsymbol{z}, \boldsymbol{y} \,|\, \boldsymbol{\phi})$. If word token w_{dn} has K topics and S paths, it has a $K * S$ length variational multinomial ϕ_{dnks}, which represents the probability that the word takes path s in topic k. The complete variational distribution is

$$q(\boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{z}, \boldsymbol{y}|\gamma, \boldsymbol{\lambda}, \boldsymbol{\phi}) = \prod_d q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d) \cdot \prod_k \prod_i q(\boldsymbol{\pi}_{ki}|\lambda_{ki}) \cdot \prod_d \prod_n q(z_{dn}, y_{dn}|\boldsymbol{\phi}_{dn}). \tag{5.3}$$

Our goal is to find the variational distribution q that is closest to the true posterior, as measured by the *Kullback-Leibler* (KL) divergence between the true posterior p and variational distribution q. This induces an “evidence lower bound”

(ELBO, $\mathcal{L}$) as a function of a variational distribution q:

$$\begin{aligned}\mathcal{L} &= \mathbb{E}_q[\log p(\boldsymbol{w}, \boldsymbol{z}, \boldsymbol{y}, \boldsymbol{\theta}, \boldsymbol{\pi})] - \mathbb{E}_q[\log q(\boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{z}, \boldsymbol{y})] \\ &= \textstyle\sum_k \sum_i \mathbb{E}_q[\log p(\boldsymbol{\pi}_{ki}|\beta_i)] + \sum_d \mathbb{E}_q[\log p(\boldsymbol{\theta}_d|\alpha)] \\ &\quad + \textstyle\sum_d \sum_n \mathbb{E}_q[\log p(z_{dn}, y_{dn}|\boldsymbol{\theta}_d, \boldsymbol{\pi}) p(w_{dn}|y_{dn})] \\ &\quad + \mathbb{H}[q(\boldsymbol{\theta})] + \mathbb{H}[q(\boldsymbol{\pi})] + \mathbb{H}[q(\boldsymbol{z}, \boldsymbol{y})],\end{aligned} \tag{5.4}$$

where $\mathbb{H}[\bullet]$ represents the entropy of a distribution.

Optimizing $\mathcal{L}$ using coordinate descent provides the following updates:

$$\begin{aligned}\phi_{dnkt} &\propto \exp\{\Psi(\gamma_{dk}) - \Psi(\textstyle\sum_k \gamma_{dk}) + \sum_{i \to j \in s} \big(\Psi(\lambda_{k,i\to j}) - \Psi(\sum_{j'} \lambda_{k,i\to j'})\big)\}; \\ \gamma_{dk} &= \alpha_k + \textstyle\sum_n \sum_{s \in \Omega^{-1}(w_{dn})} \phi_{dnkt}; \\ \lambda_{k,i\to j} &= \beta_{i\to j} + \textstyle\sum_d \sum_n \sum_{s\in\Omega'(w_{dn})} \phi_{dnkt} \mathbb{I}\left[i \to j \in s\right];\end{aligned} \tag{5.5}$$

where $\Omega'(w_{dn})$ is the set of all paths that lead to word w_{dn} in the tree, and t represents one particular path in this set. $\mathbb{I}\left[i \to j \in s\right]$ is the indicator of whether path s contains an edge from node i to j.

5.2.3 Hybrid Variational-MCMC Inference

Given the complementary strengths of MCMC and V.B., and following hybrid inference we discussed in Chapter 4, we also derive hybrid inference for ptLDA.

The transition distributions $\boldsymbol{\pi}$ are treated identically as in variational inference. We posit a variational Dirichlet distribution $\boldsymbol{\lambda}$ and choose the one that minimizes the KL divergence between the true posterior and the variational distribution.

For topic $\boldsymbol{z}$ and path $\boldsymbol{y}$, instead of variational updates, we use a Gibbs sampler within a document. We sample z_{dn} and y_{dn} conditioned on the topic and path assignments of all other document tokens, based on the variational expectation of $\boldsymbol{\pi}$,

$$q(z_{dn} = k, y_{dn} = s | \neg \boldsymbol{z}_{dn}, \neg \boldsymbol{y}_{dn}; \boldsymbol{w}) \propto (\alpha + \textstyle\sum_{m \neq n} \mathbb{I}\left[z_{dm} = k\right]) \\ \cdot \exp\{\mathbb{E}_q[\log p(y_{dn}|z_{dn}, \boldsymbol{\pi}) p(w_{dn}|y_{dn})]\}.$$

This equation embodies how this is a hybrid algorithm: the first factor resembles the Gibbs sampling term encoding how much a document prefers a topic, while the second factor encodes the expectation under the variational distribution of how much a path is preferred by this topic,

$$\mathbb{E}_q[\log p(y_{dn}|z_{dn}, \boldsymbol{\pi}) p(w_{dn}|y_{dn})] = \mathbb{I}_{[\Omega(y_{dn}) w_{dn}]} \cdot \textstyle\sum_{i \to j \in y_{dn}} \mathbb{E}_q[\log \lambda_{z_{dn}, i \to j}].$$

For every document, we sweep over all its tokens and resample their topic z_{dn} and path y_{dn} conditioned on all the other tokens' topic and path assignments $\neg \boldsymbol{z}_{dn}$ and $\neg \boldsymbol{y}_{dn}$. To avoid bias, we discard the first B burn-in sweeps and take the following M samples. We then use the empirical average of these samples update the global variational parameter $q(\boldsymbol{\pi}|\boldsymbol{\lambda})$ based on how many times we sampled these paths

$$\lambda_{k, i \to j} = \tfrac{1}{M} \textstyle\sum_d \sum_n \sum_{s \in \Omega^{-1}(w_{dn})} \big(\mathbb{I}\left[i \to j \in s\right] \cdot \mathbb{I}\left[z_{dn} = k, y_{dn} = s\right]\big) + \beta_{i \to j}. \quad (5.6)$$

For our experiments, we use the recommended settings $B = 5$ and $M = 5$ from Mimno et al. (2012).

5.3 Experiments

Topic models have two primary applications: to aid human exploration of corpora (Chang et al., 2009b) or serve as a low-dimensional representation for downstream applications. In this section, we focus on the second application, which has been fruitful for computer vision (Li Fei-Fei & Perona, 2005), computational biology (Perina et al., 2010), and information retrieval (Kataria et al., 2011).

We evaluate our polylingual tree-based topic models on the downstream task of *statistical machine translation* (SMT). In the rest of this section, we first briefly review the domain adaptation for SMT, and several past approaches using topic models as domain knowledge for SMT. We then discuss about the general setup of our SMT pipeline, configurations of our experiments, and underlying datasets. Finally, we show the performance of our polylingual tree-based topic models on domain adaptation for SMT systems. We demonstrate our model yields 1.2 BLEU score improvement over strong baselines.

Domain Adaptation for SMT Modern machine translation systems use millions of examples of translations to learn translation rules. These SMT systems are usually trained on documents with the same genre (e.g., sports, business) from a similar style (e.g., newswire, blog-posts). These are called *domains*. Translations within one domain are better than translations across domains since they vary dramatically in their word choices and style. A correct translation in one domain may be inappropriate in another domain. For example, “潜水” in a newspaper usually means “underwater diving”. On social media, it means a non-contributing “lurker”.

Systems that are robust to systematic variation in the training set are said to exhibit *domain adaptation*. Training a SMT system using diverse data requires *domain adaptation*. Early efforts focus on building separate models (Foster & Kuhn, 2007) and adding features (Matsoukas et al., 2009) to model domain information. Chiang et al. (2011) combine these approaches by directly optimizing genre and collection features by computing separate translation tables for each domain.

Topic Models as Domain Adaptation Topic models provide a solution where domains can be automatically induced from raw data: treat each topic as a domain.[1] They have been shown to be a promising solution for automatically discovering domains in machine translation corpora.

Machine translation uses inherently multilingual data: an SMT system must translate a phrase or sentence from a *source* language to a different *target* language. However, past work either relies solely on monolingual source-side models (Eidelman et al., 2012; Hasler et al., 2012; Su et al., 2012), or limited modeling of the target side (Xiao et al., 2012).

We evaluate our new polylingual tree-based topic models, ptLDA, and existing topic models—LDA, pLDA, and tLDA—on their ability to induce domains for machine translation and the resulting performance of the translations on standard

[1] Henceforth we will use the term “topic” and “domain” interchangeably: “topic” to refer to the concept in topic models and “domain” to refer to SMT corpora.

machine translation metrics.

Dataset and SMT Pipeline We use the NIST MT Chinese-English parallel corpus (NIST), excluding non-UN and non-HK Hansards portions as our training dataset. It contains 1.6M sentence pairs, with 40.4M Chinese tokens and 44.4M English tokens. We replicate the SMT pipeline of Eidelman et al. (2012): word segmentation (Tseng et al., 2005), align (Och & Ney, 2003), and symmetrize (Koehn et al., 2003) the data. We train a modified Kneser-Ney trigram language model on English (Chen & Goodman, 1996). We use CDEC (Dyer et al., 2010) for decoding, and MIRA (Crammer et al., 2006) for parameter training. To optimize SMT system, we tune the parameters on NIST MT06, and report results on three test sets: MT02, MT03 and MT05.[2]

Topic Models Configuration We compare our polylingual tree-based topic model (PTLDA) against tree-based topic models (TLDA), polylingual topic models (PLDA) and vanilla topic models (LDA).[3] We also examine different inference algorithms—Gibbs sampling (**gibbs**), variational inference (**variational**) and hybrid approach (**variational-hybrid**)—on the effects of SMT performance. In all experiments, we set the per-document Dirichlet parameter $\alpha = 0.01$ and the number of topics to 10, as used in Eidelman et al. (2012).

[2] The NIST datasets contain 878, 919, 1082 and 1664 sentences for MT02, MT03, MT05 and MT06 respectively.

[3] For Gibbs sampling, we use implementations available in Hu et al. (2013) for TLDA; and Mallet (McCallum, 2002) for LDA and PLDA.

Resources for Prior Tree To build the tree for tLDA and ptLDA, we extract the word correlations from a Chinese-English bilingual dictionary (Denisowski, 1997).[4] We filter the dictionary using the NIST vocabulary, and keep entries mapping single Chinese and single English words. The prior tree has about 1000 word pairs (*dict*).

We also extract the bidirectional word alignments between Chinese and English using GIZA++ (Och & Ney, 2003). We then remove the word pairs appearing more than 50K times or fewer than 500 times and construct a second prior tree with about 2500 word pairs (*align*).

We apply both trees to tLDA and ptLDA, denoted as tLDA-*dict*, tLDA-*align*, ptLDA-*dict*, and ptLDA-*align*. However, tLDA-*align* and ptLDA-*align* do worse than tLDA-*dict* and ptLDA-*dict*, so we omit tLDA-*align* in the results.

SMT Performance Evaluation We examine the effectiveness of using topic models for domain adaptation on standard SMT evaluation metrics—BLEU (Papineni et al., 2002) and TER (Snover et al., 2006). We report the results on three different test sets (Figure 5.2), and all SMT results are averaged over five runs.

We refer to the SMT model without domain adaptation as **baseline**.[5] LDA marginally improves machine translation (less than half a BLEU point). Polylingual topic models pLDA and tree-based topic models tLDA-*dict* are consistently better than LDA, suggesting that incorporating additional bilingual knowledge improves topic models. These improvements are not redundant: our new ptLDA-

[4] This is a two-level tree structure. However, one could build a more sophisticated tree prior with a hierarchical dictionary such as multilingual WordNet.

[5] Our replication of Eidelman et al. (2012) yields slightly higher baseline performance, but the trend is consistent.

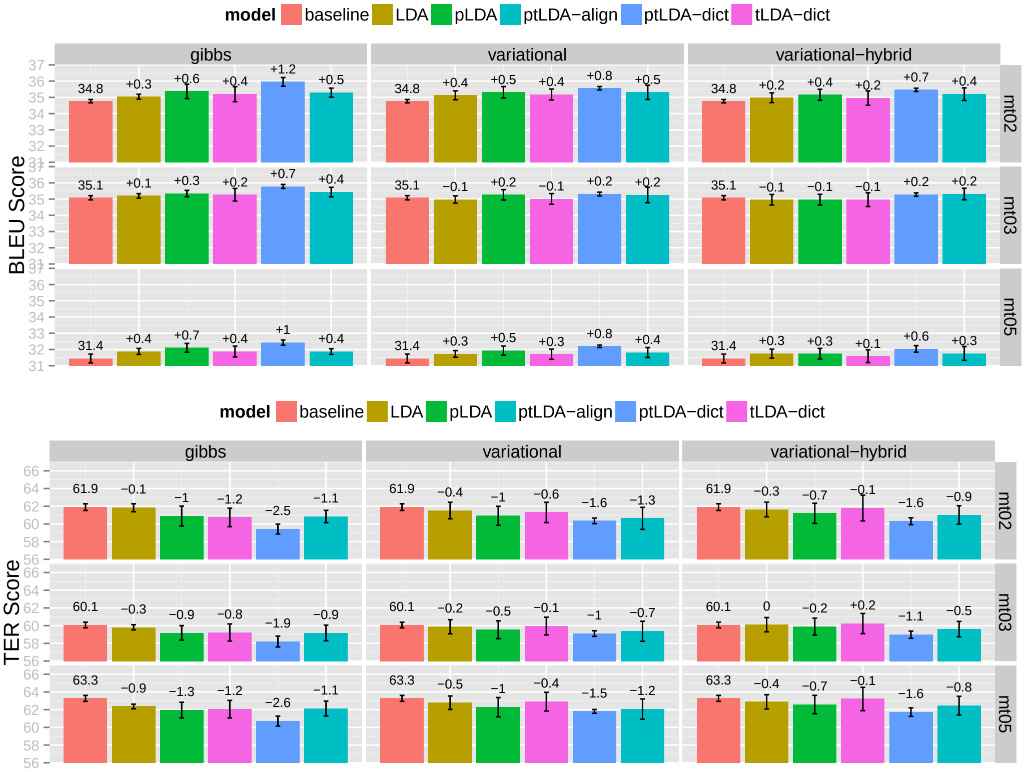


Figure 5.2: Machine translation performance for different models and inference algorithms against the baseline, on BLEU (top, higher the better) and TER (bottom, lower the better) scores. Our proposed PTLDA performs best. Results are averaged over 5 random runs. For model PTLDA-*dict* with different inference schemes, the BLEU improvement on three test sets is mostly significant with $p = 0.01$, except the results on MT03 using variational and variational-hybrid inferences.

dict model, which has aspects of both models yields the best performance among these approaches—up to a 1.2 BLEU point gain (higher is better), and -2.6 TER improvement (lower is better). The BLEU improvement is significant (Koehn, 2004) at $p = 0.01$,[6] except on MT03 with variational and variational-hybrid inference.

While PTLDA-*align* performs better than **baseline** SMT and LDA, it is worse than PTLDA-*dict*, possibly because of errors in the word alignments, making the tree priors less effective.

[6] Because we have multiple runs of each topic model (and thus different translation models), we select the run closest to the average BLEU for the translation significance test.

Scalability of Inference Methods While **gibbs** has better translation scores than **variational** and **variational-hybrid**, it is less scalable to larger datasets. With 1.6M NIST training sentences, **gibbs** takes nearly a week to run 1000 iterations. In contrast, the parallelized **variational** and **variational-hybrid** approaches, which we implement in MapReduce (Dean & Ghemawat, 2004; Wolfe et al., 2008) similar to what we discussed in Chapter 3, take less than a day to converge.

5.4 Summary

Topic models generate great interest, but their use in "real world" applications still lags; this is particularly true for multilingual topic models. As topic models become more integrated in commonplace applications, their adoption, understanding, and robustness will improve.

This chapter contributes to the deeper integration of topic models into critical applications by presenting a new multilingual topic model, PTLDA, comparing it with other multilingual topic models on a machine translation task, and showing that these topic models improve machine translation. PTLDA models both source and target data to induce domains from both dictionaries and alignments. Further improvement is possible by incorporating topic models deeper in the decoding process and adding domain knowledge to the language model.

Chapter 6

Background: Online Truncation-Free Variational Inference

As discussed in Chapter 1, other than parallelization, one other solution to scale up topic models and related probabilistic Bayesian models is to infer the latent parameters in online streaming mode. The basic idea of online learning for topic models follows the general framework of stochastic gradient descent algorithm (Diebolt & Ip, 1996; Bottou, 1998; Bottou & Le Cun, 2003).

In this chapter, we first review the online variational inference for *latent Dirichlet allocation* (Hoffman et al., 2010) in Section 6.1. Then, we briefly talk about the Dirichlet process and its generalization Pitman-Yor process in Section 6.2, which are popular Bayesian nonparametric models for discrete data. Finally, in Section 6.3, we discuss the truncation-free update mechanism that enables online hybrid inference for Bayesian nonparametric models. In following chapters, we are going to apply the online hybrid inference to topic models (Chapter 7) and adaptor grammars (Chapter 8).

6.1 Online Variational Inference

Recall the *evidence lower bound* (ELBO) $\mathcal{L}$ for LDA discussed in Chapter 2, it is computed over the entire dataset of D documents. This requires the algorithm to take a full pass through the entire dataset. It can therefore be slow to apply to very large datasets, and is not naturally suited to settings where new data are constantly arriving. To deal with this limitation, Hoffman et al. (2010) propose an online variational inference algorithm for topic models. The online algorithm is nearly as simple as the batch variational inference algorithm, but converges much faster for large datasets.

In the online setting, we assume documents arrive sequentially in time. In order to get an approximation of the gradient over entire dataset from only one document, we assume that particular document appears D times, i.e., the entire dataset contains D exactly same documents. The ELBO in this case is written as

$$\begin{aligned}\mathcal{L} =& D \cdot \sum_n \mathbb{E}_q \left[\log p(w_n|z_n, \boldsymbol{\beta})\right] + D \cdot \sum_n \mathbb{E}_q \left[\log p(z_n|\boldsymbol{\theta})\right] \\ &+ D \cdot \mathbb{E}_q \left[\log p(\boldsymbol{\theta}_d|\boldsymbol{\alpha})\right] + \sum_k \mathbb{E}_q \left[\log p(\boldsymbol{\beta}_k|\boldsymbol{\eta})\right] \\ &- D \cdot \mathbb{E}_q \left[\log q(\boldsymbol{z}_d|\boldsymbol{\phi}_d)\right] - D \cdot \mathbb{E}_q \left[\log q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d)\right] - \sum_k \mathbb{E}_q \left[\log q(\boldsymbol{\beta}_k|\boldsymbol{\lambda}_k)\right], \end{aligned} \tag{6.1}$$

and obtain a stochastic approximation over the gradient.

Given the ELBO, we are able to write down the noisy update of global variational parameter $\boldsymbol{\lambda}$ as

$$\tilde{\lambda}_{kv} = \eta_{kv} + D \cdot \left(w_v^{(d)} \phi_{kv}^{(d)}\right). \tag{6.2}$$

In such case, the online update of the global variational parameter is simply the interpolation of the old value and the noisy update:

$$\lambda = (1 - \epsilon)\lambda + \epsilon\tilde{\lambda}, \tag{6.3}$$

where ϵ is the decay factor of λ over iterations.

Such an update follows the theory of Newton's methods, which multiplies the gradient with the inverse of the Hessian $\boldsymbol{H}$ matrix for objective function using conjugate gradient techniques. In variational inference, an alternative is to use the Fisher information matrix of the variational distribution q, i.e., the Hessian of the log of the variational probability density function (Sato, 2001; Bottou & Murata, 2002). Please refer to Hoffman et al. (2010, pg. 5-6) for a detailed derivation.

The decay factor ϵ is usually set to $\epsilon = (\tau_0 + i)^{-\kappa}$, where i is the iteration counts. Learning inertia τ_0 prevents premature convergence (i.e., slows down the early iterations of the learning algorithm), and learning rate κ controls how quickly new parameter estimates replace the old ones. Variable $\kappa \in (0.5, 1]$ is required for convergence.

One common technique in stochastic learning is to consider multiple observations per update to reduce noise. In the online case, this means approximating the gradient and updating the variational parameters using minibatches that contains B documents:

$$\tilde{\lambda}_{v,k} = \eta_{v,k} + \frac{D}{B}\sum_{b}\left(w_v^{(d)}\phi_{v,k}^{(d)}\right), \tag{6.4}$$

and we recover the batch setting if $B = D$ and $\kappa = 0$.

6.2 Dirichlet Process

One model challenge during online inference, is due to the online stochastic nature—namely, the dimensionality of latent parameter space is often unknown in advance. This is a classical Bayesian nonparametric problem. Bayesian nonparametric is an appealing solution, because it models arbitrary distributions with an unbounded and possibly countably infinite support.

While Bayesian nonparametric is a broad field, we focus on the Dirichlet process (Ferguson, 1973, DP). In the following sections, we are going to briefly review the definition of a Dirichlet process (Section 6.2), and discuss how to inference the latent parameters in online fashion using truncation-free variational updates (Section 6.3).

The Dirichlet process (Ferguson, 1973, DP) is a two-parameter distribution with scale parameter α^{β} and base distribution G_0. It can be represented using *Chinese restaurant process* (Pitman, 2002, CRP), as well as a *stick-breaking process* (Sethuraman, 1994). We are going to explain these two representations in the rest of this section.

6.2.1 Chinese Restaurant Process

One of the common representations for the Dirichlet process is the Chinese restaurant process (Aldous, 1985). It does not refer to the sample $G \sim G_0$ directly, but models draws from G instead. Imaging a Chinese restaurant with possibly unbounded number of tables, each of which could fit possibly infinite number of customers. The metaphor refers to the process of customers constantly walking into the restaurant. The first customer sits at the first table with probability 1. For every new customer, he chooses to sit either on the table proportional to the number of customers already sitting on that table, or sit on a new unoccupied table with probability α^{β}. The Chinese restaurant process exhibits a clustering property. This metaphor has turned out to be useful in considering various generalizations of the Dirichlet process (Pitman, 2002) and many applications of Bayesian nonparametric methods (Teh et al., 2006).

