Title	トピックモデルのための高速スパース推論
Author(s)	Than, Quang Khoat
Citation	
Issue Date	2013-09
Type	Thesis or Dissertation
Text version	ETD
URL	http://hdl.handle.net/10119/11555
Rights	
Description	Supervisor:ホー　バオ　ツー，知識科学研究科，博士

Fast and Sparse Inference for Topic Models

by

Than Quang Khoat

submitted to
Japan Advanced Institute of Science and Technology
in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Supervisor: Ho Tu Bao

School of Knowledge Science
Japan Advanced Institute of Science and Technology

September, 2013

Abstract

Topic modeling has been increasingly maturing to be an attractive research area. Originally motivated from textual applications, it has been going beyond far from text to touch upon many amazing applications in Computer Vision, Bioinformatics, Software Engineering, Forensics, Cognitive Science, History, Politics, to name a few. It is believed to be one of the keys to automatically understanding documents written by human, and to uncovering how human knowledge is created and represented.

This thesis studies to model texts at a large scale. In other words, the thesis studies to propose models that most appropriately generate documents, and then to derive efficient methods for learning those models from a large number of available texts. To this end, the thesis systematically elucidates the two fundamental issues to be resolved: *inference of topic mixtures* and *model complexity*. The thesis then targets at developing provably fast algorithms that can recover sparse topic mixtures for documents, and developing fast algorithms to learn sparse topic models.

The first contribution is the introduction of a simple framework for inference of sparse topic mixtures, called FW, which is general and flexible enough to be employed in admixture models. The framework enjoys the following key theoretical properties: (1) inference provably converges at a linear rate to the optimal solutions; (2) prior knowledge can be easily incorporated into inference; (3) the sparsity level of topic mixtures can be directly controlled; (4) it is easy to trade off sparsity against quality and runtime. Existing inference methods do not own these properties and often work slowly. Those properties are attractive for large scale modeling.

We demonstrate the goodness and flexibility of FW by employing it to design novel methods for supervised dimension reduction. When working with very high dimensional problem, it is sometimes beneficial in efficiency and effectiveness to reduce the dimensionality of the problem, but keep or make better predictiveness of the response variable. The main result of this study is a novel method that can reach state-of-the-art performance while enjoying 30-450 times faster speed than existing methods.

The second contribution is the introduction of *Fully Sparse Topic Model* (FSTM) for modeling large collections of documents. Three key properties of the model are: (i) the

inference algorithm converges at a linear rate to the optimal solutions, (ii) it provides a principled way to directly trade off sparsity of solutions against inference quality and running time, (iii) the learning algorithm has low complexity which is near independent of dimensionality. FSTM overcomes many limitations of existing topic models, and has been demonstrated to work qualitatively on real data. The low computational complexity and low model complexity can help us work with large text collections.

The third contribution is the introduction of a fast algorithm for learning Correlated Topic Models (CTM), as well as a theory of *probable convexity* for analyzing convexity of real functions. Previous studies show that posterior inference in nonconjugate models such as CTM is intractable (NP-hard) in the worse case. However, we show that it may not be true in practice. Indeed, by introducing the concept of *probable convexity*, we show that inference of topic mixtures in CTM and many nonconjugate models is tractable in practice. Based on these findings, a novel algorithm is proposed which is surprisingly simple but is easily parallelizable or distributable. By extensive experiments, the algorithm is shown to work significantly faster than existing expensive methods while keeping comparable or better quality of the learned models.

Acknowledgments

Firstly, I would like to thank my supervisor, Ho Tu Bao. I have learnt a huge amount from him. His inspiration and feedbacks have been invaluable for the journey through grad school and for my future academic career. I am also very grateful to him for his tolerance for providing me freedom to pursue my own academic path.

Secondly, sincere thanks are due to the committee members of my dissertation, including Takashi Washio, Yoshiteru Nakamori, Riichiro Mizoguchi, and Dam Hieu Chi. I also want to thank Hiroshi Motoda very much. My writing has benefited from their feedbacks and insightful discussions. Their diverse views on my study are invaluable. By one way, those diverse views implicitly helped broaden my horizons. By some other ways, they helped me see clearly the weaknesses of the study.

I would like to thank many of the members of the Machine Learning group at JAIST for fruitful discussion and collaboration. I learned much from listening to and discussing with friends/colleagues including Doan Anh Vu, Huynh Van Nam, Nguyen Le Minh, Pham Quang Nhat Minh, Ngo Xuan Bach, Le Thi Nhan, Bui Ngoc Thang, Pham Ngoc Khanh, Nguyen Duy Khuong.

Sincere thanks go to Nguyen Xuan Long, Tomoharu Iwata and John Lafferty for insightful discussions about topic modeling. The breadth but depth in their lectures have pointed me out how far I am from being an independent researcher.

I owe a great debt of gratitude to the members of my family for their continual supports and encouragements during the hard time of the graduate study. Without them, I would not have completed this study.

Finally, I would like to express my great gratitude to MEXT (a Japanese Government scholarship) and JAIST for providing me an opportunity to join the excellent research environment at JAIST and for supporting me throughout the studies in this dissertation. I also want to thank NEC C & C Foundation and Japan Association for Mathematical Sciences for providing me financial supports.

Contents

List of Figures

List of Tables

List of Abbreviations

AIC Akaike Information Criterion

BIC Bayesian Information Criterion

CGS collapsed Gibbs sampling

CTM Correlated Topic Models

CVB collapsed variational Bayesian

DTM Dynamic Topic Model

FSTM Fully Sparse Topic Model

HDP Hierarchical Dirichlet Process

hLDA Hierarchical LDA

IFTM Independent Factor Topic Model

LDA Latent Dirichlet Allocation

LSA Latent Semantic Analysis

pLSA Probabilistic Latent Semantic Analysis

SDR supervised dimension reduction

TM Topic modeling

VB variational Bayesian

Chapter 1

Introduction

Humans are excellent at learning from text documents, and making inference on new observations based on his/her knowledge. The ability to reason correctly from little and noisy information is one of the long-standing mysteries that neuroscientists have been trying to understand. It motivates researchers in Artificial Intelligence and Machine Learning to make an intelligent computer that can mimic human abilities and behaviors. In particular, the development of algorithms that enables computers to automatically process text and natural language has always been one of the most challenges [51].

Such algorithms are even more necessary in the era of "Big Data". A large amount of valuable information are put on the web everyday, which are in various forms including news, blogs, images, musics, videos, opinions, social networks, etc. The needs of intelligent algorithms to manage, explore, and discover new knowledge from those huge data sources are increasingly arising. Hence a significant progress in the development of intelligent algorithms promises to have a strong impact on various applications ranging from information retrieval, recommendation systems, human-machine interaction, to computational social science, bioinformatics, forensics, history, and politics.

Topic modeling (TM) is a potential approach to helping organize, search, and understand vast amounts of information. It is a relatively new field whose algorithms allow us to uncover the underlying semantic structure (gists) of a document collection and use them for various tasks. This literature has seen numerous successful applications such as semantic representation [43, 58], information retrieval [32, 51], trends detection [20, 44], understanding images and audios [52, 121], historic study [46, 49, 65], discovery of hidden biological factors [36, 61, 78, 96], analysis of social networks [25, 87, 95], analysis/prediction of political issues [41, 45].

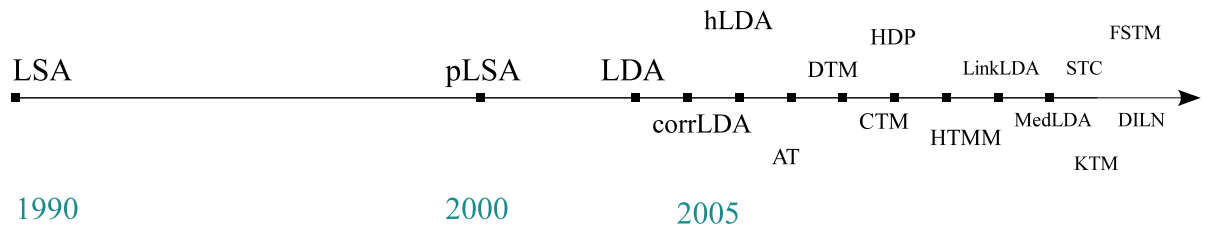


Figure 1.1: A historic illustration of the development of TM.

1.1 Topic modeling

Originated from information retrieval, the first work goes back at least to the proposal of *Latent Semantic Analysis* (LSA) in 1990 by Deerwester et al. [32]. This work shows that semantic structures hidden in text collections can be extracted by analyzing the co-occurrences of terms in documents. The next remarkable results are the introduction of probabilistic versions of LSA [51, 76] whose underlying foundation bases on Statistics. *Probabilistic Latent Semantic Analysis* (pLSA) [51] and *Latent Dirichlet Allocation* (LDA) [21] are two of the most successful developments in TM. Since then the literature has seen a flourish in both theory and application. Figure 1.1 demonstrates the historic development of TM.

