NORTHWESTERN UNIVERSITY

Improving the Usability of Topic Models

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Computer Science

By

Yi Yang

EVANSTON, ILLINOIS

August 2015

ABSTRACT

Improving the Usability of Topic Models

Yi Yang

In an age of information abundance and exploration, understanding large collections of unstructured textual documents will benefit various applications such as recommendation system, search engines and so on. Many attentions have been given to generative probabilistic topic models of textual collections, designed to identify topical representations of the documents that reduce the dimension and reveal documents statistical structure. Latent Dirichlet Allocation (LDA) is one of the most commonly used topic modeling approaches due to its capability to uncover hidden thematic patterns in textual documents with little supervision.

However, LDA has several limitations which makes it difficult for data analysis practitioner to use in practice. Firstly, the Gibbs sampling inference method for LDA runs too slow for large dataset with many topics. Secondly, the topics learned by LDA sometimes are difficult to interpret by end users. Thirdly, LDA suffers from instability problem, which occurs not only when there is new data arrives and the model needs to be update but also when the same Gibbs sampling method is run multiple times on the same data. All the above limitations undermine the usability of LDA in practice.

This thesis focuses on improving the usability of topic models. We propose a general framework, **SC-LDA**, for efficiently incorporating different kinds of knowledge into topic models. The knowledge is represented as a set of constraints which shapes the topics learned by LDA topic model. By incorporating the knowledge into topic models, users can guide the model training process so that the topics learned become more interpretable. The framework also takes advantage of topic model's sparsity to significantly reduce the computational cost of training. It is shown in experiments that SC-LDA converges much rapidly than existing baseline methods while also maintains comparable model performance.

SC-LDA alleviates the first and second limitations of topic models by efficient knowledge integration. We also build a topic model update system, none-disruptive Topic Model Update (**nTMU**), that employs SC-LDA to improve the stability issue of topic models. Evaluation results on both simulation experiments and user studies indicate that our approach significantly outperforms baseline systems in achieving high topic model stability while still maintaining high topic model quality.

Overall, this thesis presents user-centric approaches to address the usability problems of topic models. We hope the work will help topic modeling practitioners who have experienced the usability problem in their practice. Moreover,we also want to interest and inspire machine learning and data mining researchers to pay more attention to the development of user-centric data analytics algorithms to improve their usability.

Acknowledgments

I feel very fortunate to spend six years in the Department of Electrical Engineering and Computer Science at Northwestern University. It has been a long journey, and I owe my gratitude to all the people who have made this thesis possible.

First and foremost I want to thank my advisor Professor Doug Downey for all the advice and support over the past five years. It is an honor to be his first graduated Ph.D. student. He gives me great freedom to choose interesting research topics and directions, and he is very open to discuss my ideas or even random thoughts. He encourages me to do internship in the summer time to explore more research directions. His passionate in doing research, being creative, hard working and being supportive to the students set up a tremendous example for me. He also patiently listened to my questions and gave me helpful advice when I was looking for a job. I am very fortunate to work with him and learn from such an amazing advisor.

I would also want to thank Dr. Shimei Pan, who was my internship mentor at IBM Research. She gave me great support during the internship. She discussed the research projects with me over and over again and helped me to manage the progress. She made the internship going smoothly and successfully. After the internship, she still encourages my research pursuits and collaborates closely with me for all the deadlines, no matter it is weekends and late night. She always makes herself available whenever I have questions and difficulties. This dissertation would not have been possible without these two amazing

people, Doug and Shimei. I feel fortunate to have had the opportunity to work and know them.

I would also like to thank Professor Darren Gergle for serving on my thesis committee and for writing references letters for me. Thanks are also due to Professor Haoqi Zhang for serving on my thesis committee and spending his time reviewing my thesis.

I want to thank Professor Kunpeng Zhang for being a great friend to me. Every time I have confusions and difficulties in study or in life, I call him. He is my great support and company not only in study, but also on the soccer fields. We like to make fun of each other in a nice way all the time and share laughs.

Many thanks also go to my colleagues in WebSail Lab at Northwestern University. They provide invaluable intellectual and spirit support over the past five years. They are Mohammed Alam, Chandra Sekhar Bhagavatula, Dave Demeter, Michael Lucas, Thanapon Noraset and Zack Witten. And especially thank Mike Cartwright for being such an amazing office mate.

I am also very fortunate to collaborate with other researchers outside of Northwestern University that I should give a special mention. Thank Professor Jordan Boyd-Graber at University of Colorado Boulder, Dr. Jie Lu at IBM Research, Dr. Mercan Topkara at JW Player, Dr. Yangqiu Song at University of Illinois at UrbanaChampaign and Prof. Alexander Yates at Temple University. I want to thank them for spending time to discuss projects and to write papers with me.

I am also grateful to my awesome friends for the countless memories at Northwestern during the past six years. I especially want to thank Yuchen Yang and Zhongbi Chen for being great roommate.

Lastly, I would like to thank my parents Wenxiu Tian and Xiaoming Yang, for encouraging my intellectual pursuits from a young age and providing support and encouragement throughout graduate school. I could not have come this far without the encouragement and support of them. Thank you!

Table of Contents

ABSTRACT 3

Acknowledgments 5

List of Tables 10

List of Figures 12

Chapter 1. Introduction 14

1.1. Motivation 14

1.2. Having Users in the Loop 17

1.3. Summary of Contributions 20

1.4. Structure 22

Chapter 2. Previous Work 23

2.1. Latent Dirichlet Allocation (LDA) 23

2.2. SparseLDA: Efficient Topic Model Training 27

2.3. Incorporating Knowledge into Topic Models 31

2.4. User-in-the-Loop Machine Learning 41

Chapter 3. Efficient Method for Incorporating Knowledge into Topic Models 43

3.1. SC-LDA: A Factor Model for Incorporating Prior Knowledge 44

3.2. Experiments 55

3.3. Related Work 62
3.4. Conclusion 64

Chapter 4. User-directed Topic Model Update 65
4.1. Topic Model Instability 68
4.2. Non-Disruptive Topic Model Update (nTMU) 74
4.3. Example Scenarios 78
4.4. Constrained LDA (cLDA) and Evaluation 80
4.5. nTMU System Evaluation and User Study 89
4.6. Related Work 101
4.7. Discussion 104
4.8. Conclusion 107

Chapter 5. Towards Active Learning with Topic Modeling 108
5.1. Active Learning Query Generation 110
5.2. Evaluation 112
5.3. Conclusion 116

Chapter 6. Conclusion 117
6.1. Limitations and Future Work 119

Appendix A. Publications During Ph.D. Study 123

References 125

List of Tables

1.1 LDA trained a large Wikipedia dataset. Some topics are hard to interpret by users. 16

2.1 Notations Table. 26

2.2 Five topics learned by LDA on the CNN dataset. 27

3.1 Characteristics of benchmark datasets. NIPS and NYT-News datasets are used for word correlation experiments, and 20NG dataset is used for document label experiments. 55

3.2 SC-LDA runtime (in seconds) in the 1st, 50th, 100th, 200th iteration round with different number of correlations. 58

3.3 The average running time per iteration over 100 iterations, averaged over 5 seeds, on 20NG dataset. Experiments begin with 100 topics, 1000 labeled documents, and then vary one dimension: number of topics (top), and number of labeled documents (bottom). 61

4.1 Topic model instability on 20 Newsgroup and NIPS datasets. 73

4.2 Four document snippets selected from 20 Newsgroup dataset. 81

4.3 Statistics of three sub-datasets. 87

4.4 Top 10 keywords of each topic by LDA (above) and cLDA (below). 90

4.5 Topic model stability performance of different model update methods. 92

5.1 Fifteen most probable words of each topic before (above) and after active learning (below). 113

List of Figures

1.1 Traditional topic modeling diagram (above). Users are the passive consumers of the topic modeling results. User-in-the-loop topic modeling diagram (below). Users play more active roles in the topic modeling process, by providing knowledge and feedback to the system. 18

2.1 Plate representation of LDA. 25

3.1 Graphical model representation of factor model for incorporating prior knowledge into LDA. 46

3.2 Histogram of nonzero topic counts for word types in NYT-News dataset after inference. 81.9% word types have less than 50 topics with nonzero counts. 52

3.3 Model's log likelihood convergence of different methods on NIPS dataset (above) and NYT-News dataset (below). For NIPS, a 100-topic model with 100 correlations is trained. For NYT-News, a 500-topic model with 100 correlations is trained. SC-LDA reaches likelihood convergence much more rapidly than the other methods. 59

3.4 Average topic coherence and average top 20 topic coherence. The models are trained on NIPS dataset with 500-topic and 100 word correlations. SC-LDA achieves higher topic coherence than other methods. 61

4.1 Documents flow in a topic model update system. 71

4.2 Non-Disruptive Topic Model Update diagram. 75

4.3 Topic model display interface. 77

4.4 The θ of `Mix3` projected on a simplex. a)standard LDA with no constraints; b)cLDA with must-links; c)cLDA with cannot-links; d)cLDA with both must-links and cannot-links. Here, hockey documents are in red, baseball in blue and space in yellow. 89

4.5 Topic Coherence of different topic model update methods. 94

4.6 Model perplexity performance when different types of constraints are added. It shows the benefit of human guidance by providing coherent topic constraints. 96

4.7 Before-update Objective Evaluation Metrics. It shows that users' cognitive understanding of models has no significant differences with LDA and nTMU, before the model is updated. 101

4.8 After-update Objective Evaluation Metrics. It shows that after the model is updated, users who use nTMU system have better and more consistent cognitive understanding of model than the users who use LDA update system. In other words, nTMU provides users a non-disruptive topic model update environment. 102

4.9 Subjective Evaluation Metrics. 103

5.1 Diagram illustrating the topic model active learning framework. 109

5.2 Classification experiments on 20Newsgroup (left) and Reddit(right) dataset. 114

5.3 Topic Coherence experiments on 20Newsgroup (left) and Reddit(right) dataset. 115

5.4 The UI for soliciting pairwise document labels. 116

CHAPTER 1

Introduction

1.1. Motivation

In an age of information abundance and exploration, understanding large collections of unstructured textual documents remains a challenging problem. The massive amount of content available to us today necessitates some good methods for retrieval, organization, and management. In many scenarios, we need to understand the high-level themes of a corpus and explore documents of interest. For example, imagine that users are given an extremely large collection of news articles. It would be desirable for each article to have some short description or labels that could quickly tell us what the document is about. It would also be useful for these short descriptions to have representations that are consistent across the entire collection, so that the computer system may use it to determine how closely related one article is to another and recommend articles with similar topics to users. Thus, efficient algorithm for representing documents would be invaluable for searching and indexing a large text collections.

Probabilistic topic models have been widely applied to explore the large corpora by uncovering the hidden themes of the collection. Topic models have also been applied to aid information retrieval [**63**], understand scientific ideas [**20**], discover political perspectives [**46**], and many other areas.

Latent Dirichlet Allocation (LDA) is one of the most popular and simplest topic models that specifically aims to find document representations by discovering sets of words that

often appear together in the documents. These sets of words can be intuitively interpreted as "topics". Each topic is a multinomial distribution over the vocabulary words. The word types which have the highest probability in a topic convey what the topic is about. In addition, each document can be represented as a mixture of topics, which is a low dimensional description for representing what a document is about.

While LDA presents an elegant framework for document modeling, it faces several limitations when used in practice. These limitations undermine the usability of LDA.

Firstly, the training of LDA is computational costly. There are two families of inference methods for training LDA: Variational Inference and Gibbs Sampling. Both require significant time to finish for large dataset with many topics. Take Gibbs sampling for example, the time complexity of drawing a sample is linear to the number of topics. For one iteration of Gibbs sampling, the computational complexity of scanning the full dataset equals to $O(TV)$, where T is the number of topics and V denotes the total number of word tokens in the dataset. Moreover, accurate training usually takes many sampling passes over the dataset to reach the convergence. Therefore, for large dataset which has millions or even billions of tokens, Gibbs sampling takes too long to finish. It is not uncommon that the experiment results in academic papers are reported on a small or moderate size dataset with no more than hundreds of topics.

Secondly, the topics learned by LDA sometimes are not interpretable to end users. Table 1.1 shows a snippet of 4 topics extracted from a 100-topic LDA model trained on Wikipedia dataset [1]. While topic 1 and 4 convey coherent meanings, topic 2 and 3 are difficult to interpret by users. When using topic models in practice, users often face one critical problem: topics discovered by the model do not always make sense. From the users' perspective, there

[1]The results are publicly available at `http://christo.cs.umass.edu/wiki40/`

are often "bad" topics. A topic may contain thematically unrelated words. Moreover, two thematic related words may appear in different topics. This is mainly because the objective function optimized by LDA may not reflect human judgments of topic quality [**10**].

Table 1.1. LDA trained a large Wikipedia dataset. Some topics are hard to interpret by users.

Topic Index	**Top ranked keywords**
1	school, education, college, schools, high, training
2	home, fast, early, like, ronald, same
3	fire, out, down, people, through, off
4	park, forest, reserve, parks, protected, areas
....	.....

Thirdly, LDA suffers from instability problem. In machine learning, a learning algorithm is said to be *unstable* if it is sensitive to small changes in the training data. For LDA, instability can happen when there is a change in the input data (e.g., when new documents become available) and the trained model needs to be updated. It can also occur when the same Gibbs sampling algorithm is run multiple times on the same data. The unstable problem easily frustrates end user when s/he sees different results in multiple runs on the same data. For an end user, LDA instability can be felt from the changes of the topics assigned to the same documents.

All the above limitations undermine the usability of LDA from end users' perspective. For users who are not machine learning experts, topic models are often a "take it or leave it" black box. When topic models generate desirable results, end users become confident to keep using the model. However, when topic models fail to generate reasonable results, which is not uncommon, end users become frustrated and lose confidence. There is no understandable, efficient, and friendly mechanism for improving the usability of topic models. This hampers adoption and prevents topic models from being more widely used. **This**

dissertation aims at improving the usability of topic models via user-centric approaches by efficiently integrating user guidance. We stress that users can play an important role in the cycle of topic modeling system.

1.2. Having Users in the Loop

In order to overcome the weaknesses of topic models and improve its usability, we propose a user-centric framework to include users in the topic modeling loop. Therefore, instead of being topic model results consumer, users can also play a more active role. We believe that users should be able to give knowledge or feedback and improve topic models without being machine learning experts, and that this knowledge or feedback should be interactive and simple enough to allow these non-expert users to craft models that make sense for them. This is especially important for users, such as those in the social sciences, who are interested in using topic models to understand their data [**24**], and who have extensive domain knowledge but lack the machine learning expertise to modify topic modeling algorithms.

Figure 1.1 shows the traditional algorithm-centric topic modeling framework and our user-centric topic modeling framework. In traditional topic modeling framework, algorithms are the core component and are also black boxes to the end users. Users are merely passive model consumer. In the user-centric topic modeling framework, users play a more active role.

Firstly, users can provide prior knowledge or domain knowledge in order to improve the topic modeling algorithms to meet the users' needs. The knowledge is then incorporated into topic modeling algorithm along with the input data. For example, assume a medical doctor wants to use LDA to analyze a large collection of medical records, while LDA may not be able to find the correlation between "autism" and "psychiatry", the doctor can tell LDA

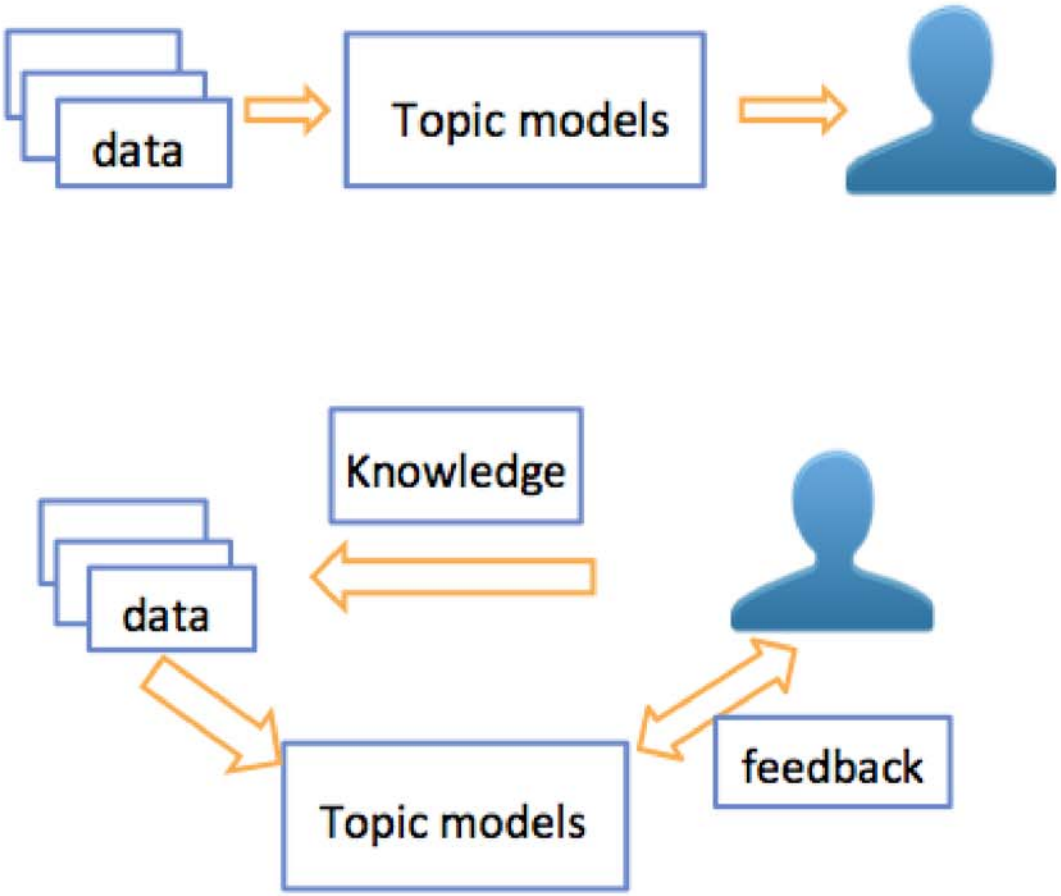


Figure 1.1. Traditional topic modeling diagram (above). Users are the passive consumers of the topic modeling results. User-in-the-loop topic modeling diagram (below). Users play more active roles in the topic modeling process, by providing knowledge and feedback to the system.

explicitly that "autism" and "psychiatry" belong to the same topic so that model can learn the correct correlation with the assistance of such domain knowledge. It is worth noting that although the knowledge only contains two words "autism" and "psychiatry", the other

words that co-occur with them will also be affected by the knowledge and thus the whole model will be affected.

Secondly, in addition to providing knowledge before the model training, users can provide feedback to the system during the training in an interactive fashion. Image the training of LDA as a process of solving an optimization problem. The objective function of classic LDA is a non-convex function (we will discuss it in details in later Chapters), therefore, finding the optimal of the objective function is difficult and is easily trapped into local optimal. Users feedback provides the guidance to help the algorithm to jump out of the local optimal so as to reach a better solution. For example, assume a LDA is trained on a collection of news articles, it is possible that the algorithm thinks "hockey" and "baseball" belong to the same topic because both words co-occur with "play", "score" and other sports-related words. However, a user who is a sports fan may hope to see separate topics on "hockey" and "baseball". By explicitly providing feedback that "baseball" and "hockey" are not belong to the same topic, algorithms can regroup two words into different topics and other words which are related to "baseball" and "hockey" can also be separated into respectively different topics.

Developing users-in-the-loop systems is also challenging and requires new design principles. Here we list three design principles that are crucial in a user-centric system.

- Friendly. Users can provide knowledge and feedback in an easy and friendly format. Since many of the topic model practitioners are not machine learning expert, it is not necessary for them to understand how LDA works and tune model parameters. Therefore, the system should solicit easy and friendly form of knowledge and feedback from end users, even though the knowledge and feedback is eventually converted into complicated constraints under the hood.

- Responsive. Any model involving interaction with users should be as computationally efficient as possible to minimize users' waiting time. In particular, [**59**], summarizing a decade of human-computer interaction research, argue that a system should respond to a user on the order of one second for a simple response, such as clicking a web link, and on the order of ten seconds for a response from a complex user-initiated activity. This design principle for an iterative model requires that we make inference as efficient as possible and to minimize the total number of iterations.
- Effective. Of course, the system should be effective in responding users' knowledge or feedback. Otherwise, users will also lose confidence of using the system if it cannot generate desirable results after efforts. Sometimes, users' knowledge or feedback are aligned with system objective function, in this case, the results will be consistent with human judgment. However, in other cases where users' knowledge is not aligned, not even against, with objective function, the system will balance user knowledge and system knowledge to find a trade-off solution.

1.3. Summary of Contributions

This dissertation makes the following contributions towards improving the usability of topic models.

- A novel factor graph framework, Sparse Constrained LDA (SC-LDA) for efficiently incorporating prior knowledge into LDA (Chapter 3). The prior knowledge is represented as a potential function over the hidden topic variables, and the model is encouraged to learn hidden topics that are accord with the prior knowledge. The factor model representation allows us to develop an efficient sampling algorithm that takes advantage of the models sparsity. SC-LDA serves as the fundamental model

for incorporating knowledge into LDA. We expect this work will interest data analysis practitioners who would like to efficiently train a big topic model with prior knowledge.

- A novel framework for alleviating the instability problem of topic models (Chapter 4). Topic models suffer from instability problem when the model is updated with new data, or the model is re-trained in different runs. This problem, which is closely related to end user experience, has been overlooked by machine learning community. We propose a user-directed non-disruptive topic model update method, nTMU, that balances the trade off between finding the model that fits the data and maintaining the stability of the model from end users perspective. We hope our work will help topic modeling practitioners who have experienced the instability problem in their practice. Moreover,we also want to interest and inspire machine learning and data mining researchers to pay more attention to the development of human-centric data analytics algorithms to improve their usability.
- A pilot study on active learning with topic modeling (Chapter 5). To help users efficiently label data, we also develop an active learning framework that interactively and iteratively acquires user annotations. We conduct experiments with both simulated inputs and real user interactions on two different datasets. Our results demonstrate that the proposed active learning method outperforms both classic LDA and passive LDA in improving topic coherence and reducing document classification errors.

1.4. Structure

Given the problems with the existing probabilistic topic models, this dissertation discusses how to represent prior knowledge, how to efficiently incorporate knowledge into statistical topic models, and how to design a user-directed topic model update system.

This dissertation is organized as follows:

- Chapter 2 reviews the background of classic LDA model. We also present several previous work on topic models, including Sparse LDA, an efficient Gibbs sampling algorithm for LDA, and a few LDA extensions that aim to incorporate knowledge into LDA.
- Chapter 3 presents the Sparse Constrained LDA (SC-LDA) framework, which is a factor graph framework that efficiently incorporates knowledge into LDA. SC-LDA serves as the fundamental framework for our work.
- Chapter 4 focuses on the instability problem that statistical topic models suffer from in practice. We demonstrate that the instability problem can be alleviated by incorporating user guidance, that leads to the user-directed topic model update method.
- Chapter 5 introduces the active learning framework in topic modeling. We conduct experiments with both simulated inputs and real user interactions on two different datasets.
- Chapter 6 concludes with a discussion of the research presented in this dissertation and suggests opportunities for future research.

CHAPTER 2

Previous Work

This chapter begins by reviewing Latent Dirichlet Allocation [**8**], one of the most popular and also the simplest topic models. Topic models, as the names suggest, are used to automatically uncover hidden topics in a large collection of textual documents. Topic models can also be used in other areas such as image recognition [**31**], but our focus is textual data. In this chapter, we first review the basic notations and the generative process of LDA , its graphical representation and its inference method. The notations used in this chapter will also be used throughout the dissertation. Even though LDA provides an elegant method for modeling textual collections, it has several intrinsic limitations that undermine its usage in practice. Therefore, some existing work addresses the limitations of LDA from different perspectives. We review Sparse LDA, an efficient method for fast LDA inference by taking advantage of sparsity property of the model. One of our contribution, Sparse Constrained LDA, is built off of Sparse LDA model. We then review several LDA extensions that aim to incorporate prior knowledge into LDA. Besides the related literature review of LDA-based topic model approaches, we also review existing work on addressing the usability issues of other machine learning systems.

2.1. Latent Dirichlet Allocation (LDA)

Topic models represent words in documents in a corpus D as mixtures of T topics, which are multinomials over a vocabulary of size V. In LDA, each document d is associated with

a multinomial distribution over topics, θ_d. The probability of a word type w given topic z is $\phi_{w|z}$. The multinomial distributions θ_d and ϕ_z are drawn from Dirichlet distributions. α and β are the hyperparameters for θ and ϕ. α is a K-dimension vector, and β is a V-dimension vector. We represent document collection D as a sequence of words $\mathbf{w}$, and topic assignments are $\mathbf{z}$. In common practice, symmetric priors α and β, which means all dimensions in the vector have the same value, are used in LDA applications due to its simplicity, but we still represent them in a vector form without losing generality.

2.1.1. Model

We first review the generative process of LDA with T topics of V vocabulary on document collection D. The generative process describes how the bag of words $\mathbf{w}$ are generated according to the dependence relationships with topics.

- For each topic $t = 1, \ldots, T$
  (1) draw a V-dimensional multinomial distribution over all words: $\phi_t \sim Dir(\beta)$
- For each document d:
  (1) draw a T-dimensional multinomial distribution over topics: $\theta_d \sim Dir(\alpha)$
  (2) for each word w in the document :
    (a) draw a topic z from the distribution θ_d: $z \sim Multi(\theta_d)$
    (b) draw a token w from the distribution ϕ_z: $w \sim Multi(\phi_z)$

Figure 2.1 shows the plate representation of LDA. In the figure, observed words are in solid circle, and all other variables are hidden. The joint probability of the model is:

$$P(\mathbf{w}, \mathbf{z}, \theta, \phi; \alpha, \beta) = \prod_t p(\phi_t|\beta) \Big[\prod_d p(\theta_d|\alpha) \Big[\prod_w p(z_w|\theta_d) p(w|\phi_{w|z}) \Big] \Big] \tag{2.1}$$

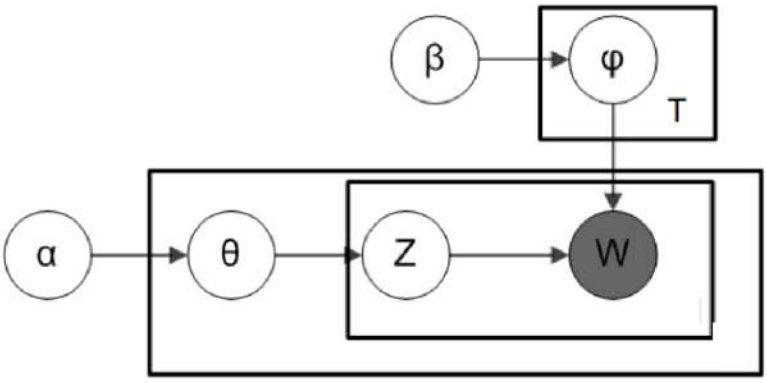


Figure 2.1. Plate representation of LDA.