6.2.2 Stick-breaking Process

Another way to express Dirichlet process is via the stick-breaking process, since the draws from a DP are composed of a weighted sum of point masses. Sethuraman (1994) made this precisely by providing a constructive definition of the DP, called the stick-breaking construction. This construction is also significantly more straightforward than many past approaches to construct DPs (Ferguson, 1973; Pitman, 2002).

A draw G from $\mathrm{DP}(\alpha^{\beta}, G_0)$—under the stick-breaking construction—is modeled with a series draws from a Beta distribution,

$$b_1, \ldots, b_i, \ldots \sim \mathsf{Beta}(1, \alpha^{\beta}), \qquad \rho_1, \ldots, \rho_i, \ldots \sim G_0.noteot \tag{6.5}$$

These individual draws from a Beta distribution are the foundation for the stick-breaking construction of the DP (Sethuraman, 1994). Each breaking point b_i models how much probability mass remains. These break points combine to form an "infinite" multinomial,

$$\beta_i \equiv b_i \prod_{j=1}^{i-1}(1 - b_j), \qquad\qquad G \equiv \sum_i \beta_i \delta_{\rho_i}, \tag{6.6}$$

where the weights β_i give the probability of selecting any particular atom ρ_i from the base distribution.

The stick-breaking process prior can be understood metaphorically as follows. Starting with a stick of unit length, we break it at b_1, assigning β_1 to be the length of stick we just broke off. Now recursively break the other portion to obtain β_2, β_3 and so forth. The stick-breaking distribution over β is often written as $\beta \sim \mathrm{GEM}(\alpha^\beta)$[1]. Due to its simplicity, the stick-breaking construction has led to a variety of extensions as well as novel inference techniques for the Dirichlet process.

6.2.3 Generalization of Dirichlet Process

The Dirichlet process is a canonical distribution over probability measures and can be generalized to the Pitman-Yor process (Pitman & Yor, 1997, PY) to model data exhibiting power-law properties (Goldwater et al., 2006; Teh et al., 2006).

A draw $H_c \equiv (\boldsymbol{\pi}_c, \boldsymbol{z}_c)$ from the Pitman-Yor process is formed by the stick-breaking process (Sudderth & Jordan, 2008, PYGEM) parametrized by scale pa-

[1] The distribution is named after initials of Griffiths, Engen and McCloskey.

rameter a, discount factor b, and base distribution G_c:

$$\begin{aligned} \pi'_k \sim& \mathrm{Beta}(1-b, a+kb), & z_k \sim& G_c, \\ \pi_k \equiv& \pi'_k \textstyle\prod_{j=1}^{k-1}(1-\pi'_j), & H \equiv& \textstyle\sum_k \pi_k \delta_{z_k}. \end{aligned} \tag{6.7}$$

Similar to Dirichlet process, the distribution H_c is a discrete reconstruction of the atoms sampled from G_c—hence, reweights G_c—but exhibits power-law behavior. The Pitman-Yor process reduces to the Dirichlet process when $b = 0$. The various representations of the DP, including the Chinese restaurant process (Section 6.2.1) and the stick-breaking construction (Section 6.2.2), have similar analogues for the Pitman-Yor process.

6.2.4 Application of Dirichlet Process

The Dirichlet process and its generalization Pitman-Yor process have been widely used in many Bayesian nonparametric models. For example, they have been applied as a Bayesian nonparametric prior in Gaussian mixture models (Rasmussen, 2000; Blei & Jordan, 2005; Wood & Black, 2008) and sequential models (Beal et al., 2002; Fox et al., 2008; Paisley & Carin, 2009) to model the number of latent mixture components or states. In the field of computer vision, they have been used in modeling the number of image segments (Sudderth et al., 2005). They have also been applied in Bayesian hierarchical models, such as topic models (Teh et al., 2006; Wang et al., 2011) and language modeling (Teh, 2006; Johnson et al., 2007; Liang et al., 2007).

6.3 Truncation-Free Variational Inference

The variational inference for Dirichlet process and Pitman-Yor process is particularly challenging, due to the infinite support. Variational inference algorithms for Bayesian nonparametric models do not operate in an unbounded latent space. One common method to inference the variational parameters in Dirichlet process and Pitman-Yor process is to "truncate" the stick-breaking process to a distribution with finite supports (Blei & Lafferty, 2005). This certainly reduces the model complexity to a manageable scale during the inference.

As noted in previous section, variational inference methods usually truncate the variational distributions to maintain tractable. This is particularly limiting in the online setting, where we hope for a Bayesian nonparametric posterior seamlessly adapting its model complexity to an endless stream of data.

Wang & Blei (2012) develop a truncation-free stochastic variational inference algorithm for nonparametric Bayesian models. It lets us more easily apply Bayesian nonparametric data analysis to massive and streaming data, which is our main focus in this dissertation. When applied to Bayesian nonparametric models, it does not require truncations and gives a principled mechanism for adapting the truncation level or model complexity of the variational distributions on the fly.

In the following chapters, we apply this method and propose the online hybrid inference framework. We use this framework to explore different parameter space and infer the latent parameters for Bayesian nonparametric models in the online setting. In Chapter 7, we propose a novel online topic model with infinite vocabulary, which significantly differs from all past approaches in a way that it allows the vocabulary to grow over time. In Chapter 8, we apply the online hybrid inference framework on an existing Bayesian nonparametric model—adaptor grammars.

Chapter 7

Online Latent Dirichlet Allocation with Infinite Vocabulary

Following the discussion in Chapter 1, a common strategy to scale up topic models—other than the parallelization approach introduced in Chapter 3—is converting batch algorithms into streaming algorithms that only make one pass over the data. In this chapter, we focus on the online streaming approach to scale up topic models. We propose a novel online LDA which supports possibly unbounded vocabulary. Our model significantly differs from all past online approaches in the way that our model allows the vocabulary to change and evolve throughout time, which is a challenge, but often overlooked problem. We derive online hybrid inference for our proposed model. We further demonstrate our model is able to incorporate new words into vocabulary and effectively refine topics over time.

Online learning of general Bayesian statistical models relies on update the model parameters in an incremental fashion. One common approach for online MCMC methods is known as *sequential Monte Carlo* (SMC) methods, also referred as *particle filters*, which are fundamentally similar to the method of *importance sampling* (Doucet et al., 2001). The basic idea is to approximate a distribution of interest using a swarm of weighted, sequentially updated samples, commonly referred as particles. SMC methods have been successfully applied to perform inference in LDA (Canini et al., 2009). However, a naïve implementation of this method may lead to a large variance in the resulting samples, which may require extra steps of resampling.

There are other online LDA approaches using MCMC methods as well.Song et al. (2005) extend the batch mode Gibbs sampler into an online algorithm—also referred as “O-LDA” in Banerjee & Basu (2007)—by sampling the topic distribution conditioning only on the topics of the words up to the end of the previous document, rather than all previous words. The algorithm requires a batch initialization and incrementally samples all the subsequent topics without resampling old topics. Canini et al. (2009) later discover that the performance of O-LDA method depends critically on the accuracy of the topics inferred during its batch initialization phase, and subsequently propose an incremental Gibbs sampler with rejuvenation to avoid the batch initialization step. AlSumait et al. (2008) introduce the idea of evolutionary matrix to model the change of the distribution over words per topic over a sliding window at any time. Although it offers the dynamics to detecting emerging trends in text streams and track their drift over time, it alters the underlying generative process of LDA and may not be easily generalized to other Bayesian statistical models.

Compared to MCMC, the online updates for variational inference can be viewed as a classical online learning problem (Bottou, 1998). Hoffman et al. (2010) extend LDA to online settings. However, this and later online topic models (Wang et al., 2011; Mimno et al., 2012) make the same simple assumption. The namesake topics, distributions over words that evince thematic coherence, are always modeled as multinomials drawn from a finite Dirichlet distribution. This assumption precludes additional words being added over time. Particularly for streaming algorithms, this is neither reasonable nor appealing. There are many reasons immutable vocabularies do not make sense: words are invented ("crowdsourcing") or words cross languages ("Gangnam"). To be flexible, online topic models must be able to capture the addition and invention of new terms in the data stream.

Allowing models to expand vocabulary to include additional words requires changing the underlying statistical formalism. Instead of assuming that topics come from a finite Dirichlet distribution, we assume that they come from a Dirichlet process (Ferguson, 1973)—which we discussed in Section 6.2—with a base distribution over all possible words, of which there are an infinite number. Bayesian nonparametric tools like the Dirichlet process allow us to reason about distributions over infinite supports. We review N-gram models of latent variable models in Section 7.1. In Section 7.2, we propose the *infinite vocabulary topic model*, which uses Bayesian nonparametric to go beyond fixed vocabularies.

In Section 7.3, we derive approximate hybrid inference for our model. We use the truncation-free variation inference approach—which was discussed in Section 6.3—to dynamically expand the vocabulary. Since emerging vocabulary are most important in non-batch settings, in Section 7.4, we extend inference to online streaming settings. We compare the coherence and effectiveness of our infinite vocabulary topic model against models with fixed vocabulary in Section 7.5.

Figure 7.1 shows how a topic evolves during online inference, which is a successful application of our model. The algorithm processes documents in subsets we call *minibatches*; after each minibatch, online hybrid inference updates our model's

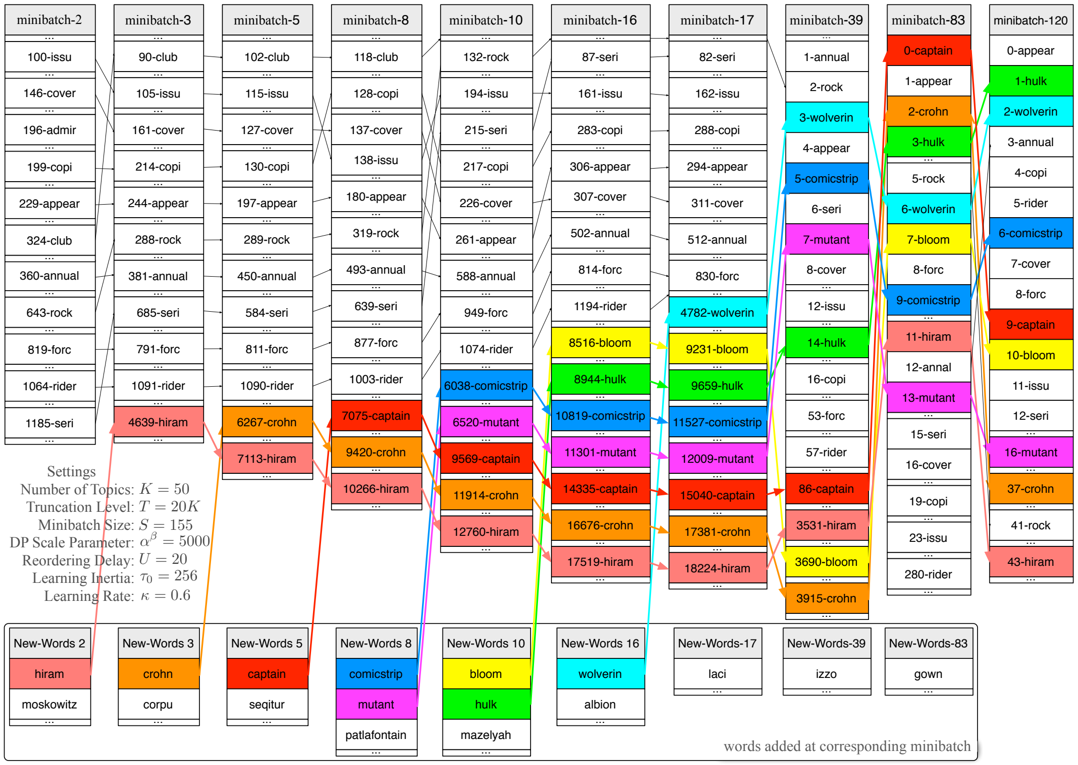


Figure 7.1: The evolution of a single "comic book" topic from the *20 newsgroups* corpus. Each column is a ranked list of word probabilities after processing a minibatch (numbers preceding words are the exact rank). The box below the topics contains words introduced in a minibatch. For example, "hulk" first appeared in minibatch 10, was ranked at 9659 after minibatch 17, and became the second most important word by the final minibatch. Colors help show words' trajectories.

parameters. This shows that *out of vocabulary words* can enter topics and eventually become *high probability words* in corresponding topics.

7.1 N-gram Models in Latent Variable Models

A strength of the probabilistic formalism is the ability to embed specialized models inside more general models. The problem of part-of-speech (POS) induction (Goldwater & Griffiths, 2007) uses morphological regularity within part of speech classes (e.g., verbs in English often end with "ed") to learn a character n-gram model for parts of speech (Clark, 2003). This has been combined within the latent variable HMM via a Chinese restaurant process (Blunsom & Cohn, 2011).

We also view latent clusters of words (topics) as a nonparametric distribution with a character n-gram base distribution, but to better support streaming data sets, we use online variational inference; previous approaches used Monte Carlo methods (Neal, 1993). Variational inference is easier to distribute (Chapter 3) and amenable to online updates (Hoffman et al., 2010).

Within the topic modeling community, there are different approaches to deal with changing word use. Dynamic topic models (Blei & Lafferty, 2006) discover evolving topics by viewing word distributions as n-dimensional points undergoing Brownian motion. These models reveal compelling topical evolution; e.g., physics moving from studies of the æther to relativity to quantum mechanics. However, the models assume **fixed vocabularies**; we show that our infinite vocabulary model discovers more coherent topics (Section 7.5.2).

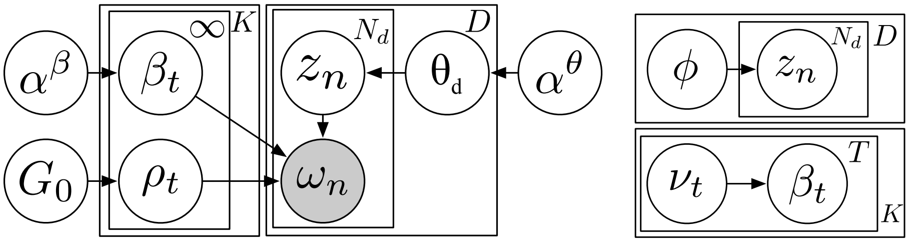
An elegant solution for large vocabularies is the “hashing trick” (Weinberger et al., 2009), which maps strings into a restricted set of integers via a hash function. These integers become the topic model’s vocabulary. While elegant, words are no longer identifiable, since multiple words might be hashed to exactly the same integer value. However, our infinite vocabulary topic model retains identifiability and better models datasets (Section 7.5.3).

7.2 Infinite Vocabulary Topic Model

The model we develop uses a base distribution over all possible words, and each topic is a draw from the Dirichlet process (Section 6.2). This approach is inspired by unsupervised models that induce parts-of-speech.

Our generative process is identical to LDA's (Chapter 2) except that topics are not drawn from a finite Dirichlet. Instead, topics are drawn from a DP with base distribution G_0 over *all* possible words:

```
for each topic $k$ do
    Draw words $\rho_{kt}, (t = \{1, 2, ...\})$ from $G_0$.
    Draw $b_{kt} \sim \mathsf{Beta}(1, \alpha^{\beta}), (t = \{1, 2, \dots\})$.
    Set stick weights $\beta_{kt} = b_{kt} \prod_{s<t}(1 - b_{ks})$.
end for
for each document $d$ in a corpus $D$ do
    Choose distribution $\theta_d$ over topics from a Dirichlet distribution $\boldsymbol{\theta}_d \sim \mathrm{Dir}(\alpha^{\theta})$.
    for each of the $n = 1, \dots, N_d$ word indexes do
        Choose a topic $z_n$ from the distribution over topics of current document $z_n \sim \mathrm{Mult}(\boldsymbol{\theta}_d)$.
        Choose a word $w_n$ from the appropriate topic's distribution over words $p(w_n|\boldsymbol{\beta}_{z_n})$.
    end for
end for
```



(a) LDA with infinite vocabulary

(b) Variational

Figure 7.2: Plate representation for latent Dirichlet allocation with infinite vocabulary (left) and its variational distribution (right).

7.2.1 A Distribution over Words

An intuitive choice for G_0 is a conventional character language model. However, such a naïve approach is unrealistic and is biased to shorter words; preliminary experiments yielded poor results. Instead, we define G_0 as the following distribution over strings

```
1: Choose a length $l \sim \text{Mult}(\boldsymbol{\zeta})$.
2: Generate character $c_i \sim p(c_i|\boldsymbol{c}_{i-n,\ldots,i-1})$.
```

This is similar to the classic n-gram language model, except that the length is first chosen from a multinomial distribution over all lengths. Estimating conditional n-gram probabilities is well-studied in natural language processing (Jelinek & Mercer, 1985).

The full expression for the probability of a word ρ consisting of the characters $c_1, c_2, \ldots$ under G_0 is

$$G_0(\rho) \equiv p_{\mathsf{WM}}(l = |\rho| \,|\, \boldsymbol{\zeta}) \prod_{i=1}^{|\rho|} p(c_i|\boldsymbol{c}_{i-n,\ldots,i-1}),$$

where $|\rho|$ is the length of the word. To avoid length bias, we chose the multinomial $\boldsymbol{\zeta}$ that minimizes the average discrepancy between word corpus probabilities $p_{\mathcal{C}}$ and the probability in our word model

$$\boldsymbol{\zeta} \equiv \arg\min_{\boldsymbol{\zeta}} \textstyle\sum_{\rho} |p_{\mathcal{C}}(\rho) - p_{\mathsf{WM}}(\rho|\,\boldsymbol{\zeta})|^2, \text{s.t.} \sum_{l} \zeta_l = 1.$$

The n-gram statistics are estimated from an English dictionary which need not be very large, since it is a language model over characters, not words.

7.3 Variational Approximation

For a corpus of D documents where the d-th document contains N_d words, the joint distribution is

$$\begin{aligned} p(\boldsymbol{W}, \boldsymbol{\rho}, \boldsymbol{\beta}, \boldsymbol{\theta}, \boldsymbol{z}) = & \textstyle\prod_{k=1}^{K} \left[\prod_{t=1}^{\infty} p(\rho_{kt}|G_0) \cdot p(\beta_{kt}|\alpha^{\beta})\right] \\ & \cdot \left[\textstyle\prod_{d=1}^{D} p(\boldsymbol{\theta}_d|\alpha^{\theta}) \prod_{n=1}^{N_d} p(z_{dn}|\boldsymbol{\theta}_d) p(\omega_{dn}|z_{dn}, \boldsymbol{\beta}_{z_{dn}})\right]. \end{aligned}$$

Let us denote the latent variables $\boldsymbol{Z} \equiv \{$corpus-level stick proportions $\boldsymbol{\beta}$, document topic distributions $\boldsymbol{\theta}$ and word topic assignments $\boldsymbol{z}\}$. We turn to variational inference (Section 2.3.1) to optimize the latent variables $\mathbf{Z}$, and then select a simpler family of distributions q.

Unlike mean-field approaches (Blei et al., 2003), which assume q is a fully factorized distribution, we integrate out the word-level topic distribution vector $\boldsymbol{\theta}$: $q(\boldsymbol{z}_d\,|\,\eta)$ is a single distribution over K^{N_d} possible topic configurations rather than a product of N_d multinomial distributions over K topics. Combined with a beta distribution $q(b_{kt}|\nu^1_{kt}, \nu^2_{kt})$ for stick-breaking points, the variational distribution q is

$$q(\boldsymbol{Z}) \equiv q(\boldsymbol{\beta}, \boldsymbol{z}) = \textstyle\prod_D q(\boldsymbol{z}_d \,|\, \eta) \prod_K q(\boldsymbol{b}_k \,|\, \boldsymbol{\nu}^1_k, \boldsymbol{\nu}^2_k). \tag{7.1}$$

However, we cannot explicitly represent a distribution over all possible strings, so we truncate our variational stick-breaking distribution $q(\boldsymbol{b} \,|\, \boldsymbol{\nu})$ to a finite set.

7.3.1 Truncation Ordered Set

Variational methods typically cope with infinite dimensionality of nonparametric models by *truncating* the distribution to a finite subset of all possible atoms that nonparametric distributions consider (Blei & Jordan, 2005; Kurihara et al., 2006; Boyd-Graber & Blei, 2009). This is done by selecting a relatively large truncation index T_k, and then stipulating that the variational distribution uses the rest of the available stick at that index, i.e., $q(b_{T_k} = 1) \equiv 1$. As a consequence, β is zero in expectation under q beyond that index.

However, directly applying such a technique is not feasible here, as truncation is not just a search over dimensionality but also over atom strings and their ordering. This is often a problem for nonparametric models, and the truncation that solves the problem matches the underlying probabilistic model: for mixture models, it is the number of components (Blei & Jordan, 2005); for hierarchical topic models, it is a tree (Wang & Blei, 2009); for natural language grammars, it is grammatons (Cohen et al., 2010). Similarly, our truncation is not just a fixed vocabulary size; it is a

truncation ordered set (TOS), which imposes an ordering over all the atoms in the truncation set. The ordering is important because the Dirichlet process is a size-biased distribution; words with lower indices are likely to have a higher probability than words with higher indices.

Each topic has a unique TOS $\mathcal{T}_k$ of limited size that maps every word type w to an integer t; thus $t = \mathcal{T}_k(w)$ is the index of the atom ρ_{kt} that corresponds to w. We defer how we choose this mapping until Section 7.3.3. More pressing is how we compute the two variational distributions of interest. For $q(z \,|\, \eta)$, we use local collapsed MCMC sampling (Mimno et al., 2012) and for $q(\boldsymbol{b} \,|\, \nu)$ we use stochastic variational inference (Hoffman et al., 2010). We describe both in turn.