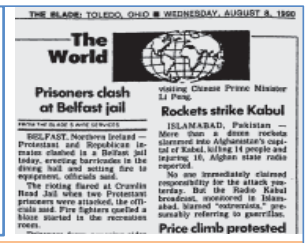
The key assumption in TM is that each document exhibits multiple topics. For example, a news which is titled "Rockets strike Kabul" may talk about a terrorist attack and damage, but is unlikely about a scientific discovery. Figure 1.2 demonstrates this intuition by drawing the composition of the news from some topics. The composition is uncovered by an *inference* process, while the topics are automatically learned from a collection of news articles. A model of topics (or semantic structures) which are hidden in documents is known as a *topic model*. For an introduction to topic modeling, we refer the readers to some excellent surveys such as [15, 19, 31, 92].

1.2 A brief of recent trends in TM

Due to the potential applicability in a wide range of areas, TM recently has attracted significant attentions from academics and industry. Significant progresses have been made in many aspects of TM after the introduction of probabilistic topic models. The followings are some of the most attractive research directions.

Large scale learning: The popularity of the Internet opens an easy way to share and

Rockets strike Kabul -- AP, August 8, 1990.
More than a dozen rockets slammed into Afghanistan's capital of Kabul today, killing 14 people and injuring 10, Afghan state radio reported. No one immediately claimed repsonsibility for the attack. But the Radio Kabul broadcast, monitored in Islamabad, blamed ``extremists,'' presumably referring to U.S.-backed guerrillas headquartered in Pakistan. Moslem insurgents have been fighting for more than a decade to topple Afghanistan's Communist-style government. In the past year, hundreds of people have died and thousands more injured in rocket assaults on the Afghan capital.


THE BLADE: TOLEDO, OHIO ■ WEDNESDAY, AUGUST 8, 1990

The World

Prisoners clash at Belfast jail

BELFAST, Northern Ireland — Protestant and Republican inmates clashed in a Belfast jail today, erecting barricades in the dining hall and setting fire to equipment, officials said.

The rioting flared at Crumlin Road Jail when two Protestant prisoners were attacked, the officials said. Fire fighters quelled a blaze started in the recreation room.

visiting Chinese Prime Minister Li Peng.

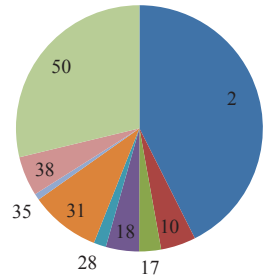
Rockets strike Kabul

ISLAMABAD, Pakistan — More than a dozen rockets slammed into Afghanistan's capital of Kabul, killing 14 people and injuring 10, Afghan state radio reported.

No one immediately claimed responsibility for the attack yesterday. But the Radio Kabul broadcast, monitored in Islamabad, blamed "extremists," presumably referring to guerrillas.

Price climb protested

How much topics contribute to the news?



Some topics previously learned from a collection of news							
2	10	18	31	38	40	42	50
police	sinhalese	contra	fire	shuttle	gorbachev	index	beirut
students	tamil	sandinistas	winds	nasa	soviet	stock	hezbollah
palestinians	iranian	chamorro	firefighters	space	republics	yen	lebanon
curfew	dam	ortega	mph	launch	politburo	points	aoun
sikh	khomeini	rebels	blaze	magellan	yeltsin	market	syrian
gaza	sri	sandinista	brush	mars	moscow	shares	militia
rangoon	cemetery	aid	homes	spacecraft	tass	trading	lebanese
moslem	accord	nicaragua	acres	telescope	party	dow	amal
israeli	wppss	managua	water	venus	treaty	unchanged	troops
militants	guerrillas	ceasefire	weather	astronauts	grigoryants	volume	wounded

Figure 1.2: Topics and their roles in a news by Associated Press. The diagram in the left-hand shows among 50 topics which may appear in and how significantly each contributes to the news. Each column in the right-hand shows a latent topic, which is a set of semantically related terms. The composition of topics in the news and the topics themselves are automatically learned from a collection of news.

collect data by various ways. A large amount of texts are available online which are in many forms including blogs, tweets, comments, opinions, reviews, etc. Hence many studies in TM have tried to develop methods that can learn valuable knowledge from those sources. State-of-the-art methods can work well with millions of documents [13, 48, 67, 71, 90, 109, 116] and collections with millions of terms (dimensions) [100, 116, 126].

Sparse modeling: This direction is motivated from both the nature of data and the need of modeling at a large scale. In reality, the inherent structure of an object is often sparse in the sense that only some components among many are sufficient to describe/generate the object. For instance, a news often talks about 2 or 3 events (topics) among a large amount of available events. Hence it is reasonable to develop models that respect the sparseness hidden in data. Unfortunately, using dense distributions to describe topics in models like pLSA and LDA seem to be unrealistic [112]. Many recent researches in TM attack the two factors of wide interests, hidden topics [89, 100, 112, 116] and topic mixtures in documents [89, 100, 120, 129]. Many of them exploit regularization techniques and sparse approximation to model sparsity.

Nonparametric models: When working with a text collection, one often does not know exactly the number of topics that collection contains. Parametric models such as pLSA and LDA cannot give a satisfying answer to this situation. Hence a number of recent

researches have tried to borrow techniques from nonparametric statistics. A significant progress has been made with the introduction of *Hierarchical Dirichlet Process* (HDP) [98] and *Hierarchical LDA* (hLDA) [22]. Various works follow this direction that can model more complicated interactions between hidden variables [16, 23, 37, 63, 120, 126].

Theoretical foundation: Over two decades of development, researches in TM often focus more on practical aspects of topic models but leave theoretical foundation open, particularly for probabilistic topic models. The first theoretical work by Papadimitriou et al. [76] explains why LSA often works well in practice. Recently, many open problems have been fulfilled such as accuracy of recovering a model from data [7, 9, 10]. Some other works concern on computational complexity such as [10, 68, 91]. Evaluation and model checking [15] also gain significant interests, since reliable and interpretable models would be very important for applications in other fields. Some pioneer works in this direction include [26, 66, 72, 73].

Incorporating meta-data: In many situations, some texts may have side information such as links, labels, tags, and weblogs. Those meta-data may contain very important information about documents, and could help us model data better [15]. Hence various works exploited meta-data to develop topic models for practical tasks including classification [14, 63, 130], authorship and influence prediction [85, 93, 106, 118], personality study [87], and community detection [83, 95].

1.3 Some challenges

Although topic modeling had an impressive development over the last two decades, there remains many open problems that should be studied further. In particular, the rise of "Big Data" poses various challenges that may require new breakthroughs in both methodology and hardware architecture. The followings are some of the challenges.

C1. Large scale learning: Exploration of a huge text collection (e.g. a century of scientific articles maintained by JSTOR, or the Google collection of more than 30 millions of books) often requires us to learn a model with hundreds of thousands of hidden topics. Learning such a model would necessarily involve billions of hidden variables and hence is very expensive. This oversize model seem to be out of reach for the state-of-the-art learning methods. More challengingly, the model and the text collection sometimes cannot fit in the available storage capacity of a supercomputer, and can preclude traditional learning methodology. Hence developing new models and scalable methods to learn semantic structures hidden in huge text collections is an urgent task.

C2. Fast inference of posteriors: The key concern when developing topic models (and probabilistic graphical models in general) is posterior distributions. The posteriors of wide interests are topic mixtures (latent representations). A topic mixture of a document shows which topics appear and how significant they are in the document. Therefore, topic mixtures are valuable sources for various applications such as thematic exploration of individual documents, information retrieval, and text classification. The posteriors provide us many advantages for making inference. In particular, posterior estimation plays a crucial role in many learning algorithms since it is often the core step [21, 48, 100, 129, 130]. As a result, a scalable algorithm for posterior estimation would promise a significant progress for TM. A pessimistic information is that posterior estimation is often intractable for LDA and many topic models [9, 91].

C3. Sparse topic mixtures: As a natural property of texts, a document may exhibit only few topics among infinite number of topics. This implies that topic mixtures for documents should be sparse —most elements are zeros. Ideally, an inference method should accurately respect this nature. However, many inference methods such as variational Bayesian [21] and Gibbs sampling [44] are unlikely to recover sparse topic mixtures, meaning that they seem not to obey the nature that each document contains only few topics. For those reasons, it is an open task to develop methods that are able to accurately recover sparse topic mixtures for documents.

C4. Theoretical guarantee: The TM literature has seen a flourish of development over the last two decades. However, there remain many open theoretical aspects. First, the quality of doing inference for a specific document is often unknown and is not theoretically guaranteed. Popular methods (e.g. variational Bayesian and Gibbs sampling) are empirically observed to do inference well, but lack a guarantee on quality. Second, the quality of the models learned is often not known. Some recent results [7, 9, 10] are very optimistic, but are limited to some restricted models. A large number of models have no guarantee on recovery quality, which as a consequence may cause some concerns when employed in other fields.

C5. Scalability in nonparametric models: The ability of nonparametric models to automatically grow their complexity with the data size is appealing. Nevertheless, it comes with the cost of complication to design efficient algorithms for learning or estimating posteriors [23, 42]. That may be the main reason of why existing learning methods often have high computational complexity. Compared with parametric models, methods for nonparametric counterparts are often much more time-consuming. Hence working with collections with millions/billions of documents is still challenging for the nonparametric approach.