To discover hidden topics $\mathbf{z}$ from observed words $\mathbf{w}$, we will consider the posterior distribution $P(\mathbf{z}|\mathbf{w})$. Because the conjugate prior of the multinomial distribution is Dirichlet Distribution, we can derive Collapsed Gibbs Sampling inference method by integrating out the topic-word distributions ϕ and the per-document topic distribution θ in the posterior distribution.

[**19**] proposes using collapsed Gibbs sampling method for evaluating this posterior distribution. The probability of a topic z in document d given an observed word type w is:

$$P(z = t|\mathbf{z}_-, w) \propto (n_{d,t} + \alpha)\frac{n_{w,t} + \beta}{n_t + V\beta} \tag{2.2}$$

where $\mathbf{z}_-$ is the total topic samples except the topic z. $n_{d,t}$ is the topic counts of document d, $n_{w,t}$ is the topic counts of word type w, and n_t is the total counts per topic. Here we omit the derivation details of Equation 2.2. Details can be found in [**22**].

It is worth noting that Variational Inference, which is the inference algorithm proposed in original LDA paper [**8**], can also be used for LDA training. However, Gibbs sampling provides

a very convenient implementation framework and it is also very suitable for parallelization, thus many topic model extensions use Gibbs sampling as the approximate inference method [**48, 42, 53**]. The reason for us to focus on Gibbs Sampling is because it is the method that has been widely adopted in practical applications like search engines and online advertising [**62**].

To sample a topic for a token in a document, we compute the probability of all topics according to Equation 2.2, and then sample a topic according to the probability mass. This process can be continued for all the token in the dataset for multiple runs till the model is converged, and then the model's parameters can be estimated with the converged samples chain.

Table 2.1 lists the symbol of notations we use throughout the dissertation.

Table 2.1. Notations Table.

Notation	**Definition**
D	a collection of documents
T	number of latent topics
V	size of vocabulary
d	one document
w	word sequence of D
z	corresponding hidden topic sequence of w
θ	document-topic distribution, dimension $D * T$
θ_d	topic distribution of document d
ϕ	topic-word distribution of, dimension $T * V$
ϕ_z	word distribution of topic z
$\phi_{w\|z}$	probability of word w given topic z
α, β	hyperparameters of Dirichlet Distribution

2.1.2. Example

Here we give an example of using LDA on a corpus of about 320 CNN news articles from October 2012 to November 2013. Since it is a small dataset for illustration purpose, we choose 5 as the number of topics.

Table 2.2. Five topics learned by LDA on the CNN dataset.

topic index	**top 10 ranked keywords**
1	violence, attacks, iraq, police, baghdad, town, car, bombs, bomb, civilians
2	sandy, storm, power, hurricane, york, water, jersey, coast, romney, impact
3	obama, tax, house, republicans, cliff, fiscal, cuts, rates, boehner, spending
4	health, insurance, website, care, obamacare, problems, million, gov, sebelius, law
5	syrian, syria, opposition, rebels, damascus, turkey, military, rebel, war, fighting

Table 2.2 shows the top ranked keywords of the five topics learned by LDA. The top ranked words of each topic are selected by ranking the topic-word distribution. As we can see, these five topics mostly capture the main themes of this corpus. The five topics can be interpreted as "Iraq Attack", "Sandy Hurricane", "Fiscal Cliff", "healthcare reform" and "Arab Spring"respectively. In addition to showing the top ranked keywords, we can also show the representative documents for each topic. The representative documents have the most dominate mass in the document-topic distribution than other documents.

LDA can be viewed as dimension reduction tool for document modeling by reducing the dataset dimension from the vocabulary size V to the number of topics T. Therefore, LDA can be potentially applied in various applications which require accurate document representation, such as document exploration, search engine, recommend system and so on.

2.2. SparseLDA: Efficient Topic Model Training

As we described above, Gibbs sampling is a popular approximated inference method for inferring hidden topics of LDA given observed word tokens. Once the sampling chain is

converged, we can use the samples to compute posterior distribution of LDA and other model parameters, such document-topic distribution and topic-word distribution. The sampler formula is presented as Equation 2.2. **Unfortunately the formula is quite slow for large dataset with many topics**. Conventional Gibbs Sampling for LDA scales linearly with the number of topics. Moreover, accurate training usually takes many sampling passes over the dataset. Therefore, for large datasets with millions or even billions of tokens, conventional Gibbs sampling takes too long to finish.

Several fast inference methods have been proposed [**47, 65, 30**] in order to accelerate the sampling. The core idea of these methods is reducing the sampling complexity for one word from $O(T)$ to sublinear or even constant. Among them, SparseLDA [**65**] has drawn many attentions due to its algorithmic simplicity. It proposes a factorization of Equation 2.2 so that each term can be computed separately, and the most importantly, some terms can be computed quite efficiently. It uses the fact that the relevant terms in the sum are sparse and only the α and β dependent terms are dense. It yields:

$$\begin{aligned} P(z = t|\mathbf{z}_-, w) &\propto (n_{d,t} + \alpha)\frac{n_{w,t} + \beta}{n_t + V\beta} \\ &\propto \frac{\alpha\beta}{n_t + V\beta} + \frac{n_{d,t}\beta}{n_t + V\beta} + \frac{(n_{d,t} + \alpha)n_{w,t}}{n_t + V\beta} \end{aligned} \tag{2.3}$$

The above equation computes the probability of a single topic assignment z. To compute the probability distribution, we need to sum up all single probabilities. It yields:

$$\sum_t P(z = t|\mathbf{z}_-, w) = \sum_t \frac{\alpha\beta}{n_t + V\beta} + \sum_t \frac{n_{d,t}\beta}{n_t + V\beta} + \sum_t \frac{(n_{d,t} + \alpha)n_{w,t}}{n_t + V\beta} \tag{2.4}$$

Note that in LDA, the document-topic counts $n_{d,t}$ and the topic-word counts $n_{w,t}$ are commonly very sparsity. [**19**] presents the Gibbs sampling method for LDA inference via a

statistical physics analogy, and the authors state that "the model favors ensembles of assignments z that form a good compromise between having few topics per document and having few words per topic, with the terms of this compromise being set by the hyperparameters α and β". This sparsity property allows us to write Equation 2.4 as:

$$\sum_t P(z = t|\mathbf{z}_-, w) = \underbrace{\sum_t \frac{\alpha\beta}{n_t + V\beta}}_{\text{s}} + \underbrace{\sum_{t,n_{d,t}>0} \frac{n_{d,t}\beta}{n_t + V\beta}}_{\text{r}} + \underbrace{\sum_{t,n_{w,t}>0} \frac{(n_{d,t} + \alpha)n_{w,t}}{n_t + V\beta}}_{\text{q}} \tag{2.5}$$

Sparse LDA splits original conditional distribution into three buckets, and all three buckets (s, r and q) sum up to 1. Instead of computing the probability mass for each topic, we first compute the probability for each topic in each of the three buckets; then we randomly sample which bucket we need and then select a topic within that bucket. Because we are still sampling from the same conditional distribution, this does not change the underlying sampling algorithm.

At a first glance, it looks like, from Equation 2.4 that we have to compute each bucket T times, so the total computational cost would be $3T$, which is still linear to the number of topics. However, due to the sparsity of the model, the three bucket terms can be computed very efficiently. The first term s is the "smoothing only" bucket that is constant for all documents. The second term r is the "document only" bucket that is shared by a document's tokens. Both s and r have simple constant time updates, and they can be computed efficiently and cached for each document and updated easily. The last term q has to be computed specifically for each token, but only for the few types with non-zero counts in a topic, which is very sparse. Since q often has the largest mass and few non-zero terms, we start the sampling from bucket q.

[**65**] propose to further speedup by sampling topics within a bucket in descending probability. The information needed to compute a probability within a bucket is stored in a data structure, which is an array in decreasing order of probability mass. Thus, on average, after selecting one of the three buckets, only a handful of topics need to be explicitly considered. To maintain topic-count tuples in sorted order within a bucket more efficiently, the topic and the count are packed into one integer (count in higher-order bits and topic in lower-order bits). Because a count change is only a small shift in the overall ordering, a bubble sort returns the array to sorted order in $O(n)$. This data structure to maintain topic-count tuples further speeds up sampling.

Sparse LDA has shown significant improvement on training LDA on large dataset with many topics. Empirical results indicate that Sparse LDA can be approximately 20 times faster than classic LDA. It has been successfully deployed in industrial applications like search engines and online advertising, where the topic model is trained on large document collections with many topics in order to capture the "long tail" of rarely used topics in big data collections. For example, while typical LDA models studied in the literature have up to 10^3 topics, in industrial applications using $10^5 \sim 10^6$ topics is not uncommon [**62**].

In addition to Sparse LDA, there are several alternative method to speed up the training of LDA. Fast LDA [**47**] presents the first fast version of traditional Gibbs sampling inference algorithm. By organizing the conditional probability, i.e., Equation 2.2 in a better way, and constructing an adaptive upper bound on the true normalization constant, Fast LDA can take advantage of the sparse and predictable nature of the topic association probabilities. This ensures both rapid improvement of the adaptive bound and that high-probability topics are visited early, allowing the sampling process to stop as soon as the sample value is located. This process gives a three to eight times factor of improvement in speed, with this factor

increasing with greater numbers of topics. Alias LDA [**30**] takes a further step to reduce the time complexity of sampling one word token $O(T)$ to an amortized constant time. At its core is the idea that dense, slowly changing distributions can be approximated efficiently by the combination of a Metropolis-Hastings step, use of sparsity, and amortized constant time sampling via Walkers alias method. Similar to the fast algorithm for Gibbs sampling, fast algorithm for Variational Inference is also proposed to reduce its computational complexity. Online LDA [**23**] develops an online variational Bayes (VB) algorithm for LDA. Online LDA is based on online stochastic optimization with a natural gradient step, which converges to a local optimum of the VB objective function. It also shows that online LDA finds topic models as good or better than those found with batch VB, and in a fraction of the time.

Moreover, in addition to the development of fast inference algorithms, parallel and distributed framework is also proposed. [**41**] presents an algorithm for distributed Gibbs sampling. The algorithm is a straightforward mapping of LDA to a distributed processor setting. In this algorithm processors concurrently perform Gibbs sampling over local data followed by a global update of topic counts. The algorithm is simple to implement and can be viewed as an approximation to Gibbs-sampled LDA. As opposed to the techniques that use Gibbs sampling, [**67**] proposes a MapReduce parallelization framework that uses variational inference as the underlying algorithm. Combined the fast inference algorithms and the parallel and distributed framework, scaling topic models to industrial-level applications which are trained on large dataset with many topics becomes feasible.

2.3. Incorporating Knowledge into Topic Models

A key fact of LDA is that words are assumed to be uncorrelated and generated independently given the topic distribution of the document. The topic assignment for each word

is irrelevant to all other words. While this assumption facilitates computational efficiency, it loses the rich correlations between words. In many applications, users have external knowledge regarding word correlation, which can be taken into account to improve the semantic coherence of topic modeling. For example, WordNet [**38**] presents a large amount of synonym relationships between words, Wikipedia [1] provides a knowledge graph by linking correlated concepts together. Moreover, in some applications, topic model practitioners have certain domain knowledge that can be incorporated into topic models, no matter whether the knowledge can improve the topic model's quality or the knowledge can meet the application's needs. All of these external knowledge can be leveraged to learn more coherent topics if we can design a mechanism to encourage similar words, correlated concepts, entities of the same category to be assigned to the same topic, and disssimilar words, uncorrelated concepts, entities of the different category to be assigned to the different topics.

In the following paragraphs, we review several existing work that aims to incorporate different kinds of knowledge into topic models.

Labeled LDA: [**48**] presents a generative model for modeling document collections where the documents are associated with labels. Unlike traditional LDA, Labeled LDA incorporates supervision by simply constraining the topic model to use only those topics that correspond to a document's observed labels. Therefore, it assumes a one to one mapping between document labels and the hidden topics. For example, if a document collection totally has 103 unique labels, then the total number of topics used in Labeled LDA would be 103 as well. If a document is associated with two labels {sports, NBA}, the word tokens in the document can only be sampled from the two label corresponding topics, rather than possibilities of 103

[1] `https://www.wikipedia.org/`

topics. Labeled LDA is also extended to Semi-Labeled LDA where parts of documents in the dataset have labels.

The generative process of Labeled LDA is as follows:

- For each topic $t = 1, \ldots, T$
  (1) draw a V-dimensional multinomial distribution over all words: $\phi_t \sim Dir(\beta)$
- For each document d:
  (1) For each topic $t = 1, \ldots, T$
    (a) draw $\delta_t \in \{0, 1\} \sim Bernoulli(\phi_t)$
    (b) $\alpha_t = \delta_t * \alpha$
  (2) draw a T-dimensional multinomial distribution over topics: $\theta_d \sim Dir(\alpha')$
  (3) for each word w in the document :
    (a) draw a topic z from the distribution θ_d: $z \sim Multi(\theta_d)$
    (b) draw a token w from the distribution ϕ_z: $w \sim Multi(\phi_z)$

Collapsed Gibbs sampling can also be applied to Labeled LDA for training. The only difference between Labeled LDA inference and traditional LDA inference is that the conditional probability for Labeled LDA would be zero for topics that are not associated with labels.

Although Labeled LDA proposes an elegant way to incorporate document labels to the traditional LDA model, its disadvantages are also obvious. It assumes that there exists a one-to-one mapping between document labels and document topics. In practice such as online blogs or news articles, many of the documents are labeled by authors in a quite arbitrary fashion. For example, one user may label a NBA article as *NBA*, while another user may only label another similar article as *basketball*. Therefore, under this one-to-one mapping

assumptions, the two articles, which are essentially about the same topics, are assigned to two different topics. In a word, Labeled LDA is very sensitive to the label quality. Ramage also extends Labeled LDA to semi-labeled LDA where only partially documents are labeled in the training dataset [**49**].

Dirichlet Forest LDA: Dirichlet Forest LDA [**3**] proposes an approach to the incorporation of word level domain knowledge into LDA. It shows that many types of knowledge can be expressed with two primitives on word pairs: word must-links and cannot-links. Word must-link relation indicates two words tend to be generated by the same topic, while word cannot-link relation prefers that two words tend to be generated by separate topics.

The generative process of Dirichlet Forest LDA is

- For each topic $t = 1, \ldots, T$
  (1) draw a Dirichlet Tree distribution $\mathbf{q} \sim DirichletForest(\beta, \eta)$
  (2) draw a multinomial distribution over all words $\phi_t \sim DirichletTree(\mathbf{q})$
- For each document d:
  (1) draw a T-dimensional multinomial distribution over topics: $\theta_d \sim Dir(\alpha)$
  (2) for each word w in the document :
    (a) draw a topic z from the distribution θ_d: $z \sim Multi(\theta_d)$
    (b) draw a token w from the distribution ϕ_z: $w \sim Multi(\phi_z)$

In the above generative process, $\mathbf{q}$ specifies a Dirichlet tree distribution, and η is the strength parameter of the domain knowledge. In order to make the constraints effective, it usually sets the strength parameter η to be much larger than the hyperparameter for the distribution over vocabulary β. This gives the constraints higher weight.

Let us intuitively explain the effect of the integration of constraints. Take must-link constraints for example, after a constraint {football, fumble} is added to the Dirichlet Forest, the probabilities of "football" and "fumble" in each topic are likely to be both high or both low. It's unlikely for "football" to have high probability in a topic and "fumble" to have a low probability in the same topic.

Collapsed Gibbs sampling can also be applied to Dirichlet Forest LDA for training. The disadvantages of this model is that the sampling is computational costly. Specifically speaking, the complexity of computing the sampling distribution is $O(TLS)$ for models with T topics, paths at most L nodes long, and at most S paths per word type. In contrast, for traditional LDA the conditional sampling distribution requires only $O(T)$. Therefore, the slow inference method makes it difficult to scale to the incorporation of a large, or even moderate size of knowledge base, where many word correlation knowledge exists.

Logic LDA: [**4**] presents a framework which combines topic modeling with First-Order Logic (FOL). A domain expert can specify her domain knowledge as First-Order Logic rules, and the model will automatically incorporate them into LDA inference to produce topics shaped by both the data and the rules. The key to the framework is to allow domain knowledge, specified in FOL, to influence the values of the hidden topics $\mathbf{z}$, indirectly influencing ϕ and θ. FOL provides a powerful and flexible way to specify domain knowledge. As an example in the paper [**4**], an analyst working on a congressional debate corpus where each speech is a document may specify the rule: $w(i, taxes) \wedge Speaker(d_i, Rep) \Rightarrow z(i, 77)$, which states that for any word token w_i="taxes" that appears in a speech by a Republican, the corresponding latent topic should be $z_i = 77$. While FOL provides flexible way to specify domain knowledge, it is also obvious that it requires too much effort for domain expert to generate logic rules.

The domain expert specifies the knowledge in the form of a weighted FOL knowledge base using predicates: KB=$\{(\lambda_1, \psi_1), ..., (\lambda_L, \psi_L)\}$. The KB is in Conjunctive Normal Form, consisting of L pairs where each rule ψ_l is an FOL clause, and $\lambda_l \geq 0$ is its weight which the domain expert sets to represent the importance of ψ_l. To combine the set of knowledge and LDA, a Markvo Random Field is defined over the hidden topic assignments $\mathbf{z}$, θ, ϕ given words $\mathbf{w}$ documents $\mathbf{d}$ and size information $\mathbf{o}$. The joint likelihood is:

$$P(\mathbf{w}, \mathbf{z}, \theta, \phi; \alpha, \beta) = \exp\Big(\sum_{l}^{L} \sum_{g \in G(\psi_l)} \lambda_l 1_g(\mathbf{z}, \mathbf{w}, \mathbf{d}, \mathbf{o})\Big) * \tag{2.6}$$

$$\prod_t p(\phi_t|\beta)\Big[\prod_d p(\theta_d|\alpha)\Big[\prod_w p(z_w|\theta_d)p(w|\phi_{w|z})\Big]\Big] \tag{2.7}$$

The first term is the total potential function of the prior knowledge from the KB, which is equivalent to a Markov Logic Network (MLN) [**50**], and the remaining terms are identical to LDA. Since exact inference is intractable for both LDA and MLN, the inference for Logic LDA is difficult as well. In the paper, authors developed a scalable inference technique using stochastic gradient descent. However, as we will see in the next chapter, Logic LDA is still quite slow to converge on large dataset with many knowledge.

Similar to using first-order logic to specify knowledge in Logic-LDA, Foulds presents latent topic networks framework that allows the development of custom latent variable topic models using probabilistic programming language with first-order logic syntax [**18**]. The framework is designed specifically to enable applied social science research. The Latent Topic Networks directly generalize LDA, but it can add prior structure, dependency relationships, and additional latent and observed variables, using hinge-loss Markov random fields.

Quad-LDA: In order to improve the coherence of the keywords per topic learned by LDA, [**42**] proposes a regularization framework to incorporate word correlation knowledge into

LDA. The regularization framework works by creating a structured prior over words that reflect broad patterns in the external data. This model might be useful for small collections or noisy text (e.g. web search result snippets or blog posts), where learned topics can be less coherent, less interpretable, and less useful. To learn better topic models for small or noisy collections, [**42**] introduces structured priors on ϕ based upon external data, which has a regularization effect on the standard LDA model. More specifically, the priors on ϕ will depend on the structural relations of the words in the vocabulary as given by external data, which will be characterized by the VV "covariance" matrix C. Intuitively, C is a matrix that captures the short-range dependencies between words in the external data. This data could come from a variety of sources, for example the corpus of 3M English Wikipedia articles. Therefore, given a matrix of word dependencies C, the joint likelihood is:

$$P(\mathbf{w}, \mathbf{z}, \theta, \phi; \alpha, \beta) = (\phi^T C \phi) \prod_t p(\phi_t|\beta) \Big[\prod_d p(\theta_d|\alpha) \Big[\prod_w p(z_w|\theta_d) p(w|\phi_{w|z}) \Big] \Big] \tag{2.8}$$

The first term is the prior knowledge of word correlations from the external data resources, and the remaining terms are identical to LDA. Collapsed Gibbs Sampling is used for Quad-LDA.

NMF-LDA: Similar to Quad-LDA, [**64**] presents a method for incorporating the external word correlation knowledge into LDA to improve the coherence of topic modeling. Very similar to Logic LDA, this paper build a Markov Random Field on the latent topic layer of LDA to encourage words labeled as similar to share the same topic label. Therefore, the topic assignment of each word is not independent, but rather affected by the topic labels of

its correlated words. The joint probability of all topic assignments $\mathbf{z}$ is:

$$P(\mathbf{z}|\theta,\lambda) = p(\mathbf{z}|\theta)\exp\{\lambda \sum_{(m,n)\in C} 1(z_m = z_n)\} \tag{2.9}$$

where $(m,n) \in C$ indicates word m and word n belong to the correlation set C. The first term is exactly the same as standard LDA, and the second term defines the potential function of the current topic assignments. [**64**] develops a variational inference method for the training of NMF-LDA.

Markov Random Topic Fields (MRTF): [**27**] presents an topic model that makes use of one or more user-specified graphs describing relationships between documents. These graph are encoded in the form of a Markov random field over topics and serve to encourage related documents to have similar topic structures. It assumes that documents in LDA are not independent if users know the relationships between the documents. It claims that related documents are more likely to have "similar" topic structures. For instance, in the scientific community, if paper A cites paper B, we would expect the topic distributions for papers A and B to be related. Similarly, if two papers share an author, we might expect them to be topically related. The model combines LDA and an Markov Random Filed specified by the document relationships graph G. Starting from the joint distribution specified by LDA (see Equation 2.1), edge potentials will be added the joint distribution for each edge in the document graph G that "encourage" the topic distributions of neighboring documents to be similar. The potentials all have the form:

$$\psi_{d_1,d_2}(\theta_{d_1},\theta_{d_2}) = \exp\Big[- l_{d_1,d_2}\rho(\theta_{d_1},\theta_{d_2})\Big] \tag{2.10}$$

where l_{d_1,d_2} is a "measure of strength" of the importance of the connection between d_1 and d_2. ρ is a distance metric measuring the dissimilarity between θ_{d_1} and θ_{d_2}. It uses Euclidean distance, but it shows that alternative distance metrics, such as Bhattacharyya and Hellinger, are preferable.

In the experiments, six types of relation graph are considered given the different kinds of meta-data in the documents. There are auth (shared author), book (shared booktitle/journal), cite (one cites the other), http (source file from same domain), time (published within one year) and year (published in the same year) .

Interactive Topic Modeling (ITM): [**26**] proposes the first interactive framework for allowing users to iteratively refine the topics discovered by LDA by adding constraints that enforce that sets of words must appear together in the same topic. The underlying algorithm for incorporating the set of words is Dirichlet Forest LDA [**3**]. For a static model, inference in ITM is the same as in Dirichlet Forest LDA. The paper presents how interactively changing constraints can be accommodated in ITM, smoothly transitioning from unconstrained LDA to constrained LDA with one constraint, to constrained LDA with two constraints, etc. Adding constraints in ITM is achieved by the unassignment of states. In the implementation of a Gibbs sampler, unassignment is done by setting a token's topic assignment to an invalid topic and decrementing any counts associated with that word. When a constraint is added, a portion of topic assignments of the words associated with the constraint will be reset to the invalid topic. How much of the existing topic assignments are reset to the invalid topic leads to different strategies. In the paper, four difference unassignment strategies are proposed. **All** revokes all state assignments, essentially starting the Gibbs sampler from scratch. Therefore, this does not allow any interactive refinement, since it does not enforce the new topics be in any way consistent with the existing topics. **Doc** performs

the unassignment operation for each word in any document containing a word added to a constraint. **Term** perform the unassignment operation only on the topic assignments of tokens whose words have added to a constraint. **None** only moves words into constraints but keep the topic assignments fixed. This is the simplest but less interactive strategy because the impact of the constraints are not immediate and users don't feel that their constraints are actually incorporated into the model.

To sum up, we briefly review several existing work on the incorporation of knowledge into LDA topic model. Incorporating knowledge into topic model has drawn many attentions and led to the development of different extensions. Due to the space limitation, we cannot enumerate each of the extensions in this section. From the above discussion, we can see that there is lack of an efficient and unified framework for incorporating different kinds of knowledge into LDA. **Labeled LDA** can only handle document label knowledge. **Dirichlet Forest LDA**, **Quad-LDA**, **NMF-LDA** and **ITM** can only handle word correlation knowledge. **MRTF** can only handle document correlation knowledge. **Logic LDA** can handle word correlation , document label knowledge and other kinds of knowledge. However, each knowledge has to be encoded as First Order Logic, and this is obviously not practical for topic model practitioners who are likely not machine learning experts. Most importantly, all the models do not scale to large dataset with many topics. This limitation greatly undermines the usability of these models.

In the next chapter, we will present the Sparse Constrained topic model (SC-LDA), a general framework for the efficient incorporation of knowledge into LDA topic models. SC-LDA can handle different kinds of knowledge such as word correlation, document correlation, document label and so on. One advantage of SC-LDA over existing methods is that it is very fast to converge which makes it practical to be used for training large topic models with

many topics. SC-LDA, due to its generality, also serves as the fundamental framework for our user-centric topic modeling which will be discussed in the later chapters.