7.3.2 Stochastic Inference

We are going to infer the latent parameters using the hybrid MCMC variational inference (Section 4.2). In our model, the conditional distribution of a topic assignment of a word with TOS index $t = \mathcal{T}_k(w_{dn})$ is

$$q(z_{dn} = k | \boldsymbol{z}_{-dn}, t = \mathcal{T}_k(w_{dn})) \propto \left(\textstyle\sum_{\substack{m=1 \\ m \neq n}}^{N_d} \mathbb{I}_{z_{dm}=k} + \alpha_k^{\theta} \right) \exp \left\{ \mathbb{E}_{q(\boldsymbol{\nu})} \left[\log \beta_{kt} \right] \right\}. \quad (7.2)$$

We iteratively sample from this conditional distribution to obtain the empirical distribution $\phi_{dn} \equiv \hat{q}(z_{dn})$ for latent variable z_{dn}, which is fundamentally different from mean-field approach (Blei et al., 2003).

There are two cases to consider for computing Eqn. (7.2)—whether a word w_{dn} is in the TOS for topic k or not. First, we look up the word's index $t = \mathcal{T}_k(w_{dn})$. If this word is in the TOS, i.e., $t \leq T_k$, the expectations are straightforward (Mimno et al., 2012)

$$q(z_{dn} = k) \propto \left(\sum_{\substack{m=1 \\ m \neq n}}^{N_d} \phi_{dmk} + \alpha_k^\theta \right) \cdot \exp\{\Psi(\nu_{kt}^1) + \sum_{s=1}^{s<t} \Psi(\nu_{ks}^2) - \sum_{s=1}^{s\leq t} \Psi(\nu_{ks}^1 + \nu_{ks}^2)\} \tag{7.3}$$

It is more complicated when a word is not in the TOS. We use the truncation-free updates discussed in Section 6.3. The conditional distribution of an unseen word ($t > T_k$) is

$$q(z_{dn} = k) \propto \left(\sum_{\substack{m=1 \\ m \neq n}}^{N_d} \phi_{dmk} + \alpha_k^\theta \right) \cdot \exp\{\sum_{s=1}^{s\leq t} (\Psi(\nu_{ks}^2) - \Psi(\nu_{ks}^1 + \nu_{ks}^2))\}. \tag{7.4}$$

This is different from finite vocabulary topic models that set vocabulary *a priori* and ignore OOV words.

7.3.3 Refining the Truncation Ordered Set

In this section, we describe heuristics to update the TOS inspired by MCMC conditional equations, a common practice for updating truncations. One component of a good TOS is that more frequent words should come first in the ordering. This is reasonable because the stick-breaking prior induces a size-biased ordering of the clusters. This has previously been used for truncation optimization for Dirichlet process mixtures and admixtures (Kurihara et al., 2007).

Another component of a good TOS is that words consistent with the underlying base distribution should be ranked higher than those not consistent with the base distribution. This intuition is also consistent with the conditional sampling equations for MCMC inference (Müller & Quintana, 2004); the probability of creating a new table with dish ρ is proportional to $\alpha^{\beta}G_0(\rho)$ in the Chinese restaurant process, which we discussed in Section 6.2.2.

Thus, to update the TOS, we define the ranking score of word t in topic k as

$$R(\rho_{kt}) = p(\rho_{kt}|G_0) \sum_{d=1}^{D} \sum_{n=1}^{N_d} \phi_{dnk} \delta_{\omega_{dn}=\rho_{kt}}, \tag{7.5}$$

sort all words by the scores within that topic, and then use those positions as the new TOS. In Section 7.4.1, we present online updates for the TOS.

7.4 Online Inference

Online variational inference seeks to optimize the ELBO $\mathcal{L}$ by stochastic gradient optimization. Because gradients estimated from a single observation are noisy, stochastic inference for topic models typically uses "minibatches" of S documents out of D total documents (Hoffman et al., 2010).

An approximation of the natural gradient of $\mathcal{L}$ with respect to ν is the product of the inverse Fisher information and its first derivative (Sato, 2001)

$$\begin{aligned} \Delta\nu^1_{kt} &= 1 + \tfrac{D}{|\mathcal{S}|} \textstyle\sum_{d\in\mathcal{S}} \sum_{n=1}^{N_d} \phi_{dnk}\delta_{\omega_{dn}=\rho_{kt}} - \nu^1_{kt}, \\ \Delta\nu^2_{kt} &= \alpha^{\beta} + \tfrac{D}{|\mathcal{S}|} \textstyle\sum_{d\in S} \sum_{n=1}^{N_d} \phi_{dnk}\delta_{\omega_{dn}>\rho_{kt}} - \nu^2_{kt}, \end{aligned} \tag{7.6}$$

which leads to an update of ν,

$$\nu^1_{kt} = \nu^1_{kt} + \epsilon \cdot \Delta\nu^1_{kt},$$

$$\nu^2_{kt} = \nu^2_{kt} + \epsilon \cdot \Delta\nu^2_{kt} \tag{7.7}$$

where $\epsilon_i = (\tau_0 + i)^{-\kappa}$ defines the step size of the algorithm in minibatch i. The **learning rate** κ controls how quickly new parameter estimates replace the old; $\kappa \in (0.5, 1]$ is required for convergence. The **learning inertia** τ_0 prevents premature convergence. We recover the batch setting if $\mathcal{S} = \mathcal{D}$ and $\kappa = 0$.

7.4.1 Updating the Truncation Ordered Set

A nonparametric streaming model should allow the vocabulary to dynamically expand as new words appear (e.g., introducing "vuvuzelas" for the 2010 World Cup), and contract as needed to best model the data (e.g., removing "vuvuzelas" after the craze passes). We describe three components of this process, expanding the truncation, refining the ordering of TOS, and contracting the vocabulary.

Determining the TOS Ordering This process depends on the ranking score of a word in topic k at minibatch i, $R_{i,k}(\rho)$. Ideally, we would compute R from all data. However, only a single minibatch is accessible. We have a per-minibatch rank estimate

$$r_{i,k}(\rho) = p(\rho|G_0) \cdot \tfrac{D}{|\mathcal{S}_i|} \textstyle\sum_{d \in \mathcal{S}_i} \sum_{n=1}^{N_d} \phi_{dnk} \delta_{\omega_{dn}=\rho}$$

which we interpolate with our previous ranking

$$R_{ik}(\rho) = (1 - \epsilon) \cdot R_{i-1,k}(\rho) + \epsilon \cdot r_{ik}(\rho). \tag{7.8}$$

We introduce an additional algorithm parameter, the **reordering delay** U. We found that reordering after every minibatch ($U = 1$) was not effective; we explore the role of reordering delay in Section 7.5. After U minibatches have been observed, we reorder the TOS for each topic according to the words' ranking score R in Eqn. (7.8); $\mathcal{T}_k(w)$ becomes the rank position of w according to the latest R_{ik}.

Expanding the Vocabulary Each minibatch contains words we have not seen before. When we see them, we must determine their relative rank position in the TOS, their rank scores, and their associated variational parameters. The latter two issues are relevant for online inference because both are computed via interpolations from previous values in Eqn. (7.8) and (7.7). For an unseen word ω, previous values are undefined. Thus, we set $R_{i-1,k}$ for unobserved words to be 0, $\boldsymbol{\nu}$ to be 1, and $\mathcal{T}_k(\omega)$ is $T_k + 1$ (i.e., increase truncation and append to the TOS).

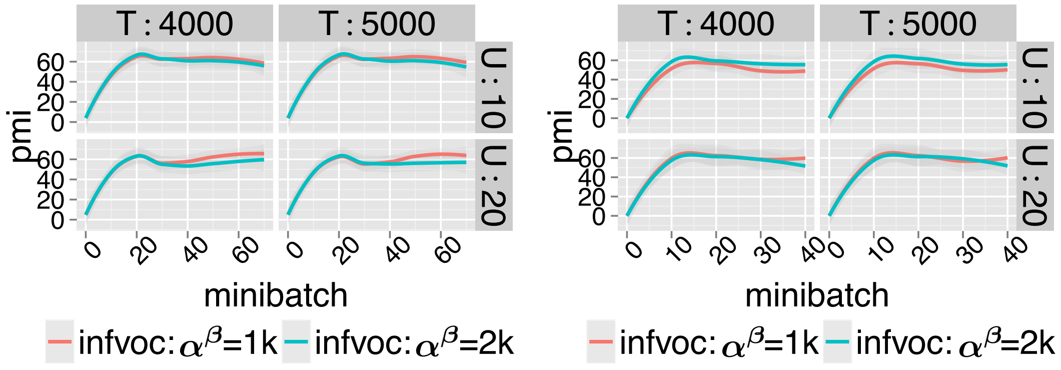
Contracting the Vocabulary To ensure tractability we must periodically prune the words in the TOS. When we reorder the TOS (after every U minibatches), we only keep the top T terms, where T is a user-defined integer. A word type ρ will be removed from $\mathcal{T}_k$ if its index $\mathcal{T}_k(\rho) > T$ and its previous information (e.g., rank and variational parameters) is discarded. In a later minibatch, if a previously discarded word reappears, it is treated as a new word.

To keep the ranking score of all the words in the vocabulary, our proposed framework bounds the memory cost to a constant factor of total number of unique words appeared in the dataset. To accommodate larger corpora, we could always discard a word's ranking score information completely, and set to the smallest score in current $\mathcal{T}_k$ if it appears again. Alternatively, Bloom filters (Bloom, 1970; Broder & Mitzenmacher, 2002; Talbot & Osborne, 2007) are another choice to maintain these statistics, as they provide a space efficient storage with strict one-sided error.

Pseudo-Code Throughout time, the vocabulary grows continuously in every iteration. However, the refinement of the vocabulary is executed periodically after a certain number of iterations in order to stabilize the vocabulary. The detailed online variational inference algorithm is listed in Algorithm 4.

Algorithm 4 Online Variational Inference

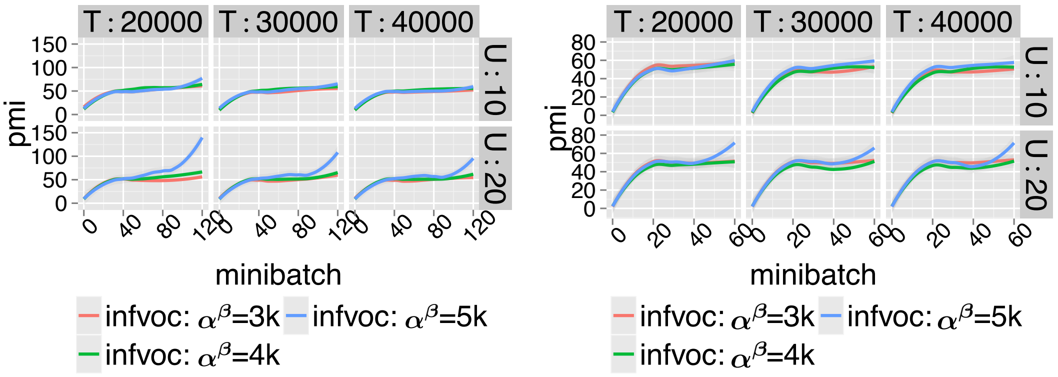
```
for each document $d$ in mini-batch $S$ do
    for every word $n$ in document $d$ do
        Empirically sample the variational distribution $q(z_{dn}|\boldsymbol{\phi}_{dn})$ using the truncation-free approach as in Eqn. (7.3) and (7.4) iteratively until converge.
    end for
end for
Update variational parameters $\boldsymbol{\nu}$ using Eqn. (7.7).
Update the ranking score according to Eqn. (7.8).
Refine the vocabulary for every topic if necessary (Section 7.4.1).
```



(a) *de-news*, $S = 140$.

(b) *de-news*, $S = 245$.

Figure 7.3: PMI score on *de-news* dataset against different settings of DP scale parameter α^β, truncation level T and reordering delay U. Common parameter settings: number of topics $K = 10$, learning rate $\kappa = 0.8$ and learning inertia $\tau_0 = 64$. Our model is more sensitive to α^β and less sensitive to T.



(a) *20 newsgroups*, $S = 155$.

(b) *20 newsgroups*, $S = 310$.

Figure 7.4: PMI score on *20 newsgroups* dataset against different settings of DP scale parameter α^β, truncation level T and reordering delay U. Common parameter settings: number of topics $K = 50$, learning rate $\kappa = 0.8$ and learning inertia $\tau_0 = 64$. Our model is more sensitive to α^β and less sensitive to T.

7.5 Experimental Evaluation

In this section, we evaluate the performance of our infinite vocabulary topic model (*infvoc*) on two corpora: *de-news*[1] and *20 newsgroups*.[2] Both corpora were

[1] A collection of daily news items between 1996 to 2000 in English. It contains 9,756 documents, 1,175,526 word tokens, and 20,000 distinct word types. Available at `homepages.inf.ed.ac.uk/pkoehn/publications/de-news`.

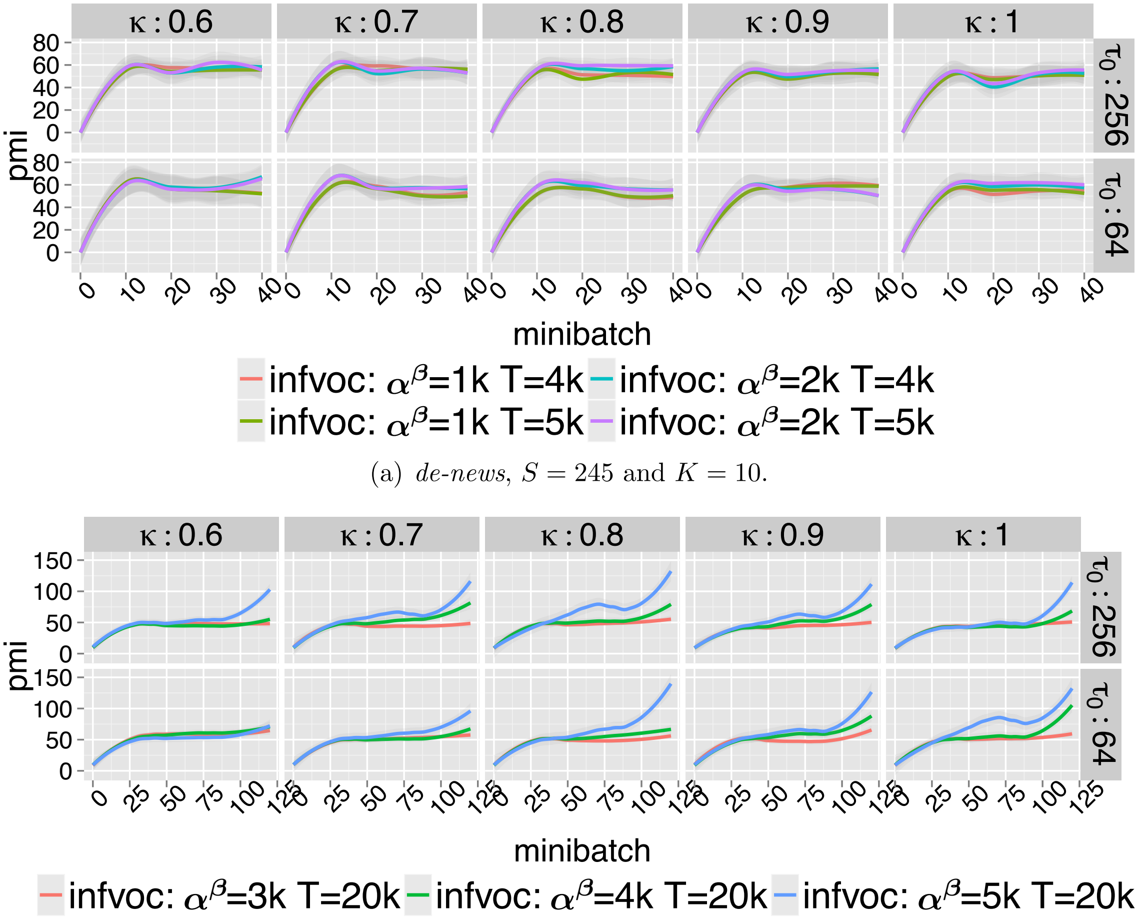
[2] A collection of discussions in 20 different newsgroups. It contains 18,846 documents and 100,000 distinct word types. It is sorted by date into roughly 60% training and 40% testing data. Available at `qwone.com/~jason/20Newsgroups`.

parsed by the same tokenizer and stemmer with a common English stopword list (Bird et al., 2009). First, we examine sensitivity to both model parameters and online learning rates. Having chosen those parameters, we then compare our model with other topic models with fixed vocabularies.

Evaluation Metric Typical evaluation of topic models is based on held-out likelihood or perplexity. However, creating a strictly fair comparison for our model against existing topic model algorithms is difficult, as traditional topic model algorithms must discard words that have not previously been observed. Moreover, held-out likelihood is a flawed proxy for how topic models are used in the real world (Chang et al., 2009a). Instead, we use two evaluation metrics: topic coherence and classification accuracy.

Pointwise mutual information (PMI), which correlates with human perceptions of topic coherence, measures how words fit together within a topic. Following Newman et al. (2009), we extract document co-occurrence statistics from Wikipedia and score a topic's coherence by averaging the pairwise PMI score (w.r.t. Wikipedia co-occurrence) of the topic's ten highest ranked words. Higher average PMI implies a more coherent topic.

Classification accuracy is the accuracy of a classifier learned from the topic distribution of training documents applied to testing documents (the topic model sees both). A higher accuracy means the unsupervised topic model better captures the underlying structure of the corpus. To better simulate real-world situations, 20-newsgroup's test/train split is by date (test documents were posted after training



(a) *de-news*, $S = 245$ and $K = 10$.

(b) *20 newsgroups*, $S = 155$ and $K = 50$.

Figure 7.5: PMI score on two datasets with reordering delay $U = 20$ against different settings of decay factor κ and τ_0. A suitable choice of DP scale parameter α^β increases the performance significantly. Learning parameters κ and τ_0 jointly define the step decay. Larger step sizes promote better topic evolution.

documents).

Comparisons We evaluate the performance of our model (*infvoc*) against three other models with fixed vocabularies: online variational Bayes LDA (*fixvoc-vb*, Hoffman et al. 2010), online hybrid LDA (*fixvoc-hybrid*, Mimno et al. 2012), and dynamic topic models (*dtm*, Blei & Lafferty 2006). Including dynamic topic models is not a fair comparison, as its inferences requires access to all of the documents in the dataset; unlike the other algorithms, it is not online.

Vocabulary For fixed vocabulary models, we must decide on a vocabulary *a priori*. We consider two different vocabulary methods: use the first minibatch to define a vocabulary (*null*) or use a comprehensive dictionary[3] (*dict*). We use the same dictionary to train *infvoc*'s base distribution.

Experiment Configuration For all models, we use the same symmetric document Dirichlet prior with $\alpha^{\theta} = 1/K$, where K is the number of topics. Online models see exactly the same minibatches. For *dtm*, which is not an online algorithm but instead partitions its input into "epochs", we combine documents in ten consecutive minibatches into an epoch (longer epochs tended to have worse performance; this was the shortest epoch that had reasonable runtime).

For online hybrid approaches (*infvoc* and *fixvoc-hybrid*), we collect 10 samples empirically from the variational distribution in E-step with 5 burn-in sweeps. For *fixvoc-vb*, we run 50 iterations for local parameter updates.

7.5.1 Sensitivity to Parameters

Figure 7.3 and Figure 7.4 show how the PMI score is affected by the DP scale parameter α^{β}, the truncation level T, and the reordering delay U. The relatively high values of α^{β} may be surprising to readers used to seeing a DP that instantiates dozens of atoms, but when vocabularies are in tens of thousands, such scale parameters are necessary to support the long tail. Although we did not investigate such approaches,

[3] `http://sil.org/linguistics/wordlists/english/`

this suggests that more advanced nonparametric distributions (Teh, 2006) or explicitly optimizing α^β may be useful. Relatively large values of U suggest that accurate estimates of the rank order are important for maintaining coherent topics.

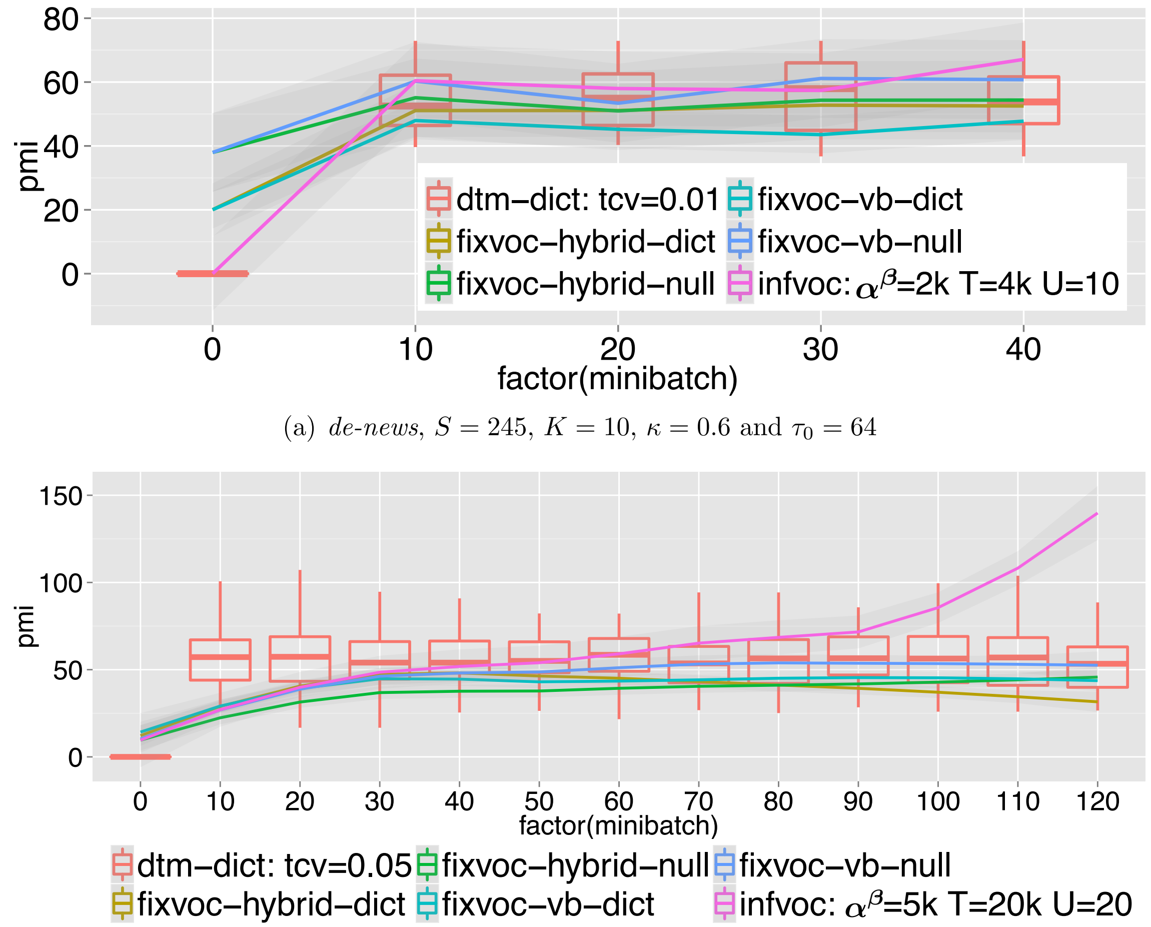
While *infvoc* is sensitive to parameters related to the vocabulary, once suitable values of those parameters are chosen, it is no more sensitive to learning-specific parameters than other online LDA algorithms (Figure 7.5), and values used for other online topic models also work well here.