C6. Visualization and user interface: When exploring a large collection of documents, it is very necessary to develop tools that enable easy navigation and interaction with users [15]. Topic models provides a new way to explore our data, at the semantic level. However, organizing intuitively a large number of semantic components in a navigator/screen would be itself challenging.

1.4 Overview of contributions

This thesis studies to model texts at a large scale. In other words, the thesis studies to propose models that most appropriately generate documents, and then to derive efficient methods for learning those models from a large number of available texts. To this end, the thesis systematically elucidates the two fundamental issues to be resolved: *inference of topic mixtures* and *model complexity*. The thesis then targets at seeking provably fast algorithms that can recover sparse topic mixtures for documents, and seeking fast algorithms to learn sparse topic models.

The first contribution is the introduction of a simple framework for inference of topic mixtures, called FW, which is general and flexible enough to be employed in admixture models. The framework enjoys the following key theoretical properties: (1) inference converges at a linear rate to the optimal solutions; (2) prior knowledge can be easily incorporated into inference; (3) the sparsity level of topic mixtures can be directly controlled; (4) it is easy to trade off sparsity against quality and runtime. Existing inference methods do not own these properties and often work slowly. Because of those attractive properties, the proposed framework provides a good answer to the three challenges C1, C2, and C3 as discussed in the last section.

We demonstrate goodness and flexibility of FW by employing it to design novel methods for *supervised dimension reduction* (SDR). When working with very high dimensional problem, it is sometimes beneficial in efficiency and effectiveness to reduce the dimensionality of the problem, but keep or make better predictiveness of the response variable. The main result of this study is a novel method that can reach state-of-the-art performance while enjoying 30-450 times faster speed than existing methods for SDR.

The second contribution is the introduction of *Fully Sparse Topic Model* (FSTM) for modeling large collections of documents. Three key properties of the model are: (i) the inference algorithm converges at a linear rate to the optimal solutions, (ii) it provides a principled way to directly trade off sparsity of solutions against inference quality and running time, (iii) the learning algorithm has low complexity which is near independent

of dimensionality. FSTM overcomes many limitations of existing topic models, and has been demonstrated to work qualitatively on real data. The low computational complexity and low model complexity can help us work with large text collections.

The third contribution is the introduction of a fast algorithm for learning *Correlated Topic Models* (CTM), as well as a theory of *probable convexity* for analyzing convexity of real functions. Modeling the interactions of hidden topics implies that we have to model two levels of unknown factors (i.e. topics and their interactions). Hence derivation of an efficient method for learning requires nontrivial efforts. Previous studies show that posterior inference in nonconjugate models such as CTM is intractable (NP-hard) in the worse case. However, we show that it may not be true in practice. Indeed, by introducing the concept of *probable convexity*, we show that inference of topic mixtures in CTM and many nonconjugate models is tractable in practice. Based on these findings, a novel algorithm is proposed which is surprisingly simple but is easily parallelizable or distributable. By extensive experiments, the algorithm is shown to work significantly faster than existing expensive methods while keeping comparable or better quality of the learned models.

As a minor contribution, the thesis introduces a new topic model in the Appendix where variational methods are used to do fast inference.

Last but not least, the thesis contributes to the public some scalable implementations of the developed models and methods. The codes are freely available at `www.jaist.ac.jp/~s1060203/codes.htm`

1.5 Organization

The thesis is organized as follows. After presenting some necessary backgrounds in the next chapter, the first contribution is presented in Chapter 3. Chapter 4 introduces the new model FSTM and detailed analysis of its properties. Chapter 5 presents the third contribution, the concept of probable convexity, analysis of nonconjugate topic models, and a fast algorithm for learning CTM. Chapter 6 summarizes the main contributions, open problems and future research.

Chapter 2

Backgrounds

2.1 Notation

Throughout the thesis, we use the following conventions and notations. Bold faces denote vectors or matrices. x_i denotes the ith element of vector $\boldsymbol{x}$, and A_{ij} denotes the element at row i and column j of matrix $\boldsymbol{A}$. Notation $\boldsymbol{A} \leq 0$ means that matrix $\boldsymbol{A}$ is negative semidefinite. For a given vector $\boldsymbol{x} = (x_1, ..., x_V)^t$, we denote $\frac{1}{\boldsymbol{x}} = (\frac{1}{x_1}, ..., \frac{1}{x_V})^t$, $\log \boldsymbol{x} = (\log x_1, ..., \log x_V)^t$, and $\log \tilde{\boldsymbol{x}} = (\log \frac{x_1}{x_V}, ..., \log \frac{x_{V-1}}{x_V})^t$. $diag(\boldsymbol{x})$ denotes the diagonal matrix whose diagonal entries are $x_1, ..., x_V$, respectively. More notations are:

- $\mathcal{V}$: vocabulary of V terms, often written as $\{1, 2, ..., V\}$.
- $\boldsymbol{d}$: a document represented as a count vector of V dimensions, $\boldsymbol{d} = (d_1, d_2, ..., d_V)$ where d_j is the frequency of term j.
- I_d: set of terms that appear in document $\boldsymbol{d}$, i.e., $I_d = \{j : d_j \neq 0\}$.
- $\mathcal{C}$: a corpus consisting of M documents, $\{\boldsymbol{d}_1, ..., \boldsymbol{d}_M\}$.
- K: number of topics.
- $\boldsymbol{\beta}_k$: a topic which is a distribution over the vocabulary $\mathcal{V}$. It is written as $\boldsymbol{\beta}_k = (\beta_{k1}, ..., \beta_{kV})^t$, where $\beta_{kj} \geq 0, \sum_{j=1}^{V} \beta_{kj} = 1$.
- $\mathbb{R}^K$: the K-dimensional Euclidean space.
- $\mathbb{E}$: the expectation of a random variable.
- Δ_K: the unit simplex in the K-dimensional space, $\Delta_K = \{\boldsymbol{x} \in \mathbb{R}^K : \sum_{k=1}^{K} x_k = 1, x_j \geq 0, \forall j\}$.
- $\overline{\Delta}_K$: the interior of Δ_K, that is $\overline{\Delta}_K = \{\boldsymbol{x} \in \mathbb{R}^K : \sum_{k=1}^{K} x_k = 1, x_j > 0, \forall j\}$.
- $\boldsymbol{e}_i$: the ith unit vector in the Euclidean space, i.e, $e_{ii} = 1$ and $e_{ij} = 0, \forall j \neq i$.

$\exp x$:	denotes e^x.
$\log x$:	the natural logarithm of x such that $\log e^x = x$.
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$:	the multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.
$Mult(\boldsymbol{x})$:	the multinomial distribution.
$x \sim \mathcal{A}(\cdot)$:	means that the random variable x follows the distribution $\mathcal{A}(\cdot)$.
Υ:	denotes the parameters $\Upsilon = \{\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}\}$.
$\mathrm{Tr}\,\boldsymbol{A}$:	the trace of matrix $\boldsymbol{A}$.
$\lambda_i(\boldsymbol{A})$	the ith largest eigenvalue of matrix $\boldsymbol{A}$.
$\mathbb{S}^K$:	the set of all symmetric matrix of size $K \times K$.
$\mathbb{S}^K_+$:	the set of all positive definite matrices of $\mathbb{S}^K$.
∇f or f':	the gradient (first partial derivative) of the given function f.
f'':	the Hessian matrix (second partial derivative) of the given function f.
$n!$:	the factorial of the positive integer n. That is $n! = 1.2.3....(n-1)n$.
$\det \boldsymbol{A}$:	the determinant of the square matrix $\boldsymbol{A}$.

2.2 Topic model

Loosely speaking, a topic is a set of semantically related words [58]. For examples, {*computer, information, software, memory, database*} is a topic about "computer"; {*jazz, instrument, music, clarinet*} may refer to "instruments for Jazz"; and {*caesar, pompay, roman, rome, carthage, crassus*} may refer to a battle in history.

Formally, we define a topic to be a distribution over a fixed vocabulary. Let $\mathcal{V}$ be the vocabulary of V terms, a topic $\boldsymbol{\beta}_k = (\beta_{k1}, ..., \beta_{kV})$ satisfies $\sum_{i=1}^{V} \beta_{ki} = 1$ and $\beta_{ki} \geq 0$ for any i. Each component β_{ki} shows the probability that term i contributes to topic k. A *topic model* is a model of the semantic structure (including topics) hidden in documents.

Each document is often assumed to be a mixture of the topics. In other words, a document is assumed to be composed from some topics with different proportions. Hence each document will have another representation, says $\boldsymbol{\theta} = (\theta_1, ..., \theta_K)$ where θ_k shows the probability that topic k appears in that document. $\boldsymbol{\theta}$ is often called *topic proportion*.

The goal of topic modeling is to automatically discover the topics from a collection of documents [15]. In reality, we can only observe the documents, while the *topic structure* including topics and topic proportions is *hidden*. The central problem for topic modeling is to use the observed documents to infer the topic structure.