2.4. User-in-the-Loop Machine Learning

Usability is a measurement of how well a system allows end users to interact with it and achieve their goals [**55**]. It is critical to the development of a user-friendly interface because it helps users to work in an effective, efficient, and manageable way [**13**]. Despite being well-established principles in UI design, new machine learning/data mining algorithms are rarely invented to specifically address the usability factor. However, for any interactive machine learning/data mining system, without proper backend support, it is impossible to achieve consistency and predictability at the UI level. For example, if the backend algorithm produces inconsistent results at different time intervals due to updates, the UI designed to surface these results will inevitably suffer from low consistency and predictability. More attention is needed during the development of backend algorithms to address these basic usability issues.

The thesis focuses on improving the usability issues of natural language processing and, in particular, LDA-based models via user-centric approaches. Besides this, getting users into the loop has also been discussed in other types of machine learning based systems. Recent work has demonstrated several applications of end user interactive machine learning systems. [**16**] propose an interactive machine-learning model that allows users to train, classify/view and correct the classifications. They also describe Image Processing with Crayons, which is a tool for creating new camera-based interfaces using a simple painting metaphor. CAPpella [**14**] system enables end-user training of a machine learning system for context detection in sensor equipped environments. Ritter and Basu [**51**] demonstrate interactive machine

learning in complex file selection tasks. CueFlik [**17**] presents a Web image search application that allows end-users to quickly create their own rules for re-ranking images based on their visual characteristics. End-users can then re-rank any future Web image search results according to their rule. WhittleSearch [**28**] proposes a mode of feedback for image search, where a user describes which properties of exemplar images should be adjusted in order to more closely match his/her mental model of the image sought. Each of these provides initial evidence of the utility of interactive machine learning. How to design effective end-user interaction with interactive machine learning systems remains a open questions for researchers. [**2**] demonstrates with CueFlik[**17**] that careful designs considering the needs of both end-users and machine learning algorithms can significantly impact the effectiveness of end-user interaction. One major distinguishing factor of this thesis with the above methods is that we are going "deeper" and "under the hood" to improve the underlying algorithms in a way that improves the usability issues. We propose algorithms in the back-end level that are able to support the front-end user interaction. In a word, our approaches aim at the back-end algorithm design while the existing work focuses on front-end system design.

CHAPTER 3

Efficient Method for Incorporating Knowledge into Topic Models

We have seen in Chapter 1 that topic models have two significant usability problems in practice. Firstly, conventional inference methods for topic modeling, such as Gibbs sampling and Variational Inference are computational costly. Take Gibbs sampling for example, Conventional Gibbs Sampling for LDA scales linearly with the number of topics. Moreover, accurate training usually takes many sampling passes over the dataset. Therefore, for large datasets with millions or even billions of tokens, conventional Gibbs sampling takes too long to finish. Secondly, LDA topic model is an unsupervised model, and it requires no annotation and discovers, without any supervision, the thematic trends in a text collection. However, this unsupervised characteristic creates potential problems when LDA is applied in practice. Often, the hidden topics learned by LDA fail to make sense to end users.

We also have seen in Chapter 2 that the limitations of topic models have drawn many attentions in the community, and several existing work aims to address the usability problem of topic models by speeding up the inference algorithm of LDA or by incorporating knowledge into LDA. However, to the best of our knowledge, no existing method achieves both scale and rich prior information. Our goal is to rectify this. We propose a factor graph model to incorporate the prior knowledge into LDA. The prior knowledge is represented as a potential function over the hidden topic variables, and the model is encouraged to learn hidden topics that are accord with the prior knowledge. The factor model representation allows us to develop an efficient sampling algorithm that takes advantage of the model's sparsity. We

conduct experiments on benchmark datasets to show that our method is able to achieve comparable performance but runs significantly faster than baseline methods. We expect this work will interest data analysis practitioners who would like to efficiently train a big topic model with prior knowledge.

3.1. SC-LDA: A Factor Model for Incorporating Prior Knowledge

LDA assumes that the hidden topic of a word is independent with other hidden topics, given document's topic distribution θ. While this assumption facilitates computational efficiency, it loses the rich correlation between words. In many scenarios, users have external knowledge regarding word correlation or document label and relations, which can be taken into account to improve the semantic coherence of topic modeling.

Prior knowledge can constrain what models discover. A correlation between two words v and w indicates that they have a similar topic distribution, i.e., $p(z|v) \sim p(z|w)$.[1] Therefore, the hidden topics for v and w tend to have a higher probability drawing from the same topics. In contrast, if v and w are uncorrelated, their hidden topics tend to have a higher probability drawing from different topics. Also, if two documents have the same label, then they might be topic related, and the hidden topics in two documents tend to be drawn from the same topics.

We denote the set of prior knowledge as M. Each prior knowledge $m \in M$ defines a potential function $f_m(z, w, d)$ of the hidden topic z of word type w in document d with which m is associated. Therefore, the whole prior knowledge M defines a score on the current

[1]In [**3**] two correlated words are taken to indicate that $p(v|z) \sim p(w|z)$. However, for word types that have very different frequencies, these two quantities would never be close, and thus we think $p(z|v) \sim p(z|w)$ is a more appropriate constraint.

topic assignments $\mathbf{z}$:

$$\psi(\mathbf{z}, M) = \prod_{z \in \mathbf{z}} \exp f_m(z, w, d) \tag{3.1}$$

If m is knowledge about word type w, then $f_m(z, w, d)$ applies to all hidden topics of word w. If m is knowledge about document d, then $f_m(z, w, d)$ applies to all topics that are in document d. The potential function assigns large values to the topic assignments that accord with the prior knowledge, while it penalizes the topic assignments that are *not* accord with the prior knowledge. Take an extreme case for example, if a prior knowledge m says word type w in document d is topic 3, then the potential function $f_m(z, w, d)$ is zero for all topics but topic 3.

Figure 3.1 shows the graphical model representation of the factor model for incorporating prior knowledge into LDA model. It is worth comparing Figure 3.1 with Figure 2.1. The only difference is the additional factor M which defines a potential function over all hidden topic assignments $\mathbf{z}$.

Since the potential function ψ is a function of $\mathbf{z}$, and it is only a real-value score of current topic assignments, we can write the joint likelihood of the model as:

$$\begin{aligned}
P(\mathbf{w}, \mathbf{z}|\alpha, \beta, M) &= P(\mathbf{w}|\mathbf{z}, \beta)P(\mathbf{z}|\alpha)\psi(\mathbf{z}, M) \\
&= \int_\theta \int_\phi p(\mathbf{w}|\mathbf{z}, \phi)p(\phi|\beta)p(\mathbf{z}|\theta)p(\theta|\alpha)\psi(\mathbf{z}, M)d\theta d\phi \\
&= \psi(\mathbf{z}, M) \int_\theta \int_\phi p(\mathbf{w}|\mathbf{z}, \phi)p(\phi|\beta)p(\mathbf{z}|\theta)p(\theta|\alpha)d\theta d\phi \\
&= \prod_{z \in \mathbf{z}} \exp f_m(z, w, d) \int_\theta \int_\phi p(\mathbf{w}|\mathbf{z}, \phi)p(\phi|\beta)p(\mathbf{z}|\theta)p(\theta|\alpha)d\theta d\phi
\end{aligned} \tag{3.2}$$

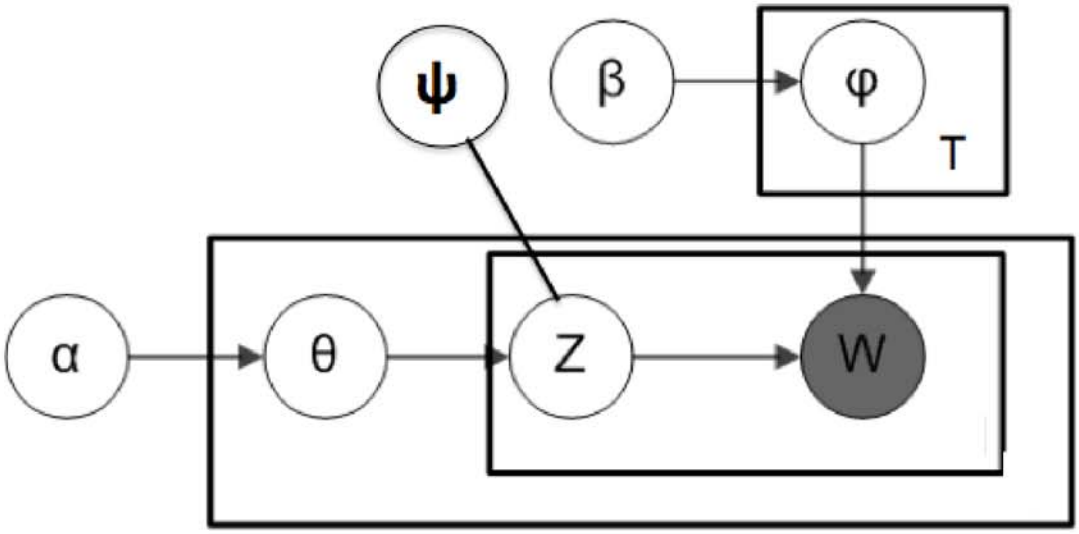


Figure 3.1. Graphical model representation of factor model for incorporating prior knowledge into LDA.

Given the joint likelihood, the goal is then to evaluate the posterior distribution $P(\mathbf{z}|\mathbf{w})$. Computing $P(\mathbf{z}|\mathbf{w})$ involves evaluating a probability distribution on a large discrete state space: $P(\mathbf{z}|\mathbf{w}) = P(\mathbf{z},\mathbf{w})/\sum_{\mathbf{z}} P(\mathbf{z},\mathbf{w})$. In [**19**], they borrow the concept from statistical physics and treat a standard LDA as a physics system where energy function of the system favors ensembles of topic assignments $\mathbf{z}$ that form a good compromise between having few topics per document and having few words per topic, with the terms of this compromise being set by the hyperparameters α and β. Our factor model representation of prior knowledge can also be intuitively explained as that the system favors ensembles of topic assignments $\mathbf{z}$ that form a good compromise between satisfying a standard LDA model as well as the given prior knowledge.

Before we derive the collapsed Gibbs sampling for the factor graph, let us first define several probability distributions.

The probability of all hidden topic variables given parameter θ is defined in a multinomial distribution

$$p(\mathbf{z}|\theta) = \prod_{d}^{D} \prod_{t}^{T} \theta_{d,t}^{n_{d,t}} \tag{3.3}$$

where $n_{d,t}$ denotes the count of words assigned to topic t in document d.

The probability of all observed word tokens given parameters ϕ and all the hidden topic assignments $\mathbf{z}$ is defined in a multinomial distribution

$$p(\mathbf{w}|\mathbf{z}, \phi) = \prod_{t}^{T} \prod_{w}^{V} \phi_{w,t}^{n_{w,t}} \tag{3.4}$$

where $n_{w,t}$ denotes the count of word w assigned to toipc t in the dataset.

Also, the prior probabilities on parameters are defined in a Dirichlet distribution

$$\begin{aligned} p(\theta|\alpha) &= \prod_{d}^{D} \Big(\frac{\Gamma(\sum_{t}^{T} \alpha_t)}{\prod_{t}^{T} \Gamma(\alpha_t)} \prod_{t}^{T} \theta_{d,t}^{\alpha_t - 1} \Big) \\ p(\phi|\beta) &= \prod_{t}^{T} \Big(\frac{\Gamma(\sum_{w}^{V} \beta_w)}{\prod_{w}^{V} \Gamma(\beta_w)} \prod_{w}^{V} \phi_{w,t}^{\beta_w - 1} \Big) \end{aligned} \tag{3.5}$$

Plugging in Equation 3.1, 3.3, 3.4 and 3.5, the collapsed Gibbs Sampling for inferring topic assignment z of word w in document d is:

$$
\begin{aligned}
& P(z = t|w, \mathbf{z}_-, M, \alpha, \beta) \\
&= \frac{P(\mathbf{w}, \mathbf{z}|\alpha, \beta, M)}{P(\mathbf{w}, \mathbf{z}_-|\alpha, \beta, M)} = \frac{P(\mathbf{w}, \mathbf{z})}{P(\mathbf{w}, \mathbf{z}_-)} \frac{\psi(\mathbf{z}, M)}{\psi(\mathbf{z}_-, M)} \\
&= \frac{\int_\theta p(\mathbf{z}|\theta) p(\theta; \alpha) d\theta}{\int_\theta p(\mathbf{z}_-|\theta) p(\theta; \alpha) d\theta} \frac{\int_\phi p(\mathbf{w}|\mathbf{z}, \phi) p(\phi; \beta) d\phi}{\int_\phi p(\mathbf{w}|\mathbf{z}_-, \phi) p(\phi; \beta) d\phi} \frac{\psi(\mathbf{z}, M)}{\psi(\mathbf{z}_-, M)} \\
&= p(z = t|\mathbf{z}_-; \alpha) p(w|\mathbf{w}, \mathbf{z}; \beta) \frac{\psi(\mathbf{z}, M)}{\psi(\mathbf{z}_-, M)} \\
&= \frac{1}{\Delta} \left\{ (n_{d,t} + \alpha) \frac{n_{w,t} + \beta}{n_t + W\beta} \right\} \frac{\psi(\mathbf{z}, M)}{\psi(\mathbf{z}_-, M)} \\
&= \frac{1}{\Delta} \left\{ (n_{d,t} + \alpha) \frac{n_{w,t} + \beta}{n_t + W\beta} \right\} \exp f_m(z, w, d)
\end{aligned} \tag{3.6}
$$

where $\Delta = \sum_t \left[\left\{ (n_{d,t} + \alpha) \frac{n_{w,t}+\beta}{n_t + W\beta} \right\} \exp f_m(z, w, d) \right]$, and recall that $\mathbf{z}_-$ is the total topic samples except the topic z.

The first part is exactly the same as standard LDA, and we can compute it efficiently by taking advantage of existing fast Gibbs sampling methods. However, if the second, $\exp f_m(z, w, d)$, is not sparse, we still need to compute it explicitly for T times because we need the summation of $P(z = t)$, i.e., Δ, for sampling. Therefore, the critical part of speeding up the sampler is whether we can find sparse representation of the second term. As long as the knowledge can be presented as constraints in a sparse structure, we are able to efficiently incorporate it into topic models. We call the framework **SC-LDA**, which is short for Sparse Constrained LDA.

The collapsed Gibbs sampling for SC-LDA framework is summarized in Algorithm 1.

Algorithm 1: Collapsed Gibbs Sampling for word type w in document d in SC-LDA, given knowledge set M

compute s, r, q with SparseLDA ;
for $t \leftarrow 1$ **to** T **do**
    update s_t, r_t, q_t
end
$p(t) = s_t + r_t + q_t$;
sample new topic assignment for w from $p(t)$;

In the following sections, we show that **natural, sparse representations of prior knowledge are possible**. We first present an efficient sparse representation of word correlation prior knowledge, and then present one for document-label knowledge.

3.1.1. Word Correlation Prior Knowledge

We now illustrate how we can encode word correlation knowledge as a set of sparse constraints $f_m(z, w, d)$ in our model. In previous work [**3**], word correlations prior knowledge are represented as word must-link constraints and cannot-link constraints. A must-link relation between two words indicates that the two words are highly related with the same topic. They either have high probability under a topic or low probability under the other topics. Its unlikely for one word to have high probability in a topic and the other one to have a low probability. In contrast, a cannot-link relation between two words indicates that these two words are not topically similar, and they should not both have large probability within any topic. For example, "quarterback" and "fumble" are both American football related terms, so they can share a must-link relation. But "fumble" and "bank" obviously mean two different topics, so they share a cannot-link.

Let us say word w is associated with a set of prior knowledge correlations M_w. Each prior knowledge $m \in M_w$ is a word pair (w, w'), and it has "topic preference" of w given its

correlation word w'. The must-link set of w is M_w^m, and the cannot-link set of w is M_w^c, i.e., $M_w = M_w^c \bigcup M_w^m$. In the example above, $M_{fumble}^m = \{quarterback\}$, and $M_{fumble}^c = \{bank\}$, so $M_{fumble} = \{quarterback, bank\}$. The topic sample of word "fumble" has higher probability to be drawn from the same topics as "quarterback", but it has less probability to be drawn from topics which "bank" are drawn from.

We define the potential score of sampling topic t to word type w, if M_w is not empty, to be:

$$f_m(z, w, d) = \sum_{u \in M_w^m} \log \max(\lambda, n_{u,z}) + \sum_{v \in M_w^c} \log \frac{1}{\max(\lambda, n_{v,z})} \tag{3.7}$$

where $\lambda > 0$ is a hyperparameter, which we call the correlation strength parameter. The intuitive explanation of Equation 3.7 is that the prior knowledge about the word type w will make an impact on the conditional probability of sampling the hidden topic z. Unlike standard LDA where every word's hidden topic is independent of other words given θ, Equation 3.7 instead increases the probability that a word w will be drawn from the same topics as those of w's must-link word set, and decreases its probability of being drawn from the same topics as those of w's cannot-link word set.

The hyperparameter λ controls the strength of each piece of prior knowledge. The smaller λ is, the stronger this correlation is. For large λ, the constraint is inactive for topics except those with the large counts. As λ decreases, the constraint becomes active for topics with lesser counts. We can adjust the value of λ for each piece of prior knowledge based on our confidence. In our experiments, for simplicity, we use the same value λ for all knowledge, and we set $\lambda = 1$.

From Equation 3.7 and Equation 3.6, the conditional probability of a topic z in document d given an observed word type w is:

$$
\begin{aligned}
&P(z = t|w, \mathbf{z}_-, M) \\
&= \frac{1}{\Delta}\left\{(n_{d,t} + \alpha)\frac{n_{w,t} + \beta}{n_t + W\beta}\right\}\left\{\prod_{u \in M_w^m} \max(\lambda, n_{u,t}) \prod_{v \in M_w^c} \frac{1}{\max(\lambda, n_{v,t})}\right\} \\
&= \frac{1}{\Delta}\left\{\frac{\alpha\beta}{n_t + V\beta} + \frac{n_{d,t}\beta}{n_t + V\beta} + \frac{(n_{d,t} + \alpha)n_{w,t}}{n_t + V\beta}\right\}\left\{\prod_{u \in M_w^m} \max(\lambda, n_{u,t}) \prod_{v \in M_w^c} \frac{1}{\max(\lambda, n_{v,t})}\right\}
\end{aligned}
\tag{3.8}
$$

$$
\begin{aligned}
\Delta &= \sum_t P(z = t|\mathbf{z}_-, w) \\
&= \left\{\underbrace{\sum_t \frac{\alpha\beta}{n_t + V\beta}}_{\text{s}} + \underbrace{\sum_{t, n_{d,t} > 0} \frac{n_{d,t}\beta}{n_t + V\beta}}_{\text{r}} + \underbrace{\sum_{t, n_{w,t} > 0} \frac{(n_{d,t} + \alpha)n_{w,t}}{n_t + V\beta}}_{\text{q}}\right\} \\
&\times \left\{\prod_{u \in M_w^m} \max(\lambda, n_{u,t}) \prod_{v \in M_w^c} \frac{1}{\max(\lambda, n_{v,t})}\right\}
\end{aligned}
\tag{3.9}
$$

As explained above, λ controls the "strength" of the prior knowledge term. If λ is a very large number, the second term essentially multiply the same value to the first term, which means the prior knowledge has little impact on the conditional probability of topic samples.

Let's return to the question whether Equation 3.7 is sparse so that Equation 3.11 can be efficiently computed. Fortunately, $n_{u,t}$ and $n_{v,t}$, which are the topic counts for must-link word u and cannot-link word v, are often sparse. For example, in a 100-topic model trained on NIPS dataset, 87.2% word types have less than 10 topics with nonzero counts. In a 500-topic model trained on a larger dataset like New York Times News dataset, 81.9% word types have less than 50 topics with nonzero counts. Moreover, the model becomes

increasingly sparse as more Gibbs sampling iterations are executed. Figure 3.2 shows the word frequency histogram of nonzero topic counts of NYT-News dataset.

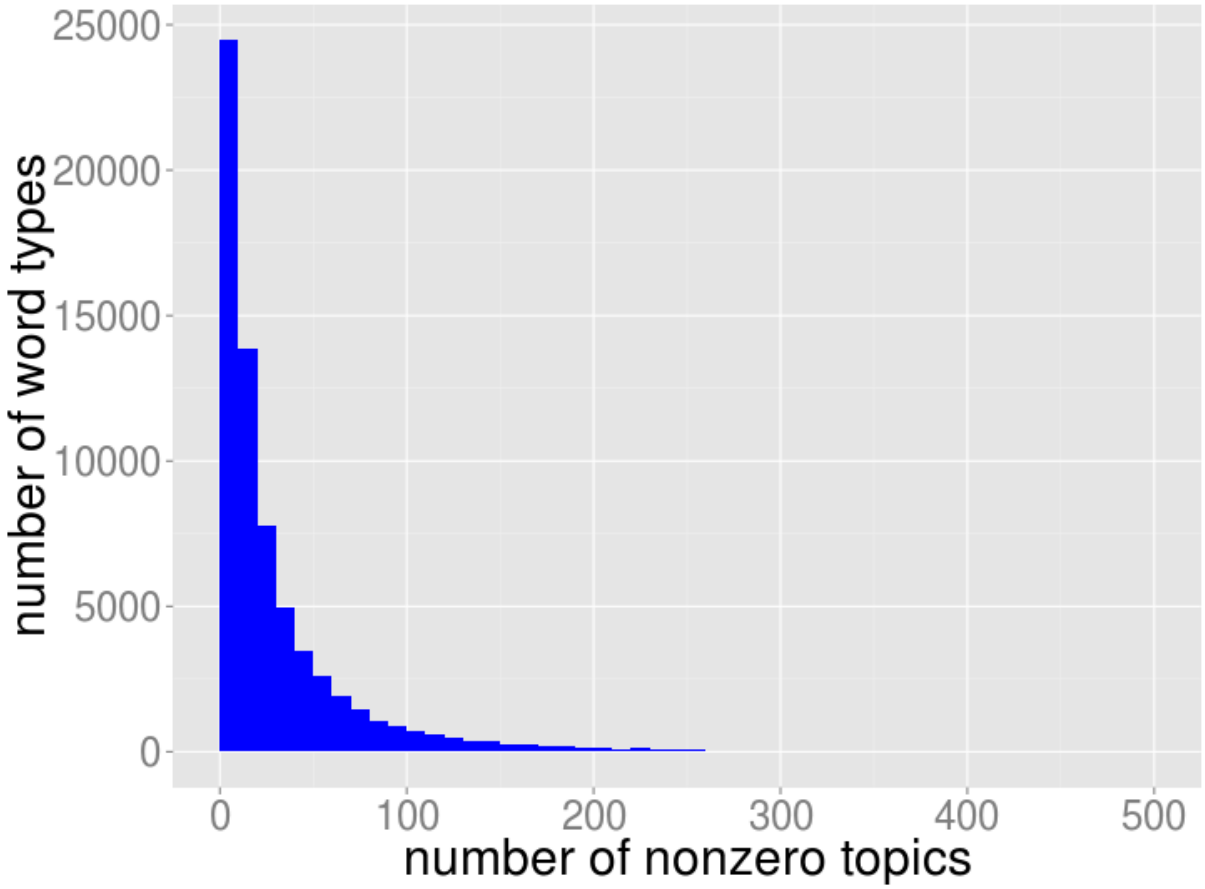


Figure 3.2. Histogram of nonzero topic counts for word types in NYT-News dataset after inference. 81.9% word types have less than 50 topics with nonzero counts.

Therefore, the computational cost of Equation 3.9 can be saved. The first term s is the "smoothing only" bucket that is constant for all documents. The second term r is the "document only" bucket that is shared by a document's tokens. Both s and r have simple constant time updates. The last term q has to be computed specifically for each token, only for the few types with non-zero counts in a topic, due to the sparsity of word-topic count.

Since q often has the largest mass and few non-zero terms, we start the sampling from bucket q. We can utilize the same technique as used in SparseLDA to efficiently compute the first term in Equation 3.9. Then for words that are associated with prior knowledge, we update s, r, q with an additional potential term. We only need to compute the potential term for the topics whose counts are greater than λ. The collapsed Gibbs sampling procedure is summarized in Algorithm 2.

Algorithm 2: Incorporating Word Correlation: Gibbs Sampling for word type w in document d, given w's correlation set M_w

compute s_t, r_t, q_t with SparseLDA ;
for $t \leftarrow 1$ **to** T **do**
    update s_t, r_t, q_t according to Eq.3.11. $\forall u \in M_w$ if $n_{u,t} > \lambda$
end
$p(t) = s_t + r_t + q_t$;
sample new topic assignment for w from $p(t)$;

3.1.2. Other Types of Prior Knowledge

The factor model framework can also handle other types of prior knowledge, such as document labels, sentence labels, and document link relations. We describe document label here.

[**48**] proposed Labeled-LDA to utilize document labels. It assumes that there is a one-to-one mapping between topics and labels, and it restricts each document's topics to be sampled only from its labels. Therefore, the idea of Labeled-LDA can be easily applied in our model. We can define

$$f_m(z, w, d) = \begin{cases} 1, & \text{if } z \in m_d \\ -\inf, & \text{else} \end{cases} \tag{3.10}$$

where m_d specifies document d's label knowledge, and it has been converted to corresponding topic labels. Since $f_m(z, w, d)$ is sparse, we can speed up the training as well. Sentence label prior knowledge can be defined in a similar way.

Given Equation 3.10, the conditional probability of a topic z in document d given and observed word type w is independent with the word type, and it is:

$$\begin{aligned} & P(z = t|w, \mathbf{z}_-, M) \\ & = \frac{1}{\Delta}\left\{\frac{\alpha\beta}{n_t + V\beta} + \frac{n_{d,t}\beta}{n_t + V\beta} + \frac{(n_{d,t} + \alpha)n_{w,t}}{n_t + V\beta}\right\} f_m(z, w, d) \\ & = \begin{cases} \frac{1}{\Delta}\left\{\frac{\alpha\beta}{n_t+V\beta} + \frac{n_{d,t}\beta}{n_t+V\beta} + \frac{(n_{d,t}+\alpha)n_{w,t}}{n_t+V\beta}\right\}, & \text{if } z \in m_d \\ 0, & \text{else} \end{cases} \end{aligned} \tag{3.11}$$

The collapsed Gibbs sampling procedure is summarized in Algorithm 3.