7.5.2 Comparing Algorithms: Coherence

Now that we have some idea of how we should set parameters for *infvoc*, we compare it against other topic modeling techniques. We used grid search to select parameters for each of the models[4] and plotted the topic coherence averaged over all topics in Figure 7.6.

While *infvoc* initially holds its own against other models, it does better and better in later minibatches, since it has managed to gain a good estimate of the vocabulary and the topic distributions have stabilized. Most of the gains in topic coherence come from highly specific proper nouns which are missing from vocabularies of the fixed-vocabulary topic models. This advantage holds even against *dtm*, which uses batch inference.

[4] For the *de-news* dataset, we select (*20 newsgroups* parameters in parentheses) minibatch size $S \in \{140, 245\}$ ($S \in \{155, 310\}$), DP scale parameter $\alpha^\beta \in \{1k, 2k\}$ ($\alpha^\beta \in \{3k, 4k, 5k\}$), truncation size $T \in \{3k, 4k\}$ ($T \in \{20k, 30k, 40k\}$), reordering delay $U \in \{10, 20\}$ for *infvoc*; and topic chain variable tcv $\in \{0.001, 0.005, 0.01, 0.05\}$ for *dtm*.



(a) *de-news*, $S = 245$, $K = 10$, $\kappa = 0.6$ and $\tau_0 = 64$

(b) *20 newsgroups*, $S = 155$, $K = 50$, $\kappa = 0.8$ and $\tau_0 = 64$

Figure 7.6: PMI score on two datasets against different models. Our model *infvoc* yields a better PMI score against *fixvoc* and *dtm*; gains are more marked in later minibatches as more and more proper names have been added to the topics. Because *dtm* is not an online algorithm, we do not have detailed per-minibatch coherence statistics and thus show topic coherence as a box plot per epoch.

7.5.3 Comparing Algorithms: Classification

For the classification comparison, we consider additional topic models. While we need the most probable topic *strings* for PMI calculations, classification experiments only need a document's topic vector. Thus, we also consider hashed vocabulary schemes. The first, which we call *dict-hashing*, uses a dictionary for the known words and hashes any other words into the same set of integers. The second, *full-hash*, used

model settings				accuracy %
$S = 155$	$\tau_0 = 64$ $\kappa = 0.6$	*infvoc*	$\alpha^{\beta} = 3k$ $T = 40k$ $U = 10$	52.683
		fixvoc	vb-dict	45.514
		fixvoc	vb-null	49.390
		fixvoc	hybrid-dict	46.720
		fixvoc	hybrid-null	50.474
		fixvoc	vb dict-hash	52.525
		fixvoc	vb full-hash $T = 30k$	51.653
		fixvoc	hybrid dict-hash	50.948
		fixvoc	hybrid full-hash $T = 30k$	50.948
	dtm-dict $tcv = 0.001$			**62.845**
$S = 310$	$\tau_0 = 64$ $\kappa = 0.6$	*infvoc*	$\alpha^{\beta} = 3k$ $T = 40k$ $U = 20$	52.317
		fixvoc	vb-dict	44.701
		fixvoc	vb-null	51.815
		fixvoc	hybrid-dict	46.368
		fixvoc	hybrid-null	50.569
		fixvoc	vb dict-hash	48.130
		fixvoc	vb full-hash $T = 30k$	47.276
		fixvoc	hybrid dict-hash	51.558
		fixvoc	hybrid full-hash $T = 30k$	43.008
	dtm-dict $tcv = 0.001$			**64.186**

Table 7.1: Classification accuracy based on 50 topic features extracted from *20 newsgroups* data. Our model (*infvoc*) out-performs algorithms with a fixed or hashed vocabulary but not *dtm*, a batch algorithm that has access to all documents.

in Vowpal Wabbit,[5] hashes *all* words into a set of T integers.

We train 50 topics for all models on the entire dataset and collect the document level topic distribution for every article. We treat such statistics as features and train a SVM classifier on all training data using Weka (Hall et al., 2009) with default parameters. We then use the classifier to label testing documents with one of the 20 newsgroup labels. A higher accuracy means the model is better capturing the underlying content.

Our model *infvoc* captures better topic features than online LDA *fixvoc* (Table 7.1) under all settings.[6] This suggests that in a streaming setting, *infvoc* can

[5] `hunch.net/~vw/`

[6] Parameters were chosen via cross-validation on a 30%/70% dev-test split from the following parameter settings: DP scale parameter $\alpha \in \{2k, 3k, 4k\}$, reordering delay $U \in \{10, 20\}$ (for *infvoc* only); truncation level $T \in \{20k, 30k, 40k\}$ (for *infvoc* and *fixvoc full-hash* models); step decay factors $\tau_0 \in \{64, 256\}$ and $\kappa \in \{0.6, 0.7, 0.8, 0.9, 1.0\}$ (for all online models); and topic chain variable $tcv \in \{0.01, 0.05, 0.1, 0.5\}$ (for *dtm* only).

better categorize documents. However, the batch algorithm *dtm*, which has access to the entire dataset performs better because it can use later documents to retrospectively improve its understanding of earlier ones. Unlike *dtm*, *infvoc* only sees early minibatches once and cannot revise its model when it is tested on later minibatches.

7.5.4 Qualitative Example

Figure 7.1 shows the evolution of a topic in *20 newsgroups* about comics as new vocabulary words enter from new minibatches. While topics improve over time (e.g., relevant words like "seri(es)", "issu(e)", "forc(e)" are ranked higher), interesting words are being added throughout training and become prominent after later minibatches are processed (e.g., "captain", "comicstrip", "mutant"). This is not the case for standard online LDA—these words are ignored and the model does not capture such information. In addition, only about 60% of the word types appeared in the SIL English dictionary. Even with a comprehensive English dictionary, online LDA could not capture all the word types in the corpus, especially named entities.

7.6 Summary

We proposed an online topic model that, instead of assuming vocabulary is known *a priori*, adds and sheds words over time. While our model is better able to create coherent topics, it does not outperform dynamic topic models (Blei & Lafferty, 2006; Wang et al., 2008) that explicitly model how topics change. It would be interesting to allow such models to—in addition to modeling the *change* of topics—also change the underlying *dimensionality* of the vocabulary.

Another possible extension is to adopt a two level hierarchical topic distribution: have one DP for the vocabulary and then another DP for each topic. This might be a better model. However, there are several challenges in doing online variational inference in this model. One approach is to have a degenerate top level distribution (Liang et al., 2007), but it is not intuitive to obtain a close form online update. Another approach is to have an indicator to connect atoms between the two levels (Wang et al., 2011), however, it is trickier in such case as atoms are identifiable since they are connected to specific strings.

To keep the ranking score of all the words in the vocabulary, our proposed framework bounds the memory cost to a linear factor of total number of unique words appeared in the database. To accommodate larger corpus, we could always discard a word's ranking score information completely, and set to the smallest score in current $\mathcal{T}_k$ if it appears again. Alternatively, Bloom filter (Bloom, 1970; Broder & Mitzenmacher, 2002; Talbot & Osborne, 2007) is another choice to maintain these statistics, as it provides a space efficient storage with strict one-sided error.

Topic models are only one example of probabilistic Bayesian models that can benefit from online inference. In the next chapter, we apply online inference to an expensive modeling framework—adaptor grammars (Johnson et al., 2007)—to capture many Bayesian nonparametric probabilistic models. We extend our online inference approach to adaptor grammars, and scale it up to handle much larger datasets.

Chapter 8

Online Adaptor Grammars with Hybrid Inference

In this chapter, we focus on adaptor grammars (Johnson et al., 2007), which are Bayesian nonparametric models based on *probabilistic context-free grammars* (PCFG). We propose a new online hybrid inference method for adaptor grammars. Our inference method is able to expand, adjust and prune the set of adapted grammar rules over time, which obviates the need for expensive preprocessing required by previous approaches. We show that our method yields significant speed-up over past approaches. This method can also be viewed as a generalization of online hybrid inference framework we proposed in Chapter 7 to a broader class of Bayesian nonparametric models.

PCFGs make a substantive assumption about the language's underlying structures, i.e., the context-free grammar rules are statistically independent of each other. When generating a new instance according to the PCFG, its structure is built up by applying a sequence of context-free grammar rules, where each rule in the sequence is selected independently at random. Therefore, the generative process of PCFG expands a symbol by completely ignoring the fact of a particular symbol has been rewritten in the past. It does not take into consideration of the information about what and how frequent a symbol has been rewritten into in the past. Adaptor grammars weaken such a strong statistical independence assumptions that PCFGs make.

The weaker statistical independence assumptions that adaptor grammars make come at the cost of expensive inference. A common approach to address this computational bottleneck is through variational inference (Wainwright & Jordan, 2008). One of the advantages of variational inference is that it can be easily parallelized (Chapter 3) or transformed into an online algorithm (Chapter 7), which often empirically converges faster than batch variational inference.

Past variational inference techniques for adaptor grammars require a preprocessing step which looks at *all available data* to establish the support of these Bayesian nonparametric distributions (Cohen et al., 2010) before starting inference. This preprocessing step becomes a critical bottleneck to scaling the algorithm to large datasets. In addition, because these past approaches require all data to be available upfront, they are not directly amenable to online inference.

Markov chain Monte Carlo (MCMC) inference (Johnson et al., 2007), an alternative to variational inference, does not have this disadvantage. MCMC is easier to implement, and it *discovers* the support of nonparametric distributions during inference rather than assuming it *a priori.*

We apply hybrid inference (Section 4.2) to adaptor grammars to get the best of both worlds. The method interleaves MCMC sampling *inside* variational inference (Section 8.2). We propose the online hybrid inference in Section 8.3, which processes examples in small batches taken from a data stream. Our algorithm dynamically extends the set of adapted grammar rules as more data are observed. This obviates the need for expensive preprocessing which is a necessary step to create an online algorithm for adaptor grammars. Our online hybrid inference approach also scales adaptor grammars up to datasets that cannot be examined exhaustively due to their size, e.g., terabytes of social media data appear every second.

We show our approach's scalability and flexibility by applying our inference framework in Section 8.4 on two tasks: unsupervised word segmentation and infinite-vocabulary topic modeling.

8.1 PCFGs and Adaptor Grammars

In this section, we review probabilistic context-free grammars and adaptor grammars.

8.1.1 Probabilistic Context-free Grammars

Probabilistic context-free grammars (PCFG) define probability distributions over derivations of a context-free grammar. We define a PCFG $\boldsymbol{\mathcal{G}}$ to be a tuple $\langle \boldsymbol{W}, \boldsymbol{N}, \boldsymbol{R}, S, \boldsymbol{\theta} \rangle$: a set of terminals $\boldsymbol{W}$, a set of nonterminals $\boldsymbol{N}$, productions $\boldsymbol{R}$, start symbol $S \in \boldsymbol{N}$ and a vector of rule probabilities $\boldsymbol{\theta}$. The rules that rewrite nonterminal c is $\boldsymbol{R}(c)$.

PCFGs typically use nonterminals with a syntactic interpretation. A sequence of terminals (the **yield**) is generated by recursively rewriting nonterminals as sequences of child symbols (either a nonterminal or a symbol). This builds a hierarchical **phrase-tree structure** for every yield.

For example, given the set of terminals $\boldsymbol{W} \equiv \{a, \ldots, z\}$ and the set of nonterminals $\boldsymbol{N} \equiv \{$Char, Chars, Word, Words, Sent$\}$ with the start symbol $S =$ Sent. The productions $\boldsymbol{R}$ and their corresponding probabilities are

Sent	↦	Words	1.0
Words	↦	Word Words	0.75
Words	↦	Word	0.25
Word	↦	Chars	1.0
Chars	↦	Char Chars	0.72
Chars	↦	Char	0.28
Char	↦	a	0.11
…			
Char	↦	z	0.01

The nonterminal Word represents a word. It gets rewritten to Chars, which can be subsequently rewritten into a sequence of nonterminals Chars,Char using different production rules according to their probabilities. The rewriting process terminates when the derivation has reached a terminal symbol such as "a".

We assume an unsupervised setting, in which only terminals are observed. Our goal is to infer the underlying phrase-structure tree.

Adaptor grammars require that the PCFG does not have self-recursive adapted nonterminals, i.e., there cannot be a path in a derivation from a given adapted nonterminal to a second appearance of that adapted nonterminal.[1] Using the above grammar as an example, the nonterminal Word captures the exactly same patterns as the nonterminal Chars does, but without any self-recursion. Therefore, it can be used as an adapted nonterminal, but Chars can not.

[1] There cannot be a path in a derivation from a given nonterminal to a second appearance of that nonterminal (see Cohen et al. (2010) for discussion).

8.1.2 Adaptor Grammars

PCFGs assume that the rewriting operations are independent given the nonterminal. This context-freeness assumption often is too strong for modeling natural language.

Adaptor grammars break this independence assumption by transforming a PCFG's distribution over trees G_c rooted at nonterminal c into a richer distribution H_c over the trees headed by a nonterminal c, which is often referred to as the *grammaton*.

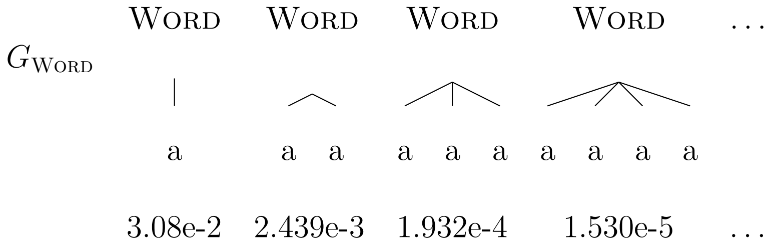
A *Pitman-Yor Adaptor grammar* (PYAG)[2] forms the adapted tree distributions H_c using a *Pitman-Yor process* (Pitman & Yor, 1997, PY), which we discussed in Section 6.2.3. A draw $H_c \equiv (\boldsymbol{\pi}_c, \boldsymbol{z}_c)$ is formed by the stick-breaking process as Equation 6.7, with scale parameter a, discount factor b, and base distribution G_c. Intuitively, the distribution H_c is a discrete reconstruction of the atoms sampled from G_c—hence, reweights G_c. Grammaton H_c assigns non-zero stick-breaking weights $\boldsymbol{\pi}$ to a countably infinite number of parse trees $\boldsymbol{z}$. We describe learning these grammatons in Section 8.2.

More formally, a PYAG is a quintuple $\boldsymbol{\mathcal{A}} = \langle \boldsymbol{\mathcal{G}}, \boldsymbol{M}, \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\alpha} \rangle$ with: a PCFG $\boldsymbol{\mathcal{G}}$; a set of adapted nonterminals $\boldsymbol{M} \subseteq \boldsymbol{N}$; Pitman-Yor process parameters a_c, b_c at each adaptor $c \in \boldsymbol{M}$ and Dirichlet parameters $\boldsymbol{\alpha}_c$ for each nonterminal $c \in \boldsymbol{N}$. We also assume an order on the adapted nonterminals, $c_1, \ldots, c_{|\boldsymbol{M}|}$ such that c_j is not reachable from c_i in a derivation if $j > i$.[3]

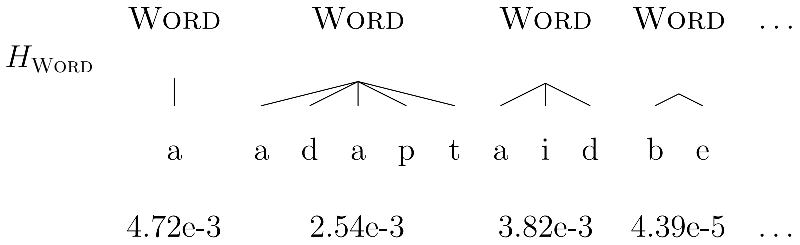
[2] Adaptor grammars, in their general form, do not have to use the Pitman-Yor process but have only been used with the Pitman-Yor process.

[3] This is possible because we assume that recursive nonterminals are not adapted.

Algorithm 5 describes the generative process of an adaptor grammar on a set of D observed sentences $x_1, \ldots, x_D$. The generative process starts with a set of PCFG rules $\boldsymbol{\mathcal{G}}$. The PCFG $\boldsymbol{\mathcal{G}}$ specifies a distribution over all possible derivation trees under each nonterminal. For example, the nonterminal Word has a distribution G_{Word} over derivation trees according to $\boldsymbol{\mathcal{G}}$:



For every adapted nonterminal c, we reweight its distribution according to Pitman-Yor process, i.e., $H_c \sim \text{PYGEM}(a_c, b_c, G_c)$. In this case, the distribution under the adapted nonterminal Word has been reweighted to



For each observation in the dataset, we then generate it by recursively rewriting nonterminals according to the set of rules specified by PCFG $\boldsymbol{\mathcal{G}}$ and H_c's until terminals.

Given a PYAG $\boldsymbol{\mathcal{A}}$, the joint probability for a set of sentences $\boldsymbol{X}$ and its collection of trees $\boldsymbol{T}$ is

$$p(\boldsymbol{X}, \boldsymbol{T}, \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{z} | \boldsymbol{\mathcal{A}}) = \prod_{c \in \boldsymbol{M}} p(\boldsymbol{\pi}_c | a_c, b_c) p(\boldsymbol{z}_c | G_c) \cdot \prod_{c \in \boldsymbol{N}} p(\boldsymbol{\theta}_c | \alpha_c) \prod_{x_d \in \boldsymbol{X}} p(x_d, t_d | \boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{z}),$$

where x_d and t_d represent the d^{th} observed string and its corresponding parse.

Algorithm 5 Generative process of adaptor grammar.

```
For nonterminals $c \in \boldsymbol{N}$, draw rule probabilities $\boldsymbol{\theta}_c \sim \text{Dir}(\boldsymbol{\alpha}_c)$ for PCFG $\boldsymbol{\mathcal{G}}$.
for adapted nonterminal $c$ in $c_1, \ldots, c_{|\boldsymbol{M}|}$ do
    Draw grammaton $H_c \sim \text{PYGEM}(a_c, b_c, G_c)$ according to 6.7, where $G_c$ is defined by the PCFG rules $\boldsymbol{R}$.
end for
For $i \in \{1, \ldots, D\}$, generate a phrase-structure tree $t_{S,i}$ by constantly sampling grammar rule from set of PCFG rules $\boldsymbol{R}(e)$ at non-adapted nonterminal $e$ and the grammatons $H_c$ at adapted nonterminals $c$.
The yields of trees $t_1, \ldots, t_D$ are observations $x_1, \ldots, x_D$.
```

The multinomial PCFG parameter $\boldsymbol{\theta}_c$ is drawn from a Dirichlet distribution at nonterminal $c \in \boldsymbol{N}$. At each adapted nonterminal $c \in \boldsymbol{M}$, the stick-breaking weights $\boldsymbol{\pi}_c$ are drawn from a PYGEM (Equation 6.7). Each weight has an associated atom $z_{c,i}$ from base distribution G_c, a subtree rooted at c. The probability $p(x_d, t_d \,|\, \boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{z})$ is the PCFG likelihood of yield x_d with parse tree t_d.

8.2 Hybrid Variational-MCMC Inference

Discovering the latent variables of the model—trees, adapted probabilities, and PCFG rules—is a problem of posterior inference given observed data. Previous approaches use MCMC (Johnson et al., 2007) or variational inference (Cohen et al., 2010).

MCMC discovers the support of nonparametric models during the inference, but does not scale to larger datasets (due to tight coupling of variables). Variational inference, however, is inherently parallel and easily amendable to online inference, but requires preprocessing to discover the adapted productions. We combine the best of both worlds and propose a hybrid variational-MCMC inference algorithm for adaptor grammars.

Variational inference—as introduced in Section 2.3.1—posits a variational distribution over the latent variables in the model; this in turn induces an "evidence lower bound" (ELBO, $\mathcal{L}$) as a function of a variational distribution q, a lower bound on the marginal log-likelihood. Variational inference optimizes this objective function with respect to the parameters that define q.

In this section, we derive coordinate-ascent updates for these variational parameters. A key mathematical component is taking expectations with respect to the variational distribution q. We strategically use MCMC sampling to compute the expectation of q *over parse trees* z. Instead of explicitly computing the variational distribution for all parameters, one can sample from it (Section 4.2). This produces a sparse approximation of the variational distribution, which benefits both scalability and performance Mimno et al. (2012). Moreover, because the sparse representation can flexibly adjust the support for the Pitman-Yor process, it is a necessary prerequisite to online inference (Section 8.3).