Definition 1 (Topic mixture)**.** *Consider a topic model $\mathfrak{M}$ with K topics. Each document $\boldsymbol{d}$ will be represented by $\boldsymbol{\theta} = (\theta_1, ..., \theta_K)^t \in \Delta_K$, where θ_k indicates the proportion that topic*

k contributes to $\boldsymbol{d}$*.* $\boldsymbol{\theta}$ *is called* topic mixture *(or* topic proportion *or* latent representation*) of* $\boldsymbol{d}$*.*

Inference essentially refers to the process of making inference for a new document, given a topic model. The aim of inference may vary according to the specific task we are concerning on. Since topic mixtures contain valuable information about documents and have various applications, we will focus on inferring topic mixtures for documents from now on.

Definition 2 (ML Inference)**.** *Consider a topic model* $\mathfrak{M}$*, and a given document* $\boldsymbol{d}$*. The ML inference problem is to find the topic mixture* $\boldsymbol{\theta}$ *that maximizes the likelihood* $P(\boldsymbol{d}|\boldsymbol{\theta}, \mathfrak{M})$*.*

Definition 3 (MAP Inference)**.** *Consider a topic model* $\mathfrak{M}$*, and a given document* $\boldsymbol{d}$*. The MAP inference problem is to find the topic mixture* $\boldsymbol{\theta}$ *that maximizes the posterior probability* $P(\boldsymbol{\theta}|\boldsymbol{d}, \mathfrak{M})$*.*

Note that topic mixtures are hidden in documents, and are defined as probability distributions over topics. Hence ML inference and MAP inference are actually *posterior estimation problems.*

Learning a topic model is the problem of estimating the parameters of the model from a training corpus. Various schemes are employed for learning such as variational methods [21], Gibbs sampling [44], and stochastic Gibbs sampling [67]. Hidden topics are often of wide interests for the aim of understanding and exploring corpora.

When discussing about sparsity of a vector $\boldsymbol{x}$, various interpretations can be made. The most popular interpretation is that $\boldsymbol{x}$ is called sparse if many of its elements are 0. We will use this interpretation throughout this thesis. Based on this interpretation, we define concepts of sparsity for documents and topics as follows.

Definition 4 (Document sparsity)**.** *Consider a topic model* $\mathfrak{M}$ *with* K *topics, and a corpus* $\mathcal{C}$ *with* M *documents. Let* $\boldsymbol{\theta}_m$ *be the topic proportion of document* $\boldsymbol{d}_m \in \mathcal{C}$*. Then the* document sparsity *of* $\mathcal{C}$ *under the model* $\mathfrak{M}$ *is defined as the proportion of non-zero entries of the new representation of* $\mathcal{C}$*, i.e.,*

$$\textit{document sparsity} = \frac{\#\textit{non-zeros of } (\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_M)}{M.K}.$$

Definition 5 (Topic sparsity)**.** *Consider a topic model* $\mathfrak{M}$ *with* K *topics* $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K)$*.*

Algorithm 1 Frank-Wolfe algorithm

Input: objective function $f(\boldsymbol{\theta})$.
Output: $\boldsymbol{\theta}$ that maximizes $f(\boldsymbol{\theta})$ over Δ_K.
Pick as $\boldsymbol{\theta}_0$ the vertex of Δ_K with largest f value.
for $\ell = 0, ..., \infty$ **do**
  $i' := \arg\max_i \nabla f(\boldsymbol{\theta}_\ell)_i$;
  $\alpha' := \arg\max_{\alpha \in [0,1]} f(\alpha \boldsymbol{e}_{i'} + (1-\alpha)\boldsymbol{\theta}_\ell)$;
  $\boldsymbol{\theta}_{\ell+1} := \alpha' \boldsymbol{e}_{i'} + (1-\alpha')\boldsymbol{\theta}_\ell$.
end for

Topic sparsity *of* $\mathfrak{M}$ *is defined as the proportion of non-zero entries in* $\boldsymbol{\beta}$*, i.e.,*

$$\textit{topic sparsity} = \frac{\#\textit{non-zeros of } \boldsymbol{\beta}}{V.K}.$$

2.3 Concave maximization over simplex and sparse approximation

Consider a concave function $f(\boldsymbol{\theta}) : \mathbb{R}^K \to \mathbb{R}$ which is twice differentiable over the unit simplex Δ_K. We are interested in the following problem, *concave maximization over simplex*,

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \Delta_K} f(\boldsymbol{\theta}) \tag{2.1}$$

Convex optimization has been extensively studied in the optimization literature. There has been various excellent results such as [57, 70]. However, we are interested in sparse approximation algorithms specialized for problem (2.1). More specifically, we focus on the Frank-Wolfe algorithm [28].

Loosely speaking, the Frank-Wolfe algorithm is an approximation one for problem (2.1). Starting from a vertex of the simplex Δ_K, it iteratively selects the most potential vertex of Δ_K to change the current solution closer to that vertex in order to maximize $f(\boldsymbol{\theta})$. Details are presented in Algorithm 1 for optimization over the unit simplex Δ_K. It has been shown that the algorithm converges at a linear rate to the optimal solution. Moreover, at each iteration, the algorithm finds a provably good approximate solution lying in a face of Δ_K.

Theorem 1. *[28] Let f be a continuously differentiable, concave function over Δ_K, and denote C_f be the largest constant so that $f(\alpha \boldsymbol{x}' + (1-\alpha)\boldsymbol{x}) \geq f(\boldsymbol{x}) + \alpha(\boldsymbol{x}' - \boldsymbol{x})^t \nabla f(\boldsymbol{x}) -$*

$\alpha^2 C_f, \forall \boldsymbol{x}, \boldsymbol{x}' \in \Delta_K$, $\alpha \in [0,1]$. *After ℓ iterations, the Frank-Wolfe algorithm finds a point $\boldsymbol{\theta}_\ell$ on an $(\ell+1)-$dimensional face of Δ_K such that*

$$\max_{\boldsymbol{\theta} \in \Delta_K} f(\boldsymbol{\theta}) - f(\boldsymbol{\theta}_\ell) \leq \frac{4C_f}{(\ell+3)}. \tag{2.2}$$

It is worth noting some observations about the algorithm:

- It achieves a linear rate of convergence, and has provably bounds on the goodness of approximate solutions. These are crucial for practical applications.
- Overall running time mostly depends on how complicated f and ∇f are.
- It provides an explicit bound on the dimensionality of the face of Δ_K in which an approximate solution lies. After ℓ iterations, $\boldsymbol{\theta}_\ell$ is a convex combination of at most $\ell+1$ vertices of Δ_K, i.e., at most $\ell+1$ out of K components of $\boldsymbol{\theta}_\ell$ are non-zero. This implies that we can find an approximate solution to the problem (2.1) which are sparse.
- It is easy to directly control the sparsity level of such approximate solutions by trading off sparsity against quality. The fewer the number of iterations, the sparser the solution. This characteristic makes the algorithm more attractive for resolving high dimensional problems.
- It is possible to accelerate the speed, but keep convergence rate of the algorithm. In Algorithm 1, we have to repeatedly search for auxiliary variable α. Those searches can be avoided by selecting a sequence of predefined values as suggested by [28]. We can choose $\alpha := 2/(\ell+3)$ at iteration ℓ. Such a choice is capable of maintaining the bound on approximation error (2.2), but reducing computations considerably.

Algorithm 1 presents the Frank-Wolfe algorithm in simple form for optimization over the unit simplex. In fact, the algorithm is very general so that it can be extended easily to the case that the domain is a simplex (convex hull). Indeed, let $\boldsymbol{b}_1, ..., \boldsymbol{b}_K$ be vectors in $\mathbb{R}^V$ and $\Delta = conv(\boldsymbol{b}_1, ..., \boldsymbol{b}_K)$ be the convex hull of those vectors. Then the problem $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \Delta} f(\boldsymbol{x})$ can be solved as follows: Starting from a vertex of the simplex Δ, iteratively selects the most potential vertex of Δ to change the current solution closer to that vertex in order to maximize $f(\boldsymbol{x})$. Results similar to Theorem 1 can be established.

Theorem 2. *[28] Let f be a twice differentiable concave function over Δ, and denote $C_f = -\frac{1}{2}\sup_{\boldsymbol{y},\boldsymbol{z} \in \Delta; \tilde{\boldsymbol{y}} \in [\boldsymbol{y},\boldsymbol{z}]} (\boldsymbol{y}-\boldsymbol{z})^t.\nabla^2 f(\tilde{\boldsymbol{y}}).(\boldsymbol{y}-\boldsymbol{z})$. After ℓ iterations, the Frank-Wolfe*

algorithm finds a point $\boldsymbol{x}_\ell$ on an $(\ell+1)-$dimensional face of Δ such that

$$\max_{\boldsymbol{x}\in\Delta} f(\boldsymbol{x}) - f(\boldsymbol{x}_\ell) \leq \frac{4C_f}{\ell+3}. \tag{2.3}$$

2.4 Random matrices

For a comprehensive introduction to matrix algebra and random matrices, we refer to the book by [1] and the surveys by Izenman [53], Tracy and Widom [104], Tropp [105], Vershynin [108]. Here we review some concepts and results that are most necessary for the following chapters.