Algorithm 3: Incorporating Document Label Knowledge: Gibbs Sampling for word type w in document d, given d's label set m_d

```
compute s_t, r_t, q_t with SparseLDA ;
for t ← 1 to T do
    if t ∈ m_d then
        do nothing;
    else
        set s_t, r_t, q_t = 0;
    end
end
p(t) = s_t + r_t + q_t ;
sample new topic assignment for w from p(t) ;
```

There also exist another type of side information about document, such as link relation. For example, a scientific paper and its citations might be similar in topics. We can use Equation 3.7 and replace word-topic counts $n_{v,z}$ with document-topic counts $n_{d,z}$. By doing

that, we encourage related documents to have similar topic structures. Moreover, document-topic count is also sparse, which fits into the efficient learning framework.

Therefore, for different types of prior knowledge, as long as we can define $\psi(\mathbf{z}, M)$ appropriately so that $f(z, w, d)$ is sparse, we are able to speed up the learning.

3.2. Experiments

In this section, we demonstrate the effectiveness of our model SC-LDA by comparing it with several baseline methods on three benchmark datasets. We first evaluate the convergence rate of each method, and then we evaluate the learned model parameter ϕ, topic-word distribution, in terms of topic coherence. We show that SC-LDA can achieve comparable results with the baseline models, but runs significantly faster. We set up all experiments on a 8-Core 2.8GHz CPU, 16GB RAM machine.

3.2.1. Dataset

Table 3.1. Characteristics of benchmark datasets. NIPS and NYT-News datasets are used for word correlation experiments, and 20NG dataset is used for document label experiments.

Dataset	docs	type	token(approx)
NIPS	1,500	12,419	1,900,000
NYT-News	3,000,000	102,660	100,000,000
20NG	18,828	21,514	1,946,000

We use the NIPS and NYT-News datasets from the UCI bag of words data collections[2] in our experiment. These two datasets have no document labels, and we use them for word correlation experiments. We also use the 20Newsgroup (20NG) dataset [3], which has

[2] https://archive.ics.uci.edu/ml/datasets/Bag+of+Words
[3] http://qwone.com/~jason/20Newsgroups/

document labels, and we use it for document label experiment. The detailed characteristics of each dataset are listed in Table 3.1. Since NIPS and NYT-News have already been preprocessed, to ensure repeatability, we use the data "as they are" from the sources. For 20NG, we perform tokenization and stopword removal using Mallet [**35**], and we also remove words which appears less than 10 times.

3.2.2. Prior Knowledge Generation

Word Correlation Prior Knowledge. Previous work propose two methods to automatically generate prior word correlation knowledge from external sources. [**25**] use WordNet 3.0 to obtain synsets for word types, and then if a synset is also in the vocabulary, they add a must-link correlation between the word type and the synset. [**64**] use a different method that takes advantage of an existing pretrained word embedding. Each word embedding is a real-valued vector capturing its semantic meaning based on distributional similarity. If the similarity between two word types in the vocabulary exceeds a threshold, they generate a correlation between the two words.

In our experiments, we adopt a hybrid method that combines the above two methods. For a noun word type, we first obtain its synsets from WordNet 3.0. We also obtain the embeddings of each word from word2vec [**37**]. If the synset is also in the vocabulary, and the similarity between the synset and the word is higher than a threshold, which in our experiment is 0.2, we generate a correlation between them. Empirically, this hybrid method is able to obtain high quality correlated words. For example, for the NIPS dataset, the correlated words we obtain for *randomness* are {noise, entropy, stochasticity}.

Document Label Prior Knowledge. Since documents in the 20NG dataset are associated with labels, we use the labels directly as prior knowledge.

3.2.3. Baselines

The baseline methods for incorporating word correlation prior knowledge in our experiments are as follows:

DF-LDA: incorporates word must-links and cannot-links knowledge using a Dirichlet Forest prior in LDA [**3**]. Here we use [**25**]'s efficient implementation *FAST-RB-SDW* for DF-LDA.
Logic-LDA: encodes general domain knowledge as first-order logic and incorporates it in LDA [**4**]. Logic-LDA has been utilized for word correlations, and document label knowledge.
MRF-LDA: encodes word correlation prior knowledge in LDA as Markov random field [**64**].

We also use Mallet's SparseLDA implementation for vanilla LDA in topic coherence experiment. We use a symmetric Dirichlet prior for all models. We set $\alpha = 1.0$, $\beta = 0.01$. For DF-LDA, $\eta = 100$.[4] For Logic-LDA, we use the default parameter setting in the package, which the sample rate was set to 1.0, and the step rate was set to 10.0. For MRF-LDA, we use the default setting where $\gamma = 1.0$. The meaning of the parameters can be found in the papers respectively.

3.2.4. Convergence Rate

The main advantage of our method over other existing methods is efficiency. In this experiment, we show the change of model's loglikelihood over time. In topic model, the loglikelihood change is a good indicator of whether a model is converged or not. Figure 2 shows the loglikelihood change over time for SC-LDA and three baseline methods on NIPS and

[4]It is also worth noting that we do not optimize hyperparameter during the inference, although Mallet provides hyperparameter optimization for training LDA. The reason is that hyperparameter optimization is not available in other baseline methods, so to make the experiment comparison fair and to not introduce extra computational cost for SparseLDA, we choose not to optimize hyperparameter.

NYT-News dataset. It can be seen from the figure that SC-LDA converges at a faster rate than all the other methods.

We also conduct an experiments on SC-LDA with different number of word correlations. Table 3.2 shows the Gibbs sampling iteration time on the 1st, 50th, 100th and the 200th iteration. We also incorporate different number of word correlations to SC-LDA. It can be seen from the table that SC-LDA runs faster in the later iterations than the beginning iteration. This is because as model becomes more sparse, the more speedup we can gain from the sparsity. The table also shows that more correlation knowledge does increase the iteration time, but the impact is relatively small.

Table 3.2. SC-LDA runtime (in seconds) in the 1st, 50th, 100th, 200th iteration round with different number of correlations.

	Number of Word Correlations			
round	C0	C100	C500	C1000
1st iteration	2.02	2.14	2.30	2.50
50th iteration	0.53	0.56	0.58	0.62
100th iteration	0.48	0.50	0.53	0.56
200th iteration	0.48	0.49	0.52	0.56

3.2.5. Topic Coherence

Topic models are often evaluated using perplexity on held-out test data. However, recent researches [**10**] have shown that human judgment sometimes is contrary to the perplexity measure. Following [**39**], we employ Topic Coherence, a metric which was shown to be highly consistent with human judgment, to measure a topic model's quality. Topic t's coherence is defined as:

$$C(t : V^{(t)}) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{F(v_m^{(t)}, v_l^{(t)}) + \epsilon}{F(v_l^{(t)})} \tag{3.12}$$

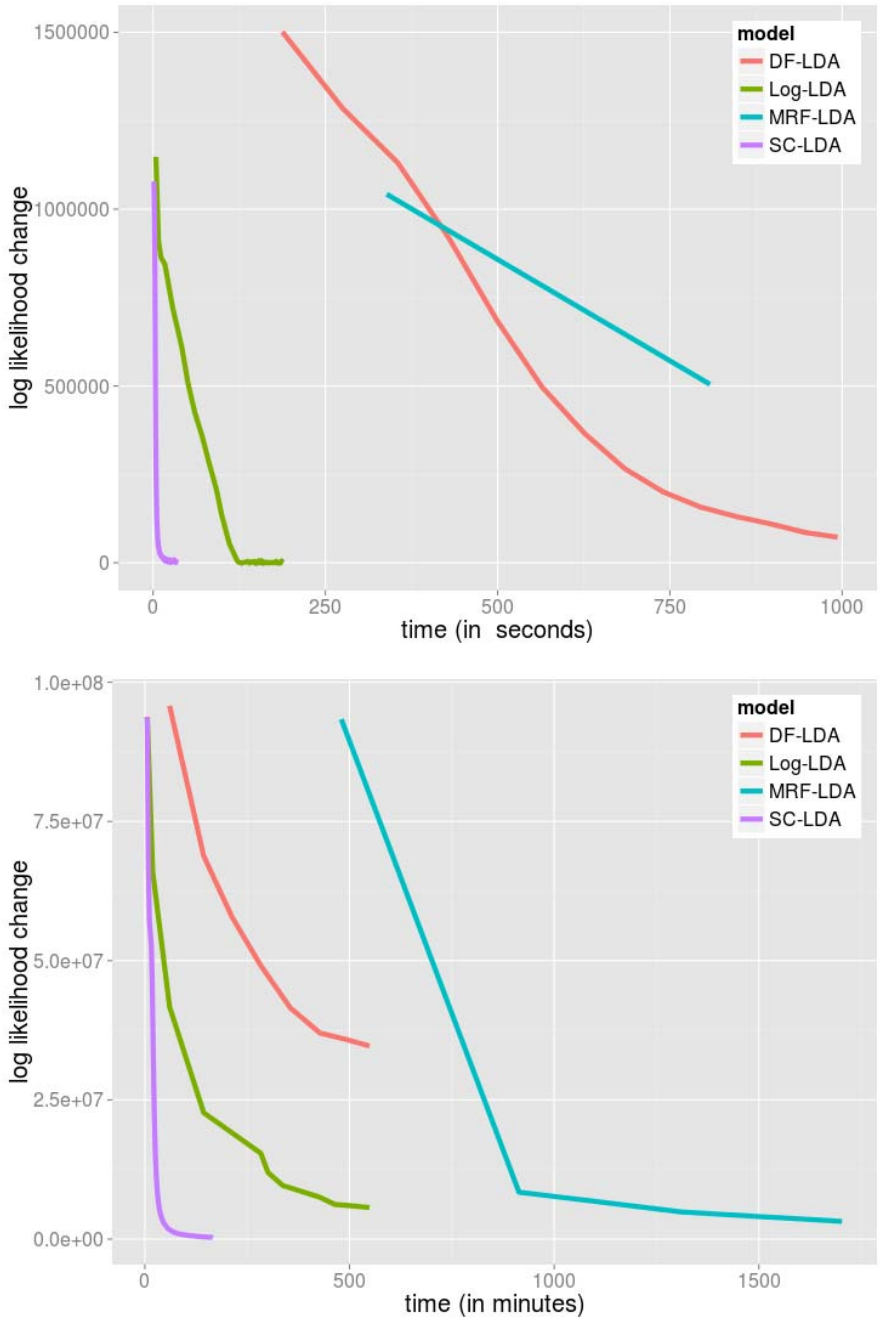


Figure 3.3. Model's log likelihood convergence of different methods on NIPS dataset (above) and NYT-News dataset (below). For NIPS, a 100-topic model with 100 correlations is trained. For NYT-News, a 500-topic model with 100 correlations is trained. SC-LDA reaches likelihood convergence much more rapidly than the other methods.

where $F(v)$ is the document frequency of word type v, $F(v, v')$ is the co-document frequency of word type v and v', and $V^{(t)} = (v_1^{(t)}, ..., v_M^{(t)})$ is a list of the M most probable words in topic t. In our experiments, we choose the 10 most probable words to compute topic coherence, i.e., $M = 10$. In [**39**] $\epsilon = 1$, but [**52**] shows that by setting ϵ to a small number such as 10^{-12}, the coherence score becomes more stable, so we set $\epsilon = 10^{-12}$. The larger topic coherence value, the more coherent the topic keywords, and the more interpretable for human users.

We train a 500-topic model on NIPS dataset with different methods and compare the average topic coherence score and the average of top 20 topic coherence score. Since the topics learned by topic model often contains "bad" topics [**39**] which do not make sense to end users, evaluating top 20 topic quality can also reflect the model's performance. We let each model to train for one hour and then stop. Figure 3.4 shows the topic coherence performance of each methods. It can be seen that SC-LDA has about the same average topic coherence with LDA, but it has higher coherence score (-36.6) for the top 20 topics than LDA (-39.1). The reason for this is because by incorporating word correlation knowledge, words that are highly correlated tend to also have high probability under the same topic, thus improving the coherence score. For the other methods, however, because they cannot converge within an hour time limit, the topic coherence scores are worse than SC-LDA and LDA. This again demonstrates the efficiency of SC-LDA over other baselines.

3.2.6. Document Label Prior Knowledge

SC-LDA can also handle other types of prior knowledge. We compare it with Label-LDA [**48**]. Since Label-LDA uses Gibbs sampling for parameter estimation, we are able to compare average iteration time of the sampler.

Figure 3.4. Average topic coherence and average top 20 topic coherence. The models are trained on NIPS dataset with 500-topic and 100 word correlations. SC-LDA achieves higher topic coherence than other methods.

Table 3.3. The average running time per iteration over 100 iterations, averaged over 5 seeds, on 20NG dataset. Experiments begin with 100 topics, 1000 labeled documents, and then vary one dimension: number of topics (top), and number of labeled documents (bottom).

Number of Topics				
	T50	T100	T200	T500
Label-LDA	0.93	1.89	3.60	8.05
SC-LDA	0.38	0.45	0.51	0.72
Number of Labeled Documents				
	C500	C1000	C2000	C5000
Label-LDA	1.95	1.88	1.75	1.48
SC-LDA	0.51	0.45	0.41	0.31

Table 3.3 shows the average running time per iteration for Label-LDA and SC-LDA. Because document labels apply sparsity to the document-topic counts, the average running

time per iteration decreases as the number of labeled document increases. Table 3.3 also shows that SC-LDA gains more speedup as the number of topics increases. When $T = 500$,[5] SC-LDA is running more than 10 times faster than Label-LDA.

3.3. Related Work

This work is closely related to two areas of topic modeling.

The first related area is incorporating prior knowledge into topic models. [**34**] proposes a supervised version of LDA where document labels are known, and it uses variational EM algorithm to estimate the model parameter. [**48**] adopts an intuitive way of utilizing document labels. It assumes that there is a one-to-one mapping between topics and labels, and it restricts each document's topics to be sampled only from its labels. Therefore, collapsed Gibbs sampling can be directly used in Labeled-LDA. However, when document labels are overlapped and messy, the performance of Labeled-LDA drops. [**27**] presents a topic model that makes use of user-specified graphs describing relationships between documents. These graph are encoded in the form of a Markov random field over topics. Word correlation knowledge is also studied to incorporate with topic model. [**3**] incorporates word correlation knowledge using Dirichlet Forest prior in LDA. The prior is a mixture of Dirichlet tree distributions with special structures. [**4**] proposes a general framework which allows the model to incorporate general domain knowledge in First-Order Logic. [**64**] incorporates word correlation to LDA by building a Markov Random Field regularization on top of hidden topic variables, and they use variational inference to learn the parameters. [**42**] proposes a regularized method to improve topic coherence by creating a structured prior over words

[5]For 20NG dataset, it may overfit the data with 500 topics, but here we use it to demonstrate the scalability.

that reflect broad patterns in the external data. All the experiments in the above work are reported on a relatively small dataset.

The second related area is efficient learning of topic models so that the model can scale to large corpora and large number of topics. Variational Inference and Collapsed Gibbs Sampling are two popular learning methods for topic modeling. Due to the simplicity of Collapsed Gibbs Sampling, many topic model extensions resort to this method. [**47**] proposes Fast-LDA by constructing an adaptive upper bound on the sampling distribution and achieves a faster inference. [**65**] presents Sparse LDA, which takes advantage of the sparsity of model, and it significantly reduces the sampling complexity. Moreover, since it benefits from sparsity, the larger the topic space, the greater the speedup Sparse LDA can achieve over conventional LDA. Sparse LDA has been deployed in the commercial topic modeling system [**62**]. [**30**] reports a even faster sampling techniques by constructing an alias table before sampling. [**66**] presented an efficient constant time sampling algorithm for building big topic models. In addition to speeding up Collapsed Gibbs Sampling, [**23**] developed an online variational Bayes algorithm for inference LDA. All the above work aims to speed up learning algorithm for LDA. [**25**] is the first work to speed up topic model with prior knowledge. It extends Sparse LDA inference scheme for Tree-based LDA, but the sampler still needs to sample along the tree, which makes it still slow to converge. There is also work on paralleling the training across multiple processors so as to achieve faster learning. In this paper, we only focus on single-processor learning, but existing parallelization techniques [**41**] are applicable to our model. The primary distinction of our work with previous work is that we present a factor graph framework with efficient sampling algorithm to incorporate prior knowledge so that it can scale to large corpora and many topics.

3.4. Conclusion

We present a factor graph framework for incorporating prior knowledge into topic models. By constraining prior knowledge on hidden topics directly, we are able to take advantage of model's sparsity and speed up the training. We demonstrate in experiments that our model runs significantly faster than the other alternative models and achieves comparable performance in terms of topic coherence. Efficient algorithm for incorporating prior knowledge with large topic models will benefit several downstream applications. For example, interactive topic modeling becomes feasible because the fast model update can reduce the user's waiting time, and therefore improve the user experience. Personalized topic modeling is also an interesting future direction in which the model will generate a personalized topic structure based on the user's preferences or interests. For all these applications, efficient learning algorithm plays a fundamental role. We expect this work will interest data analysis practitioners who would like to efficiently train a large topic model with prior knowledge.

CHAPTER 4

User-directed Topic Model Update

In an age of information abundance and explosion, providing interactive interfaces to guide users through the navigation and exploration of the vast information space becomes a necessity for information applications [**45, 32**]. To achieve this goal, applications commonly employ statistical topic modeling techniques to structuralize content collections by grouping documents into coherent topics, and utilize interactive visualizations at the interfaces to support topic-centric navigation and exploration.

LDA [**8**] is one of the most commonly used approaches to uncover hidden thematic patterns in text. Since exact LDA inference is intractable, statistical inference methods such as Gibbs Sampling and Variational Inference, are often used. Although these methods are versatile solutions for a wide range of intractable inference problems, they may produce different results each time they are run. For example, in LDA, multiple runs of Gibbs Sampling on the same input text may return different results. This problem is more severe during LDA model update. Due to the highly dynamic nature of many content collections where new documents are frequently added, updates to existing topic models are necessary in order to capture the changes to topics (e.g. the emergence of new topics and the evolution of existing topics) over time. To handle update, standard (batch) LDA requires the topic model to be regenerated from scratch based on the updated dataset (including both old and new content) [**65**]. Because of the unpredictability of the statistical inference methods and the influence of the new content, the topic assignments of the old documents may vary

significantly before and after the update. When topic model updates introduce substantial changes in the topic assignments of existing documents, the users' mental maps between documents and topics may be disrupted, leading to increased cognitive load of the users and negatively impacting the usability of the applications. Therefore, how to maintain *topic model stability* becomes a critical issue.

Let us use an example to illustrate why the lack of topic model stability may pose a serious problem. Here we focus on the update scenario where new documents become available. A user named Jane frequently needs to explore publications in certain fields for her work. She has been using a software tool that trains an LDA topic model on all the papers from a publication database to classify the collection of papers into three topics: Natural Language Processing (NLP), Speech Processing (SP), and Information Management (IM). Realizing that the topic model was created a year ago, Jane requested an update. The tool updated the topic model based on the current publication database, which contains all the previous papers as well as the new papers published since last year. After the update however, Jane cannot locate some of the papers she frequently visited before, because they are re-assigned to a different topic. For instance, some old NLP papers are now under the same topic as other SP papers. In a way, the mental map Jane has built for the paper collection is disrupted, resulting in confusion and frustration. The tool has become less useful to Jane unless she puts in some effort to update her mental map, which significantly increases her cognitive load.

Although other alternatives such as online LDA [**56, 5, 9, 23, 68**] can alleviate this problem, they can not solve the problem entirely. For example, for online LDA, we can keep the model relatively stable in the short term by making small incremental updates to the model. But over time, the topic model may still undergo significant changes. Since topic

model stability is not an optimization goal for online LDA, for the same document, its topic assignment a few months ago can still be different from that inferred based on the current model. So far, there has not been much work systematically addressing the stability issue of LDA. The stability and usability of the model has increasingly become one of the main reasons for not using LDA in practice [**11**].

In this chapter, we introduce a new method that explicitly encodes topic stability as an additional constraint in LDA topic models. In this study, we focus on ensuring topic model stability during model update by employing a novel user-directed non-disruptive topic model update method (**nTMU**). We mainly focus on the stability issue during topic model update because it has even more severe instability problem than none-update case where the model is re-trained on the same dataset. The proposed topic model update method, nTMU, directly incorporates topic stability constraints to minimize the changes to the topic assignments of the same documents before and after model update. As a result, it can effectively maintain the stability of document-to-topic mappings and minimize the disruption to the mental maps of end users. The core module of nTMU runs a novel constrained topic modeling algorithm cLDA. It allows LDA to incorporate pair-wise document constraints such as document must-links and cannot-links. Evaluation on a benchmark dataset as well as a user study have demonstrated the effectiveness of our approach in the topic model update scenario. We believe that both the nTMU system and the cLDA algorithm are significant contributions to the field since as far as we know, there isn't any other system that addresses the topic model stability issue explicitly.

4.1. Topic Model Instability

In machine learning, a learning algorithm is said to be *unstable* if it is sensitive to small changes in the training data. Turney [**60**] describes a situation in which Decision Tree algorithms were used by engineers to help them understand the sources of low yield in a manufacturing process: "The engineers frequently have good reasons for believing that the causes of low yield are relatively constant over time. Therefore the engineers are disturbed when different batches of data from the same process result in radically different decision trees. The engineers lose confidence in the decision trees, even when we can demonstrate that the trees have high predictive accuracy."

The problem topic models suffer from is the same or even worse. For topic models, similar to decision tree learning, instability can happen when there is a change in the input data (e.g., when new documents become available). It can also occur when the same inference algorithm is run multiple times on the same data. For an end user, topic model instability can be felt from the changes of the topics assigned to the same documents. Internally, if a topic model also assigns a topic to each word token in each document, the instability can also be observed in the changes of the topics assigned to the same word tokens in the same documents.

To quantify the severity of the instability problem of topic models, we conduct a few experiments. In these experiments, we focus on topic models that use Gibbs sampling. In the discussion section, we will talk about whether this can be generalized to topic models that use other inference methods such as Variational Inference [**8**].

The reason for us to focus on Gibbs Sampling is because it is the method that has been widely adopted in practical applications like search engines and online advertising [**62**]. In

practice, efficiency and scalability are essential since we frequently need to process a massive amount of data. With Gibbs Sampling, novel techniques exist to significantly reduce its computational complexity [**65**]. Moreover, efficient parallelization framework exists to make it even more scalable.

In the following, we firstly review how to assign topic label to a document and how to align two topic models, we then define the measures we used to assess the stability of a topic model. Then we describe the experiments we conducted to quantify the severity of the instability problem of topic models.

4.1.1. Topic Label Assignments for Documents

Given document d's topic distribution θ_d, we can assign d's topic label l_d to be the topic that has the maximum value in θ_d. For example, assume a document's topic distribution learned by a 5-topic LDA model is $\{0.1, 0.1, 0.05, 0.7, 0.05\}$, then we will assign 4 [1] to be the topic label of this document since this topic has the highest proability value.

However, it is possible that some topics learned by LDA are only "background" topics which have significant non-trivial probabilities over many documents [**57**]. Since background topics are often uninteresting, a weighted topic distribution could be used to filter them. Therefore, we normalize θ_{dt} by the sum of the tth topic value over all documents. This idea is similar to the inverse document frequency used in tf-idf, which normalizes the weight of a word by its overall occurrences in all documents. A weighted tth component θ'_{dt} for document d is computed as follows:

$$\theta'_{dt} = \theta_{dt} / \sum_{i=0}^{D} \theta_{it}. \tag{4.1}$$

[1] Here we follow the convention that topic index $\in \{1, ..., T\}$

Then, instead of assigning the document label to the topic that has the maximum value, we choose the topic that has the maximum normalized value. Therefore, the topic label of document d is $l_d = \arg\max_{t \in T} \theta'_{dt}$.

Note that there are other methods of assigning a topic label to a document. For example, we can use the per-document topic distribution as a feature vector in an unsupervised clustering algorithm such as k-means clustering, where the number of clusters equals to the number of topics in LDA. In our work, for simplicity and to prevent the confound effect of another machine learning algorithm (e.g., clustering) when computing model stability, we utilize the straightforward method described above for topic label assignment.

4.1.2. Topic Alignment

In LDA, we assume that words are generated by topics (by fixed conditional distributions) and that those topics are infinitely exchangeable within a document. Due to the exchangeability of statistical topic models, even two models with exactly the same parameters could have different topic indexes. Moreover, since Gibbs sampler randomly initialize topic samples for each document (to avoid local optima), the topic label of the same topic in two models can be different (e.g., in the 1st model, the topic index for the "NLP" topic is "2" while in the 2nd model, the topic index of the same topic is "5"). To facilitate the comparison of two topic models, we need to align the topic indexes to make the labels consistent. To efficiently solve the matching problem, we use the Hungarian algorithm [2] to align the topic indexes of two models. The Hungarian algorithm will find a permutation of the topic indexes so that two models have the closest document label assignments. In the following sections, when comparing the stability of two topic models, we assume that the models have been aligned.

[2] http://en.wikipedia.org/wiki/Hungarian_algorithm

4.1.3. Stability Measures

In the experiments, we use two measures to quantify the stability of a topic model: *document-level stability* and *token-level stability*. Specifically, we use the Equation 4.2 to define *document-level stability*. Here we denote D_1 as the dataset used to train the old topic model, D_2 as the new dataset and $D_3 = D_1 \cup D_2$ as the total dataset used to train the new topic model. Moreover, we define l_{1i} as the topic label assigned to the ith document based on the old topic model, and l_{2i} as the topic label assigned to the same document after the model is updated to the new topic model.

$$S_d = (1 - \frac{\sum_{d_i \in D_1} I(l_{1i} \neq l_{2i})}{|D_1|}) * 100\%, \tag{4.2}$$

where $I(.)$ is the indication function. S_d equals to 100% when the topic labels for all the documents in D_1 remain the same after the update.