8.2.1 Variational Lower Bound

We posit a mean-field variational distribution:

$$q(\boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{T}|\boldsymbol{\gamma}, \boldsymbol{\nu}, \boldsymbol{\phi}) = \prod_{c\in\boldsymbol{M}}\prod_{i=1}^{\infty} q(\pi'_{c,i}|\nu^1_{c,i}, \nu^2_{c,i}) \cdot \prod_{c\in\boldsymbol{N}} q(\boldsymbol{\theta}_c|\boldsymbol{\gamma}_c) \prod_{x_d\in\boldsymbol{X}} q(t_d|\boldsymbol{\phi}_d), \tag{8.1}$$

where $\pi'_{c,i}$ is drawn from a variational Beta distribution parametrized by $\nu^1_{c,i}, \nu^2_{c,i}$; and $\boldsymbol{\theta}_c$ is from a variational Dirichlet prior $\boldsymbol{\gamma}_c \in \mathbb{R}_+^{|\boldsymbol{R}(c)|}$.[4] Index i ranges over a possibly infinite number of adapted rules. The parse for the d^{th} observation, t_d is modeled by a multinomial $\boldsymbol{\phi}_d$, where $\phi_{d,i}$ is the probability generating the i^{th} phrase-structure tree $t_{d,i}$.

The variational distribution over latent variables induces the following ELBO on the likelihood:

$$\begin{aligned}\mathcal{L}(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{T}, \boldsymbol{D}; \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\alpha}) = {} & H[q(\boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{T})] + \textstyle\sum_{c \in \boldsymbol{N}} \mathbb{E}_q[\log p(\boldsymbol{\theta}_c | \boldsymbol{\alpha}_c)] \\ & + \textstyle\sum_{c \in \boldsymbol{M}} \sum_{i=1}^{\infty} \mathbb{E}_q[\log p(\pi'_{c,i} | a_c, b_c)] + \textstyle\sum_{c \in \boldsymbol{M}} \sum_{i=1}^{\infty} \mathbb{E}_q[\log p(z_{c,i} \,|\, \boldsymbol{\pi}, \boldsymbol{\theta})] \\ & + \textstyle\sum_{x_d \in \boldsymbol{X}} \mathbb{E}_q[\log p(x_d, t_d \,|\, \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{z})],\end{aligned} \tag{8.2}$$

where $H[\bullet]$ is the entropy function.

To make this lower bound tractable, we truncate the distribution over π to a finite set (Blei & Jordan, 2005) for each adapted nonterminal $c \in \boldsymbol{M}$, i.e., $\pi'_{c,K_c} \equiv 1$ for some index K_c. Because the atom weights π_k are deterministically defined by Equation 6.7, this implies that $\pi_{c,i}$ is zero beyond index K_c. Each weight $\pi_{c,i}$ is associated with an atom $z_{c,i}$, a subtree rooted at c. We call the ordered set of $z_{c,i}$ the *truncated nonterminal grammaton* (TNG). Each adapted nonterminal $c \in \boldsymbol{M}$ has its own TNG$_c$. The i^{th} subtree in TNG$_c$ is denoted TNG$_c(i)$.

[4] Note that the variable $\boldsymbol{\phi}$ and $\boldsymbol{\gamma}$ are different from the notation we used in all previous chapters. In the context of LDA, $\boldsymbol{\gamma}$ refers to the variational Dirichlet parameters for topic distribution per document, and $\boldsymbol{\phi}$ refers to the variational multinomial parameters for topic distribution of each word in a documents. In the context of adaptor grammars, $\boldsymbol{\gamma}$ refers to the variational Dirichlet parameters for PCFG, and $\boldsymbol{\phi}$ refers to the variational multinomial parameters for the parse trees.

In the rest of this section, we describe approximate inference to maximize $\mathcal{L}$. The most important update is $\phi_{d,i}$, which we update using stochastic MCMC inference (Section 8.2.2). Past variational approaches for adaptor grammars (Cohen et al., 2010) rely on a preprocessing step and heuristics to define a *static* TNG. In contrast, our model dynamically discovers trees. The TNG grows as the model sees more data, allowing online updates (Section 8.3).

The remaining variational parameters are optimized using expected counts of adaptor grammar rules. These expected counts are described in Section 8.2.3, and the variational updates for the variational parameters excluding $\phi_{d,i}$ are described in Section 8.2.4.

8.2.2 Stochastic MCMC Inference

Each observation x_d has an associated variational multinomial distribution $\boldsymbol{\phi}_d$ over trees $\boldsymbol{t}_d$ that can yield observation x_d with probability $\phi_{d,i}$. Holding all other variational parameters fixed, the coordinate-ascent update (Mimno et al., 2012; Bishop, 2006) for $\phi_{d,i}$ is

$$\phi_{d,i} \propto \exp\{\mathbb{E}_q^{\neg\boldsymbol{\phi}_d}[\log p(t_{d,i}|x_d, \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{z})]\}, \tag{8.3}$$

where $\phi_{d,i}$ is the probability generating the i^{th} phrase-structure tree $t_{d,i}$ and $\mathbb{E}_q^{\neg\boldsymbol{\phi}_d}[\bullet]$ is the expectation with respect to the variational distribution q, excluding the value of $\phi_{d,i}$.

Instead of computing this expectation explicitly, we turn to stochastic variational inference (Section 4.2) to sample from this distribution. This produces a set of sampled trees $\boldsymbol{\sigma}_d \equiv \{\sigma_{d,1}, \ldots, \sigma_{d,k}\}$. From this set of trees we can approximate our variational distribution over trees $\boldsymbol{\phi}$ using the empirical distribution $\boldsymbol{\sigma}_d$, i.e.,

$$\phi_{d,i} \propto \mathbb{I}[\sigma_{d,j} = t_{d,i}, \forall \sigma_{d,j} \in \boldsymbol{\sigma}_d]. \tag{8.4}$$

This leads to a sparse approximation of variational distribution $\boldsymbol{\phi}$.[5]

Sampling requires a derived PCFG $\boldsymbol{\mathcal{G}'}$ that approximates the distribution over tree derivations conditioned on a yield. It includes the original PCFG rules $\boldsymbol{R} = \{c \rightarrow \beta\}$ that define the base distribution and the new adapted productions $\boldsymbol{R}' = \{c \Rightarrow z, z \in \textsc{tng}_c\}$. Under $\boldsymbol{\mathcal{G}'}$, the probability θ' of adapted production $c \Rightarrow z$ is

$$\log \theta'_{c\Rightarrow z} = \begin{cases} \mathbb{E}_q[\log \pi_{c,i}], & \text{if } \textsc{tng}_c(i) = z \\ \mathbb{E}_q[\log \pi_{c,K_c}] + \mathbb{E}_q[\log \theta_{c\Rightarrow z}], \text{otherwise} \end{cases} \tag{8.5}$$

where K_c is the truncation level of $\textsc{tng}_c$ and π_{c,K_c} represents the left-over stick weights in the stick-breaking process for adaptor $c \in \boldsymbol{M}$. Variable $\theta_{c\Rightarrow z}$ represents the probability of generating tree $c \Rightarrow z$ under the base distribution.

The expectation of the Pitman-Yor multinomial $\pi_{c,i}$ under the truncated variational stick-breaking distribution is

$$\mathbb{E}_q[\log \pi_{a,i}] = \Psi(\nu^1_{a,i}) - \Psi(\nu^1_{a,i} + \nu^2_{a,i}) + \textstyle\sum_{j=1}^{i-1}(\Psi(\nu^2_{a,j}) - \Psi(\nu^1_{a,j} + \nu^2_{a,j})), \tag{8.6}$$

[5] In our experiments, we use ten samples.

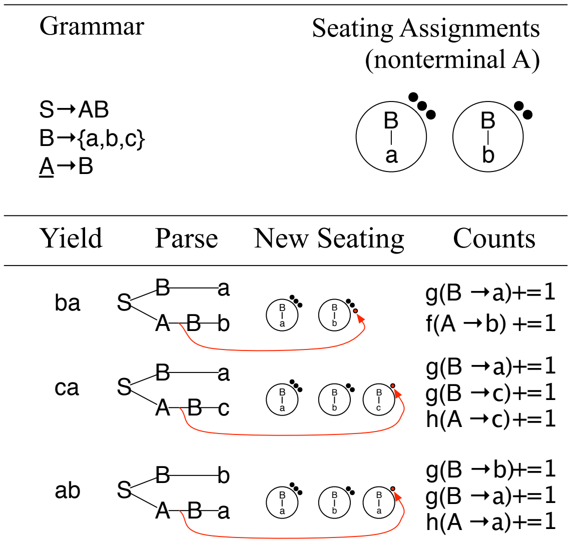


Figure 8.1: Given an adaptor grammar, we sample derivations given an approximate PCFG and show how these affect counts. The sampled derivations can be understood via the Chinese restaurant metaphor (Johnson et al., 2007). Existing cached rules (elements in the TNG) can be thought of as occupied tables; this happens in the case of the yield "ba", which increases counts for unadapted rules g and for entries in TNG_A, f. For the yield "ca", there is no appropriate entry in the TNG, so it must use the base distribution, which corresponds to sitting at a new table. This generates counts for g, as it uses the unadapted rule and for h, which represents entries that could be included in the TNG in the future. The final yield, "ab", shows that even when compatible entries are in the TNG, it might still create a new table, changing the underlying base distribution.

and the expectation of generating the phrase-structure tree $a \Rightarrow z$ based on PCFG productions under the variational Dirichlet distribution is

$$\mathbb{E}_q[\log \theta_{a\Rightarrow z}] = \textstyle\sum_{c\to\beta\in a\Rightarrow z} \left(\Psi(\gamma_{c\to\beta}) - \Psi(\textstyle\sum_{c\to\beta'\in \boldsymbol{R}_c} \gamma_{c\to\beta'})\right) \qquad (8.7)$$

where $\Psi(\bullet)$ is the digamma function, and $c \to \beta \in a \Rightarrow z$ represents all PCFG productions in the phrase-structure tree $a \Rightarrow z$.

This PCFG can compose arbitrary subtrees and thus discover new trees that better describe the data, even if those trees are not part of the TNG. This is equivalent to "creating a new" table in MCMC inference and provides *truncation-free* variational updates (Wang & Blei, 2012) by sampling a unseen subtree with adapted nonterminal $c \in \boldsymbol{M}$ at the root. This frees our model from preprocessing to initialize truncated grammatons in Cohen et al. (2010). This stochastic approach has the advantage of creating sparse distributions (Wang & Blei, 2012): few unique trees will be represented.

8.2.3 Calculating Expected Rule Counts

For every observation x_d, the hybrid approach produces a set of sampled trees, each of which contains three types of productions: adapted rules, original PCFG rules, and potentially adapted rules. The last set is most important, as these are new rules discovered by the sampler. These are explained using the Chinese restaurant metaphor in Figure 8.1. The multiset of all adapted productions is $M(t_{d,i})$ and the multiset of non-adapted productions that generate tree $t_{d,i}$ is $N(t_{d,i})$. We compute three counts:

1: f is the expected number of productions within the TNG. It is the sum over the probability of a tree $t_{d,k}$ times the number of times an *adapted* production appeared in $t_{d,k}$,

$$f_d(a \Rightarrow z_{a,i}) = \textstyle\sum_k \left(\phi_{d,k} \underbrace{|a \Rightarrow z_{a,i} : a \Rightarrow z_{a,i} \in M(t_{d,k})|}_{\text{count of rule } a \Rightarrow z_{a,i} \text{ in tree } t_{d,k}}\right).$$

2: g is the expected counts of PCFG productions $\boldsymbol{R}$ that defines the *base* distribution of the adaptor grammar,

$$g_d(a \to \beta) = \textstyle\sum_k \left(\phi_{d,k} \left| a \to \beta : a \to \beta \in N(t_{d,k}) \right|\right).$$

3: Finally, a third set of productions are newly discovered by the sampler and not in the TNG. These subtrees are rules that *could be adapted*, with expected counts

$$h_d(c \Rightarrow z_{c,i}) = \textstyle\sum_k \left(\phi_{d,k} \left| c \Rightarrow z_{c,i} : c \Rightarrow z_{c,i} \notin M(t_{d,k}) \right|\right).$$

These subtrees—lists of PCFG rules sampled from Equation 8.5—correspond to adapted productions not yet present in the TNG. Once these rules are added to the TNG, their h counts become f counts.

8.2.4 Variational Updates

Given the sparse vectors $\boldsymbol{\phi}$ sampled from the hybrid MCMC step, we update all variational parameters using gradient descent:

$$\begin{aligned}
\gamma_{a\to\beta} =& \alpha_{a\to\beta} + \textstyle\sum_{x_d \in \boldsymbol{X}} g_d(a \to \beta) + \sum_{b \in \boldsymbol{M}} \sum_{i=1}^{K_b} n(a \to \beta, z_{b,i}), \\
\nu^1_{a,i} =& 1 - b_a + \textstyle\sum_{x_d \in \boldsymbol{X}} f_d(a \Rightarrow z_{a,i}) + \sum_{b \in \boldsymbol{M}} \sum_{k=1}^{K_b} n(a \Rightarrow z_{a,i}, z_{b,k}), \\
\nu^2_{a,i} =& a_a + i b_a + \textstyle\sum_{x_d \in \boldsymbol{X}} \sum_{j=1}^{K_a} f_d(a \Rightarrow z_{a,j}) + \sum_{b \in \boldsymbol{M}} \sum_{k=1}^{K_b} \sum_{j=1}^{K_a} n(a \Rightarrow z_{a,j}, z_{b,k}),
\end{aligned}$$

where $n(r, t)$ is the expected number of times production r is in tree t, collected during sampling. For the detailed derivations of all variational parameters, please refer to Appendix A.

Hyperparameter Update We update our PCFG hyperparameter $\boldsymbol{\alpha}$, PYGEM hyperparameters $\boldsymbol{a}$ and $\boldsymbol{b}$ as in Cohen et al. (2010).

8.3 Online Variational Inference

Online inference for probabilistic models requires us to update our posterior distribution as new observations arrive. Unlike batch inference algorithms, we do not assume we always have access to the entire dataset. Instead, as stated in Chapter 7, we assume that observations arrive in small groups called *minibatches*. The advantage of online inference is threefold: a) it does not require retaining the whole dataset in memory; b) each online update is fast; and c) the model often converges faster empirically (Hoffman et al., 2010). All of these make adaptor grammars scalable to larger datasets.

Our approach is based on the stochastic variational inference for topic models (Hoffman et al., 2013). This inference strategy uses a form of stochastic gradient descent (Bottou, 1998): using the gradient of the ELBO, it finds the sufficient statistics necessary to update variational parameters (which are mostly expected counts calculated using the inside-outside algorithm), and interpolates the result with the current model.

We assume data arrive in minibatches $\boldsymbol{B}$ (a set of sentences). We accumulate expected counts

$$\tilde{f}^{(l)}(a \Rightarrow z_{a,i}) = (1-\epsilon) \cdot \tilde{f}^{(l-1)}(a \Rightarrow z_{a,i}) + \epsilon \cdot \frac{|\boldsymbol{X}|}{|\boldsymbol{B}_l|} \sum_{x_d \in \boldsymbol{B}_l} f_d(a \Rightarrow z_{a,i}), \tag{8.8}$$

$$\tilde{g}^{(l)}(a \rightarrow \beta) = (1-\epsilon) \cdot \tilde{g}^{(l-1)}(a \rightarrow \beta) + \epsilon \cdot \frac{|\boldsymbol{X}|}{|\boldsymbol{B}_l|} \sum_{x_d \in \boldsymbol{B}_l} g_d(a \rightarrow \beta), \tag{8.9}$$

with *decay factor* $\epsilon \in (0, 1)$ to guarantee convergence. We set it to $\epsilon = (\tau + l)^{-\kappa}$, where l is the minibatch counter. The *decay inertia* τ prevents premature convergence, and *decay rate* κ controls the speed of change in sufficient statistics (Hoffman et al., 2010). We recover batch variational approach when $\boldsymbol{B} = \boldsymbol{D}$ and $\kappa = 0$.

The variables $\tilde{f}^{(l)}$ and $\tilde{g}^{(l)}$ are accumulated sufficient statistics of adapted and unadapted productions after processing minibatch $\boldsymbol{B}_l$. They update the approximate gradient. The updates for variational parameters become

$$\gamma_{a \rightarrow \beta} = \alpha_{a \rightarrow \beta} + \tilde{g}^{(l)}(a \rightarrow \beta) + \sum_{b \in \boldsymbol{M}} \sum_{i=1}^{K_b} n(a \rightarrow \beta, z_{b,i}), \tag{8.10}$$

$$\nu^1_{a,i} = 1 - b_a + \tilde{f}^{(l)}(a \Rightarrow z_{a,i}) + \sum_{b \in \boldsymbol{M}} \sum_{k=1}^{K_b} n(a \Rightarrow z_{a,i}, z_{b,k}), \tag{8.11}$$

$$\nu^2_{a,i} = a_a + ib_a + \textstyle\sum_{j=1}^{K_a} \tilde{f}^{(l)}(a \Rightarrow z_{a,j}) + \sum_{b \in \boldsymbol{M}} \sum_{k=1}^{K_b} \sum_{j=1}^{K_a} n(a \Rightarrow z_{a,j}, z_{b,k}), \tag{8.12}$$

where K_a is the size of the TNG at adaptor $a \in \boldsymbol{M}$.

8.3.1 Refining the Truncation

As we observe more data during inference, our TNGs need to change to capture more patterns, since the expected number of total adapted rules for any adapted nonterminal under the Pitman-Yor process is proportional to the log value of the total data. This is one of the critical differences of our online hybrid inference technique to the batch mode variational inference method (Cohen et al., 2010). By allowing the TNGs to grow, our approach does not require the preprocessing step, instead, it is able to adjust the supports of the Pitman-Yor process over time based on the data. In additional to adding new rules to TNG, we also need to remove useless rules to prevent the TNG growing unbounded, and update the derivations for existing rules over time. In this section, we describe heuristics for performing each of these operations.

Adding Productions Sampling can identify productions that are not adapted but were instead drawn from the base distribution. These are candidates for the TNG. For every nonterminal a, we add these potentially adapted productions to TNG_a after each minibatch. The count associated with candidate productions is now associated with an adapted production, i.e., the h count contributes to the relevant f count. This mechanism dynamically expands TNG_a.

Sorting and Removing Productions Our model does not require a preprocessing step to initialize the TNGs, rather, it constructs and expands all TNGs on the fly. To prevent the TNG from growing unwieldy, we prune TNG after every u minibatches. As a result, we need to impose an ordering over all the parse trees in the TNGs to establish a ranking on how useful each adapted rule is and to remove the less useful rules. The underlying PYGEM distribution implicitly places an ranking over all the atoms according to their corresponding sufficient statistics (Kurihara et al., 2007), as shown in 8.8. It measures the "usefulness" of every adapted production throughout inference process.

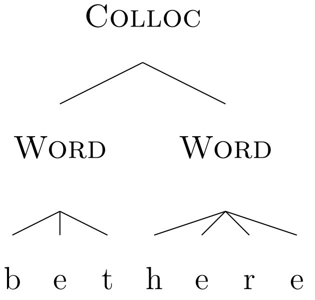
In addition to accumulated sufficient statistics, Cohen et al. (2010) add a secondary term to discourage short constituents (Mochihashi et al., 2009). We impose a reward term for longer phrases in addition to $\tilde{f}$ and sort all adapted productions in $\textsc{tng}_a$ using the ranking score

$$\Lambda(a \Rightarrow z_{a,i}) = \tilde{f}^{(l)}(a \Rightarrow z_{a,i}) \times \log(\epsilon \cdot |s| + 1),$$

where $|s|$ is the number of yields in production $a \Rightarrow z_{a,i}$. Because ϵ decreases each minibatch, the reward for long phrases diminishes. This is similar to an annealed version of Cohen et al. (2010) – where the reward for long phrases is fixed, see also Mochihashi et al. (2009). After sorting, we remove all but the top K_a adapted productions.

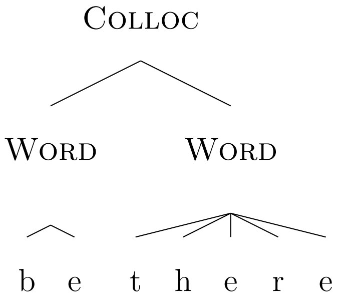
Rederiving Adapted Productions For MCMC inference, Johnson & Goldwater (2009) observe that atoms already associated with a yield may have trees that do not explain their yield well. They propose *table label resampling* to rederive yields. When dealing with hierarchical grammars, e.g, *collocation* grammars, the adaptor grammar sampler itself maintains a hierarchy of Chinese Restaurant Processes or Pitman-Yor Processes, one per adapted nonterminal, to cache the adapted rules, each of which is a derivation tree. The idea of table label resampling is to resample these derivation trees in these adaptors—table labels in the restaurant metaphor. It potentially changes the analysis of many sentences at once.

For example, each Colloc in the *collocation* grammar can occur in many Sent, and each Word can occur in many Colloc. There are many possible derivation trees available for Colloc ↦ "bethere". The current derivation of this collocation is



(8.13)

Table label resampling may alter the structure of this derivation tree, e.g.,



(8.14)

It can subsequently change the way it is analyzed into Word, thus changing the analysis of all of the Sent containing that Colloc.