A matrix $\boldsymbol{A}$ is positive semidefinite if and only if the least eigenvalue $\lambda_{\min}(\boldsymbol{A})$ is nonnegative. If $\boldsymbol{A}$ has K eigenvalues, its trace satisfies $\operatorname{Tr}\boldsymbol{A} = \sum_{i=1}^K A_{ii} = \sum_{i=1}^K \lambda_i(\boldsymbol{A})$. If $\boldsymbol{A}$ is a random matrix, we have trace-expectation relation $\operatorname{Tr}\mathbb{E}\boldsymbol{A} = \mathbb{E}(\operatorname{Tr}\boldsymbol{A})$.

Consider a function $f:\mathbb{R}\to\mathbb{R}$. We define a map on a diagonal matrix $\boldsymbol{A}$ as $f(\boldsymbol{A}) = diag(f(A_{11}),...,f(A_{KK}))$. Similarly, a function of a symmetric matrix $\boldsymbol{A}$ is defined by using the eigenvalue decomposition:

$$f(\boldsymbol{A}) = \boldsymbol{Q}.f(\boldsymbol{\Lambda}).\boldsymbol{Q}^t, \text{ where } \boldsymbol{A} = \boldsymbol{Q}.\boldsymbol{\Lambda}.\boldsymbol{Q}^t \text{ and } \boldsymbol{\Lambda} \text{ is a diagonal matrix.}$$

The ***spectral mapping theorem*** states that each eigenvalue of $f(\boldsymbol{A})$ is equal to $f(\lambda)$ for some eigenvalue λ of $\boldsymbol{A}$. If f is nondecreasing, then $\lambda_k(f(\boldsymbol{A})) = f(\lambda_k(\boldsymbol{A}))$ for any k whenever $\lambda_k(\boldsymbol{A})$ exists.

We will work with *matrix exponential* which is defined for an $\boldsymbol{A}\in\mathbb{S}^K$ by

$$e^{\boldsymbol{A}} = \sum_{i=0}^{\infty} \frac{\boldsymbol{A}^i}{i!}.$$

Note that $\lambda_k(e^{\boldsymbol{A}}) = e^{\lambda_k(\boldsymbol{A})}$ for any k provided that $\lambda_k(\boldsymbol{A})$ exists. The logarithm of a matrix $\boldsymbol{A}\in\mathbb{S}_+^K$ is a matrix, denoted by $\log\boldsymbol{A}$, such that $e^{\log\boldsymbol{A}} = \boldsymbol{A}$.

Theorem 3 (Golden-Thompson inequality)**.** *For $\boldsymbol{A},\boldsymbol{B}\in\mathbb{S}^K$, we have*

$$\operatorname{Tr} e^{\boldsymbol{A}+\boldsymbol{B}} \leq \operatorname{Tr}(e^{\boldsymbol{A}}.e^{\boldsymbol{B}}).$$

This is a standard result and can be found in [105, 119]. Note that $e^{\boldsymbol{A}}$ and $e^{\boldsymbol{B}}$ are

positive definite which implies $\operatorname{Tr}(e^{\boldsymbol{A}}.e^{\boldsymbol{B}}) \leq \operatorname{Tr} e^{\boldsymbol{A}}.\operatorname{Tr} e^{\boldsymbol{B}}$, since according to Yang and Feng [122], $\operatorname{Tr}(\boldsymbol{A}.\boldsymbol{B}) \leq \operatorname{Tr} \boldsymbol{A}.\operatorname{Tr} \boldsymbol{B}$ if $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{S}_+^K$. Hence we have the following.

Corollary 1. *For $\boldsymbol{A}, \boldsymbol{B} \in \mathbb{S}^K$, we have $\operatorname{Tr} e^{\boldsymbol{A}+\boldsymbol{B}} \leq \operatorname{Tr} e^{\boldsymbol{A}}.\operatorname{Tr} e^{\boldsymbol{B}}$.*

The next theorem was shown by Tropp [105].

Theorem 4 (Laplace transform method)**.** *Let $\boldsymbol{B}$ be a random matrix of $\mathbb{S}^K$. For any real t, we have*

$$\Pr(\lambda_1(\boldsymbol{B}) \geq t) \leq \inf_{a>0}\{e^{a.t}\mathbb{E}\operatorname{Tr} e^{a.\boldsymbol{B}}\}.$$

Lemma 1. *Consider a matrix $\boldsymbol{B} \in \mathbb{S}^K$ and a nonnegative real a. We have*

$$\mathbb{E}\operatorname{Tr} e^{a.\boldsymbol{B}} \leq K\mathbb{E}e^{a\lambda_1(B)}.$$

Proof. Since the trace of $\boldsymbol{B}$ equals the sum of its eigenvalues, we have $\operatorname{Tr}\boldsymbol{B} \leq K\lambda_1(\boldsymbol{B})$. Hence $\mathbb{E}\operatorname{Tr} e^{a.\boldsymbol{B}} \leq K\mathbb{E}\lambda_1(e^{a\boldsymbol{B}}) \leq K\mathbb{E}e^{\lambda_1(a\boldsymbol{B})} = K\mathbb{E}e^{a\lambda_1(\boldsymbol{B})}$, where the last inequality is derived by using the spectral mapping theorem. (Q.E.D.)

Chapter 3

Fast inference of sparse topic mixtures

This chapter addresses the posterior estimation of topic mixtures, which is one of the key problems when developing topic models. The framework introduced in this chapter was employed in various works including those in the subsequent chapters.

3.1 Introduction

We are interested in the two important problems relating to recovery of topic mixtures for documents:[1] *sparsity* and *time*. The sparsity problem is to infer sparse topic mixtures, while the second problem asks for an efficient algorithm for recovering topic mixtures. The topic mixture of a document shows which topics appear and how significantly each contributes to the document. This suggests that topic mixtures are valuable sources for doing many tasks such as understanding individual documents, information retrieval [51, 116, 117], dimensionality reduction [21], and text classification [21, 129, 130]. Further estimation of topic mixtures plays as the core step in many algorithms for learning topic models [21, 48, 51, 100, 129]. Therefore, these two problems have been attracting significant interest in recent years, because of their significant impacts and non-trivial nature.

Inference is an integral part of any topic models, and is often NP-hard [91]. Various methods for efficient inference have been proposed such as folding-in [51], *variational*

[1] We will interchange the use of topic mixture, latent representation, and topic proportion. Those terms are all about the $\boldsymbol{\theta}$ of a document.

Bayesian (VB) [21], *collapsed variational Bayesian* (CVB) [12, 99], *collapsed Gibbs sampling* (CGS) [44]. Sampling-based methods are guaranteed to converge to the underlying distributions, but with unknown rate. VB and CVB are much faster, and CVB0 [12] often performs the best. Although these inference methods are significant developments for topic models, they remain two common limitations. First, there has been no theoretical upper bound on convergence rate and approximation quality of inference. Second, the inferred latent representations of documents are often dense, which may consume considerable memory for storage.[2]

Previous researches that have attacked the sparsity problem can be categorized into two main directions. The first direction is probabilistic [120] for which probability distributions or stochastic processes are employed to control sparsity. The other direction is non-probabilistic for which regularization techniques are employed to induce sparsity [59, 89, 129]. Although those approaches have gained important successes, they suffer from some drawbacks. Indeed, the probabilistic approach often requires extension of core topic models to be more complex, thus complicating learning and inference. Meanwhile, the non-probabilistic one often changes the objective functions of inference to be non-smooth which complicates doing inference, and requires some more auxiliary parameters associated with regularization terms. Such parameters necessarily require us to do model selection to find an acceptable setting for a given dataset, which is sometimes expensive. Furthermore, a common limitation of these two approaches is that the sparsity level of the topic mixtures is a priori unpredictable, and cannot be directly controlled.

There is inherently a tension between sparsity and time in the previous inference approaches. Some approaches focusing on speeding up inference [12, 21, 99] often ignore the sparsity problem. The main reason may be that a zero contribution of a topic to a document is implicitly prohibited in some models, in which Dirichlet distributions [21] or logistic-normal distributions [18] are employed to model latent representations of documents. Meanwhile, the approaches to the sparsity problem often result in time-consuming methods, e.g., [59, 120].[3] Note that in many practical applications, e.g., information retrieval and computer vision, fast inference of sparse latent representations of documents is of substantial significance. Hence resolving this tension is necessary.

[2]Some attempts have been initiated to speed up inference time and to attack the sparsity problem for Gibbs sampling [67, 123]. Sparsity in those methods does not lie in the latent representations of documents, but lies in sufficient statistics of Gibbs samples. Two main limitations of those methods are that we cannot directly control the sparsity level of sufficient statistics, and that there has been no theory for inference quality and convergence rate.

[3]The model by Zhu and Xing [129] is an exception, for which inference is potentially fast. Nonetheless, their inference method cannot be applied to probabilistic topic models, since unnormalization of latent representations is required.