If we only re-train the topic model on the same dataset, then D_2 is empty and D_1 is the same as D_3. So re-training a topic model on the same dataset is a special case of updating the topic model with additional new documents. Figure **??** illustrates the flow of new data.

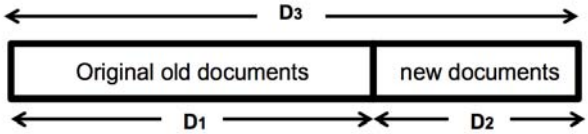


Figure 4.1. Documents flow in a topic model update system.

In addition, since LDA with Gibbs Sampling also assigns a topic to each word token in each document, we also define a *token-level stability* measure. Although token-level topic assignments are internal to LDA and invisible to end users, we compute this to provide additional insight into this problem. Here we denote l_{1ij} as the topic assignment for the jth

word token in ith document based on the old topic model and 2_{1ij} as the topic assignment for the jth word token in the ith document based on the new topic model.

$$S_t = (1 - \frac{\sum_{d_i \in D_1} \sum_{j \in d_i} I(l_{1ij} \neq l_{2ij})}{\sum_{d_i \in D_1} \sum_{j \in d_i}}) * 100\%, \tag{4.3}$$

where $I(.)$ is the indication function. S_t equals to 100% when the topics for all the word tokens in all the documents in D_1 remain the same after the model is re-trained.

4.1.4. Experiments

We compute the above two stability measures in two different scenarios: (1) None-Update scenario in which we re-train the topic model on the same dataset, (2) Update scenario in which we re-train the topic model after new documents are added to the existing dataset. As we have already discussed, the none-Update scenario is a special case of the update scenario, when the new document dataset D_2 is empty. In the update scenario, in order to create the update environment, we split each dataset into two halves. We first train a before-model using one half of the data and then train an after-model after we add the second half of the data. Since in the before-model, we don't have the topic analysis results for the new documents, we can only compute the stability measures based on the documents in the first half of the data. Moreover, since the same topic may have different indexes in the before and after models, we apply topic alignment to match the topic indexes from different models. We conduct the experiments using two commonly used benchmark datasets: the 20 Newsgroup dataset[3] and the NIPS dataset[4]. During the experiments, we use a typical parameter setting employed in previous studies: the number of topics was set to 20, the number of iterations

[3] `http://qwone.com/~jason/20Newsgroups/`
[4] `http://psiexp.ss.uci.edu/research/programsdata/toolbox.htm`

was 1000. We use a uniform α with a value of 1.0, a uniform β with a value of 0.01. Table 4.1 shows the results.

From the table, it can be seen that in the first scenario where the input data remain the same, only around 60% of the documents have the same topic assignments between the before and after models (it was 52.7% on 20 Newsgroup data and 60.3% on NIPS data). The problem is even worse in scenario 2, the update scenario. In this case, less than 45% of the documents were assigned the same topic labels (it was 44.5% on 20 Newsgroup and 43.3% on NIPS data). Similarly, at the token level, in scenario 1, 42.2% and 39.2% of the word tokens were kept the same topic assignments in the 20 Newsgroup and NIPS data respectively. Again, the problem is worse in the update scenario in which only 38.2% of the word tokens in the 20 Newsgroup data and 30.8% of the word tokens in the NIPS data still have the same topic assignments.

Table 4.1. Topic model instability on 20 Newsgroup and NIPS datasets.

Dataset	**20 Newsgroup**		**NIPS**	
scenarios	None-Update	Update	None-Update	Update
document-level stability	52.7%	44.5%	60.3%	43.3%
token-level stability	42.2%	38.2%	39.2%	30.8%

We speculate that there might be several reasons behind the instability problem in topic modeling: *different local optima*, *model convergence* and *new data*. First, in topic modeling, since computing the posterior distributions of model parameters is computationally intractable, approximated inference methods such as Gibbs sampling are often used. Since LDA is a non-convex model, when initialized with different random seeds, different runs of these methods may converge to different local optima. Thus, even with the same input data, the same inference algorithm may produce different results on two separate runs, which may cause the instability of the inference results. Second, for Gibbs Sampling, frequently

there is no specific criteria to test the convergence of the model. Thus in practice, we often use a pre-determined iteration number (e.g., 1000). Since with different random seeds, different runs of the same inference algorithm may have different convergence speed, using a pre-determined iteration number may cause some runs to end prematurely. Models that ended prematurely may produce very different results, which may cause the instability of the inference results. Third, since fitting the input data is the main optimization criteria in model training, when new data are added into the input, it is expected that the model would need to adjust its parameters to fit the new data. New model parameters will cause change in the topic assignments to documents as well as to word-tokens.

In summary, these results also show that topic instability is a significant problem for topic modeling. On average, almost half of documents/word-tokens will receive a different topic assignment when a topic model is re-trained. Instability problem may severely hinder a topic model's usability and its adoption in practice. Since instability ratio is higher in the second scenario where new documents are added into the dataset, in the rest of the paper, we focus on developing solutions that maintain topic stability during model update. Same methods should also apply to the first scenario where the input data remain the same (It is a special case in which zero new document is added to the dataset). In the following section, we describe a novel user-directed non-disruptive topic model update algorithm (nTMU) that can maintain topic stability when new data are added to the input data.

4.2. Non-Disruptive Topic Model Update (nTMU)

The standard topic model update systems do not have users in the loop. When new data are coming in, based on different topic model update strategies [**65**], topic models are re-trained on either both old data and new data, or only the new data. As we will

show in the System Evaluation section, such topic model update strategies have poor topic model stability since their main objective is only to fit the data into the model. To alleviate the instability problem, we propose a user-directed topic model update method. Figure 4.2 illustrates our system nTMU (user-directed **n**on-disruptive **T**opic **M**odel **U**pdate). It consists of four main components and includes users as a critical part. Below we describe these components in counter-clockwise order as shown in the figure.

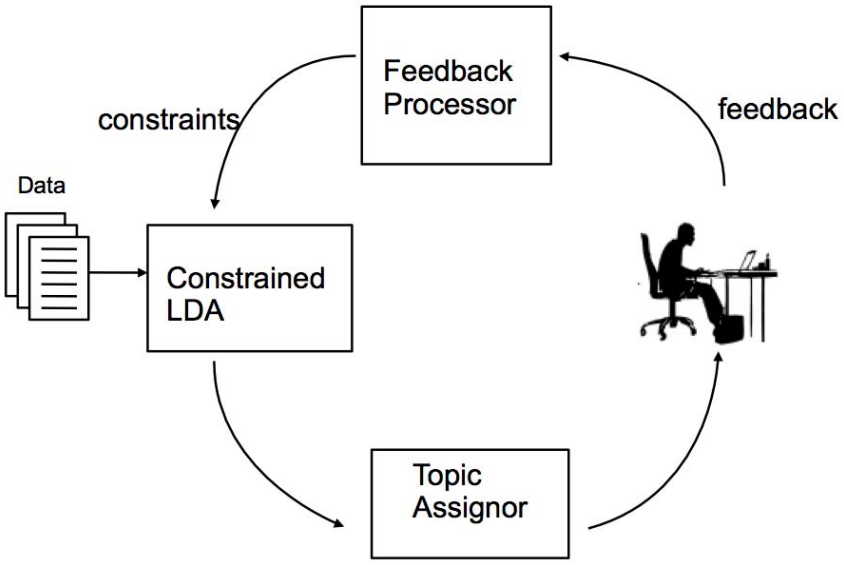


Figure 4.2. Non-Disruptive Topic Model Update diagram.

Data, such as text collections, arrive at the nTMU system in an online fashion. The topic model is created and updated by the *Constrained LDA* component, which employs the cLDA algorithm (described in detail in the next section). Unlike unsupervised standard LDA, cLDA is capable of taking user feedback encoded as document must-links and cannot-links. In the first round when the nTMU system is newly started with an initial set of data and no user feedback, cLDA behaves the same as standard LDA without any constraints.

The system does allow users to provide prior domain knowledge/labels, but we assume that there is no prior information available in the current setting.

Given the topic modeling results from the *Constrained LDA* component, the *Topic Assignor* component assigns a topic label to each document using the method discussed in Section 4.1.1. Since a Gibbs sampler randomly initializes topic samples for each document, the topic label of the same topic before and after update can be different, and we need to align the topic label to make the labels consistent. As we have discussed in section 4.1.2, we use the Hungarian algorithm to match the topic indexes. It will find an optimal matching that maximizes the number of the same document label before and after the update.

Next, the *User Interface* component presents the labeled results to the end users. It displays for each topic the most probable words based on topic-word distributions (Equation 2), along with a treemap-like visualization showing the twenty most representative documents that are assigned to this topic. Users can also click on the "Show More button" to see more representative documents. The title of each document is displayed, and if a user clicks on a document, a summary window will pop out to show the detailed information of this document, including time, author and highlights from the content. Figure 4.3 provides a screen shot of the user interface. The treemap-like visualization has been demonstrated to be effective to support topic-centric navigation and exploration of document collections [**29**].

The system allows the user to provide feedback about the topics through the interface. In particular, for each topic, the system asks the user whether the topic is coherent and whether s/he wants to keep it. For example, if the content of the representative documents of the topic is consistent with this topic, the user would respond with *Yes* to the system. However, if the user decides that s/he does not care about the topic or the topic has incoherent keywords or inconsistent representative documents, he/she would response *No* to the system.

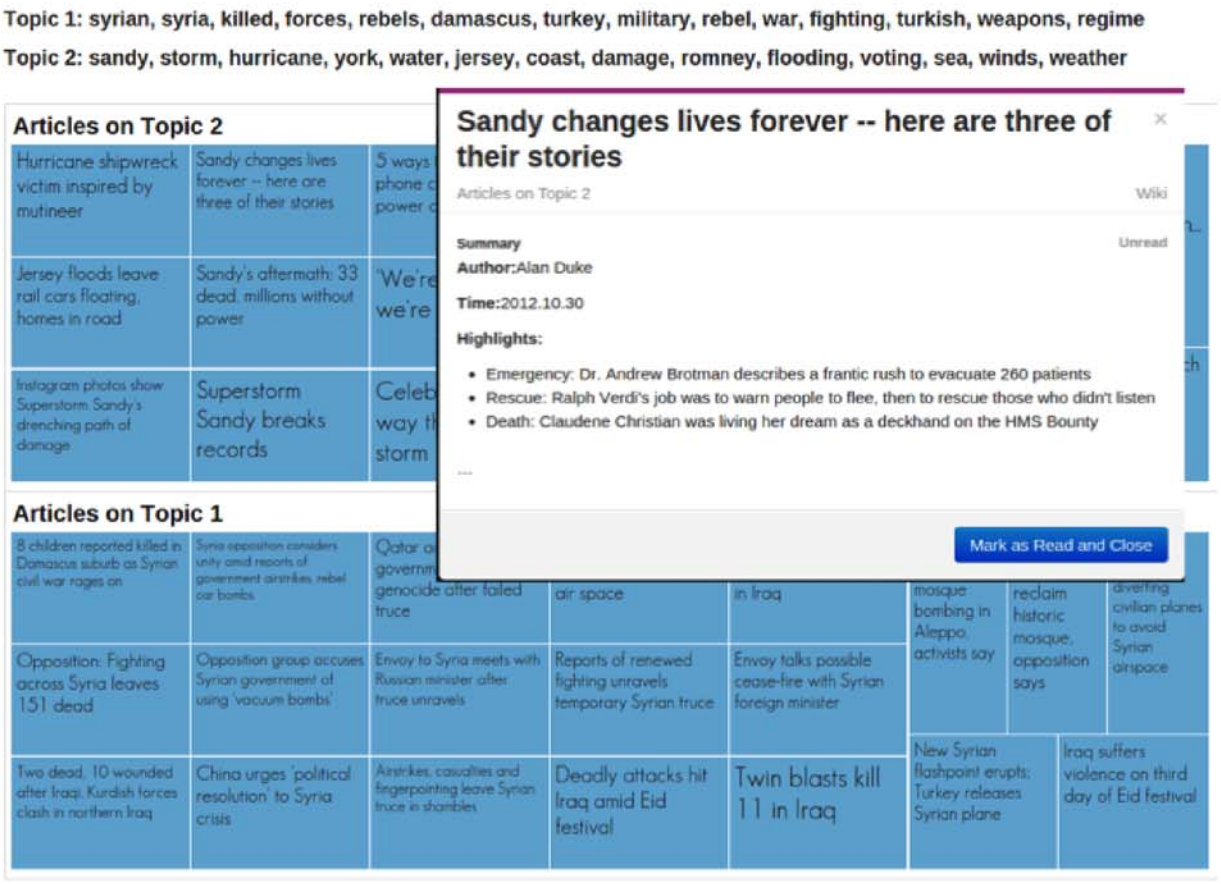


Figure 4.3. Topic model display interface.

After the feedback is collected, the *Feedback Processor* component converts the user's *Yes* or *No* feedback to topic stability constraints. With standard LDA, if an existing model is re-trained on the same dataset or refreshed with additional documents, it is possible that the model's parameters may change. As a result, the top keywords of the same topic and the topic label assignments of the same documents may change. Given this, there are two different ways to maintain topic model stability: (1) keeping the top keywords associated with each topic unchanged; (2) keeping the topic labels assigned to the same documents unchanged. We believe the first approach is too rigid since it does not allow a

topic to evolve. For example, early papers on Natural Language Processing (NLP) focused on linguistic methods. As a result, the top keywords of the old NLP topic may include "grammar, syntax, semantics, noun, verb." Nowadays, more and more NLP systems employ data-driven approaches. Statistical and machine learning based NLP methods have become mainstream. Thus, after updating the topic model with new NLP publications, we expect that the top NLP keywords may include "statistics, learning, unigram, frequency, corpus." Since the change of top topic keyword is the result of topic evolution, minimizing the change of top topic keywords does not seem to be a good solution. Thus, our strategy to maintain topic stability is to keep the topic assignments to the same documents stable.

To keep the document topic assignments stable, we convert a user's *Yes* or *No* feedback to a set of document must-link and cannot-link constraints. A **must-link** constraint between two documents indicates that they should belong to the same topic, and a **cannot-link** constraint indicates that the documents should belong to different topics. For a topic that the user responses with Yes, must-link constraints are added for each pair of the documents that have this topic label, and cannot-link constraints are added between documents with this topic label and documents with a different topic label. In contrast, for a topic that the user responses with No, neither must-links nor cannot-links are added. All the must-links and cannot-links are fed to the *Constrained LDA* component for constrained topic model update, which completes the loop.

4.3. Example Scenarios

In this section, we provide two example scenarios to illustrate the topic model stability problem during dynamic topic-centric exploration of online news articles. For both scenarios, we also show how the nTMU system we have developed aids the user, named Alice, by

interactively incorporating her feedback during topic model update to keep the topic model stable. We develop the scenarios on a corpus of about 320 CNN news articles from October 2012 to November 2013. This corpus covers five topics, including *Fiscal Cliff*, *Obamacare* and *Hurricane Sandy*, according to the category labels of the articles.

In the first scenario, news articles related to *Fiscal Cliff* and *Obamacare* are put under two different topics based on the initial topic model created. Alice explores the articles under both topics but focuses more on Obamacare which is more closely related to her personal interests. A few weeks later, as more news articles become available, the topic model needs to be updated. When standard LDA is used to update the model, the *Fiscal Cliff* and *Obamacare* articles are put under a single topic as they are both related to government issues, which means that the articles about two previously separate topics are now mixed together. After this update, Alice is unable to easily relocate the old Obamacare articles and has difficulty identifying the changes happening to Obamacare between several weeks ago and now. In contrast, using the nTMU system which conducts constrained topic model update, Alice can indicate that she wants to keep the *fiscal cliff* and *Obamacare* articles separate since she wants to follow the news on these two events separately. nTMU incorporates this feedback during topic model update, so the articles about *fiscal cliff* and *Obamacare* remain under separate topics after the update to satisfy Alice's preference.

In the second scenario, the initial model includes (among other topics) two topics, one about *Hurricane Sandy* and the other about *Fiscal Cliff*. Alice reads several articles related to Hurricane Sandy. A few weeks later, the topic model is updated to include a new set of news articles. In this new article set, some *Hurricane Sandy* articles discuss government funding situations to aid recovery after the hurricane, which happen to share many common keywords as the articles about *Fiscal Cliff*. As a result, some previous *Hurricane Sandy* articles are

put under the same topic as the articles related to *Fiscal Cliff* when standard LDA is used to create the updated model. This totally confuses Alice as she does not understand why those articles about Sandy are not with other Sandy articles but mixed with those about *Fiscal Cliff*. As a result, she questions the accuracy of the system and deems the system not useful. In comparison, the nTMU system will ensure that all *Hurricane Sandy* articles remain together and are not mixed with *Fiscal Cliff* articles after the update.

4.4. Constrained LDA (cLDA) and Evaluation

In the previous section, we have introduced nTMU, the Non-Disruptive Topic Model Update system, that is capable of incorporating user feedback into standard LDA, but we treat the Constrained LDA module as a black box. In this section, we will present the Constrained LDA algorithm in details, and we will also show that cLDA can be encoded into SC-LDA framework.

We focus on two types of document level constraints: must-links and cannot-links. A **must-link** constraint indicates that two documents should share the same topics (e.g., between two Sports articles), and a **cannot-link** constraint indicates that two documents should have different topics (e.g., between a Sports and a Politics article). For example, Table 4.2 shows the snippets of four documents. We can see that `doc1`, `doc2` and `doc4` are about Sports articles that the first one is about hockey and the rest two are about baseball. Also, `doc3` is a space topic article. Therefore, from our definition on must-link and cannot-link, users can add must-link between {`doc2`, `doc4`} and add cannot-link between {`doc1`, `doc3`}, {`doc2`, `doc3`} and {`doc3`, `doc4`}. However, different users may have different opinion on hockey topic and baseball topic articles. Some may say both are about Sport topic so there should be a must-link between them, while others who look for more fine-grained topics

Table 4.2. Four document snippets selected from 20 Newsgroup dataset.

doc1	...in Salt Lake City this past Sunday, the local ABC station decided not to televise the **hockey** games...
doc2	...Hello, my friends and I are running the Homewood Fantasy Baseball League (pure fantasy **baseball** teams)...
doc3	...According the IAU Circular #5744, Comet Shoemaker-Levy 1993e, may be temporarily in orbit around **Jupiter**...
doc4	...He has obtained the play by play records for every major league **baseball** game for the past several years...

may argue they are two different topics so they share cannot-link relation. Both arguments reflect users understanding of the documents. Therefore, we hope that constrained LDA can take constraints as soft preferences rather than hard constraints.

Specifically, we denote $\mathcal{M}_d \in \mathcal{D}$ as the set of documents sharing must-links with document d, and $\mathcal{C}_d \in \mathcal{D}$ as the set of documents sharing cannot-links with document d. The idea behind cLDA is that, we want to make documents share must-links close to each other, while documents share cannot-links far away from each other, in the T-dimensional space, by controlling documents' topic distribution prior.

4.4.1. The Role of Concentration Parameters

In LDA, the prior of θ is a Dirichlet distribution, which is denoted by $Dir(\vec{\alpha}_g) = Dir(\alpha_{g,1}, ..., \alpha_{g,T})$. $\vec{\alpha}_g$ is the global hyperparameter, and $\alpha_{g,t}$ determines how "concentrated" the probability mass of a sampled θ is likely to be on topic t. If all the $\alpha_{g,t}$ are less than one, the probability mass tends to be concentrated on a few topics. If all the $\alpha_{g,t}$ are greater than one, the probability mass tends to be more uniformly distributed. Meanwhile, smaller $\alpha_{g,t}$ attracts more concentration on topic t. A simple and commonly used Dirichlet distribution is the symmetric Dirichlet distribution, where $\alpha_{g,1} = \alpha_{g,2} = ... = \alpha_{g,T} = \alpha$.

If there is no information about relationships of the documents, the concentration parameters are normally set to be equal. Then the model can update the likelihood by learning the posterior of θ. When we have knowledge about a document's topic distribution, e.g., two documents have similar topic distribution, the topic distributions of documents cannot be assumed to be independently sampled. To achieve this, we manipulate the Dirichlet prior over document-topic distribution such that document must-link and cannot-link constraints can be incorporated into the topic model. In such a way, we use an asymmetric Dirichlet prior over document-topic while keep the Dirichlet prior over topic-word symmetric [**61**]. In the following, we explain how to incorporate the prior knowledge encoded in document must-links and cannot-links into an LDA topic model.

4.4.2. Must-link Constraint

A must-link between two documents d_1 and d_2 suggests that d_1 and d_2 should share the same topics, e.g., both are Sports news. Thus θ_1 should be similar to θ_2, and both distributions on the T-dimensional space should be close to each other.

Given the documents in $\mathcal{M}_d$, we introduce an auxiliary variable $\vec{\alpha}_d^{\mathcal{M}}$:

$$\vec{\alpha_d}^{\mathcal{M}} = K * \frac{1}{|\mathcal{M}_d|} \sum_{i \in \mathcal{M}_d} \theta_i, \tag{4.4}$$

where K controls the concentration parameters. The larger the value of K is, the closer θ_d is to the average of θ_i's. $\mathcal{M}_d \in \mathcal{D}$ is the set of documents sharing must-links with document d.

4.4.3. Cannot-link Constraint

A cannot-link between documents d_1 and d_2 suggests that d_1 and d_2 should not have the same topics, for example, one is about Sports and the other is about Politics. Thus, θ_1 should *not* be similar to θ_2, and both distributions on the T-dimensional space should be really far away from each other.

Given the documents in $\mathcal{C}_d$, we introduce the following auxiliary variable:

$$\vec{\alpha_d}^{\mathcal{C}} = K * \arg_{\theta_d} \max \min_{i \in \mathcal{C}_d} KL(\theta_d, \theta_i), \tag{4.5}$$

where $KL(\theta_d, \theta_i)$ is the KL-divergence between two distributions θ_d and θ_i. This means we choose a vector that is maximally far away from $\mathcal{C}_d$, in terms of KL divergence to its nearest neighbor in $\mathcal{C}_d$. $\mathcal{C}_d \in \mathcal{D}$ is the set of documents sharing cannot-links with document d.

Then in each iteration, we draw a θ_d from the following distribution:

$$\theta_d \sim Dir(\eta_g \vec{\alpha}_g + \eta_{\mathcal{M}} \vec{\alpha_d}^{\mathcal{M}} + \eta_{\mathcal{C}} \vec{\alpha_d}^{\mathcal{C}}) = Dir(\vec{\alpha}_d). \tag{4.6}$$

Here, η_g, $\eta_{\mathcal{M}}$ and $\eta_{\mathcal{C}}$ are the weights to control the trade-off among the three terms. Note that in the first iteration of learning, it is possible that all the θ_d's are initialized by drawing from the global α_g solely. In our experiment, we choose $K = 100$, $\eta_g = \eta_{\mathcal{M}} = \eta_{\mathcal{C}} = 1$.

4.4.4. Inference with Gibbs Sampling

Given a set of document must-links $\mathcal{M}$ and cannot-links $\mathcal{C}$, we infer the values of the hidden variables $\mathbf{z}$ using collapsed Gibbs sampling as in LDA [**19**]. In each iteration, topic assignments z of word w in document d is sampled based on all the other variables. We directly

present the conditional probability as following distribution:

$$\begin{aligned} P(z = t|\mathbf{z}_-, \mathbf{w}, \mathcal{M}_d, \mathcal{C}_d) &= \frac{P(\mathbf{z}, \mathbf{w}|\mathcal{M}_d, \mathcal{C}_d)}{P(\mathbf{z}_-, \mathbf{w}|\mathcal{M}_d, \mathcal{C}_d)} \\ &= \frac{(n_{d,t} + \alpha_{d,t})}{n_d + \sum_{t=1}^{T} \alpha_{d,t}} \frac{n_{w,t} + \beta}{n_t + V\beta} \propto (n_{d,t} + \alpha_{d,t}) \frac{n_{w,t} + \beta}{n_t + V\beta} \end{aligned} \tag{4.7}$$

It is worth comparing Equation 4.7 with Equation 2.2. In LDA, the hyperparameter $\vec{\alpha}$ is not updated during the training. Moreover, in most of the real world LDA system, for simplicity reason, $\vec{\alpha}$ is a uniformed vector with the same value for each vector component over all documents. However, in cLDA, $\vec{\alpha}$ is not a fixed value but is updated in every iteration based on the constraint sets, and different documents will have different hyperparameter which also reflects the relationship among documents.

The new iterative Gibbs sampling process is shown in Algorithm 4.

Algorithm 4: cLDA Gibbs Sampling for document d with constraint set $\mathcal{M}_d$, $\mathcal{C}_d$

$\vec{\alpha}_d = \eta_g \vec{\alpha}_g + \eta_{\mathcal{M}} \vec{\alpha_d}^{\mathcal{M}} + \eta_{\mathcal{C}} \vec{\alpha_d}^{\mathcal{C}}$;
for *each word token* w *in document* d **do**
  for *each topic index* $t \in \mathcal{T}$ **do**
    $p(z = t|\mathbf{z}_-, \mathbf{w})$=$(n_{d,t} + \alpha_{d,t}) \frac{n_{w,t}+\beta}{n_t+V\beta}$;
  end
  sample new topic assignment for w from $p(z)$;
end

The difference between the cLDA's Gibbs sampling and the standard LDA's Gibbs sampling lies in the first line of Algorithm 4. In standard LDA, document d's topic distribution prior $\vec{\alpha}_d$ is a vector with fixed values. However, in cLDA, $\vec{\alpha}_d$ is updated in every iteration based on d's constraints set $\mathcal{M}_d$, $\mathcal{C}_d$. If d's constraints set $\mathcal{M}_d$, $\mathcal{C}_d$ are empty sets, then the cLDA Gibbs sampling for d is reduced to LDA Gibbs sampling.

4.4.5. Encoded in SC-LDA Framework

We have shown an intuitive way to represent document must-link and cannot-link constraints/knowledge and develop an inference algorithm to incorporate the constraints into LDA topic models. In this section, we show that the way we incorporating constraints by changing the model priors can also be represented in SC-LDA framework.