In our approach this is equivalent to "mutating" some derivations in a tng. After pruning rules every u minibatches, we perform table label resampling for

Algorithm 6 Online inference for adaptor grammars

```
Random initialize all variational parameters.
for minibatch of $l = 1, 2, \ldots$ do
    Construct approximate PCFG $\boldsymbol{\theta}'$ of $\boldsymbol{\mathcal{A}}$ as in Eq. 8.5.
    for input sentence $d = 1, 2, \ldots, D_l$ do
        Accumulate inside probabilities from approximate PCFG $\boldsymbol{\theta}'$.
        Sample phrase-structure trees $\boldsymbol{\sigma}$ and update the tree distribution $\boldsymbol{\phi}$ (Equation 8.4).
    end for
    For every adapted nonterminal $c$, append adapted productions to TNG$_c$.
    Accumulate sufficient statistics (Equation 8.8 and 8.9).
    Update $\boldsymbol{\gamma}$, $\boldsymbol{\nu}^1$, and $\boldsymbol{\nu}^2$ (Equation 8.10-8.12).
    Refine and prune the truncation every $u$ minibatches.
end for
```

adapted nonterminals from general to specific (i.e., a topological sort). This provides better expected counts $n(r, \bullet)$ for rules used in phrase-structure subtrees. Empirically, we find table label resampling only marginally improves the word-segmentation result.

Initialization Our inference begins with random variational Dirichlets and empty TNGs, which obviates the preprocessing step in Cohen et al. (2010). Our model constructs and expands all TNGs on the fly. It mimics the *incremental initialization* of Johnson & Goldwater (2009). Algorithm 6 summarizes the pseudo-code of our online approach.

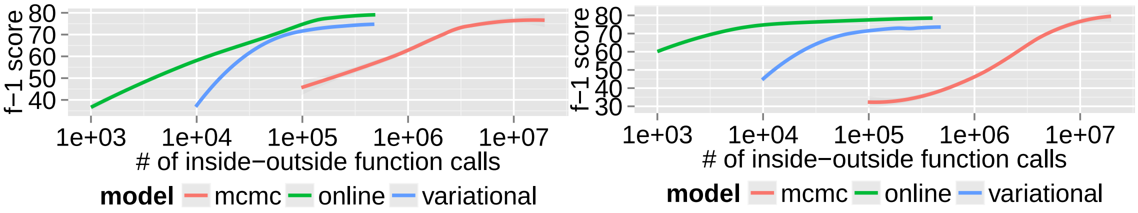
8.3.2 Complexity

Inside and outside algorithm calls dominate execution time for adaptor grammar inference. Comparing to both MCMC sampling and hybrid inference approaches—which only need to compute the inside algorithm and then sample parse trees out of the distribution—the variational approach needs to compute inside-outside algorithms and estimate the expected counts for every possible tree derivation (Cohen et al., 2010). For a dataset with D observations, variational inference requires $O\big(DI\big)$ calls to *inside-outside* algorithm, where I is the number of iterations, typically in the tens.

In contrast, MCMC only needs to accumulate inside probabilities, and then sample a tree derivation (Chappelier & Rajman, 2000). The sampling step is negligible in processing time compare to the inside algorithm. MCMC inference requires $O\big(DI\big)$ calls to the *inside* algorithm—hence every iteration is much faster than variational approach—but I is usually on the order of thousands.

Likewise, our hybrid approach also only needs the less expensive inside algorithm to sample trees. Comparing to the batch mode variational inference (Cohen et al., 2010) which requires the complete inside-outside algorithm, our online hybrid inference only inside algorithm. In addition, our approach can achieve reasonable results with only a single pass through the data. And thus only requires $O(D)$ calls to the *inside* algorithm.

Because the inside-outside algorithm is fundamental to each of these algorithms, we will use it as a common basis for comparison across different implementations. This is over-generous to variational approaches, as the full inside-outside computation is more expensive than the inside probability computation required for sampling.



(a) *unigram* grammar.

(b) *collocation* grammar.

Figure 8.2: Word segmentation accuracy measured by word token F_1 scores on *brent* corpus of three approaches against number of inside-outside function call using *unigram* (upper) and *collocation* (lower) grammars in Table 8.1.[7]

Unigram			Collocation			Infinite LDA			
Sent	↦	Words	Sent	↦	Colloc Collocs	Sent	↦	Doc$_j$	$j = 1, \dots, D$
Words	↦	Word Words	Sent	↦	Colloc	Doc$_j$	↦	$_{-j}$ Topic$_i$	$j = 1, \dots, D;$
Word	↦	Chars	Colloc	↦	Word Words				$i = 1, \dots, K$
Chars	↦	Char Chars	Colloc	↦	Words	Topic$_i$	↦	Word	$i = 1, \dots, K$
Chars	↦	Char	Words	↦	Word Words	Word	↦	Chars	
			Word	↦	Word	Chars	↦	Char Chars	
			Word	↦	Chars	Chars	↦	Char	
			Chars	↦	Char Chars				
			Chars	↦	Char				

Table 8.1: Grammars used in our experiments. For all grammars, the nonterminal Char is a non-adapted rule that expands to all characters used in the data. For Chinese segmentation, all Chinese character; for topic modeling, all Latin characters. Following Johnson & Goldwater (2009), we underline adapted nonterminals. For the topic model grammar, D is the total number of strings and K is the number of topics.

8.4 Experiments and Discussion

We implement our online adaptor grammar model (online) in Python[8] and compare it against both MCMC (Johnson & Goldwater, 2009, MCMC) and the variational inference (Cohen et al., 2010, variational). We use the released implementation of MCMC sampler for adaptor grammars,[9] and simulate variational approach using our implementation by setting the minibatch size $\boldsymbol{B} = \boldsymbol{D}$ and

[7] Our online settings are batch size $B = 20$, decay inertia $\tau = 128$, decay rate $\kappa = 0.6$ for *unigram* grammar; and minibatch size $B = 5$, decay inertia $\tau = 256$, decay rate $\kappa = 0.8$ for *collocation* grammar. TNGs are refined at interval $u = 50$. Truncation size is set to $K_{\mathsf{Word}} = 1.5k$ and $K_{\mathsf{Colloc}} = 3k$. The settings are chosen from cross validation. We observe similar behavior under $\kappa = \{0.7, 0.9, 1.0\}$, $\tau = \{32, 64, 512\}$, $B = \{10, 50\}$ and $u = \{10, 20, 100\}$.

[8] Implementation available at `http://www.umiacs.umd.edu/~zhaike/`.

[9] `http://web.science.mq.edu.au/~mjohnson/code/py-cfg-2013-02-25.tgz`

$\kappa = 0$.[10] For MCMC approach, we use the best settings reported in Johnson & Goldwater (2009) with *incremental initialization* and *table label resampling.*

We examine our online adaptor grammar on two different aspects—the scalability and flexibility. In Section 8.4.1, we look into the scalability of our proposed approach, and compare it against other batch learning techniques. In Section 8.4.2, we use the online topic models with infinite vocabulary—which we discussed in Chapter 7—as an example, and demonstrate how to use our online adaptor grammars to quickly prototype it.[11] This can be easily applied to other online Bayesian nonparametric models without loss of generality.

8.4.1 Word Segmentation

We evaluate our online adaptor grammar on the task of word segmentation, which focuses on identifying word boundaries from a sequence of characters. This is especially the case for Chinese, since characters are written in sequence without word boundaries.

We first evaluate all three models on the standard Brent version of the Bernstein-Ratner corpus (Bernstein-Ratner, 1987; Brent & Cartwright, 1996, *brent*). The dataset contains $10k$ sentences, $1.3k$ distinct words, and 72 distinct characters. We compare the results on both *unigram* and *collocation* grammars listed in Table 8.1

[10] Note that this is not exactly the same as the variational inference approach proposed by Cohen et al. (2010), instead, we are using hybrid inference, i.e., only compute the inside algorithm during parsing, and approximate the variational distribution by taking samples of parse trees.

[11] The main purpose of this section is to demonstrate the flexibility of our proposed online adaptor grammars, in terms of quickly prototyping or validating a particular Bayesian nonparametric model, without spending time on deriving the inference. The adaptor grammar approach is slower than the method we discussed in Chapter 7, but it is much easier to get running without going through the parameter derivation process.

introduced in Johnson & Goldwater (2009).

Figure 8.2 illustrates the word segmentation accuracy in terms of word token F_1-scores on *brent* against the number of inside-outside function calls for all three approaches using *unigram* and *collocation* grammars. In both cases, our ONLINE approach converges faster than MCMC and VARIATIONAL approaches, yet yields comparable or better performance when seeing more data.

In addition to the *brent* corpus, we also evaluate three approaches on three other Chinese datasets compiled by Xue et al. (2005) and Emerson (2005):[12]

- Chinese Treebank 7.0 (*ctb7*): 162*k* sentences, 57*k* distinct words, 4.5*k* distinct characters;
- Peking University (*pku*): 183*k* sentences, 53*k* distinct words, 4.6*k* distinct characters; and
- City University of Hong Kong (*cityu*): 207*k* sentences, 64*k* distinct words, and 5*k* distinct characters.

We compare our inference method against other approaches on F_1 score. While other unsupervised word segmentation systems are available (Mochihashi et al. (2009), inter alia),[13] our focus is on a direct comparison of inference techniques for adaptor grammar, which achieve competitive (if not state-of-the-art) performance.

Table 8.2 shows the word token F_1-scores and negative likelihood on held-out test dataset of our model against MCMC and VARIATIONAL approach. For held-out test data, we randomly sample 30% of the sentences for testing and the rest for

[12] We use all punctuation as natural delimiters (i.e., words cannot cross punctuation).

[13] Their results are not directly comparable: they use different subsets and assume different preprocessing.

training. We compute the held-out likelihood of the most likely sampled parse trees out of each model.[14] Our ONLINE approach consistently better segments words than VARIATIONAL and achieves comparable or better results than MCMC.

For MCMC, Johnson & Goldwater (2009) discovered that *incremental initialization*—or online updates in general—results in more accurate word segmentation, even though the trees have lower posterior probability. Similar to that, our ONLINE approach initializes and learns them on the fly, instead of initializing the grammatons and parse trees for all data upfront as for VARIATIONAL. This uniformly outperforms batch initialization on the word segmentation tasks.

[14] Note that this is only an approximation to the true held-out likelihood, since it is impossible to enumerate all the possible parse trees and hence compute the likelihood for a given sentence under the model.

Model and Settings			*ctb7*		*pku*		*cityu*	
			unigram	*collocation*	*unigram*	*collocation*	*unigram*	*collocation*
MCMC		500 iter	72.70 (2.81)	50.53 (2.82)	72.01 (2.82)	49.06 (2.81)	74.19 (3.55)	63.14 (3.53)
		1000 iter	72.65 (2.83)	62.27 (2.79)	71.81 (2.81)	62.47 (2.77)	74.37 (3.54)	70.62 (3.51)
		1500 iter	72.17 (2.80)	69.65 (2.77)	71.46 (2.80)	70.20 (2.73)	74.22 (3.54)	72.33 (3.50)
		2000 iter	71.75 (2.79)	71.66 (2.76)	71.04 (2.79)	72.55 (2.70)	74.01 (3.53)	73.15 (3.48)
ONLINE	κ	τ	$K_{\mathsf{Word}} = 30k$	$K_{\mathsf{Colloc}} = 100k$	$K_{\mathsf{Word}} = 40k$	$K_{\mathsf{Colloc}} = 120k$	$K_{\mathsf{Word}} = 50k$	$K_{\mathsf{Colloc}} = 150K$
	0.6	32	70.17 (2.84)	68.43 (2.77)	69.93 (2.89)	68.09 (2.71)	72.59 (3.62)	69.27 (3.61)
		128	72.98 (2.72)	65.20 (2.81)	72.26 (2.63)	65.57 (2.83)	74.73 (3.40)	64.83 (3.62)
		512	72.76 (2.78)	56.05 (2.85)	71.99 (2.74)	58.94 (2.94)	73.68 (3.60)	60.40 (3.70)
	0.8	32	71.10 (2.77)	70.84 (2.76)	70.31 (2.78)	70.91 (2.71)	73.12 (3.60)	71.89 (3.50)
		128	72.79 (2.64)	70.93 (2.63)	72.08 (2.62)	72.02 (2.63)	74.62 (3.45)	72.28 (3.51)
		512	72.82 (2.58)	68.53 (2.76)	72.14 (2.58)	70.07 (2.69)	74.71 (3.37)	72.58 (3.49)
	1.0	32	69.98 (2.87)	70.71 (2.63)	69.42 (2.84)	71.45 (2.67)	73.18 (3.59)	72.42 (3.45)
		128	71.84 (2.72)	71.29 (2.58)	71.29 (2.67)	72.56 (2.61)	73.23 (3.39)	72.61 (3.41)
		512	72.68 (2.62)	70.67 (2.60)	71.86 (2.63)	71.39 (2.66)	74.45 (3.41)	72.88 (3.38)
VARIATIONAL			69.83 (2.85)	67.78 (2.75)	67.82 (2.80)	66.97 (2.75)	70.47 (3.72)	69.06 (3.69)

Table 8.2: Word segmentation accuracy measured by word token F_1 scores and negative log-likelihood on held-out test dataset in the brackets (lower the better, on the scale of 10^6) for our ONLINE model against MCMC approach Johnson et al. (2007) on various dataset using the *unigram* and *collocation* grammar.[16]

8.4.2 Infinite Vocabulary Topic Modeling

Many topic models can be represented using a PCFG. Nonparametric models such as adaptor grammars can be used to capture extensions such as topical collocations and sticky topics (Johnson, 2010). Still, there is a strong limitation to all these models—the vocabulary is assumed to be fixed, even with online algorithms (Hoffman et al., 2010).

In Chapter 7, we argue that this is a strong constraint that violates the fundamental assumption in online algorithms: new words are introduced as more data are streamed to the algorithm. We also introduce an inference framework, InfVoc, to discover words from a Dirichlet process with a character n-gram base distribution.

We take the best of both worlds, and model a similarly flexible vocabulary using our online adaptor grammar inference algorithm. Our extension to InfVoc generalizes the static character n-gram model (Section 7.2.1), learning the base distribution (i.e., how words are composed from characters) from data.

This is an attractive testbed for our online inference. Within a topic, we can verify that the words we discover are relevant to the topic and that new words rise in importance in the topic over time if they are indeed relevant. For these experiments, we treat each token (with its associated document pseudo-word $_j$) as a single sentence, and each minibatch contains only one sentence (token).

[16] For online inference, we parallelize each minibatch with four threads with settings: batch size $B = 100$ and tng refinement interval $u = 100$. online approach runns for two passes over datasets. variational runs fifty iterations, with the same truncation level as in online. For negative log-likelihood evaluation, we train the model on a random 70% of the data, and hold out the rest

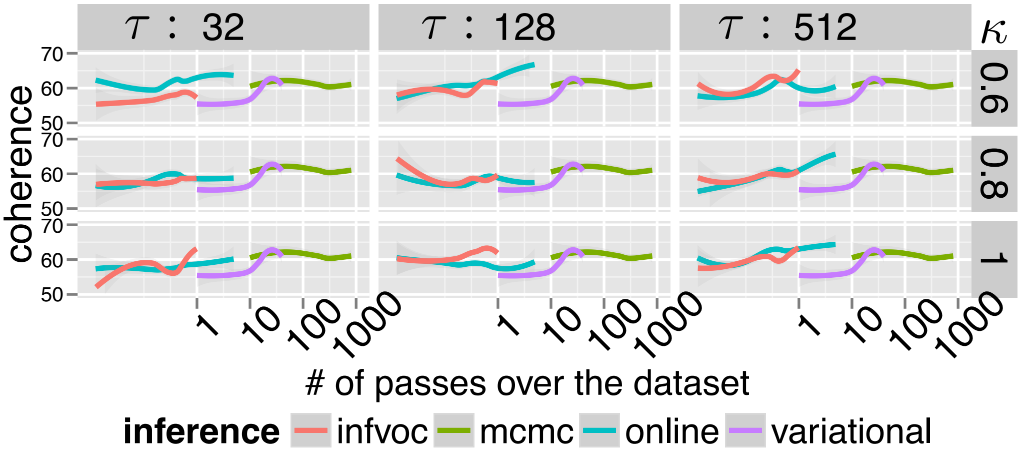


Figure 8.3: The average coherence score of topics on *de-news* datasets against InfVoc approach and other inference techniques (MCMC, variational) under different settings of decay rate κ and decay inertia τ using the *InfVoc LDA* grammar in Table 8.1. The horizontal axis shows the number of passes over the entire dataset.[18]

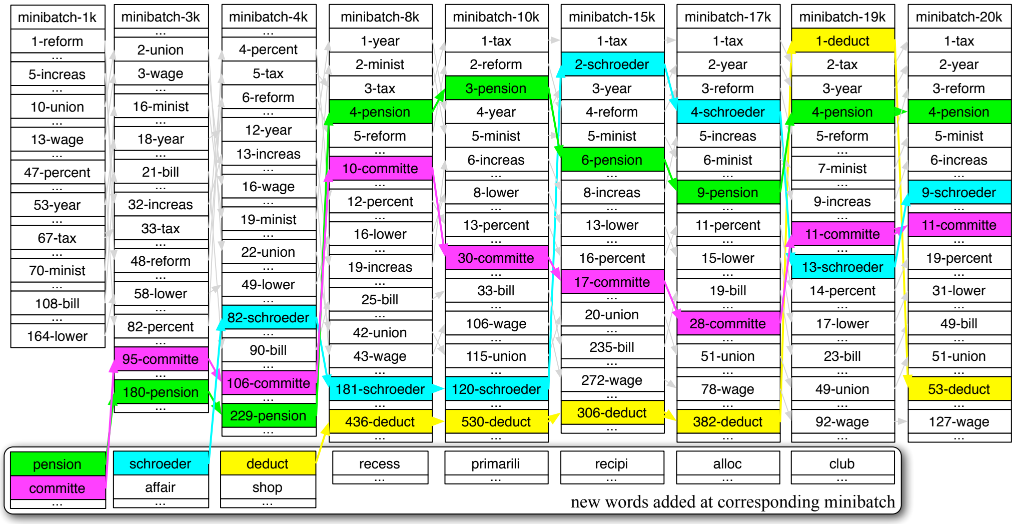


Figure 8.4: The evolution of one topic—concerning tax policy—out of five topics learned using online adaptor grammar inference on the *de-news* dataset. Each minibatch represents a word processed by this online algorithm; time progresses from left to right. As the algorithm encounters new words (bottom) they can make their way into the topic. The numbers next to words represent their overall rank in the topic. For example, the word "pension" first appeared in mini-batch 100, was ranked at 229 after minibatch 400 and became one of the top 10 words in this topic after 2000 minibatches (tokens).[20]

for testing. We observe similar behavior for our model under $\kappa = \{0.7, 0.9\}$ and $\tau = \{64, 256\}$.

[18] We train all models with 5 topics with settings: TNG refinement interval $u = 100$, truncation size $K_{\text{Topic}} = 3k$, and the mini-batch size $B = 50$. We observe a similar behavior under $\kappa \in \{0.7, 0.9\}$ and $\tau \in \{64, 256\}$.

[20] The plot is generated with truncation size $K_{\text{Topic}} = 2k$, mini-batch size $B = 1$, truncation

Quantitatively, we evaluate three different inference schemes and the InfVoc approach[21] on a collection of English daily news snippets (*de-news*).[22] We used the *InfVoc LDA* grammar as shown in Table 8.1. For all approaches, we train the model with five topics, and evaluate how coherent the topics are. This is measured using coherence score (Newman et al., 2009), which correlates well with human understanding about topic interpretability (Chang et al., 2009a). We collect the co-occurrence counts from Wikipedia and compute the average pairwise *pointwise mutual information* (PMI) score between the top 10 ranked words of every topic. Figure 8.3 illustrates the PMI score for both approaches. Our approach yields comparable or better results against all other approaches under most conditions.

Qualitatively, Figure 8.4 shows an example of a topic evolution using online adaptor grammar for the *de-news* dataset. The topic is about “tax policy”. The topic improves over time; words like “year”, “tax” and “minist(er)” become more prominent. More importantly, the online approach discovers new words and incorporates them into the topic. For example, “schroeder” (former chancellor of Germany) first appeared in minibatch 300, was successfully picked up by our model and became one of the top ranked words in the topic.

pruning interval $u = 50$, decay inertia $\tau = 256$, and decay rate $\kappa = 0.8$. All PY hyper-parameters are optimized.

[21] Implementation available at `http://www.umiacs.umd.edu/~zhaike/`.

[22] The *de-news* dataset is randomly selected subset of $2.2k$ documents from `http://homepages.inf.ed.ac.uk/pkoehn/publications/de-news/`. It contains $6.5k$ unique types and over $200k$ word tokens. Tokenization and stemming provided by NLTK (Bird et al., 2009).

8.5 Summary

Probabilistic modeling, particularly generative models, are a useful tool in understanding unstructured data or data where the structure is latent, like language. However, developing these models is often a difficult process, requiring significant machine learning expertise.

In this chapter, we focus on the application of adaptor grammars, which are an expensive but flexible Bayesian nonparametric modeling framework. They offer a flexible tool to quickly prototype and test new models. Despite expensive inference, adaptor grammars have been used for topic modeling (Johnson, 2010), discovering perspective (Hardisty et al., 2010), segmentation (Johnson & Goldwater, 2009), and grammar induction (Cohen et al., 2010).

In this chapter, we presented an online, hybrid inference scheme for adaptor grammars. Unlike previous approaches, it does not require extensive preprocessing. It is also able to faster discover useful structure in text; with further development, these algorithms could further speed the development and application of new nonparametric models to large datasets.

We have explored the effectiveness and efficiency of online learning on topic models (Chapter 7) and adaptor grammars (Chapter 8). It provides one other popular solution—in addition to parallelization approach (Chapter 3)—to scale up probabilistic Bayesian models. In next chapter, we will discuss a few possible future research directions in both modeling and scaling.

Chapter 9

Conclusion

In this dissertation, we show several approaches of scaling up topic models and related probabilistic models. While topic models (Chapter 2) provide an unsupervised way to scan through a collection of unstructured documents, it comes at a price of slow inference. Two popular solutions to speed up inference process are parallelization and online updates. This dissertation focuses on how to apply these two methods to topic models and adaptor grammars. In Section9.1, we recap our contributions throughout this dissertation, and discuss some of the possible future works in Section 9.2. In Section 9.3, we summarize the generalizable knowledge from this dissertation.