In this work, we make the contributions as follows:

- First, we resolve both problems in a unified way. Particularly, we introduce a simple framework for inference in topic models, called FW, which is general and flexible enough to be employed in mixture models. Our framework enjoys the following key theoretical properties: (1) inference converges at a linear rate to the optimal solutions; (2) prior knowledge can be easily incorporated into inference; (3) the sparsity level of topic mixtures can be directly controlled; (4) it is easy to trade off sparsity against quality and time. We would like to remark that the last two properties are unspecified for existing inference methods.[4]

- Second, we employ FW to design the *two-phase* framework for doing supervised dimension reduction (SDR). The framework is (i) general and flexible so that it can be easily adapted to unsupervised topic models, (ii) able to inherit scalability of unsupervised topic models, and (iii) can exploit well label information and local structure of data when searching for a new space. The main consequence of this study is an effective method for SDR, namely FSTM^c. From extensive experiments, we find that FSTM^c can reach the state-of-the-art performance while enjoying 30-450 times faster speed than existing methods for SDR.

Organization: We introduce the FW framework for inference in Section 3.2. We also discuss when inference by FW is equivalent to doing ML and MAP inference. Further, we briefly discuss how FW can be applied to PLSA and LDA. Section 3.3 describes our experiments to see practical behaviors of the FW framework. Section 3.4 describes the application of FW to designing effective algorithms for doing dimensionality reduction under the presence of supervised information (labels).

3.2 Framework for fast and sparse inference

Given a document $\boldsymbol{d}$, we would like to find a desired topic proportion $\boldsymbol{\theta}$ of $\boldsymbol{d}$. The latent representation $\boldsymbol{\theta}$ depends heavily on the objective of inference. The most popular objective is the likelihood of $\boldsymbol{d}$. In many situations, our objective may differ far from the likelihood

[4]Regularization techniques [103] provide a way to impose sparsity on latent representations, by adding a regularization term to the objective function $f(x)$ to get $g(x) = f(x) + \lambda h(x)$, where $h(x)$ plays a role as a regularization inducing sparsity. Increasing the parameter, λ, associated with the regularization term may result in sparser solutions. However, it is not always provably true. Further, one cannot a priori decide a desired number of non-zero components of a solution. Hence regularization techniques provide only an indirect control over sparsity. The same holds for the existing probabilistic inference approaches.

Algorithm 2 The FW framework

Input: document $\boldsymbol{d}$ and topics $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K$.
Output: latent representation $\boldsymbol{\theta}$.
Step 1: select an appropriate objective function $f(\boldsymbol{\theta})$ which is continuously differentiable, concave over Δ_K.
Step 2: maximize $f(\boldsymbol{\theta})$ over Δ_K by the Frank-Wolfe algorithm.

solely. One example is supervised dimension reduction for which the new representations should be discriminative, i.e, the new representation of a document should remain the most discriminative characteristics of the class to which the document belongs.

To serve various objectives of inference, we discuss the FW framework which is presented in Algorithm 2. Loosely speaking, to do inference for a given document $\boldsymbol{d}$, one first chooses an appropriate objective function $f(\boldsymbol{\theta})$ which is continuously differentiable, concave over the unit simplex Δ_K. Then one uses sparse approximation such as the Frank-Wolfe algorithm [28] to find topic proportion $\boldsymbol{\theta}$. This algorithm follows the greedy approach, and has been proven to converge at a linear rate to the optimal solutions (see subsection 2.3). Moreover, at each iteration, the algorithm finds a provably good approximate solution lying in a face of the simplex Δ_K.

As inherited from the Frank-Wolfe algorithm, FW has many interesting properties:

- Inference by FW achieves a linear rate of convergence, and has provably bounds on the goodness of approximate solutions. These are crucial for practical applications.
- There is an explicit bound on the dimensionality of the face of Δ_K in which an approximate solution lies. After ℓ iterations, $\boldsymbol{\theta}_\ell$ is a convex combination of at most $\ell + 1$ vertices of Δ_K, i.e., at most $\ell + 1$ out of K components of $\boldsymbol{\theta}_\ell$ are non-zero. This implies that we can find an approximate solution to the problem (2.1) which are sparse.
- It is easy to directly control the sparsity level of such approximate solutions by trading off sparsity against quality. The fewer the number of iterations, the sparser the solution. This characteristic makes FW more attractive for resolving high dimensional problems.

We would like to remark that the FW framework is very general and flexible. It can be readily modified in various ways. For example, one can replace the second step by using other approximation algorithms such as sequential greedy approximation [127] or forward basis selection [125]. In addition, the first step offers us flexibility to customize objectives of inference.

Perhaps, the most difficult step in the FW framework is to choose a suitable objective function which can serve our purpose well. Various ways can be considered, however we appeal to the following principle for probabilistic topic models:

$$f(\boldsymbol{\theta}) = L(\boldsymbol{d}|\boldsymbol{\theta}) + \lambda.h(\boldsymbol{\theta}), \tag{3.1}$$

where $L(\boldsymbol{d}|\boldsymbol{\theta})$ is the log likelihood function of a given document, and $h(\boldsymbol{\theta})$ is a function of the latent representation $\boldsymbol{\theta}$, λ is a constant. This principle in turn bears resemblance to regularization techniques [103] which are widely used for sparse modeling. In fact, this principle is implicitly employed in some existing inference methods such as folding-in [51] and VB [21], as shown later. We will discuss in details some applications of this principle to PLSA, LDA and other models in the next subsections. The following states some key properties of our framework for inference, which is a corollary of Theorem 1.

Corollary 2. *Consider a topic model with K topics, and a document $\boldsymbol{d}$. Let $f(\boldsymbol{\theta})$ be continuously differentiable, concave over the simplex Δ_K. Let C_f be defined as in Theorem 1. Then inference by FW converges to the optimal solution at a linear rate. In addition, after ℓ iterations, the inference error is at most $4C_f/(\ell+3)$, and the topic proportion $\boldsymbol{\theta}$ has at most $\ell+1$ non-zero components.*

Note that the convergence rate of inference by our framework is linear, i.e., $O(1/\ell)$. It is possible to speed up convergence rate to sub-linear if the Frank-Wolfe algorithm is replaced with forward basis selection [125]. In addition, if we do not want to work with derivatives ∇f, replacing the Frank-Wolfe algorithm by sequential greedy algorithm [127] is appropriate. Nonetheless, such extensions are left open for future research.

3.2.1 ML and MAP inference

Next we would like to discuss two of the most popular inference problems: ML inference where there is no explicit prior over topic proportions; and MAP inference where topic proportions are endowed with a prior distribution. Note that inference for PLSA is ML inference whereas that for LDA and CTM is MAP inference [91]. We will show how our framework is naturally applicable to ML and MAP inference. Besides, a suitable choice of the objective function implies that inference by the framework is in fact MAP inference.

Before making analysis in details, we make the following assumptions on topic models and corpus:

Bag-of-words assumption. the documents are represented as bag-of-words, meaning

that the order of words in documents is ignored.

Mixture assumption. we assume that the occurrences of words in document $\boldsymbol{d}$ follow the generative process, given topics $\boldsymbol{\beta}$, as:

- Generate topic mixture $\boldsymbol{\theta}$ from a distribution $\mathcal{A}$ (depending on a specific model)
- For the nth word in $\boldsymbol{d}$:
  1. Pick topic index z_n from the multinomial distribution $Mult(\boldsymbol{\theta})$.
  2. Generate the word w_n from the multinomial distribution $Mult(\boldsymbol{\beta}_{z_n})$.

Most existing topic models take these two assumptions into account. Examples include admixture models [18, 20, 21, 51, 80, 93, 129, 130] and nonparametric models [23, 98, 112, 120].

Lemma 2. *Consider a topic model with K topics $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K$, and a given document $\boldsymbol{d}$. The ML inference problem can be reformulated as the following concave maximization problem, over the simplex Δ_K:*

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta} \in \Delta_K} \sum_{j \in I_d} d_j \log \sum_{k=1}^{K} \theta_k \beta_{kj}. \tag{3.2}$$

Proof. Denote by $P(w_j|z_k) = \beta_{kj}$ the probability that the term w_j appears in topic k, and by $P(z_k|\boldsymbol{d}) = \theta_k$ the probability that topic k contributes to document $\boldsymbol{d}$. For a given document $\boldsymbol{d}$, the probability that a term w_j appears in $\boldsymbol{d}$ can be expressed as $P(w_j|\boldsymbol{d}, \boldsymbol{\theta}, \boldsymbol{\beta}) = \sum_{k=1}^{K} P(w_j|z_k)P(z_k|\boldsymbol{d}) = \sum_{k=1}^{K} \theta_k \beta_{kj}$. Hence the log likelihood of document $\boldsymbol{d}$ is

$$\log P(\boldsymbol{d}|\boldsymbol{\theta}, \boldsymbol{\beta}) = \log \prod_{j \in I_d} P(w_j|\boldsymbol{d}, \boldsymbol{\theta}, \boldsymbol{\beta})^{d_j} = \sum_{j \in I_d} d_j \log P(w_j|\boldsymbol{d}, \boldsymbol{\theta}, \boldsymbol{\beta}) = \sum_{j \in I_d} d_j \log \sum_{k=1}^{K} \theta_k \beta_{kj}.$$

Note that $\boldsymbol{\theta} \in \Delta_K$, since $\sum_k \theta_k = 1$, $\theta_k \geq 0, \forall k$. As a result, the inference task is in turn the problem of finding $\boldsymbol{\theta} \in \Delta_K$ that maximizes the objective function $\sum_{j \in I_d} d_j \log \sum_{k=1}^{K} \theta_k \beta_{kj}$.
(Q.E.D.)