We set the potential function of topic assignment z in document d as

$$f(z = t|d) = \ln \frac{n_{d,t} + \alpha_{d,t}}{n_{d,t} + \alpha} \tag{4.8}$$

Recall that the conditional probability of SC-LDA for word w in document d given knowledge m is

$$P(z = t|w, \mathbf{z}_-, M) \propto \left\{ (n_{d,t} + \alpha) \frac{n_{w,t} + \beta}{n_t + W\beta} \right\} \exp f_m(z, w, d) \tag{4.9}$$

Replacing $f_m(z, w, d)$ in Equation 4.9 with Equation 4.8, we get

$$P(z = t|w, \mathbf{z}_-, M) \propto (n_{d,t} + \alpha_{d,t}) \frac{n_{w,t} + \beta}{n_t + V\beta} \tag{4.10}$$

which is exactly the same as the cLDA inference method in Equation 4.7.

4.4.6. Put It Together: User-directed Topic Model Update

In the above paragraphs, we have unveiled the algorithm details of the Constrained LDA module which we treated as a black box in section 4.2. Now, we describe the details of the user-directed non-disruptive topic model update procedure in Algorithm 5. To be consistent with our experiments where we split the dataset in halves by chronological order, we display

one round of topic model update in the algorithm, i.e., the model is updated once. Obviously it can be easily extended to multiple rounds of updates.

Algorithm 5: User-directed Topic Model Update (nTMU)

Data: First half document collection D_1, and second half document collection D_2
train LDA model on D_1;
assign topic label to documents in D_1, and display LDA model to end user;
$\mathcal{M} = \emptyset$, $\mathcal{C} = \emptyset$;
for *each topic index* $j = \{1, ..., T\}$ **do**
  if *user keeps topic* j **then**
    randomly select two documents u, v with the same topic label j, and add u, v to $\mathcal{M}$;
  else
    do nothing
  end
end
randomly select two documents u, v with different topic labels, and add u, v to $\mathcal{C}$;
train cLDA model on $D_1 \bigcup D_2$ with constraints set $\mathcal{M}$ and $\mathcal{C}$;
assign topic label to documents in $D_1 \bigcup D_2$, and display the updated model to end user;

Existing topic model update systems try to fit topic models to the new data without taking model stability into consideration. The main differences between nTMU and other topic model update systems are two folds. Firstly, nTMU allows users to select the topics they would like to keep. If a topic is incoherent (e.g., consists of incoherent keywords and inconsistent documents), it is unlikely that a user would like to keep it stable. Secondly, user's feedback is translated into document must-link and cannot-link constraints that will be incorporated into the topic model update procedure. In summary, the nTMU algorithm is designed to optimize the update procedure based on two goals simultaneously: keeping topics stable given user's choices, as well as fitting the model to the new data.

4.4.7. cLDA Evaluation

We conduct several experiments to evaluate the effectiveness of cLDA in incorporating document must-link and cannot-link constraints. Here we use the 20 Newsgroups dataset[5]. The corpus is partitioned evenly into 20 different newsgroups, each corresponding to a topic. After preprocessing, we follow [**6**] to create two sub-datasets: `Sim3` which consists of three newsgroups on similar topics (comp.graphics, comp.os.ms-windows, comp.windows.x); `Diff3` which consists of three newsgroups on different topics (alt.atheism, rec.sport.baseball, sci.space). In addition, we also create a new `Mix3` which consists of two similar topics (rec.sport.hockey, rec.sport.baseball) and one distinctive topic (sci.space). We expect that `Sim3` is the most difficult dataset for topic modeling due to overlapping vocabularies while `Diff3` is the easiest. The characteristics of the three sub-dataset are shown in Table 4.3.

Table 4.3. Statistics of three sub-datasets.

	`Sim3`	`Diff3`	`Mix3`
D_{train}	1,768	1,670	1,790
# Vocabulary	4,222	4,822	4,685
D_{test}	1,178	1,110	1,190

We simulate user inputs using the documents' ground truth labels. If two documents have the same label, we add a must-link between them. Similarly, We also add a cannot-link between two documents with different labels.All these constraints are added into a constraint pool. We also augment the constraint pool with derived constraints. For example, due to transitivity, if there is a must-link between (a, b) and (b, c), then we add a must link between (a, c). We simulate the process to acquire 1,000 pairwise constraints before we train cLDA.

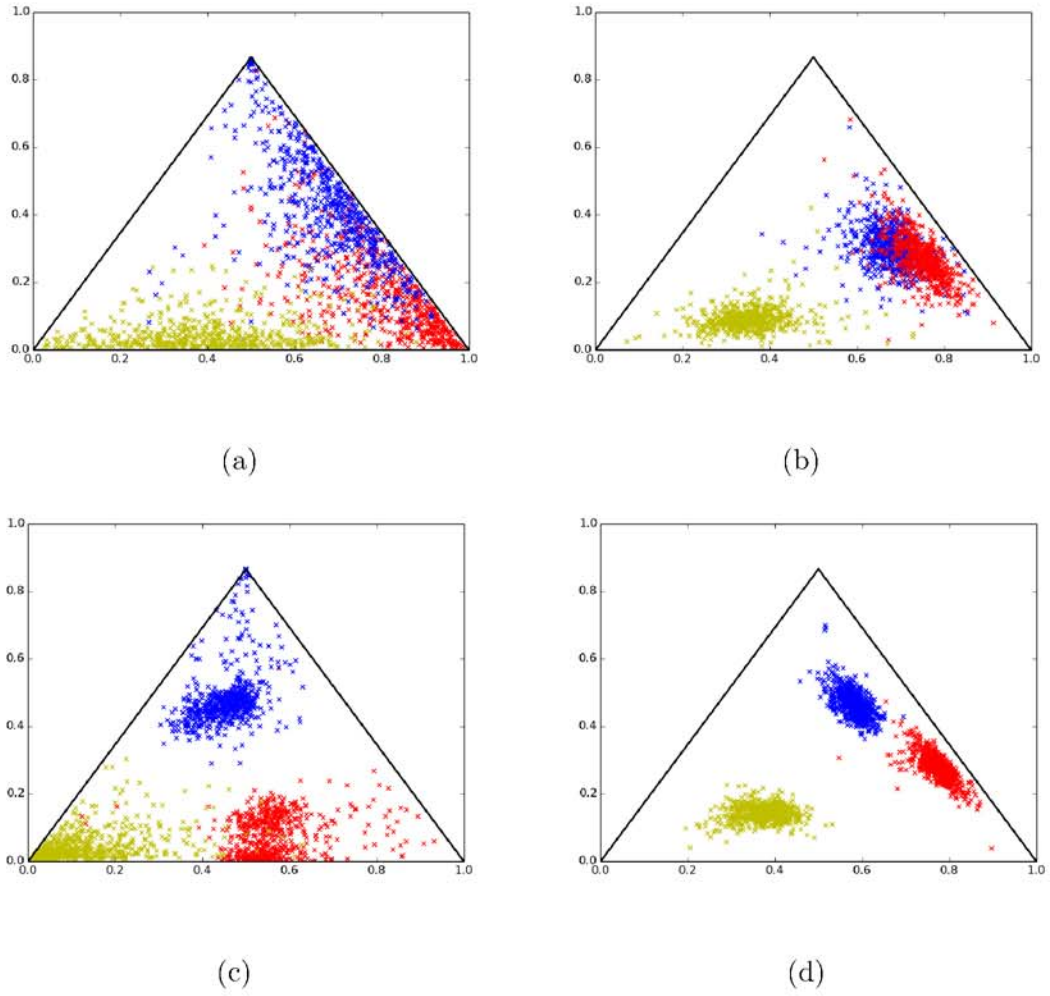
In the experiment, we evaluate the effects of constraints on the main LDA parameters: document-topic distribution θ and topic-word distribution ϕ. If cLDA is effective, we would

[5]Available at `http://people.csail.mit.edu/jrennie/20Newsgroups(20news-bydate.tar.gz)`

expect that on the Dirichlet simplex, the inferred θs associated with the documents connected by must-links are clustered together while the θs associated with the documents connected by cannot-links are separated apart. In addition, if cLDA is effective, we also expect to see more coherent topic keywords derived based on ϕ than those derived by standard LDA.

Based on the topic label of each document, we randomly generate 10,000 must-links between documents in the same newsgroup and cannot-links between those in different newsgroups. Figure 4.4 shows the inferred θs of `Mix3` projected on a simplex. Figure 4.4(a) shows that although standard LDA did capture some thematic patterns among the documents, it could not clearly separate these three newsgroups, especially the articles in Hockey and Baseball. After we apply the must-links using cLDA however, the documents within each newsgroup become more tightly grouped in 4.4(b). Similarly, after we apply the cannot-links, documents from different newsgroups become more separated in 4.4(c). Finally, when we apply both the must-links and cannot-links in 4.4(d), our system discover three tightly grouped document clusters with well-defined boundaries between them. This result confirm that cLDA can indeed incorporate pairwise document constraints effectively when inferring θ.

To verify that with additional document constraints cLDA can also infer ϕ more accurately, we present the top 10 words representing each of the three topics in `Mix3` using standard LDA and cLDA. For the keywords derived by both LDA and cLDA, we perform a post-processing step to re-rank them using [**57**]. As shown in Table 4.4, standard LDA has more trouble learning the keywords related to the *Baseball* topic. cLDA, however, successfully separated all three topics, and the keywords representing each topic are quite coherent and informative.



(a) (b)

(c) (d)

Figure 4.4. The θ of `Mix3` projected on a simplex. a)standard LDA with no constraints; b)cLDA with must-links; c)cLDA with cannot-links; d)cLDA with both must-links and cannot-links. Here, hockey documents are in red, baseball in blue and space in yellow.

4.5. nTMU System Evaluation and User Study

In this section, we evaluate our nTMU system's performance from two aspects, topic model quality and topic model stability. From a user's standpoint of view, a good topic model update system should not only be able to fit the data but can also maintain a stable topic model transition.

Table 4.4. Top 10 keywords of each topic by LDA (above) and cLDA (below).

Topic	Words
baseball	writes think good know better even baseball really going anyone
space	space nasa launch orbit satellite moon earth data mission shuttle
hockey	game hockey playoff division pittsburgh run detroit canada boston pick
baseball	baseball run base thanks brave pitching cub ball hitter yankee
space	space nasa system launch orbit satellite moon science center earth
hockey	play hockey goal playoff period pittsburgh leaf wing detroit ranger

For evaluation, we use the NIPS dataset[6], a benchmark dataset commonly use in text analysis. It contains 1,740 papers from the NIPS conferences between 1987 and 1999. Documents in the dataset are sorted based on their timestamps. We use the first 844 documents (from 1987 to 1993) as D_1, next 896 documents (from 1994 to 1999) as D_2. Note that the documents in this dataset do not have ground-truth topic labels. After removing stopwords and stemming, the dataset has 6,448 unique words. For topic modeling parameters, we choose 20 as the number of topics since it is commonly reported in previous literatures. We also set the global hyperparameter $\alpha_g = 1$, and $\beta = 0.01$. To make sure the Gibbs sampling chain converges, we average over 10 runs for each experiment, and for each run, 500 iterations. The first two experiments are conducted without having user in the loop. Thus, the feedback processor component automatically generates document constraints. Must-links are added for documents with the same topic label, while cannot-links are added between documents with different topic labels. In the experiments, for each document, 10 must-links and 10 cannot-links are randomly sampled.

[6] `http://psiexp.ss.uci.edu/research/programsdata/toolbox.htm`

4.5.1. Experiment 1: Topic Stability

In the first experiment, we test nTMU's capability in maintaining topic stability. We compare nTMU against three baseline systems:

(1) Standard LDA: it jointly resamples all topics for the entire collection, including both the old and the new documents, and old documents' topic samples are not reused. Therefore, it re-trains the whole topic model on the entire collections from scratch.

(2) Fixed Fold-in: it jointly resamples topics for all the new documents, but it keeps old documents' topic samples fixed. Therefore, old documents' topic samples are used to initialize the topic model, but they are not updated. Note that although this method keeps all old documents' topic sample fixed, it does not necessarily mean that the topic labels for the documents will be the same before and after the update. There are two reasons that the topic label would be changed. First, document's topic distribution $\hat{\theta}$ is a weighted distribution (Equation 3), and its value is normalized over all the documents, including the new documents. Second, since topic labels are aligned by Hungarian algorithm which looks for a "maximum matching" on all documents, it is likely that the old documents' label indexes are swapped, which changes the topic labels for old documents.

(3) Rejuvenated Fold-in: in addition to jointly resampling topics for all new documents, it also updates old documents' topic samples. Therefore, after being used to initialize topic model, old documents' topic samples will be updated.

Both (2) and (3) are inspired by the ideas behind sampling-based inference strategies for new documents discussed in [**65**]. In fact, the difference among the three baseline methods is how to initialize a new topic model given an old topic model. In standard LDA, old

documents' topic samples have no impact on the new topic model, while in Fixed Fold-in and Rejuvenated Fold-in, old documents' topic samples will have different impacts.

Table 4.5 shows the *document-level stability* of three baseline systems and nTMU since document-level stability is what most users care about. Among all the topic update methods, nTMU significantly outperforms the other methods in maintaining high topic stability. In particular, it improves the standard LDA's topic model stability by 103%, and it also improves the second best Fixed fold-in method by 46.3%. Note that the results are averaged over 10 runs for each topic model update method.

Table 4.5. Topic model stability performance of different model update methods.

Model update method	stability
Standard LDA	43.3%
Fixed fold-in	60.2%
Rejuvenated fold-in	57.8%
nTMU	**88.1%**

4.5.2. Experiment 2: Topic Coherence

In addition to topic stability metric, we also evaluate our nTMU using topic coherence metric. Recent research [**10**] has shown that topic coherence is highly consistent with human judgement than held-out data likelihood. Thus, here we use topic coherence to assess a topic model's quality. Following [**39**], topic t's coherence is defined as $C(t : V^{(t)}) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{F(v_m^{(t)}, v_l^{(t)})+1}{F(v_l^{(t)})}$, where $F(v)$ is the document frequency of word type v, $F(v, v')$ is the co-document frequency of word type v and v', and $V^{(t)} = (v_1^{(t)}, ..., v_M^{(t)})$ is a list of the M most probable words in topic t. In our experiments, we choose the 20 most probable words to compute topic coherence, i.e., $M = 20$. In addition, since LDA usually generates common background topics which appear in many documents and thus un-interesting, we also filtered

those topics based on the method proposed in [**57**] before we compute the coherence scores for all the methods.

As shown in Figure 4.5, nTMU and Standard LDA achieve similar topic coherence score, and both methods perform better than Fixed Fold-in and Rejuvenated Fold-in methods with statistical significance at $p = 0.05$, using Chi-Square test. Although nTMU is slight worse than standard LDA, but there is no statistical significance. Since this is a simulation experiment, and the constraints are added automatically for every topic, it is inevitable that 1). mistakes are introduced to the system for inconsistent or uncorrelated topics; 2). the model is over-constrained by the constraints. For example, Fixed Fold-in has the worst topic coherence performance because it does not allow topic samples in old documents to update. Therefore, this system is over-constrained and has a poor topic model quality. However, we can see from the figure that the nTMU system can still achieve topic coherence nearly as good as standard LDA update system, which does not have any constraints applied and have the most freedom to fit new data.

To sum up, unlike Fixed Fold-in and Rejuvenated Fold-in who achieve higher topic stability than standard LDA at the cost of lower topic coherence, nTMU achieves the highest topic stability score among all the methods without sacrificing any topic coherence.

4.5.3. Experiment 3: Incorporating User Guidance

In the above two experiments, nTMU system simulates user interaction by adding constraints for each topic automatically. In this experiment, instead of the simulations, we demonstrate the benefit of keeping users in the loop and allowing them to guide topic model update.

Since nTMU employs a constrained topic modeling framework, it needs to satisfy two goals simultaneously: (1) to satisfy as many constraints as possible (2) to derive a topic

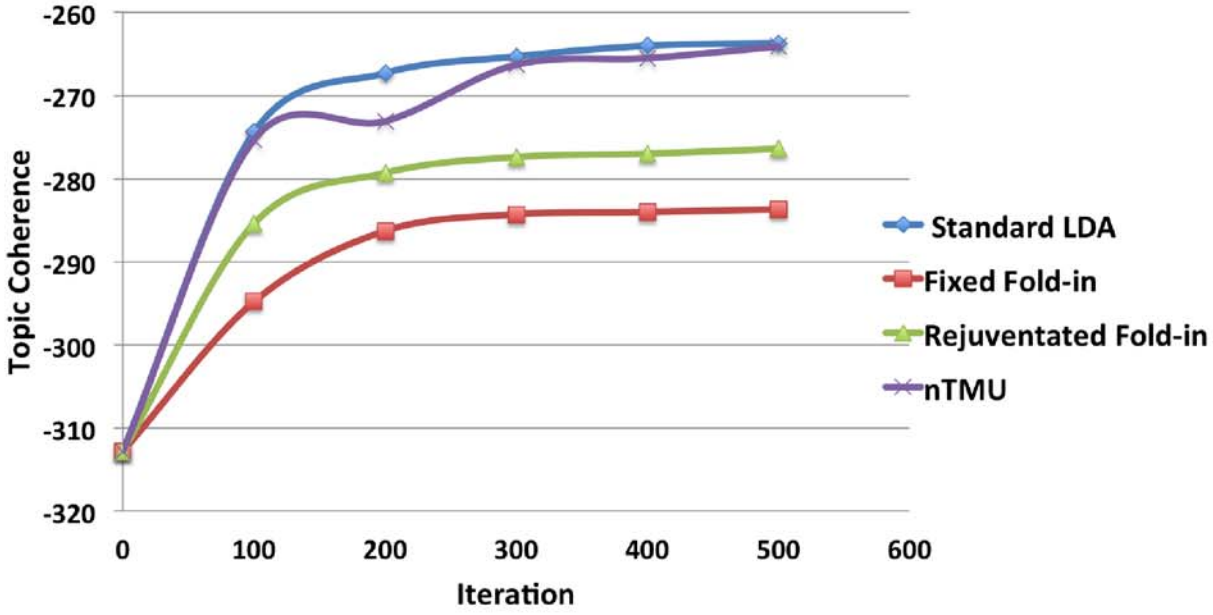


Figure 4.5. Topic Coherence of different topic model update methods.

model that fits the data. In general, if goal (1) matches goal (2), the constraints will help the system to meet goal (2) (e.g., the ground-truth labels). But if goal (1) is competing with goal (2) (e.g., minimizing end user disruption in nTMU), goal (1) may distract the system from meeting goal (2). Here is an example on how goal (1) in nTMU might distract the system from achieving goal (2): With an old topic model, a paper on language modeling was categorized as an SP paper since the method was first adopted in the speech recognition community. Over the years, language modeling has become very popular in the NLP community for tasks such as machine translation and POS tagging. As a result, now, it is more appropriate to categorize this as an NLP paper. But due to the topic stability constraints, our system will insist on categorizing this as an SP paper.

Giving the competing nature of the two goals in nTMU, the more constraints we add to the system, the less freedom it has to fit the topic model to the data. Without user

feedback, by default, nTMU will keep all the topics stable. This may over-constrain the system. For example, since not all the topics are equally interesting, there is no need to ensure the stability of topics that a user does not care about. Similarly, not all the topics are equally good. Some of the inferred topics may even be incoherent. Enforcing the stability of incoherent topics can hurt the performance.

To validate the above concerns and to demonstrate the benefit of keeping users in the loop, we designe the following experiment. First, we allow a user to choose a subset of the topics (e.g., 3) that are important to him. Frequently, user selects topics are coherent topics. In the experiment, this user is one of the authors who is familiar with the topics in this dataset. We also compare this model with two alternatives: (1) a model which kept three incoherent topics stable (2) a model which kept all the topics stable. With this setting, we want to verify that (1) keeping incoherent topic stable will hurt the system's capability to adapt to the data (2) keeping all the topics stable will over-constrain the system and prevent it from fitting the data. Here, since we want to measure how well the learned model fits the test dataset, rather than the consistence with human judgement, we use *perplexity*, not topic coherence, as the metric to assess the topic model's quality. As shown in Figure 4.6, nTMU that uses coherent topic stability constraints performed very well. At the end of the 2000 iterations, there is no difference between the perplexity of this model and the perplexity of a model that has no constraint. In contrast, the performance of the model that enforces the stability of incoherent topics deteriorated noticeably with statistical significance. Finally, the default model that enforces the stability constraints on all the topics had the worst performance. This result has clearly demonstrated the benefit of incorporating user guidance in managing topic model update.

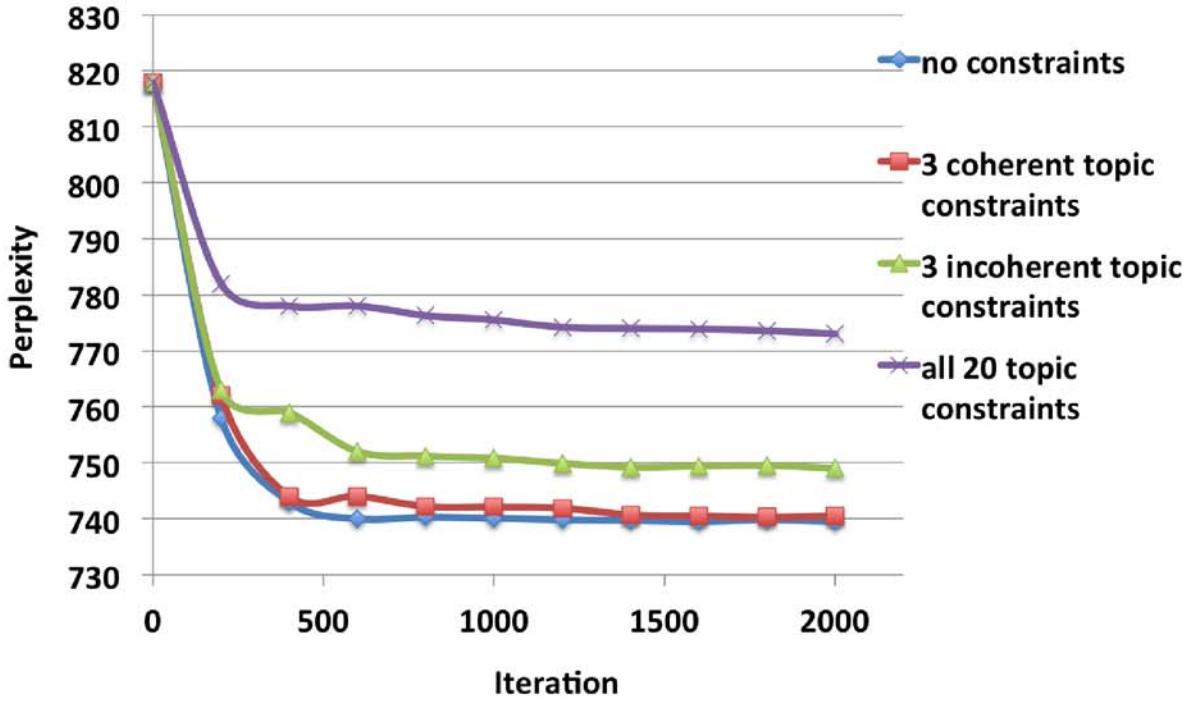


Figure 4.6. Model perplexity performance when different types of constraints are added. It shows the benefit of human guidance by providing coherent topic constraints.

4.5.4. Setting Model Parameters

In our current system, we need to set multiple model parameters, such as the number of topics, the value of concentration parameters, and the weights of must-link and cannot-link constraints. Although we can use heuristics to estimate the parameters for a dataset based on its characteristics (e.g., size of the data set), in reality, the parameters will still need to be tuned empirically for each new dataset. Optimizing such parameters for each dataset automatically is an active research topic in Machine Learning and beyond the scope of this study.

4.5.5. User Study Experiment Design

We also conduct a user study to evaluate the impact of topic stability in an automated document exploration system. We use Amazon Mechanical Turk (MTurk) as our study platform. In order to design tasks that are not too difficult for the workers (called Turkers), we decide to use a general news dataset instead of a scientific publication dataset. Our news dataset includes CNN news articles from October, 2012 to November, 2013, covering five prominent topics at the time including *"Fiscal Cliff", "Hurricane Sandy", "Violence in Iraq", "Obamacare", and "Syrian Civil War"*. Since it is impossible for a Turker to explore a very large dataset during the time given at the study, we limit the dataset to include 320 articles.

We sort the articles based on their timestamps and then split the dataset into two halves. The first half is used to train an initial topic model using LDA. Then we add the second half when updating the topic model. We employ a between-subject design, testing two different update algorithms, one used standard LDA, the other used nTMU. Due to the similarities of some of the topics in the dataset (e.g., both "Obamacare" and "Fiscal Cliff" are about US politics), the topic model is not very stable during update. For example, before update, "Fiscal Cliff" and "Obamacare" are two different topics. After the update using LDA, they merge into one general topic of "US Government Affairs". With nTMU, the system still maintains two separate topics in the updated topic model.

We design two Human Intelligence Tasks (HITs), one for each test condition. Each HIT is divided into two parts: before-update tasks and after-update tasks. Before-update tasks is designed to help a Turker to build a mental model of the system. The before-update tasks are exactly the same in two test conditions. For example, before an update, a Turker is

asked to choose the best label for a given topic by inspecting the top topic keywords and the top articles on that topic, both inferred by the system. S/he is also asked to select a few articles s/he would like to read and write down a few details from each article such as its title. The after-update tasks are designed to test how topic stability affects (a) a Turker's comprehension of the topics in the updated model (2) a Turker's ability to recall and relocate the articles s/he has chosen before the update.

We use both subjective and objective metrics to evaluate the effectiveness of these methods. Our objective evaluation metrics include

(1) $KTLA\_BU$ (Keyword-based topic label accuracy before update). This measures whether a Turker is able to pick a correct topic label based on the top topic keywords before model update.

(2) $ATLA\_BU$(Article-based topic label accuracy before update). This measures whether a Turker is able to pick a correct topic label based on the top articles in a topic before model update. The top articles were automatically inferred by the system.

(3) $BTLA\_BU$ (Topic label accuracy based on both topic keywords and top articles before model update). This measures whether a Turker is able to pick a correct topic label based on both the topic keywords and the top articles in a topic before model update.