9.1 Contributions

In Chapter 3, we parallelize an existing topic model in MapReduce—Mr. LDA.[1] Our implementation relies on previously proposed variational inference method and takes advantage of the independent structure during inference. We show that our implementation is scalable to large datasets and yields better performance than Mahout. We also demonstrate the flexibility of our implementation using two different extensions—informed prior to incorporate human prior knowledge into topics and polylingual LDA to model topics in multilingual environment.

[1] This work has been previously published in Zhai et al. (2012).

Chapter 5 focuses on the area modeling topics in mulitlingual environment. We propose novel polylingual tree-based topic models and develop three different inference schemes to infer the latent parameters.[2] Unlike past approaches which only use monolingual information in multilingual data, our model discovers meaningful topic out of multilingual corpus. We scale up our model using MapReduce and evaluate our model using a downstream task of statistical machine translation. We show significant improvement of 1.2 BLEU on the translation performance.

In addition to the parallelization approach, we also explore the online updating approach to scale up topic models and related Bayesian models. In Chapter 7, we focus on the online streaming approach to scale up topic models and propose a novel online LDA which supports possibly infinite vocabulary.[3] Unlike all past approaches, our model addresses a challenge, but often overlooked problem—the vocabulary is constantly changing and evolving throughout time in online setting. We propose online hybrid inference to correct the inconsistency between the data and variational inference method. We demonstrate our proposed model is able to effectively incorporate new words into vocabulary and discovers more meaningful topics over time.

In Chapter 8, we generalize our online hybrid inference method to adaptor grammars—originally proposed by Johnson et al. (2007)—which are a broader class of Bayesian nonparametric models.[4] We develop online hybrid inference for adaptor grammars. We show that our implementation is able to scale up to much larger

[2] This work has been previously published in Hu et al. (2014).

[3] This work has been previously published in Zhai & Boyd-Graber (2013).

[4] This work has been previously published in Zhai et al. (2014).

datasets than all past approaches.

9.2 Future Work

While we explore the effectiveness and efficiency of both parallelization and online updating inference approaches on topic models and adaptor grammars, there are many other interesting extensions toward this direction, both on scalability and modeling. For example, on scaling up other Bayesian models, these two different approaches are not mutually exclusive, instead, they can be jointly applied to achieve more significant speed-ups. On the modeling side, although adaptor grammars provide us a quick and easy way to prototype new Bayesian nonparametric models, one could further extend the model to encode much richer hierarchies. This promotes Bayesian nonparametric probabilistic models to capture more latent information and structures of the data in a completely data-driven fashion. In this chapter, we expand these ideas and discuss a few possible directions for future extensions.

Distributed Online Learning Two approaches we discussed to scale up an algorithm—parallelization and online learning—have their relative merits and drawbacks, but they are not mutually exclusive. These two methods address different aspects of a Bayesian statistical model. The former one emphasizes on the independent structure and implementation framework, while the latter one focuses more on the internal update mechanism of the model itself. They can be jointly apply to achieve more speed-up Bayesian statistical models. Applying these two approaches together can possibly scale up more complicated Bayesian models (Hu et al., 2012;

Zhai & Williams, 2014) to much larger datasets.

Hierarchical Infinite Vocabulary Topic Models Although the online approach we discussed in Chapter 7 relaxes the assumption of fixed vocabulary in topic models, it uses a static base distribution which is approximated using a language model. Hence, the base distribution is not adaptive as new data come. In addition to explicitly modeling the change of topics over time, it is also possible to model additional structure within a topic. Rather than a fixed, immutable base distribution, modeling each topic with a hierarchical character n-gram model would capture regularities in the corpus that would, for example, allow certain topics to favor different orthographies (e.g., a technology topic might prefer words that start with "i"). While some past topic models have attempted to capture orthography for multilingual applications (Boyd-Graber & Blei, 2009), our approach would be more robust and scalable to larger datasets.

Nonparametric Topic Models Latent Dirichlet allocation (Blei et al., 2003, LDA) assumes a fixed number of topics as well as a pre-defined vocabulary. A few past works have been proposed in relaxing the first constraint and extending LDA into a nonparametric Bayesian model, e.g., *Dirichlet process mixture models* (Blei & Jordan, 2005) and *hierarchical Dirichlet process* (Teh et al., 2006; Wang et al., 2011, HDP). In Chapter 7, we relax the second constraint and are able to dynamically expand or contract the vocabulary on the fly. It is interesting to introduce another level of hierarchy into the model, and extend HDP into a fully nonparametric model.

This additional level of hierarchy further extends the topic distributions as a mixture of possibly infinite number of words, given by the character level N-gram model. By doing this, the underlying topic models are able to infer not only the distribution over words per topic, but also how many topics are presented in the data.

Jointly Model Adaptor Grammar with Infinite PCFG As discussed in Chapter 8, adaptor grammar (Johnson et al., 2007) extends the classic PCFG framework to Bayesian nonparametric. It imposes a Dirichlet process prior on the number of grammar rules, which allows the program to discover new grammar rules. Another alternative approach to "nonparametricfy" PCFG is the *infinite* PCFG (Liang et al., 2007) framework. It imposes a Dirichlet process prior on the categories of non-terminals, which potentially allows infinite number of non-terminals. As we combine the idea of adaptor grammars and infinite PCFG, we are allowing not only the number of rules to expand, but also the types of the non-terminals to increase. This well connects to the previous discussion about nonparametric topic models, as a completely nonparametric topic models would allow both the number of topics as well as the vocabulary size to expand.

9.3 Summary

As hackneyed as the term "big data" has become, researchers and industry alike require algorithms that are scalable and efficient. Bayesian probabilistic models are no different. In summary, this dissertation discusses two different approaches—parallelization and online update—to scale up complicated Bayesian probabilistic models to large datasets.

On the parallelization approach, we argue that variational inference methods for Bayesian probabilistic models are inherently easier to distribute than MCMC approaches. This is due to variational methods break dependencies among latent variables in the model, hence reduce the information to be shared and minimize message synchronization across different machines in the cluster.

In the setting of online streaming approach, data are assumed to arrive constantly over time, which is a common setting for many industrial applications. Due to the nature of data, many Bayesian probabilistic models become *nonparametric*. For example, in batch mode, topic models usually assume the vocabulary is fixed, since all data are available prior to the inference. Therefore, the models are *parametric*—parameter space does not change. However, in online case, such a parametric assumption makes less sense, and the models become nonparametric due to the nature of data. All past variational approaches ignore this natural characteristic either by making the parametric assumption (e.g., a constant vocabulary for topic models) or reducing the problem to finite dimension (e.g., the preprocessing step for adatpor grammars).

In online inference case, as a result, the underlying model should match the charasteristic of data. We propose a online hybrid inference framework which ensures the inference assumptions are met by the model. Our method corrects the inconsistency between the data and many past approaches, i.e., our online hybrid inference is able to "preserve" such the nonparametric nature of data. In addition, our method can be generalized to many other Bayesian probabilistic models as well.

We use topic models and adaptor grammars as examples to demonstrate the effectiveness and efficiency of these two approaches, but the methods can be easily generalized to other models as well. We believe these are appealing approaches to both industrial and academia applications.

Appendix A

Variational Inference for Adaptor Grammars

A.1 Evidence Lower Bound

Recall Equation 8.3, the variational distribution over latent variables induces the following ELBO on the likelihood:

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{z}, \boldsymbol{\pi}, \boldsymbol{\theta}, \boldsymbol{T}, \boldsymbol{D}; \boldsymbol{a}, \boldsymbol{b}, \boldsymbol{\alpha}) = & \underbrace{\sum_{c \in \boldsymbol{M}} \mathbb{H}_q \left[q(\boldsymbol{\theta}_c) \right] + \sum_{c \in \boldsymbol{M}} \sum_{i=1}^{\infty} \mathbb{H}_q \left[q(\pi'_{c,i}) \right]}_{\text{Entropy Terms}} \\
& + \sum_{c \in \boldsymbol{N}} \underbrace{\mathbb{E}_q \left[\log p(\boldsymbol{\theta}_c \,|\, \boldsymbol{\alpha}_c) \right]}_{\text{PCFG rules}} + \sum_{c \in \boldsymbol{M}} \sum_{i=1}^{\infty} \underbrace{\mathbb{E}_q \left[\log p(\pi'_{c,i} \,|\, a_c, b_c) \right]}_{\text{PY stick}} \\
& + \sum_{c \in \boldsymbol{M}} \sum_{i=1}^{\infty} \underbrace{\mathbb{E}_q \left[\log p(z_{c,i} \,|\, \boldsymbol{\pi}, \boldsymbol{\theta}) \right]}_{\text{PY atoms}} + \sum_{d \in \boldsymbol{D}} \underbrace{\mathbb{E}_q \left[\log p(x_d, t_d \,|\, \boldsymbol{\pi}, \boldsymbol{\theta}) \right]}_{\text{Observations}} \qquad \text{(A.1)}
\end{aligned}
$$

where $H[\bullet]$ is the entropy function. We expand each term in the ELBO as following:

PCFG Rule Each nonterminal has a distribution over rules θ_c; the ELBO term associated with this multinomial is

$$
\begin{aligned}
\mathbb{E}_q \left[\log p(\boldsymbol{\theta}_c | \boldsymbol{\alpha}_c) \right] = & \log \Gamma \left(\textstyle\sum_{c \to \beta \in \boldsymbol{R}_c} \alpha_{c \to \beta} \right) - \sum_{c \to \beta \in \boldsymbol{R}_c} \log \Gamma \left(\alpha_{c \to \beta} \right) \\
& + \sum_{c \to \beta \in \boldsymbol{R}_c} (\alpha_{c \to \beta} - 1) \mathbb{E}_q \left[\log \theta_{c \to \beta} \right], \qquad \text{(A.2)}
\end{aligned}
$$

which can be further expanded using the expectation of a Dirichlet,

$$\mathbb{E}_q \left[\log \theta_{c \to \beta}\right] = \Psi\left(\gamma_{c \to \beta}\right) - \Psi\left(\textstyle\sum_{c \to \beta' \in \boldsymbol{R}_c} \gamma_{c \to \beta'}\right) \tag{A.3}$$

PY Stick The Pitman-Yor distribution has two components, a weighting over atoms and the atoms themselves. The ELBO term corresponding to the distribution over atom weights, π, is

$$\begin{aligned}\mathbb{E}_q \left[\log p(\pi'_{c,i} \,|\, a_c, b_c)\right] =& \log \Gamma\left(1 - b_c + a_c + i b_c\right) - \log \Gamma\left(1 - b_c\right) - \log \Gamma\left(a_c + i b_c\right) \\ & - b_c \mathbb{E}_q \left[\log \pi'_{c,i}\right] + (a_c + i b_c - 1)\mathbb{E}_q \left[\log(1 - \pi'_{c,i})\right] \end{aligned} \tag{A.4}$$

PY Atoms The atoms weighted by the Pitman-Yor distribution in the ELBO term

$$\begin{aligned}\mathbb{E}_q \left[\log p(z_{c,i} | \boldsymbol{\pi}, \boldsymbol{\theta})\right] =& \sum_{b \to \beta \in N(z_{c,i})} g(b \to \beta, z_{c,i}) \mathbb{E}_q \left[\log \theta_{b \to \beta}\right] \\ & + \sum_{b \Rightarrow z_{b,k} \in M(z_{c,i})} f(b \Rightarrow z_{b,k}, z_{c,i}) \mathbb{E}_q \left[\log \pi_{b,k}\right]. \end{aligned} \tag{A.5}$$

Observations Finally, observed trees are described by both adapted and unadapted rules which contribute to the ELBO,

$$\mathbb{E}_q \left[\log p(x_d, t_d \,|\, \boldsymbol{\theta}, \boldsymbol{\pi}, \boldsymbol{Z})\right] = \sum_{b \Rightarrow z_{b,k}} \mathbb{E}_q \left[\log p(\pi_{b,k})\right] + \sum_{b \to \beta} \mathbb{E}_q \left[\log p(\theta_{b \to \beta})\right] \tag{A.6}$$

Entropy Terms Entropy terms for Dirichlet distribution

$$\mathbb{H}_q\left[\theta_c \,|\, \boldsymbol{\gamma}_c\right] = -\log\Gamma\left(\textstyle\sum_{c\to\beta\in\boldsymbol{R}_c}\gamma_{c\to\beta}\right) + \sum_{c\to\beta\in\boldsymbol{R}_c}\log\Gamma\left(\gamma_{c\to\beta}\right) \\ - \sum_{c\to\beta\in\boldsymbol{R}_c}(\gamma_{c\to\beta}-1)\mathbb{E}_q\left[\log\theta_{c\to\beta}\right] \tag{A.7}$$

Entropy term for Pitman-Yor process

$$\mathbb{H}_q\left[\pi'_{c,i} \,|\, \nu_{c,i}\right] = -\log\Gamma\left(\nu^1_{c,i}+\nu^2_{c,i}\right) + \log\Gamma\left(\nu^1_{c,i}\right) + \log\Gamma\left(\nu^2_{c,i}\right) \\ -(\nu^1_{c,i}-1)\mathbb{E}_q\left[\log\pi'_{c,i}\right] - (\nu^2_{c,i}-1)\mathbb{E}_q\left[\log(1-\pi'_{c,i})\right] \tag{A.8}$$

A.2 Update for Global Variational Parameters

By taking the derivative of $\mathcal{L}$ with respect to the corresponding variational parameters, we would be able to optimize the global variational parameters in turn using gradient descent. These are exactly the same updates as Cohen et al. (2010).

Optimize γ The update for the variational parameter governing the probability over unadapted rules is

$$\gamma_{c\to\beta} = \underbrace{\alpha_{c\to\beta}}_{\text{prior}} + \underbrace{\sum_{d\in\boldsymbol{D}} g_d(c\to\beta)}_{\text{rules in data}} + \underbrace{\sum_{b\in\boldsymbol{M}}\sum_{i=1}^{K_b}\left|c\to\beta : c\to\beta\in N(z_{b,i})\right|}_{\text{rules in adapted rules}}. \tag{A.9}$$

Optimize ν The update for the variational parameter governing the stick-breaking weight for the i-th atom associated with nonterminal a is

$$\nu^1_{c,i} = \underbrace{\sum_{b \in \boldsymbol{M}} \sum_{k=1}^{K_b} |c \Rightarrow z_{c,i} : c \Rightarrow z_{c,i} \in M(z_{b,k})|}_{\text{Adapted rules of nonterminal } c \text{ used in } b\text{'s rules}} + \underbrace{\sum_{d \in \boldsymbol{D}} f_d(c \Rightarrow z_{c,i}) - b_c + 1}_{\text{Adapted rules in corpus}} \quad \text{(A.10)}$$

$$\nu^2_{c,i} = \sum_{b \in \boldsymbol{M}} \sum_{k=1}^{K_b} \sum_{j=1}^{K_c} |c \Rightarrow z_{c,j} : c \Rightarrow z_{c,j} \in M(z_{b,k})| \quad \text{(A.11)}$$

$$+ \sum_{d \in \boldsymbol{D}} \sum_{j=1}^{K_c} f_d(c \Rightarrow z_{c,j}) + a_c + i b_c. \quad \text{(A.12)}$$

Bibliography

Ahmed, Amr, Aly, Mohamed, Gonzalez, Joseph, Narayanamurthy, Shravan, and Smola, Alexander. Scalable inference in latent variable models. In *WSDM*, pp. 123–132, 2012.

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

AlSumait, Loulwah, Barbará, Daniel, and Domeniconi, Carlotta. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *International Conference on Data Mining*, Washington, DC, USA, 2008. IEEE Computer Society.

Andrzejewski, David, Zhu, Xiaojin, and Craven, Mark. Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In *Proceedings of the International Conference of Machine Learning*, 2009.

Asuncion, Arthur, Smyth, Padhraic, and Welling, Max. Asynchronous distributed learning of topic models. In *Proceedings of Advances in Neural Information Processing Systems*, 2008.

Asuncion, Arthur, Welling, Max, Smyth, Padhraic, and Teh, Yee Whye. On smoothing and inference for topic models. In *Proceedings of Uncertainty in Artificial Intelligence*, 2009.

Banerjee, Arindam and Basu, Sugato. Topic models over text streams: a study of batch and online unsupervised learning. In *Proceedings of SIAM International Conference on Data Mining*, 2007.

Beal, M. J., Ghahramani, Z., and Rasmussen, C. E. The infinite hidden Markov model. In *Proceedings of Advances in Neural Information Processing Systems*. MIT Press, 2002.

Bernstein-Ratner, Nan. The phonology of parent child speech. volume 6, pp. 159–174, 1987.

Bhattacharya, Indrajit and Getoor, Lise. A latent dirichlet model for unsupervised entity resolution. In *Proceedings of SIAM International Conference on Data Mining*, 2006.

Bird, Steven, Klein, Ewan, and Loper, Edward. *Natural Language Processing with Python*. O'Reilly Media, 2009.

Bishop, Christopher M. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

Blei, David M. and Jordan, Michael I. Modeling annotated data. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 127–134, New York, NY, USA, 2003. ACM Press. ISBN 1581136463. doi: 10.1145/860435.860460.

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

Blei, David M. and Lafferty, John D. Correlated topic models. In *Proceedings of Advances in Neural Information Processing Systems*, 2005.

Blei, David M. and Lafferty, John D. Dynamic topic models. In *Proceedings of the International Conference of Machine Learning*, 2006.

Blei, David M. and McAuliffe, Jon D. Supervised topic models. In *NIPS*. 2007.

Blei, David M., Ng, Andrew, and Jordan, Michael. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 2003.

Bloom, Burton H. Space/time trade-offs in hash coding with allowable errors. pp. 422–426, New York, NY, USA, July 1970. ACM.

Blunsom, Phil and Cohn, Trevor. A hierarchical Pitman-Yor process HMM for unsupervised part of speech induction. In *Proceedings of the Association for Computational Linguistics*, 2011.

Bottou, Léon. Online learning and stochastic approximations, 1998.

Bottou, Léon and Le Cun, Yann. Large scale online learning. In *Proceedings of Advances in Neural Information Processing Systems*. 2003.

Bottou, Léon and Murata, Noboru. Stochastic approximations and efficient learning. *The Handbook of Brain Theory and Neural Networks, Second edition,. The MIT Press, Cambridge, MA*, 2002.

Boyd-Graber, Jordan and Blei, David M. Syntactic topic models. In *Proceedings of Advances in Neural Information Processing Systems*, 2008.

Boyd-Graber, Jordan and Blei, David M. Multilingual topic models for unaligned text. In *Proceedings of Uncertainty in Artificial Intelligence*, 2009.

Boyd-Graber, Jordan and Resnik, Philip. Holistic sentiment analysis across languages: Multilingual supervised latent Dirichlet allocation. In *Proceedings of Emperical Methods in Natural Language Processing*, 2010.

Boyd-Graber, Jordan, Blei, David M., and Zhu, Xiaojin. A topic model for word sense disambiguation. In *Proceedings of Emperical Methods in Natural Language Processing*, 2007.

Brants, Thorsten, Popat, Ashok C., Xu, Peng, Och, Franz J., and Dean, Jeffrey. Large language models in machine translation. In *Proceedings of Emperical Methods in Natural Language Processing*, 2007.

Brent, Michael R. and Cartwright, Timothy A. Distributional regularity and phonotactic constraints are useful for segmentation. volume 61, pp. 93–125, 1996.

Broder, Andrei and Mitzenmacher, Michael. Network applications of bloom filters: A survey. In *Internet Mathematics*, pp. 636–646, 2002.

Canini, Kevin R., Shi, Lei, Neuroscience, Helen Wills, and Griffiths, Thomas L. Online inference of topics with latent dirichlet allocation. In *Proceedings of Artificial Intelligence and Statistics*, 2009.

Chang, Jonathan, Boyd-Graber, Jordan, and Blei, David M. Connections between the lines: Augmenting social networks with text. In *Knowledge Discovery and Data Mining*, 2009a.

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

Chappelier, Jean-Cédric and Rajman, Martin. Monte-carlo sampling for NP-hard maximization problems in the framework of weighted parsing. In *Natural Language Processing*, pp. 106–117, 2000.

Chen, Stanley F. and Goodman, Joshua. An empirical study of smoothing techniques for language modeling. In *Proceedings of the Association for Computational Linguistics*, 1996.

Chiang, David, DeNeefe, Steve, and Pust, Michael. Two easy improvements to lexical weighting. In *Proceedings of the Human Language Technology Conference*, 2011.

Chu, Cheng-Tao, Kim, Sang Kyun, Lin, Yi-An, Yu, YuanYuan, Bradski, Gary, Ng, Andrew, and Olukotun, Kunle. Map-Reduce for machine learning on multicore. In *Advances in Neural Information Processing Systems 19 (NIPS 2006)*, pp. 281–288, Vancouver, British Columbia, Canada, 2007.

Clark, Alexander. Combining distributional and morphological information for part of speech induction. 2003.

Cohen, Shay B. and Smith, Noah A. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2009.

Cohen, Shay B., Blei, David M., and Smith, Noah A. Variational inference for adaptor grammars. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2010.

Crammer, Koby, Dekel, Ofer, Keshet, Joseph, Shalev-Shwartz, Shai, and Singer, Yoram. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7:551–585, 2006. ISSN 1532-4435.

Dean, Jeffrey and Ghemawat, Sanjay. MapReduce: Simplified data processing on large clusters. In *Symposium on Operating System Design and Implementation*, 2004.

Deerwester, Scott, Dumais, Susan, Landauer, Thomas, Furnas, George, and Harshman, Richard. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

Denisowski, Paul. CEDICT, 1997. http://www.mdbg.net/chindict/.

Diebolt, Jean and Ip, Eddie H.S. *Markov Chain Monte Carlo in Practice*, chapter Stochastic EM: method and application. Chapman and Hall, London, 1996.

Doucet, Arnaud, De Freitas, Nando, and Gordon, Neil (eds.). *Sequential Monte Carlo methods in practice.* Springer Texts in Statistics. 2001.