This lemma tells us that $f(\boldsymbol{\theta}) = \sum_{j \in I_d} d_j \log \sum_{k=1}^{K} \theta_k \beta_{kj}$ is the objective of ML inference, which is concave w.r.t $\boldsymbol{\theta}$. So this objective follows the principle (3.1). For MAP inference we need an employment of Bayes' rule to see clearly the objective function.

Lemma 3. *Consider a topic model with K topics $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K$, in which topic proportions are assumed to be samples from a prior distribution. Assume further that the prior distribution belongs to an exponential family, parameterized by α, whose density function can*

be expressed as $p(\boldsymbol{\theta}|\alpha) \propto \exp(\alpha.t(\boldsymbol{\theta}) - G(\alpha))$. *Then the MAP inference problem of a given document* $\boldsymbol{d}$ *can be reformulated as*

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\Delta_K} \sum_{j\in I_d} d_j \log \sum_{k=1}^{K} \theta_k \beta_{kj} + \alpha.t(\boldsymbol{\theta}). \tag{3.3}$$

Proof. MAP inference is to maximize the posterior probability $P(\boldsymbol{\theta}|\boldsymbol{d}, \boldsymbol{\beta}, \alpha)$ given a document $\boldsymbol{d}$. Bayes' rule says that $P(\boldsymbol{\theta}|\boldsymbol{d}, \boldsymbol{\beta}, \alpha) = P(\boldsymbol{d}|\boldsymbol{\theta}, \boldsymbol{\beta})P(\boldsymbol{\theta}|\alpha)/P(\boldsymbol{d})$. Hence

$$\begin{aligned}
\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}\in\Delta_K} P(\boldsymbol{\theta}|\boldsymbol{d}, \boldsymbol{\beta}, \alpha) = \arg\max_{\boldsymbol{\theta}\in\Delta_K} \log P(\boldsymbol{\theta}|\boldsymbol{d}, \boldsymbol{\beta}, \alpha) \\
&= \arg\max_{\boldsymbol{\theta}\in\Delta_K} \log P(\boldsymbol{d}|\boldsymbol{\theta}, \boldsymbol{\beta}) + \log P(\boldsymbol{\theta}, \alpha) \\
&= \arg\max_{\boldsymbol{\theta}\in\Delta_K} \log P(\boldsymbol{d}|\boldsymbol{\theta}, \boldsymbol{\beta}) + \alpha.t(\boldsymbol{\theta}) - G(\alpha).
\end{aligned}$$

Ignoring constants and rewriting the likelihood complete the proof. (Q.E.D.)

Essentially, this lemma reveals that $f(\boldsymbol{\theta}) = \sum_{j\in I_d} d_j \log \sum_{k=1}^{K} \theta_k \beta_{kj} + \alpha.t(\boldsymbol{\theta})$ is the objective function of MAP inference, which is exactly of the form (3.1), where $t(\boldsymbol{\theta})$ is the sufficient statistics of the prior over $\boldsymbol{\theta}$. However such a function is not always concave. An example is the MAP inference in LDA for which $\alpha.t(\boldsymbol{\theta}) = \sum_{k=1}^{K}(\alpha_k - 1)\log\theta_k$ is not concave if $\alpha < 1$, as noted before by [91]. We next show that with an appropriate choice of the objective function in the form (3.1), inference by FW is in fact MAP inference.

Theorem 5. *Consider a topic model* $\mathcal{M}$*, and a document* $\boldsymbol{d}$*. Let* $f(\boldsymbol{\theta}) = L(\boldsymbol{d}|\boldsymbol{\theta}) + \lambda.h(\boldsymbol{\theta})$*, where* $L(\boldsymbol{d}|\boldsymbol{\theta})$ *is the log likelihood of the document,* $h(\boldsymbol{\theta})$ *is a continuously differentiable, concave function over* Δ_K*,* $\lambda > 0$*. Then maximizing* $f(\boldsymbol{\theta})$ *over* Δ_K *is an MAP inference problem.*

Proof. Consider the marginal distribution of the random variable $\boldsymbol{\theta}$ whose density function is of the form $p(\boldsymbol{\theta}|\lambda) \propto \exp(\lambda.h(\boldsymbol{\theta}))$. Then

$$\begin{aligned}
\boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}\in\Delta_K} P(\boldsymbol{\theta}|\boldsymbol{d}, \mathcal{M}) = \arg\max_{\boldsymbol{\theta}\in\Delta_K} \log P(\boldsymbol{\theta}|\boldsymbol{d}, \mathcal{M}) \\
&= \arg\max_{\boldsymbol{\theta}\in\Delta_K} \log P(\boldsymbol{d}|\boldsymbol{\theta}, \mathcal{M}) + \log P(\boldsymbol{\theta}|\lambda) \\
&= \arg\max_{\boldsymbol{\theta}\in\Delta_K} \log P(\boldsymbol{d}|\boldsymbol{\theta}, \mathcal{M}) + \lambda.h(\boldsymbol{\theta}).
\end{aligned}$$

The objective of this optimization problem is exactly the function $f(\boldsymbol{\theta})$, completing the proof. (Q.E.D.)

3.2.2 Application to PLSA and LDA

We now discussed how FW can be adapted to the two of the most influential topic models, PLSA [51] and LDA [21]. Lemma 2 provides us a connection between ML inference and concave optimization. As a consequence, inference in PLSA can be reformulated as an *easy* optimization problem, and can be seamlessly resolved by FW. Combining this with Corollary 2, we obtain the following.

Corollary 3. *Consider PLSA with K topics, and a document $\boldsymbol{d}$. Then there exists an algorithm for inference that converges to the optimal solution at a linear rate, and that allows us to efficiently find a sparse topic proportion $\boldsymbol{\theta}$ with a guaranteed bound on inference error.*

Note that according to Lemma 2, the objective function of inference in PLSA is $f(\boldsymbol{\theta}) = \sum_{j \in I_d} d_j \log \sum_{k=1}^{K} \theta_k \beta_{kj}$. This objective turns out to be of the form (3.1) where $h(\boldsymbol{\theta}) \equiv 0$. It is easy to check that this function is continuously differentiable, concave over the simplex Δ_K if $\boldsymbol{\beta} > 0$. Hence, the Frank-Wolfe algorithm can be exploited for inference. One can handily do MAP inference for PLSA by modifying the objective function to be of the form (3.1). While MAP inference for PLSA has been studied by [89] and [59], their methods result in concave-convex objective functions and thus have no guaranteed bound for convergence.

We next turn our consideration to LDA [21]. It is known [91] that finding a topic proportion for a given document in LDA is an MAP inference problem, where the objective function is $f(\boldsymbol{x}) = \sum_{j \in I_d} d_j \log \sum_{k=1}^{K} \theta_k \beta_{kj} + \sum_{k=1}^{K} (\alpha_k - 1) \log \theta_k$. This objective is of the same form with (3.1), where $h(\boldsymbol{\theta}) = (\log \theta_1, ..., \log \theta_K)^t$ and $\lambda = (\alpha_1 - 1, ..., \alpha_K - 1)$. $h(\boldsymbol{\theta})$ and λ originally come from the Dirichlet prior over topic proportions. One can interpret $\lambda.h(\boldsymbol{\theta})$ to be a regularization term which induces *sparse* solutions for $\lambda < 1$. However, such a regularization does not always result in a concave objective function, and hence causes the inference in LDA to be NP-hard [91]. Furthermore, such a regularization requires all topics to have non-zero contributions to a specific document, since the function $\log \theta_k$ requires $\theta_k > 0$ to be well-defined. Hence, LDA cannot infer latent representations which are sparse in common sense.

To find sparse latent representations in LDA, some modifications are necessary. One can readily apply the FW framework to LDA where the objective is the log likelihood function. Other employments of the FW framework can yield MAP inference for LDA as suggested by Theorem 5. In those cases, it amounts to endowing new priors other than Dirichlet over topic proportions.

3.3 Empirical evaluation

In this section, we explore how well the FW framework works compared with existing inference methods. We first investigate some fundamental characteristics of FW, including sparsity of the inferred topic proportions, inference time, and inference quality. In addition to theoretical analysis and demonstration, we made a library for use in practice that is very easy for researchers/users to incorporate our framework into their customized models, just by writing their own objective functions. This may help substantially reduce complication and time for researchers when designing new topic models. The library is general enough to be applicable to inference in other literatures than topic modeling.[5]

The flexibility of the FW framework is evidenced by two specific applications. In the first one, we successfully develop *fully sparse topic models* (FSTM) [100] which is a simplified variant of PLSA and LDA. FSTM has been demonstrated to work well and has various attractive properties for dealing with large data. In the second application, we employ FW to design effective methods for supervised dimension reduction [101, 102] which will be described in the next section.