(4) $KTLA\_AU$(Keyword-based topic label accuracy after model update). This measures whether a Turker is able to pick a correct topic label based on the top topic keywords after model update.

(5) $ATLA\_AU$(Article-based topic label accuracy after model update). This measures whether a Turker is able to pick a correct topic label based on the top articles in a topic after model update.

(6) $BTLA\_AU$(Topic label accuracy based on both topic keywords and top articles after model update). This measures whether a Turker is able to pick a correct topic label based on both the topic keywords and the top articles in a topic after the model update.

(7) $ARSR\_AU$(Article relocating success rate after update) This measures whether a Turker is able to relocate the articles s/he found before an update.

In addition, we include four subjective evaluation metrics :

(1) TD (Topic difficulty) This measures how difficult it is to understand the system-derived topics.

(2) ARD(Article relocating difficulty) This measures how difficult it is to relocate the articles a user found before.

(3) UA(Use Again) This measures the likelihood of a user to use the system if it is available.

(4) RF(Recommend Friends) This measures the likelihood of a user to recommend the system to others.

All the survey questions are rated on a 5-point Likert scale with 1 being the least desirable and 5 the most desirable.

4.5.6. Experiment Results

Overall, we have collected data from 80 Turkers, 40 for each test condition. Figure 4.7 shows the before-update objective metrics. Since the before-update tasks are exactly the same in the two test conditions, we do not expect much differences. Moreover, any significant

difference in these scores may indicate an irregularity of the test conditions and thus require further investigation.

As shown in Figure 4.7, since there isn't any statistically significant differences between the test conditions, we are assured of the validity of the data . In contrast, since the after-update metrics are design to capture the impact of topic stability on a user's ability to comprehend topics and recall articles, if maintaining topic stability is important and our system is effective, we expect that our method will result in better scores than those using the LDA method. Our study results have also confirmed this. With our method, the topic keywords are more coherent, which makes it easier for a user to understand a topic and select a correct topic label for it (0.688 with nTMU v.s. 0.338 with LDA). Since the top articles selected by the system are also more topic-relevant, which makes it easier for a user to pick the correct topic label based on top articles (0.875 with nTMU versus 0.388 with LDA). Finally, keeping the topic model stable can also significantly improve a user's chance to relocate an article (0.775 with nTMU v.s. 0.525 with LDA). As shown in Figure 4.8, our system outperforms the baseline in all the after-update evaluation dimensions. The differences are all statistically significant with $p < 0.001$ based on independent sample t-test. Therefore, it again demonstrates that our nTMU system can provide users a non-disruptive topic model update process, compared to the real world system that based on standard LDA.

In addition, as shown in Figure 4.9, our system also outperformes the baseline in three out of the four subjective evaluation dimensions. For example, with our system, the users think that they can understand the topics better. It is also easier for them to relocate an article they found before. They are also more likely to recommend the system to their friends.

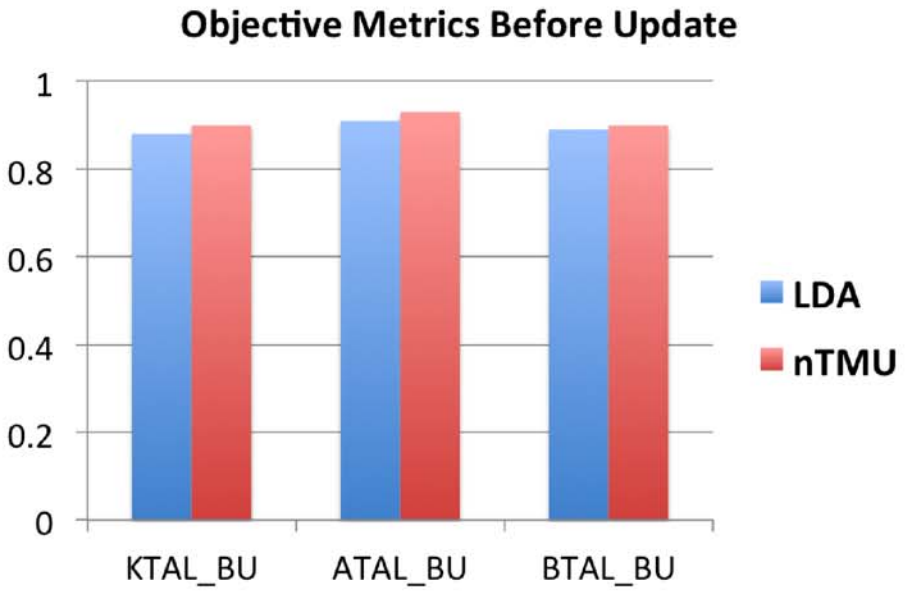


Figure 4.7. Before-update Objective Evaluation Metrics. It shows that users' cognitive understanding of models has no significant differences with LDA and nTMU, before the model is updated.

4.6. Related Work

Our work is closely related to three areas of research: user interface design, topic model update and topic model with prior knowledge.

4.6.1. User Interface (UI) Design

Usability is critical to the development of a user-friendly interface because it helps users to work in an effective, efficient, and manageable way [**13**]. Learnability, which is the ease with which a software application or product can be picked up and understood by users, is a key attribute of usability [**43**]. Because consistency and predictability are considered design factors that contributed to increased learnability [**15**], they have become two of the most important principles in UI design [**44**].

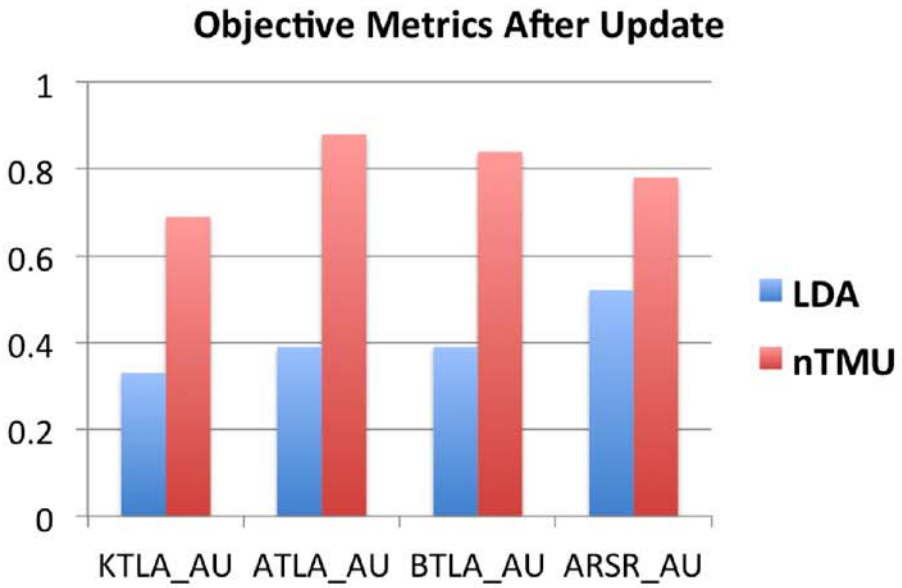


Figure 4.8. After-update Objective Evaluation Metrics. It shows that after the model is updated, users who use nTMU system have better and more consistent cognitive understanding of model than the users who use LDA update system. In other words, nTMU provides users a non-disruptive topic model update environment.

Despite being well-established principles in UI design, consistency and predictability have received little attention in the machine learning and data mining communities. New machine learning/data mining algorithms are rarely invented to specifically address these factors. However, for any interactive machine learning/data mining system, without proper backend support, it is impossible to achieve consistency and predictability at the UI level. For example, if the backend algorithm produces inconsistent results at different time intervals due to updates, the UI designed to surface these results will inevitably suffer from low consistency and predictability. More attention is needed during the development of backend algorithms to address these basic usability issues. Our work is the first in this direction.

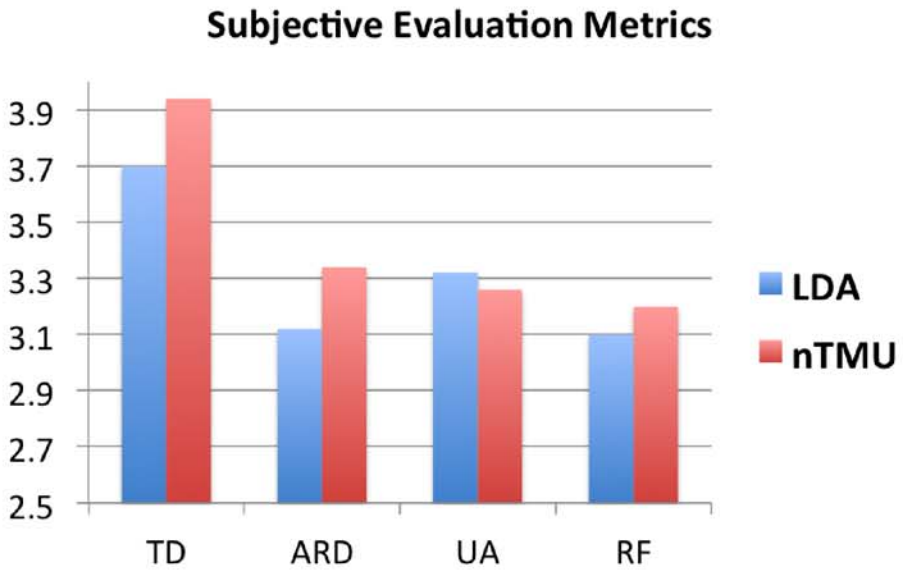


Figure 4.9. Subjective Evaluation Metrics.

4.6.2. Topic Model Update

Online LDA has been the main solution to topic model update. It incrementally updates the topic model when new documents arrive. Among existing online LDA algorithms, some use Variational Bayes (VB) [**23, 68**] to infer latent variables, others employ sampling methods [**56, 5, 9**]. Although online LDA sometimes can perform as well as batch LDA, it suffers from the same *topic model stability* problem as batch LDA since it mainly focuses on trying to approximate batch LDA's performance without requiring a full scan of the entire dataset. In the short term, since online LDA makes small incremental updates to the model when new data become available, this can keep the model relatively stable. But over a long period of time, the topic model may still experience significant changes. Since topic model stability is not an optimization goal for online LDA, for the same document, its topic assignment a few months ago may still be different from that based on the current topic model. In this paper, we propose a new LDA-based algorithm that employs a novel constrained topic

modeling approach to topic model update. With this approach, the algorithm can maintain topic model stability and minimize end user disruption by directly incorporating stability constraints in topic modeling.

4.6.3. Topic Model with Prior Knowledge

Recently, several algorithms have been developed to extend the standard LDA algorithm to incorporate user feedback or domain knowledge. Among them, some support seed constraints (e.g., document topic labels) [**7, 48**] while others support pair-wise word constraints (e.g., word must-links and cannot-links) [**3, 4, 26**]. [**36**] presents a topic modeling framework with network regularization, but it cannot easily handle cannot-links. To ensure topic model stability and at the same time to avoid over-constraining the new topic model, it is desirable to encode the stability constraints as pair-wise document constraints (e.g., document must-link and cannot-link constraints). However, none of these algorithms supports general pair-wise document constraints without further extension.

There have also been work focusing on incorporating specific link relations (e.g., citations between documents) to facilitate social network-based topic analysis [**58, 40, 33**]. Since all of them employ generative models, they cannot directly formulate cannot-link constraints between documents. Our constrained LDA algorithm is more general and can handle both types of document constraints (must-links and cannot-links).

4.7. Discussion

4.7.1. Generalization

In this paper, we have focused on addressing the stability and usability issues of LDA topic models that employ Gibbs Sampling. We focus on Gibbs Sampling because it is widely

adopted in practical applications. However, the instability problem is not specific to just Gibbs Sampling or topic modeling. Variational Inference is another example that suffers from the same problem. It is proposed in the original LDA paper [**8**] as an approximate inference method. Because LDA is a non-convex model, there are many local optima. As a result, different runs of the Variational Inference Algorithm may still converge to different local optima, which will produce different inference results. This has been verified in our experiments as well. We run LDA with Variational Inference on the 20 Newsgroup and the NIPS datasets. Even when the input data is kept the same, the document-level stability for Variational Inference on 20 Newsgroup is 57.5% and 62.3% on NIPS. This means that between two random runs of LDA with Variational Inference, close to 40% of the documents are assigned to different topics.

Furthermore, since statistical inference methods, such as Gibbs Sampling or Variational Inference, are not specific to topic modeling and are widely used in the field of machine learning and data mining to solve intractable Bayesian learning problems, the same instability issues may emerge in other probabilistic graph models as well.

4.7.2. Scalability

For today's constantly expanding document collections, a topic modeling system that supports topic-centric navigation and exploration needs to be quickly updated when new documents arrive in an online fashion. Therefore, scalability is a critical issue for any topic modeling update system. Our nTMU system relies on a constrained LDA algorithm cLDA, which is carried out using Gibbs sampling. Conventional Gibbs Sampling for LDA scales linearly with the number of topics. Moreover, accurate training usually takes many sampling passes over the dataset. Therefore, for large datasets with millions or even billions of

tokens, conventional Gibbs sampling takes too long to finish. Recently, fast Gibbs sampling methods [**65, 30, 66**] have made it possible to deploy LDA in industrial applications like search engines and online advertising, where the topic model is trained on large document collections with many topics. For example, while typical LDA models have up to 10^3 topics, in industrial applications, a model with $10^5 \sim 10^6$ topics is not uncommon [**62**]. Therefore, to achieve scalability and improve the efficiency of cLDA, we can also take advantage of existing fast Gibbs sampling methods.

4.7.3. Personalization

In addition to non-disruptive topic model update, the proposed constrained topic modeling, cLDA, can also be used for personalized topic modeling. The personalized topic model would be very useful in practice by providing individual users the freedom to explore the dataset differently to match their own interests and preferences. We believe that personalization, along with user interaction, is an important future research direction. To illustrate how cLDA enables personalized topic modeling, let us assume that two users want to explore a newsgroup dataset, such as New York Times news article dataset. The first user who is a sports fan desires fine-grained topics for sports articles. As a result, this user is likely to add a cannot-link between articles about Baseball and those about Hockey. In contrast, the second user who has little interest in sports just wants a coarse-grained topic for sports articles. Therefore, this user may add a must-link between Baseball and Hockey articles. Although these two users are training the same dataset with cLDA, they will get different topic summarizations based on their different feedback, which allow them to explore the dataset in a personalized way.

4.8. Conclusion

Topic model stability, if neglected, may significantly impact a topic model's usability, especially during model update. Existing topic model update methods (e.g., online LDA) have long neglected such an issue. In this chapter, we present an approach to directly addresses topic model stability to enhance its usability. Included in this approach are 1) a novel constrained LDA algorithm cLDA which enables LDA to incorporate general pair-wise document constraints that none of the existing methods are capable of handling effectively, and 2) a new user-directed non-disruptive topic model update system nTMU which collects user feedback, converts them to pair-wise document constraints and employs cLDA to achieve a smooth topic model transition. Evaluation results on both simulation experiments and user studies indicate that our approach significantly outperforms baseline systems in achieving high topic model stability while still maintaining high topic model quality. We hope our work will help topic modeling practitioners who have experienced the instability problem in their practice. Moreover, we also want to interest and inspire machine learning and data mining researchers to pay more attention to the development of human-centric data analytics algorithms to improve their usability.

CHAPTER 5

Towards Active Learning with Topic Modeling

Interactively soliciting users feedback during topic modeling also provides an efficient way to have users in the loop. [**26**] proposed *interactive LDA*, an interactive topic modeling framework that allows users to add word pairwise annotations. For example, the user could assert that "Mars" and "Venus" should be in the same topic. However, since the vocabulary size of a large document collection can be large, users may need to annotate a large number of word annotations for this method to be effective. In addition, polysemes present significant challenges for interactive LDA. For example, the word "pound" can refer to either a currency or a unit of mass. If a user adds a must-link between "pound" and another financial term, then he/she cannot add a must-link between "pound" and any measurement terms. Since word must-links are added without context, there is no way to disambiguate them. As a result, word annotations are frequently not as effective as document annotations.

Active learning [**54**] is a general framework which allows users to iteratively provide annotations to a learning model to improve its quality. In general, with the same amount of human labeling, active learning often results in a better model than that learned by a passive learning method. Therefore, instead of obtaining a batch of user labels before topic modeling, soliciting and incorporating the knowledge during the training can also benefit the model.

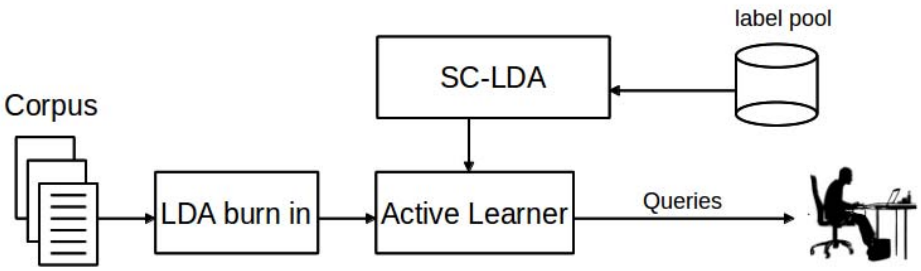


Figure 5.1. Diagram illustrating the topic model active learning framework.

In this chapter, we conduct a pilot study on active learning with topic modeling. We develop an active learning algorithm that interactively and iteratively solicits user annotations to improve the quality of a topic model. We propose different active learning strategies for seed and pairwise document labels. We also evaluate the results on several benchmark datasets using both simulated user inputs and real user interactions. The results demonstrate that the proposed active learning method with SC-LDA outperforms both classical LDA and passive LDA. In Chapter 3, we introduce a general framework SC-LDA to incorporate knowledge into LDA. The knowledge is encoded as constraints that shape the topics learned by LDA. We also present the integration of document label knowledge and document pairwise knowledge, which is treated as prior knowledge and is incorporated into LDA before the training.

As shown in Figure 5.1, given a document collection, the framework first runs the classic LDA with a burn-in component. Based on the burn-in result, the system dynamically generates *queries* for user to label. In general, different types of queries are encoded as different types of user labels which will be added to a label pool. Then we update the topic model using SC-LDA with the labels from the pool. In the next iteration, the system will choose

another query for users to annotate based on an assessment of the current topic model. This process continues until the user is satisfied with the resulting topic model. In the following, we explain how we solicit user inputs using dynamically generated queries.

5.1. Active Learning Query Generation

5.1.1. Seed Label Query Generation

A seed query solicits a topic label for a dynamically selected document sample. A user is given the definitions of all the topics in the current topic model as well as a document sample. Then she chooses a topic label for the given document based on her assessment of the relationship between the current document and the topics in the current model. We consider two active learning strategies for choosing d.

- SEED-MAXENTROPY: The entropy of a document d is computed as

$$H_d = -\sum_{t=1}^{T} \theta_{dt} \log \theta_{dt} \tag{5.1}$$

  Using this strategy, the system will select a document with the highest entropy since this is the one the model is the most confused about. By definition, a document with uniformed θ will be chosen first since currently the model has no information about the topic distribution of the document.

- SEED-MINLIKELIHOOD: The likelihood of a document d is computed as

$$L_d = (\sum_{i=1}^{N} \sum_{t=1}^{T} \phi_{ti}\theta_{dt})/N \tag{5.2}$$

  where N is the number of tokens in d, and ϕ is model's topic-word distribution. Using this strategy, the system will select a document with the minimum likelihood.

Since the overall likelihood of input documents is the objective function LDA aims to maximize, using this criteria, the system will select a document which is most difficult to predict based on the current model.

5.1.2. Pairwise Label Query Generation

A pairwise label query consists of a pair of two document samples. Users judge the content similarity of the two documents and label this query with either a *must-link* or a *cannot-link*.

- PAIRWISE-TARGETANCHOR: Each pairwise label query includes one target document and a few anchor documents. The target document is one on which the active learner solicits user feedback. The anchor documents are representatives of topics in the current model. Specifically, for each topic t, the active learner selects an anchor document who has minimum Euclidean distance with an ideal anchor θ'_t, where in the ideal anchor θ'_t, all the components are zero except the value of the t^{th} component is 1. Currently, we use SEED-MAXENTROPY to select a target document since based on our preliminary evaluation, it performed better than SEED-MINLIKELIHOOD.
- PAIRWISE-KMEANS: We apply Kmeans clustering using the document's topic distribution inferred by the current model as features. We compute the distance between each cluster's centroid. We then either select a document pair within a single cluster, or from two clusters with distance less a pre-defined threshold. The intuition behind this strategy is that if LDA has already done a good job separating documents into different clusters, it is not necessary to acquire additional user feedback on those pairs. In contrast, for document pairs that are not well separated, additional user feedback might be need to help the model to learn cluster boundary faster.

5.2. Evaluation

In this section, we conduct two experiments to evaluate the active learning framework, one with simulated user annotations which we obtain from dataset's ground-truth labels. We evaluate the effectiveness of seed labels and pairwise labels and different active learning strategies with classical LDA and baseline passive learning.

Passive Learning Instead of obtaining user annotations with the above active learning strategies, we can also randomly select queries for user to annotate. In contrast to active learning, the random sampling is a passive method, and thus we considered it as baseline in the evaluation. Since we focus on seed and pairwise user annotations, we create two baseline passive learning methods: Seed-Passive and Pairwise-Passive.

5.2.1. Simulation Experiments

We use the following datasets in the simulation experiments, and the preprocessing is applied to all datasets using Stanford CoreNLP tools.

- 20 Newsgroup: The 20 Newsgroup dataset[1] is organized into 20 different newsgroups, each corresponding to a different topic. Some of the newsgroups are very closely related to each other, while others are not. It totally contains 11,314 documents.
- Reddit: It contains 3,126 forum posts extracted from `www.reddit.com` in 7 categories.[2]

The documents in both datasets have a category label, thus we can use the ground-truth label to simulate user input. Starting with 100 burn-in LDA iterations, we perform

[1]Available at `http://people.csail.mit.edu/jrennie/20Newsgroups`

[2]We extract seven topics: breakup, conspiracy, findbostonbombers, gaming, iphone, politics, and religion.

Table 5.1. Fifteen most probable words of each topic before (above) and after active learning (below).

Topic	Words
1	space subject organization lines nasa launch writes moon orbit data article earth system shuttle work
2	subject organization lines writes article year good team university game time baseball players ll ve
3	hockey team play nhl game lines games subject university organization season period pittsburgh la win
1	space nasa launch moon orbit data earth system work shuttle lunar satellite research pat program
2	baseball period david runs hit home gm bob lost night won braves dave guy pitching
3	hockey nhl pittsburgh la cup pts canada points leafs detroit wings rangers flyers montreal playoffs

successive rounds of refinement. In each round, a new label is added corresponding to the dataset labels. We keep the total number of training iterations the same for all methods.

We first conduct an experiment to assess how well the discovered latent topic structure can reproduce the dataset's inherent structure. In particular, we use the per-document topic distribution as feature vector in a supervised multi-class classifier[**21**]. The lower the classification error rate, the better the model has captured the structure of the corpus. Note that here our goal is to study regularized topic models and the corresponding active learning strategies, we do not use state of the art features (tf-idf unigrams) to do the classification, and therefore the classification results are below state of the art. We find that:

(1) All of the passive learning and active learning methods with document annotations outperform classical LDA. The more annotations, the better classification performance. All of the active learning methods have better classification performance

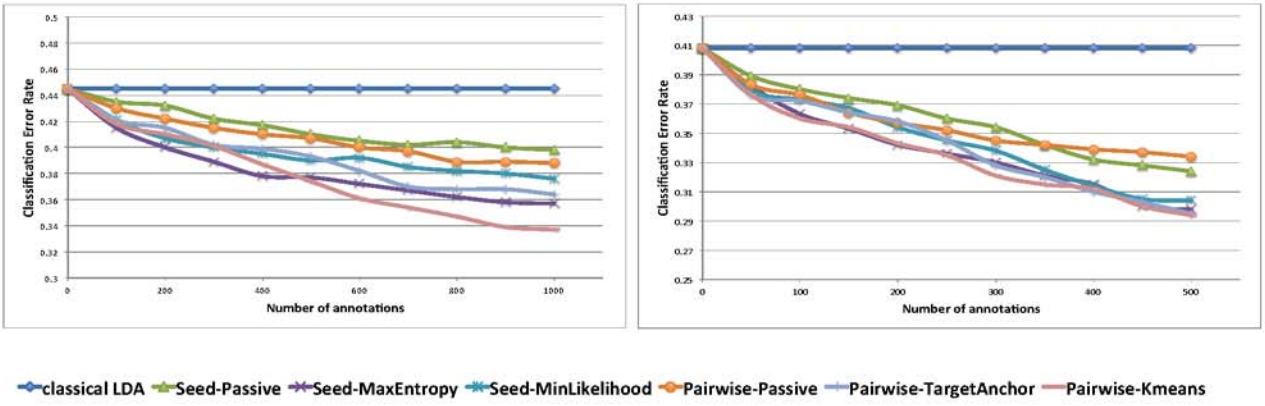


Figure 5.2. Classification experiments on 20Newsgroup (left) and Reddit(right) dataset.

than the two baseline methods, SEED-PASSIVE and PAIRWISE-PASSIVE, with statistical significance.[3]

(2) For seed label, MAXENTROPY outperforms MINLIKEHOOD with statistical significance. For pairwise label, KMEANS outperforms TARGETANCHOR. Moreover, PAIRWISE-KMEANS performs the best among all four active learning strategies. For 20 Newsgroup dataset, it offers 24.1%, 11.4%, 7.6% error rate reduction over classical LDA, passive learning, and SEED-MAXENTROPY respectively. For the Reddit dataset, PAIRWISE-KMEANS does not statistical significantly outperform PAIRWISE-TARGETANCHOR and SEED-MAXENTROPY, but it does offer 27.9% and 9.8% error rate reduction over classical LDA and passive learning with statistical significance respectively.

Secondly, we conduct an experiment to assess how well the discovered topic keywords matches with human judgement. Previously, topic models are often evaluated using perplexity on held-out test data. However, recent researches have shown that human judgement

[3]All statistical analyses are performed using Chi-Square test, $p < 0.05$.