Dyer, Chris, Cordova, Aaron, Mont, Alex, and Lin, Jimmy. Fast, easy and cheap: Construction of statistical machine translation models with MapReduce. In *Workshop on SMT (ACL 2008)*, Columbus, Ohio, 2008.

Dyer, Chris, Lopez, Adam, Ganitkevitch, Juri, Weese, Jonathan, Ture, Ferhan, Blunsom, Phil, Setiawan, Hendra, Eidelman, Vladimir, and Resnik, Philip. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of ACL System Demonstrations*, 2010.

Eidelman, Vladimir, Boyd-Graber, Jordan, and Resnik, Philip. Topic models for dynamic translation model adaptation. In *Proceedings of the Association for Computational Linguistics*, 2012.

Emerson, Tom. Sighan bakeoff 2005. *http://www.sighan.org/bakeoff2005*, 2005.

Falush, D., Stephens, M., and Pritchard, J. K. Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies. *Genetics*, 164(4):1567–1587, 2003. ISSN 0016-6731.

Ferguson, Thomas S. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2), 1973.

Finkel, Jenny Rose, Grenager, Trond, and Manning, Christopher D. The infinite tree. In *Proceedings of the Association for Computational Linguistics*, 2007.

Foster, George and Kuhn, Roland. Mixture-model adaptation for smt. In *Proceedings of the Second Workshop on Statistical Machine Translation*, 2007.

Foundation, Apache Software, Drost, Isabel, Dunning, Ted, Eastman, Jeff, Gospodnetic, Otis, Ingersoll, Grant, Mannix, Jake, Owen, Sean, and Wettin, Karl. Apache Mahout, 2010. http://mloss.org/software/view/144/.

Fox, Emily B., Sudderth, Erik B., Jordan, Michael I., and Willsky, Alan S. An HDP-HMM for systems with state persistence. In *Proceedings of the International Conference of Machine Learning*, 2008.

Geman, S. and Geman, D. *Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images*, pp. 452–472. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. ISBN 1-55860-125-2.

Ghemawat, Sanjay, Gobioff, Howard, and Leung, Shun-Tak. The google file system. In *Proceedings of the nineteenth ACM symposium on Operating systems principles*, SOSP '03, pp. 29–43, New York, NY, USA, 2003. ACM. ISBN 1-58113-757-5. doi: 10.1145/945445.945450. URL `http://doi.acm.org/10.1145/945445.945450`.

Goldwater, Sharon and Griffiths, Thomas L. A fully Bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the Association for Computational Linguistics*, 2007.

Goldwater, Sharon, Griffiths, Thomas L., and Johnson, Mark. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the Association for Computational Linguistics*, 2006.

Griffiths, T. L., Canini, K. R., Sanborn, A. N., and Navarro, D. J. Unifying rational models of categorization via the hierarchical Dirichlet process. In *Proceedings of the 29th Annual Conference of the Cognitive Science Society*, 2007.

Griffiths, Thomas L. and Steyvers, Mark. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl 1):5228–5235, 2004.

Griffiths, Thomas L., Steyvers, Mark, Blei, David M., and Tenenbaum, Joshua B. Integrating topics and syntax. In *Proceedings of Advances in Neural Information Processing Systems*. 2005.

Griffiths, T.L. and Ghahramani, Z. Infinite latent feature models and the Indian buffet process. In *Proceedings of Advances in Neural Information Processing Systems*, 2005.

Hall, David, Jurafsky, Daniel, and Manning, Christopher D. Studying the history of ideas using topic models. In *Proceedings of Emperical Methods in Natural Language Processing*, 2008.

Hall, Mark, Frank, Eibe, Holmes, Geoffrey, Pfahringer, Bernhard, Reutemann, Peter, and Witten, Ian H. The WEKA data mining software: An update. *SIGKDD Explorations*, 11, 2009.

Hardisty, Eric, Boyd-Graber, Jordan, and Resnik, Philip. Modeling perspective using adaptor grammars. In *Proceedings of Emperical Methods in Natural Language Processing*, 2010.

Hasler, Eva, Haddow, Barry, and Koehn, Philipp. Sparse lexicalised features and topic adaptation for SMT. In *Proceedings of IWSLT*, 2012.

Hoffman, Matthew, Blei, David M., and Bach, Francis. Online learning for latent Dirichlet allocation. In *NIPS*, 2010.

Hoffman, Matthew, Blei, David M., Wang, Chong, and Paisley, John. Stochastic variational inference. In *Journal of Machine Learning Research*, 2013.

Hofmann, Thomas. Probabilistic latent semantic analysis. In *Proceedings of Uncertainty in Artificial Intelligence*, 1999.

Hu, Diane and Saul, Lawrence K. A probabilistic model of unsupervised learning for musical-key profiles. In *International Society for Music Information Retrieval Conference*, 2009.

Hu, Yuening. *Expressive Knowledge Resources in Probabilistic Models*. PhD thesis, 2014.

Hu, Yuening and Boyd-Graber, Jordan. Efficient tree-based topic modeling. In *Proceedings of the Association for Computational Linguistics*, 2012.

Hu, Yuening, Boyd-Graber, Jordan, and Satinoff, Brianna. Interactive topic modeling. In *Proceedings of the Association for Computational Linguistics*, 2011.

Hu, Yuening, Zhai, Ke, Williamson, Sinead, and Boyd-Graber, Jordan. Modeling images using transformed Indian buffet processes. In *Proceedings of the International Conference of Machine Learning*, 2012.

Hu, Yuening, Boyd-Graber, Jordan, Satinoff, Brianna, and Smith, Alison. Interactive topic modeling. *Machine Learning Journal*, 2013.

Hu, Yuening, Zhai, Ke, Eidelman, Vladimir, and Boyd-Graber, Jordan. Polylingual tree-based topic models for translation domain adaptation. In *Proceedings of the Association for Computational Linguistics*, 2014.

Jelinek, F. and Mercer, R. Probability distribution estimation from sparse data. *IBM Technical Disclosure Bulletin*, 28:2591–2594, 1985.

Johnson, Mark. PCFGs, topic models, adaptor grammars and learning topical collocations and the structure of proper names. In *Proceedings of the Association for Computational Linguistics*, 2010.

Johnson, Mark and Goldwater, Sharon. Improving nonparameteric Bayesian inference: experiments on unsupervised word segmentation with adaptor grammars. In *Conference of the North American Chapter of the Association for Computational Linguistics*, Boulder, Colorado, 2009.

Johnson, Mark, Griffiths, Thomas L., and Goldwater, Sharon. Adaptor grammars: A framework for specifying compositional nonparametric Bayesian models. In *Proceedings of Advances in Neural Information Processing Systems*, 2007.

Jordan, Michael I., Ghahramani, Zoubin, Jaakkola, Tommi S., and Saul, Lawrence K. An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183–233, 1999. ISSN 0885-6125. doi: 10.1023/A:1007665907178.

Kataria, Saurabh S., Kumar, Krishnan S., Rastogi, Rajeev R., Sen, Prithviraj, and Sengamedu, Srinivasan H. Entity disambiguation with hierarchical topic models. In *kdd*, New York, NY, USA, 2011. ACM.

Kirk, David B. and Hwu, Wen-mei W. *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2010. ISBN 0123814723, 9780123814722.

Koehn, Philipp. Statistical significance tests for machine translation evaluation. In *Proceedings of Emperical Methods in Natural Language Processing*, 2004.

Koehn, Philipp. *Statistical Machine Translation*. Cambridge University Press, 2009.

Koehn, Philipp, Och, Franz Josef, and Marcu, Daniel. Statistical phrase-based translation. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 2003.

Koppel, Moshe, Schler, J., Argamon, Shlomo, and Pennebaker, J. Effects of age and gender on blogging. In *In AAAI 2006 Symposium on Computational Approaches to Analysing Weblogs*, 2006.

Kurihara, Kenichi, Welling, Max, and Vlassis, Nikos. Accelerated variational Dirichlet process mixtures. In *Proceedings of Advances in Neural Information Processing Systems*, 2006.

Kurihara, Kenichi, Welling, Max, and Teh, Yee Whye. Collapsed variational Dirichlet process mixture models. In *International Joint Conference on Artificial Intelligence*, 2007.

Landauer, Thomas K., McNamara, Danielle S., Marynick, Dennis S., and Kintsch, Walter (eds.). *Probabilistic Topic Models*. Laurence Erlbaum, 2006.

Li, Aaron, Ahmed, Amr, Ravi, Sujith, and Smola, Alexander J. Reducing the sampling complexity of topic models. In *ACM Conference on Knowledge Discovery and Data Mining (KDD)*, 2014.

Li Fei-Fei and Perona, Pietro. A Bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition*, pp. 524–531, 2005. ISBN 0-7695-2372-2.

Liang, Percy, Petrov, Slav, Jordan, Michael, and Klein, Dan. The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of Emperical Methods in Natural Language Processing*, 2007.

Lin, Chenghua and He, Yulan. Joint sentiment/topic model for sentiment analysis. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2009. ISBN 978-1-60558-512-3. doi: http://doi.acm.org/10.1145/1645953.1646003.

Lin, Jimmy and Dyer, Chris. *Data-Intensive Text Processing with MapReduce*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2010.

Lowe, David G. Object recognition from local scale-invariant features. In *ICCV '99: Proceedings of the International Conference on Computer Vision-Volume 2*, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0769501648.

Malewicz, Grzegorz, Austern, Matthew H., Bik, Aart J.C, Dehnert, James C., Horn, Ilan, Leiser, Naty, and Czajkowski, Grzegorz. Pregel: a system for large-scale graph processing. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, SIGMOD '10, pp. 135–146, New York, NY, USA, 2010. ACM. ISBN 978-1-4503-0032-2. doi: 10.1145/1807167.1807184. URL `http://doi.acm.org/10.1145/1807167.1807184`.

Maskeri, Girish, Sarkar, Santonu, and Heafield, Kenneth. Mining business topics in source code using latent dirichlet allocation. In *ISEC*, 2008. ISBN 978-1-59593-917-3. doi: http://doi.acm.org/10.1145/1342211.1342234.

Matsoukas, Spyros, Rosti, Antti-Veikko I., and Zhang, Bing. Discriminative corpus weight estimation for machine translation. In *Proceedings of Emperical Methods in Natural Language Processing*, 2009.

McCallum, Andrew Kachites. Mallet: A machine learning for language toolkit. http://www.cs.umass.edu/ mccallum/mallet, 2002.

Mimno, David, Wallach, Hanna, Naradowsky, Jason, Smith, David, and McCallum, Andrew. Polylingual topic models. In *Proceedings of Emperical Methods in Natural Language Processing*, 2009.

Mimno, David, Hoffman, Matthew, and Blei, David. Sparse stochastic inference for latent Dirichlet allocation. In *Proceedings of the International Conference of Machine Learning*, 2012.

Minka, Thomas P. Estimating a dirichlet distribution. Technical report, Microsoft, 2000. http://research.microsoft.com/en-us/um/people/minka/papers/dirichlet/.

Mochihashi, Daichi, Yamada, Takeshi, and Ueda, Naonori. Bayesian unsupervised word segmentation with nested pitman-yor language modeling. In *Proceedings of the Association for Computational Linguistics*. Association for Computational Linguistics, 2009.

Müller, Peter and Quintana, Fernando A. Nonparametric Bayesian data analysis. *Statistical Science*, 19(1), 2004.

Nallapati, Ramesh, Cohen, William, and Lafferty, John. Parallelized variational EM for latent Dirichlet allocation: An experimental evaluation of speed and scalability. In *ICDMW*, 2007.

Neal, Radford M. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, 1993.

Newman, David, Asuncion, Arthur, Smyth, Padhraic, and Welling, Max. Distributed Inference for Latent Dirichlet Allocation. In *Proceedings of Advances in Neural Information Processing Systems*. 2008.

Newman, David, Karimi, Sarvnaz, and Cavedon, Lawrence. External evaluation of topic models. In *Proceedings of the Aurstralasian Document Computing Symposium*, 2009.

NIST. Trec special database 22, 1994. http://www.nist.gov/srd/nistsd22.htm.

Och, Franz and Ney, Hermann. A systematic comparison of various statistical alignment models. In *Computational Linguistics*, volume 29(21), pp. 19–51, 2003.

Paisley, John and Carin, Lawrence. Hidden markov models with stick-breaking priors. *Trans. Sig. Proc.*, October 2009.

Papadimitriou, Christos H., Raghavan, Prabhakar, Tamaki, Hisao, Vempala, S., and Vempala, Santosh. Latent semantic indexing: A probabilistic analysis. In *ACM Symposium on Principles of Database Systems*, pp. 159–168. ACM press, 1998.

Papineni, Kishore, Roukos, Salim, Ward, Todd, and Zhu, Wei-Jing. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Association for Computational Linguistics*, pp. 311–318, 2002.

Paul, Michael and Girju, Roxana. A two-dimensional topic-aspect model for discovering multi-faceted topics. In *Association for the Advancement of Artificial Intelligence*, 2010.

Pennebaker, James W. and Francis, Martha E. *Linguistic Inquiry and Word Count*. Lawrence Erlbaum, 1 edition, August 1999. ISBN 156321203X.

Perina, Alessandro, Lovato, Pietro, Murino, Vittorio, and Bicego, Manuele. Biologically-aware latent dirichlet allocation (balda) for the classification of expression microarray. In *Proceedings of the 5th IAPR international conference on Pattern recognition in bioinformatics*, PRIB'10, Berlin, Heidelberg, 2010. Springer-Verlag.

Pitman, J. and Yor, M. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25(2):855–900, 1997.

Pitman, Jim. Combinatorial stochastic processes. Technical Report 621, UC Berkeley Dept. of Statistics, 2002. URL `http://stat.berkeley.edu/users/pitman/621.pdf`.

Rasmussen, Carl Edward. The infinite Gaussian mixture model. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 554–560, 2000.

Robert, Christian and Casella, George. *Monte Carlo Statistical Methods*. Springer Texts in Statistics. Springer-Verlag, New York, NY, 2004.

Sato, Masa-Aki. Online model selection based on the variational Bayes. *Neural Computation*, 13(7):1649–1681, July 2001.

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

Shringarpure, Suyash and Xing, Eric P. mStruct: a new admixture model for inference of population structure in light of both genetic admixing and allele mutations. In *Proceedings of the International Conference of Machine Learning*, 2008. ISBN 978-1-60558-205-4.

Shu, Liangcai, Long, Bo, and Meng, Weiyi. A latent topic model for complete entity resolution. In *icde*, 2009.

Sivic, J., Russell, B., Efros, A., Zisserman, A., and Freeman, W. Discovering object categories in image collections. Technical report, CSAIL, Massachusetts Institute of Technology, 2005.

Smola, Alexander J. and Narayanamurthy, Shravan. An architecture for parallel topic models. *International Conference on Very Large Databases*, 3, 2010.

Snover, Matthew, Dorr, Bonnie, Schwartz, Richard, Micciulla, Linnea, and Makhoul, John. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*, 2006.

Song, Xiaodan, Lin, Ching-Yung, Tseng, Belle L., and Sun, Ming-Ting. Modeling and predicting personal information dissemination behavior. In *Knowledge Discovery and Data Mining*, pp. 479–488, New York, NY, USA, 2005. ACM. ISBN 1-59593-135-X.

Stan Development Team. Stan: A c++ library for probability and sampling, version 2.4, 2014. URL `http://mc-stan.org/`.

Stolcke, Andreas. An efficient probabilistic context-free parsing algorithm that computes prefix probabilities. *Computational Linguistic*, 21(2):165–201, June 1995. ISSN 0891-2017.

Su, Jinsong, Wu, Hua, Wang, Haifeng, Chen, Yidong, Shi, Xiaodong, Dong, Huailin, and Liu, Qun. Translation model adaptation for statistical machine translation with monolingual topic information. In *Proceedings of the Association for Computational Linguistics*, 2012.

Sudderth, Erik B. and Jordan, Michael I. Shared segmentation of natural scenes using dependent Pitman-Yor processes. In *Proceedings of Advances in Neural Information Processing Systems*, 2008.

Sudderth, Erik B., Torralba, Antonio, Freeman, William T., and Willsky, Alan S. Describing visual scenes using transformed dirichlet processes. In *Proceedings of Advances in Neural Information Processing Systems*, 2005.

Talbot, David and Osborne, Miles. Smoothed bloom filter language models: Terascale lms on the cheap. In *ACL*, pp. 468–476, 2007.

Teh, Yee Whye. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the Association for Computational Linguistics*, 2006.

Teh, Yee Whye, Jordan, Michael I., Beal, Matthew J., and Blei, David M. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 2006.

Thusoo, Ashish, Sarma, Joydeep Sen, Jain, Namit, Shao, Zheng, Chakka, Prasad, Anthony, Suresh, Liu, Hao, Wyckoff, Pete, and Murthy, Raghotham. Hive: a warehousing solution over a map-reduce framework. *International Conference on Very Large Databases*, 2(2):1626–1629, August 2009. ISSN 2150-8097.

Toutanova, Kristina and Johnson, Mark. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In Platt, J.C., Koller, D., Singer, Y., and Roweis, S. (eds.), *Proceedings of Advances in Neural Information Processing Systems*, pp. 1521–1528. MIT Press, Cambridge, MA, 2008.

Tseng, Huihsin, Chang, Pichuan, Andrew, Galen, Jurafsky, Daniel, and Manning, Christopher. A conditional random field word segmenter. In *SIGHAN Workshop on Chinese Language Processing*, 2005.

Wainwright, Martin J. and Jordan, Michael I. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2): 1–305, 2008.

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

Wang, Chong and Blei, David. Variational inference for the nested Chinese restaruant process. In *Proceedings of Advances in Neural Information Processing Systems*, 2009.

Wang, Chong and Blei, David M. Truncation-free online variational inference for bayesian nonparametric models. In *Proceedings of Advances in Neural Information Processing Systems*, 2012.

Wang, Chong, Blei, David M., and Heckerman, David. Continuous time dynamic topic models. In *Proceedings of Uncertainty in Artificial Intelligence*, 2008.

Wang, Chong, Blei, David, and Fei-Fei, Li. Simultaneous image classification and annotation. In *Computer Vision and Pattern Recognition*, 2009a.

Wang, Chong, Paisley, John, and Blei, David. Online variational inference for the hierarchical Dirichlet process. In *Proceedings of Artificial Intelligence and Statistics*, 2011.

Wang, Yi, Bai, Hongjie, Stanton, Matt, Chen, Wen-Yen, and Chang, Edward Y. PLDA: parallel latent Dirichlet allocation for large-scale applications. In *International Conference on Algorithmic Aspects in Information and Management*, 2009b.

Wei, Xing and Croft, Bruce. LDA-based document models for ad-hoc retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, 2006.

Weinberger, K.Q., Dasgupta, A., Langford, J., Smola, A., and Attenberg, J. Feature hashing for large scale multitask learning. In *Proceedings of the International Conference of Machine Learning*, pp. 1113–1120. ACM, 2009.

White, Tom. *Hadoop: The Definitive Guide (Second Edition)*. O'Reilly, 2 edition, 2010.

Winn, John and Bishop, Christopher M. Variational message passing. *Journal of Machine Learning Research*, 6:661–694, 2005. ISSN 1532-4435.

Wolfe, Jason, Haghighi, Aria, and Klein, Dan. Fully distributed EM for very large datasets. In *Proceedings of the International Conference of Machine Learning*, pp. 1184–1191, 2008. ISBN 978-1-60558-205-4. doi: http://doi.acm.org/10.1145/1390156.1390305.

Wood, Frank and Black, Michael J. A non-parametric Bayesian alternative to spike sorting. *Journal of Neuroscience Methods*, 173:1–12, 2008.

Xiao, Xinyan, Xiong, Deyi, Zhang, Min, Liu, Qun, and Lin, Shouxun. A topic similarity model for hierarchical phrase-based translation. In *Proceedings of the Association for Computational Linguistics*, 2012.

Xue, Naiwen, Xia, Fei, Chiou, Fu-dong, and Palmer, Marta. The Penn Chinese TreeBank: Phrase structure annotation of a large corpus. *Natural Language Engineering*, 11:207–238, June 2005. ISSN 1351-3249.

Yan, Feng, Xu, Ningyi, and Qi, Yuan. Parallel inference for latent dirichlet allocation on graphics processing units. In *Proceedings of Advances in Neural Information Processing Systems*, pp. 2134–2142. 2009.

Yao, Limin, Mimno, David, and McCallum, Andrew. Efficient methods for topic model inference on streaming document collections. In *Knowledge Discovery and Data Mining*, 2009. ISBN 978-1-60558-495-9.

Zhai, Ke and Boyd-Graber, Jordan L. Online latent Dirichlet allocation with infinite vocabulary. *Proceedings of the International Conference of Machine Learning*, 2013.

Zhai, Ke and Williams, Jason D. Discovering latent structure in task-oriented dialogues. In *Proceedings of the Association for Computational Linguistics*, 2014.

Zhai, Ke, Boyd-Graber, Jordan, Asadi, Nima, and Alkhouja, Mohamad. Mr. LDA: A flexible large scale topic modeling package using variational inference in mapreduce. In *Proceedings of World Wide Web Conference*, 2012.

Zhai, Ke, Boyd-Graber, Jordan, and Cohen, Shay. Online adaptor grammars with hybrid inference. *Transactions of the Association for Computational Linguistics*, 2 (0):465–476, 2014.

Zhang, Haizheng, Qiu, Baojun, Giles, C. Lee, Foley, Henry C., and Yen, John. An lda-based community structure discovery approach for large-scale social networks. In *ISI*, pp. 200–207. IEEE, 2007. URL `http://dblp.uni-trier.de/db/conf/isi/isi2007.html#ZhangQGFY07`.

Zhao, Bing and Xing, Eric P. BiTAM: Bilingual topic admixture models for word alignment. In *Proceedings of the Association for Computational Linguistics*, 2006.