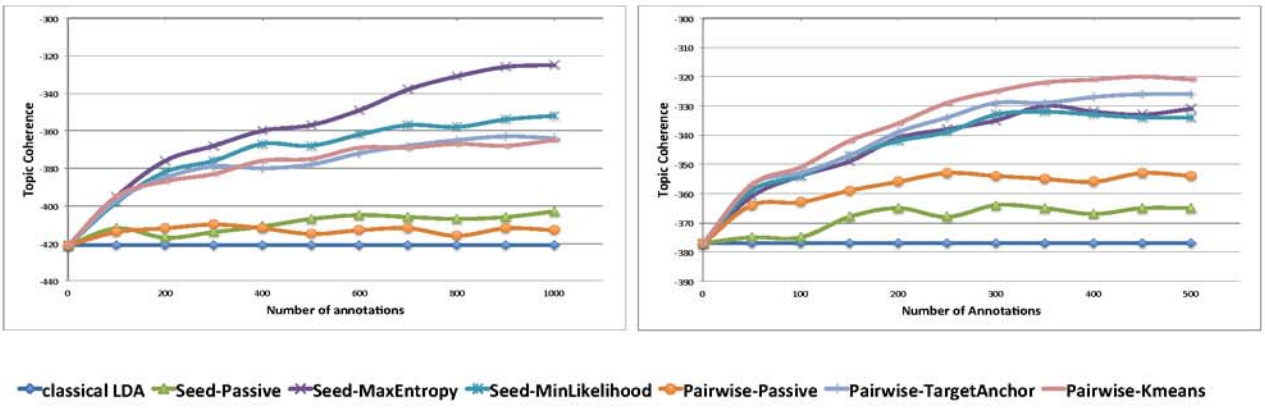


Figure 5.3. Topic Coherence experiments on 20Newsgroup (left) and Reddit(right) dataset.

sometimes is contrary to the perplexity measure. Following [**39**], we employ Topic Coherence, a metric which was shown to be highly consistent with human judgement, to measure a topic model's quality.

We can also see in Figure 5.3 that all active learning methods have better topic coherence than passive learning with statistical significance. For 20 Newsgroup dataset, SEEDMAXENTROPY has the best topic coherence score, and for Reddit dataset, PAIRWISEKMEANS achieves the best topic coherence performance.

The above preliminary experiments with simulated user input demonstrate the potential advantages of using active learning strategy to solicit user annotations. To further demonstrate the usefulness of the active learning framework, we need to conduct real user study, and this is a future research item. For now, we design and implement a user interface for soliciting user pairwise document annotations. Figure 5.4 shows a screenshot of the UI that can be used to solicit pairwise document annotations from users. The topics in the current model are displayed as a ranked list of topic keywords. To provide an overview of each document, we summarize the content of documents in a tag cloud. Users can click the "show

original text" button to read details. After inspecting the topic definitions as well as the content of the anchor and target document, a user clicks either the "similar" or "not similar" button to provide feedback.

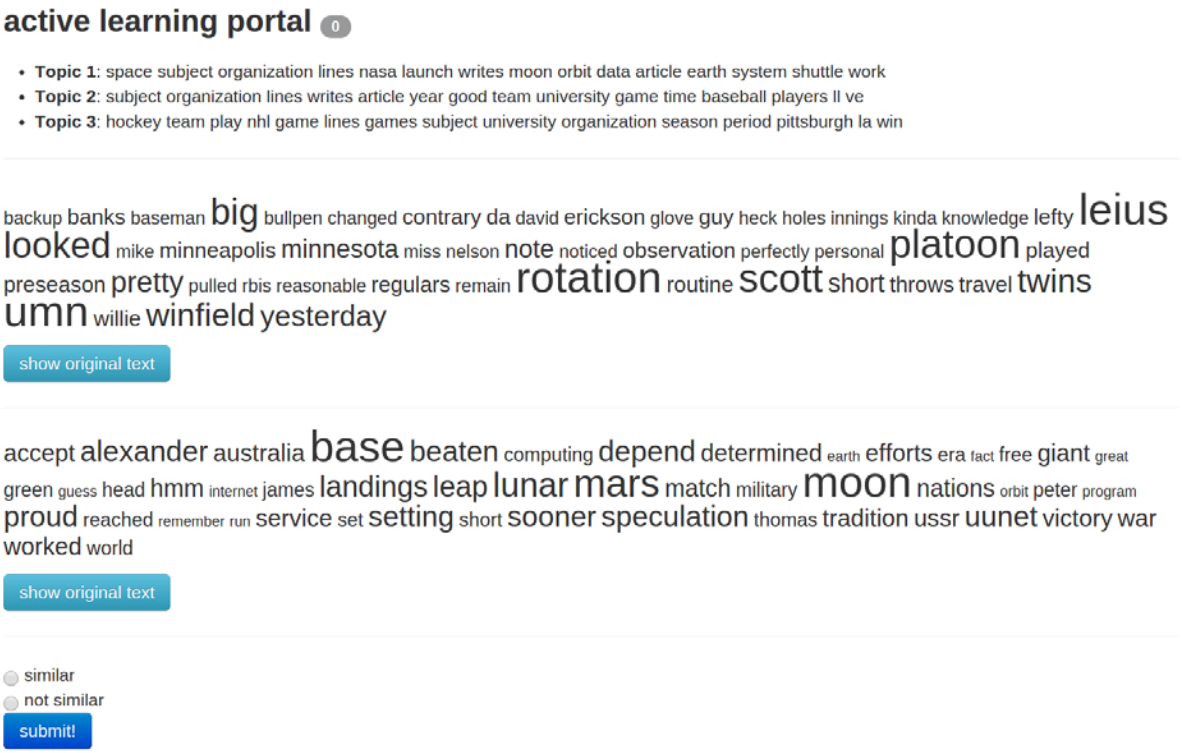


Figure 5.4. The UI for soliciting pairwise document labels.

5.3. Conclusion

In this chapter, We introduce an active learning framework that interactively and iteratively acquires user annotations. To demonstrate the effectiveness of our methods, we conduct pilot experiments with simulated inputs on two different datasets. Our results demonstrate that the proposed active learning method outperforms both classic LDA and passive LDA in improving topic coherence and reducing document classification errors.

CHAPTER 6

Conclusion

Being able to understand large collections of unstructured textual documents will benefit various applications in an age of information abundance and exploration. Probabilistic topic models, which are designed to identify topical representations of the documents that reduce the dimension and reveal documents structure, have attracted attentions in researchers, data analysts from different principles. While many data analysis practitioners hope to use topic models directly to discover document collections, they often encounter unexpected and undesirable results. It is not uncommon that after many hours of training, the topics learned by the model do not make sense to them. For example, a topic is represented with semantically incoherent words, or words that are semantically related are in different topics. Moreover, topic models suffer from the instability problem when the model is updated with new content or even the model is retrained on the same content. These undesirable results severely undermine the usability of topic models, and it hampers adoption and prevents topic models from being more widely used.

To address the usability problems of topic models, we propose user-centric approaches in which users play critical roles in the topic modeling system. Users can provide prior knowledge or domain knowledge to topic models that adapt to meet their needs and desires. Users can also direct the topic model update by the integration of their guidance so as to keep the stability of the update. The core of our user-centric approach is a factor graph framework SC-LDA, which is able to efficiently incorporate knowledge with LDA. The knowledge is

encoded as soft constraints that shape the topics learned by LDA. The advantage of SC-LDA over existing methods are two folds. Firstly, its training is fast. Experiments on large dataset with many topics show that SC-LDA has a more rapid convergence rate than baseline methods. Efficient training makes the training of large topic models on large document collection become feasible. Secondly, it presents an unified framework for handling different types of knowledge, and we show that natural, sparse representations of prior knowledge are possible. This is the first work achieves both scale and rich prior information integration. We expect this work will interest data analysis practitioners who would like to efficiently train a big topic model with prior knowledge.

We also investigate the instability problem of topic models which is closely related to end users experience but has been overlooked by the machine learning community. Instability is an intrinsic problem of LDA partly because the model is a non-convex function with many local optimals. For an end user, topic model instability can be felt from the changes of the topics assigned to the same documents. To address the instability problem, we present a user-directed topic model update system, nTMU, which solicits users feedback based on their preferences of the stability of topics. Users feedback, which is translated into document pairwise constraints, is then incorporated into LDA under SC-LDA framework. User experiments show that nTMU can effectively maintain the stability of document-to-topic mappings and minimize the disruption to the mental maps of end users. We hope our work will help topic modeling practitioners who have experienced the instability problem in their practice.

In this big data era, as the scale of data becomes larger and larger, the underlying machine learning algorithms to handle the data become more and more sophisticate. Machine learning can give us ability to use massive data by uncovering useful insights from raw data.

However, the complexity of machine learning has largely restricted its use to experts and skilled developers. For many end users who are not machine learning experts, they use the algorithm as a black box. When the output from the black box fails to make sense, the end users become frustrated and lose confidence in the algorithm, even when the machine learning researchers can demonstrate that the algorithms have high predictive accuracy. Therefore, we stress that algorithm usability should always be an important factor to take into consideration. This thesis expects to interest and inspire machine learning and data mining researchers to pay more attention to the development of human-centric data analytics algorithms to improve their usability.

6.1. Limitations and Future Work

The methods and theoretical results we presented have a number of limitations which could be addressed in future work. Many of the theoretical results rely on strong assumptions. One assumption is that users can generate high-quality input. In SC-LDA, we obtain prior knowledge from existing high quality document labels and knowledge base such as WordNet. Although SC-LDA provides a unified framework to incorporate prior knowledge or domain knowledge, we don't know yet how domain experts interact with SC-LDA to provide knowledge. In nTMU, we conduct user study to demonstrate that user guidance can help to maintain the stability of documents, but the user study is conducted on Amazon Mechanical Turk at a relatively small scale. While users may perform well on small scale dataset, we don't know yet if they will perform equally well on large scale dataset in a topic model update system. In the following paragraphs, I will list several research directions that can potentially further improve the usability of a topic model system or even a machine learning system.

The nTMU system allows users to decide which topics to keep stable based on their needs or preferences. The underlying topic model update system also assumes a fixed number of topics. However, as more and more data are coming, users may want to merge/split topics to find fine-grained topics. Therefore, a topic model with fixed number of topics may suffer from evolution and performance poorly. Non-parametric topic model which treats the topic number as a random variable will be more practical and useful in this scenario. Developing non-parametric model that incorporates user's feedback as document pairwise constraints is non-trivial task and is a further research item.

Another limitation of the thesis is the lack of front-end user study to validate the improvement in metrics like speed (Chapter 3) or stability (Chapter 4) will eventually translate into better user experience. Our primary focus is the design of back-end level algorithms to support front-end user interaction. For example, the SC-LDA model allows efficient incorporation of different kinds of knowledge. The development of back-end level algorithm is critical because without proper back-end support, it is impossible to achieve usability at the front-end level. This is also the significant distinguish between our work and previous interactive machine learning work which primarily focuses on front-end usability issue. However, we don't know to what extend the improvement on back-end system will result in better usability to the system users. For example, how much user satisfaction will be improved if we can speed up the underlying algorithm by 10 times? We need to conduct more user studies to fully understand the usability and user experience relationship with underlying algorithms.

Topic Modeling Visualization Visualization tools provide a user-friendly visual analysis interface for end users to review the topic model results, filter "good" topics from "bad" topics and diagnose learned models. Recent work on topic model visualization [**12, 1**] focuses

on displaying topic model parameters, such as per-document distribution and topic-word distribution, with document clusters and word clusters. Instead of showing top ranked words per each topic for end users, these visualization tool can also present underlying document and word similarity network so that end users can evaluate the model quality more rapid and accurate. However, these visual tools only passively display the model results to end users and they cannot solve the intrinsic problems of topic models. For end users, the underlying topic model is still black box to them. One future work is the integration of current topic modeling visualization tools with user interaction of knowledge. Users can see the changes on the interface when they provide feedback or knowledge to the topic models. That way the black box becomes "transparent" to the end users. Users can control and adjust the topic models that adapt to meet their needs and desires.

Intelligent User Interaction In the Chapter 5 of this thesis, we conduct a pilot study on active learning with topic modeling. In the proposed framework, we investigate the strategies of soliciting document seed label and document pairwise label from end users respectively. Our assumption here is that users, like oracle, can always generate desirable labels of the instances. In practice, it is *not* true. The label annotators do not always generate correct answers, especially when more and more data collection tasks are deployed on Amazon Mechanic Turk, where the annotators have different background and expertise. Therefore, we need a more intelligent active learning or user interaction mechanism to solicit users' annotations. The motivation for user modeling with active learning is that we hope the active learning algorithm is able to take an annotator's expertise/preference into consideration when asking the annotator questions. To achieve this, instead of selecting a query that the system is currently most confused with, we select a query that maximizes the system confusion as well as the annotator's expertise. Moreover, since annotator expertise usually forms a hierarchical

structure, instead of asking the annotator questions about the labels of instances, we can also ask him to answer a pairwise question, i.e., do these two instances belong or not belong to the same class, or a relative question, i.e., is instance a more similar to instance b than to instance c. Lastly, given annotator's answers, we update the annotator's expertise model so that the active learning system can further take advantage of. An annotator's expertise model is a latent variable model which consists of group expertise variable and individual expertise variable. The group expertise variable is shared by the crowd annotators, while the individual variable is annotator specific. The expertise latent variable directly connect to observed variable which is the answers the annotator generates. Further investigation of user modeling with active learning is a key item for future work of intelligent user interaction.

APPENDIX A

Publications During Ph.D. Study

- **Yi Yang**, Shimei Pan, Jie Lu, Mercan Topkara and Yangqiu Song. The Stability and Usability of Statistical Topic Models. (under submission)
- **Yi Yang**, Doug Downey and Jordan Boyd-Graber. Efficient Algorithm for Incorporating Knowledge into Topic Models. EMNLP, 2015.
- Doug Downey, Chandra Sekhar Bhagavatula and **Yi Yang**. A Latent Variable Document Model for Large Topic Hierarchies. ACL, 2015.
- **Yi Yang**, Shimei Pan, Yangqiu Song, Jie Lu and Mercan Topkara. User-directed Non-disruptive Topic Model Update for Effective Exploration of Dynamic Content. IUI, 2015. ***Best Long Paper Honorable Mention**
- **Yi Yang**, Shimei Pan, Jie Lu, Mercan Topkara, and Doug Downey. Incorporating User Input with Topic Modeling. Workshop on Interactive Mining for Big Data. CIKM, 2014.
- Hongyu Gao, **Yi Yang**, Kai Bu, Yan Chen, Doug Downey, Kathy Lee and Alok Choudhary. Spam aint as Diverse as It Seems: Throttling OSN Spam with Templates Underneath. 2014 Annual Computer Security Applications Conference. ACSAC, 2014.
- **Yi Yang**, Shimei Pan, Doug Downey and Kunpeng Zhang. Active Learning with Constrained Topic Model. Workshop on Interactive Language Learning, Visualization, and Interfaces. ACL, 2014.

- Fei Huang, Arun Ahuja, Doug Downey, **Yi Yang**, Yuhong Guo, and Alexander Yates. Learning Representations for Weakly Supervised Natural Language Processing Tasks. Computational Linguistics, 2014.
- Thanapon Noraset, Chandra Sekhar Bhagavatula and **Yi Yang** and Doug Downey. WebSail: English Entity Linking. Text Analysis Conference (TAC), 2013.
- **Yi Yang**, Doug Downey and Alexander Yates. Overcoming the Memory Bottleneck in Distributed Training of Latent Variable Models of Text. NAACL-HLT, 2013.

References

[1] Eric Alexander, Joe Kohlmann, Robin Valenza, Michael Witmore, and Michael Gleicher. Serendip: Topic model-driven visual exploration of text corpora. In *PRE-PRINT: Proceedings of the 2014 IEEE Conference on Visual Analytics Science and Technology (VAST)*, November 2014.

[2] Saleema Amershi, James Fogarty, Ashish Kapoor, and Desney S. Tan. Effective end-user interaction with machine learning. In Wolfram Burgard and Dan Roth, editors, *AAAI*. AAAI Press, 2011.

[3] David Andrzejewski, Xiaojin Zhu, and Mark Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pages 25–32, New York, NY, USA, 2009. ACM.

[4] David Andrzejewski, Xiaojin Zhu, Mark Craven, and Benjamin Recht. A framework for incorporating general domain knowledge into latent dirichlet allocation using first-order logic. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Two*, IJCAI'11, pages 1171–1177, 2011.

[5] Arindam Banerjee and Sugato Basu. Topic models over text streams: A study of batch and online unsupervised learning. In *SDM*, 2007.

[6] Sugato Basu, A. Banjeree, ER. Mooney, Arindam Banerjee, and Raymond J. Mooney. Active semi-supervision for pairwise constrained clustering. In *SDM*, pages 333–344, 2004.

[7] David Blei and Jon McAuliffe. Supervised topic models. In *NIPS*, pages 121–128, 2008.

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

[9] Kevin R. Canini, Lei Shi, and Thomas L. Griffiths. Online inference of topics with latent dirichlet allocation. In *AISTAT*, 2009.

[10] Jonathan Chang, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. Reading tea leaves: How humans interpret topic models. In Y. Bengio, D. Schuurmans, J. Lafferty, C. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 288–296. 2009.

[11] Jaegul Choo, Changhyun Lee, Chandan K. Reddy, and Haesun Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):1992–2001, December 2013.

[12] Jason Chuang, Christopher D. Manning, and Jeffrey Heer. Termite: Visualization techniques for assessing textual topic models. In *Advanced Visual Interfaces*, 2012.

[13] M. S. Crowther, C. C. Keller, and G. L. Waddoups. Improving the quality and effectiveness of computer-mediated instruction through usability evaluations. In *British Journal of Educational Technology, 35(3)*, pages 289–303, 2004.

[14] Anind K. Dey, Raffay Hamid, Chris Beckmann, Ian Li, and Daniel Hsu. A cappella: Programming by demonstration of context-aware applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 33–40, New York, NY, USA, 2004. ACM.

[15] A. Dix, J. Finlay, G. Abowd, and Beale R. Human-computer interaction. In *Prentice Hall, Upper Saddle River, NJ, USA.*, 1998.

[16] Jerry Alan Fails and Dan R. Olsen, Jr. Interactive machine learning. In *Proceedings of the 8th International Conference on Intelligent User Interfaces*, IUI '03, pages 39–45, New York, NY, USA, 2003. ACM.

[17] James Fogarty, Desney Tan, Ashish Kapoor, and Simon Winder. Cueflik: Interactive concept learning in image search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 29–38, New York, NY, USA, 2008. ACM.

[18] James Foulds, Shachi Kumar, and Lise Getoor. Latent topic networks: A versatile probabilistic programming framework for topic models. In *International Conference on Machine Learning (ICML)*, 2015.

[19] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(Suppl. 1):5228–5235, 2004.

[20] David Hall, Daniel Jurafsky, and Christopher D. Manning. Studying the history of ideas using topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, pages 363–371, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.

[21] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD*, 2009.

[22] G. Heinrich. Parameter estimation for text analysis. 2005.

[23] Matthew D. Hoffman, David M. Blei, and Francis Bach. Online learning for latent dirichlet allocation. In *In NIPS*, 2010.

[24] Daniel J. Hopkins. The exaggerated life of death panels: The limits of framing effects on health care attitudes, 2012.

[25] Yuening Hu and Jordan Boyd-Graber. Efficient tree-based topic modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 275–279, 2012.

[26] Yuening Hu, Jordan Boyd-Graber, and Brianna Satinoff. Interactive topic modeling. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 248–257, 2011.

[27] Hal Daum III. Markov random topic fields. In *ACL/IJCNLP (Short Papers)*, pages 293–296. The Association for Computer Linguistics, 2009.

[28] Adriana Kovashka, Devi Parikh, and Kristen Grauman. Whittlesearch: Interactive image search with relative attribute feedback. *CoRR*, abs/1505.04141, 2015.

[29] Jennifer Lai, Jie Lu, Shimei Pan, Danny Soroker, Mercan Topkara, Justin Weisz, Jeff Boston, and Jason Crawford. Expediting expertise: Supporting informal social learning in the enterprise. In *IUI*, pages 133–142, 2014.

[30] Aaron Q. Li, Amr Ahmed, Sujith Ravi, and Alexander J. Smola. Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 891–900, 2014.

[31] Fei-Fei Li and Pietro Perona. A bayesian hierarchical model for learning natural scene categories. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 524–531, Washington, DC, USA, 2005. IEEE Computer Society.

[32] Shixia Liu, Michelle X. Zhou, Shimei Pan, Yangqiu Song, Weihong Qian, Weijia Cai, and Xiaoxiao Lian. Tiara: Interactive, topic-based visual text summarization and analysis. *ACM Trans. Intell. Syst. Technol.*, 3(2):25:1–25:28, 2012.

[33] Yan Liu, Alexandru Niculescu-Mizil, and Wojciech Gryc. Topic-link lda: Joint models of topic and author community. In *ICML*, pages 665–672, 2009.

[34] Jon D. Mcauliffe and David M. Blei. Supervised Topic Models. In *Advances in Neural Information Processing Systems*, pages 121–128, 2008.

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

[36] Qiaozhu Mei, Deng Cai, Duo Zhang, and ChengXiang Zhai. Topic modeling with network regularization. In *WWW*, pages 101–110, 2008.

[37] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[38] George A. Miller. Wordnet: A lexical database for english. *Commun. ACM*, 38(11):39–41, November 1995.

[39] David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. Optimizing semantic coherence in topic models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 262–272, 2011.

[40] Ramesh M. Nallapati, Amr Ahmed, Eric P. Xing, and William W. Cohen. Joint latent topic models for text and citations. In *KDD*, pages 542–550, 2008.

[41] David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. Distributed algorithms for topic models. *J. Mach. Learn. Res.*, 10:1801–1828, December 2009.

[42] David Newman, Edwin V. Bonilla, and Wray L. Buntine. Improving topic coherence with regularized topic models. In *NIPS*, pages 496–504, 2011.

[43] J Nielsen. Usability engineering. In *Academic Press, Boston, MA, USA.*, 1993.

[44] Donald A. Norman and Jakob Nielsen. 10 heuristics for user interface design. `http://www.nngroup.com/articles/ten-usability-heuristics/`, 2013. [Online; Retrieved 31-August-2013].

[45] Shimei Pan, Michelle X. Zhou, Yangqiu Song, Weihong Qian, Fei Wang, and Shixia Liu. Optimizing temporal topic segmentation for intelligent text visualization. In *IUI*, pages 339–350, 2013.

[46] Michael J. Paul and Roxana Girju. A two-dimensional topic-aspect model for discovering multi-faceted topics. In Maria Fox and David Poole, editors, *AAAI*. AAAI Press, 2010.

[47] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed gibbs sampling for latent dirichlet allocation. In *Proceedings*

of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '08, pages 569–577, 2008.

[48] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 248–256, 2009.

[49] Daniel Ramage, Christopher D. Manning, and Susan Dumais. Partially labeled topic models for interpretable text mining. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 457–465, New York, NY, USA, 2011. ACM.

[50] Matthew Richardson and Pedro Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, February 2006.

[51] Alan Ritter and Sumit Basu. Learning to generalize for complex selection tasks. In *Proceedings of the 14th International Conference on Intelligent User Interfaces*, IUI '09, pages 167–176, New York, NY, USA, 2009. ACM.

[52] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the Space of Topic Coherence Measures. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining - WSDM '15*, pages 399–408, New York, New York, USA, February 2015. ACM Press.

[53] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *UAI*, pages 487–494, 2004.

[54] Burr Settles. Active learning literature survey. Technical report, 2010.

[55] B Shackel. Ergonomics in design for usability. In *Proceedings of the Second Conference of the British Computer Society, Human Computer Interaction Specialist Group on People and Computers: Designing for Usability*, pages 44–64, New York, NY, USA, 1986. Cambridge University Press.

[56] Xiaodan Song, Ching-Yung Lin, Belle L. Tseng, and Ming-Ting Sun. Modeling and predicting personal information dissemination behavior. In *KDD*, pages 479–488, 2005.

[57] Yangqiu Song, Shimei Pan, Shixia Liu, Michelle X. Zhou, and Weihong Qian. Topic and keyword re-ranking for lda-based topic modeling. In *CIKM*, pages 1757–1760, 2009.

[58] Elena Erosheva Stephen, Stephen Fienberg, and John Lafferty. Mixed membership models of scientific publications. In *Proceedings of the National Academy of Sciences*, page 2004, 2004.

[59] James J. Thomas and Kristin A. Cook. *Illuminating the Path: The Research and Development Agenda for Visual Analytics*. National Visualization and Analytics Ctr, 2005.

[60] Peter D. Turney. Technical note: Bias and the quantification of stability. *Machine Learning*, 20(1-2):23–33, 1995.

[61] Hanna M. Wallach, David M. Mimno, and Andrew McCallum. Rethinking lda: Why priors matter. In *NIPS*, pages 1973–1981, 2009.

[62] Yi Wang, Xuemin Zhao, Zhenlong Sun, Hao Yan, Lifeng Wang, Zhihui Jin, Liubin Wang, Yang Gao, Ching Law, and Jia Zeng. Peacock: Learning Long-Tail Topic Features for Industrial Applications. page 23, May 2014.

[63] Xing Wei and W. Bruce Croft. Lda-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '06, pages 178–185, New York, NY, USA, 2006. ACM.

[64] Pengtao Xie, Diyi Yang, and Eric P Xing. Incorporating word correlation knowledge into topic modeling. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics*, 2015.

[65] Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, page 937. ACM Press, June 2009.

[66] Jinhui Yuan, Fei Gao, Qirong Ho, Wei Dai, Jinliang Wei, Xun Zheng, Eric P. Xing, Tie-Yan Liu, and Wei-Ying Ma. LightLDA: Big Topic Models on Modest Compute Clusters. December 2014.

[67] Ke Zhai, Jordan Boyd-Graber, Nima Asadi, and Mohamad L. Alkhouja. Mr. lda: A flexible large scale topic modeling package using variational inference in mapreduce. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 879–888, 2012.

[68] Ke Zhai and Jordan L. Boyd-Graber. Online latent dirichlet allocation with infinite vocabulary. In *ICML*, pages 561–569, 2013.