3.3.1 Time, sparsity, and quality

Analyses in the previous section have shown that inference by our framework is both fast and provably good, if provided a suitable choice of the objective function. In this section, we demonstrate empirically that even with the modest choice, say likelihood, our framework infers comparably well. Three inference methods were taken in comparison: Folding-in [51], Variational Bayesian [21], denoted by VB, and FW.[6] The objective function for FW is the log likelihood function. Five corpora were used in the investigation, of which some statistics are shown in Table 3.1.[7] For each corpus, we first trained the LDA model on the training part. We then did inference on the test set with the same criteria of convergence.[8]

[5]The library is freely available at `www.jaist.ac.jp/~s1060203/codes/FW/`

[6]CVB, CVB0, and CGS were not included for some reasons. CVB is often slower than VB [68]; CVB0 is faster than VB but works on documents which are not in bag-of-words representation; CGS is often slowest. Futhermore, these methods can achieve comparable quality as long as suitable parameter settings are chosen [12]. Hence VB is selected to be a representative.

[7]AP was retrieved from `http://www.cs.princeton.edu/~blei/lda-c/ap.tgz`.
KOS, NIPS, and Enron were from `http://archive.ics.uci.edu/ml/datasets/`.
Grolier was from `http://cs.nyu.edu/~roweis/data.html`

[8]At most 1000 iterations are allowed for inference, and the algorithm will converge if the relative change of the objective is less than 10^{-6}.

Table 3.1: Data for experiments.

Data	Training size	Testing size	#Terms
AP	2021	225	10473
KOS	3087	343	6906
NIPS	1350	150	12419
Grolier	23044	6718	15276
Enron	35875	3986	28102

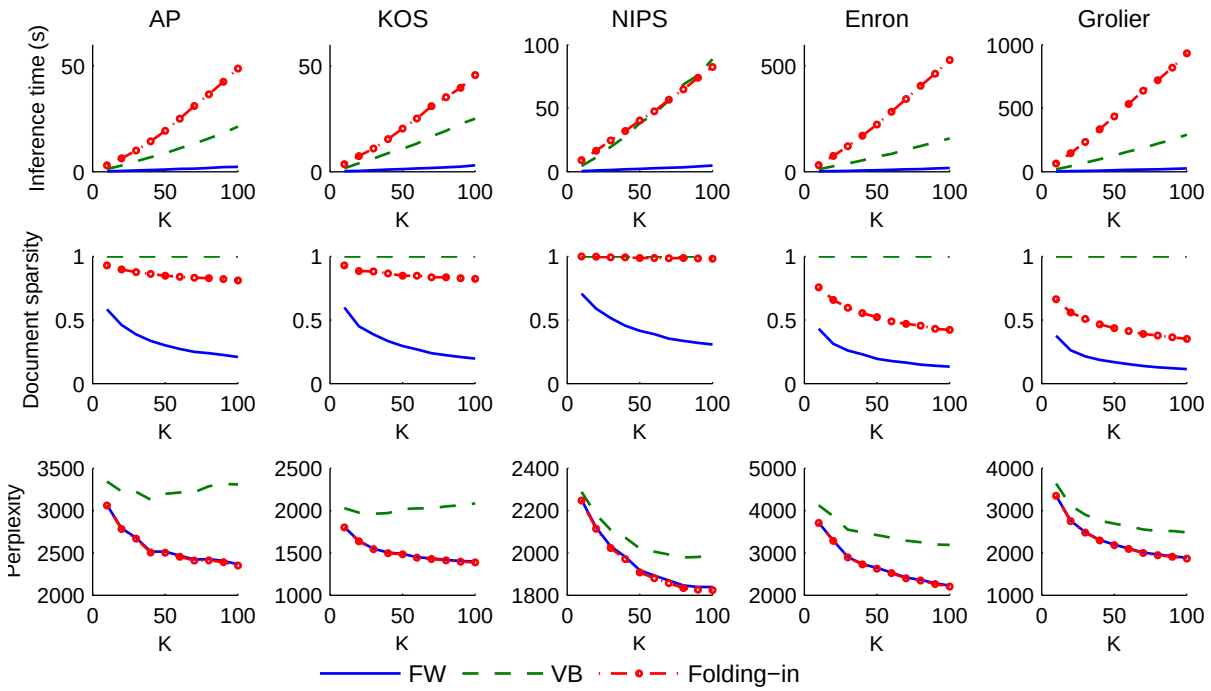


Figure 3.1: Comparison of inference methods as the number of topics increases. Lower is better.

Inference time: the first measure for comparison is inference time. Figure 3.1 depicts the results of inference on 5 corpora. We observe that Folding-in did slowest. VB did much more quickly than Folding-in. Each iteration of Folding-in took very few computations, much less than that of VB. However, VB often reached convergence in much less steps than Folding-in. That is why overall VB did more quickly. Compared with Folding-in and VB, our framework did inference significantly faster. FW often reached convergence in a few tens of iterations. Note that complexity of our framework heavily depends on how complicated the objective is. In this case, the objective is the log likelihood which needs few computations to be evaluated. One can realize that the inference time of FW was not quickly scaled up as the number of topics K increases, while VB and Folding-in increased much faster. This suggests that our framework is substantially more scalable than Folding-in and VB.

Document sparsity: we next consider how sparse the inferred topic proportions are.

Sparsity of a given document is the fraction of nonzero elements in the inferred latent representation. It is averaged for each test set, and is depicted in the second row of Figure 3.1. Note that inference by our framework always found very sparse topic proportions. The sparsity level increases as we model with more topics. Surprisingly, inference by Folding-in sometimes achieves sparse topic proportions. One possible reason is that Folding-in may inherit sparsity of original data, since inference by Folding-in simply does addition and multiplication on sparse data. Nevertheless, it is not always for Folding-in to achieve sparse solutions without a principled mechanism. Unsurprisingly, VB did not find any sparse latent representations of documents.

Perplexity: Corollary 2 suggests that inference by our framework theoretically finds provably good solutions. This theoretical result is further supported by experiments. The last row of Figure 3.1 shows the goodness of different inference methods in terms of perplexity [18, 21]. Loosely speaking, perplexity is the inverse of the geometric mean of the probabilities of words appearing in the testing documents, and is calculated on the testing set $\mathcal{D}$ by $Perplexity(\mathcal{D}) = \exp\left(-\sum_{\boldsymbol{d}\in\mathcal{D}} \log P(\boldsymbol{d}) / \sum_{\boldsymbol{d}\in\mathcal{D}} ||\boldsymbol{d}||_1\right)$. Observing Figure 3.1, we see that Folding-in and FW achieved comparably good predictive power. They performed much better than VB even though they were given the same models which had been trained before.

To explain this phenomenon, more thorough investigations are necessary. We observed that in all cases, LDA learned very small parameters α of the Dirichlet priors. Remember that when $\alpha < 1$, inference in LDA is NP-hard [91]. The NP-hardness may prevent the variational method from quickly inferring good solutions. This may be the main reason for the inferior performance of VB. Note further that inference in LDA is MAP inference, whose objective is different from the likelihood. But perplexity mainly relates to likelihood. Therefore, asynchronous objective functions for inference is another reason for inferior performance of VB in terms of perplexity.

Separability of documents in the topical space: topic models are often expected to provide us a soft clustering of documents in the space of topics, i.e., clustering documents into topical clusters. Hence we would like to see how well inference methods cluster the testing documents. A good method should cluster documents into topics *separately*. In other words, in the topical space, the documents should be separately clustered. To see this, we use the inferred latent representations of documents, and visualize the first 3 dimensions. Figure 3.2 shows the distribution of documents in the topical space. One can observe that the documents projected by VB spread around the axes, and they were not separated clearly into clusters. Similar phenomenon can be observed for Folding-in. Meanwhile, when projected by FW, each document focused more on few topics, and

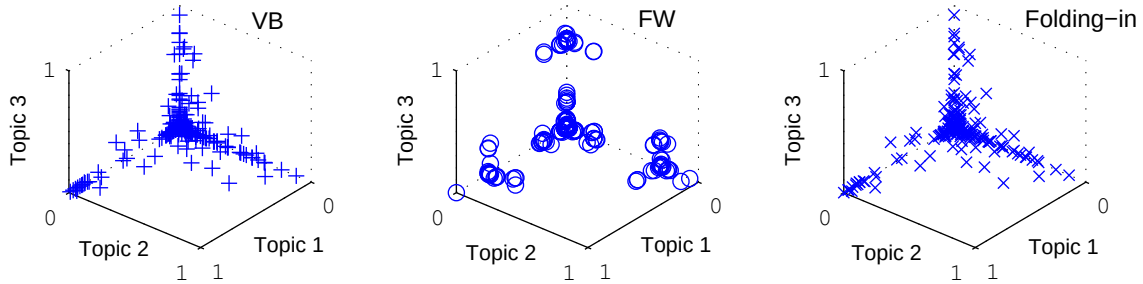


Figure 3.2: Separability of documents in the space of topics, inferred by different methods on AP with $K = 10$. Folding-in and VB do not provide separate clusters of documents. Meanwhile, FW separates documents explicitly into clusters associated with latent topics.

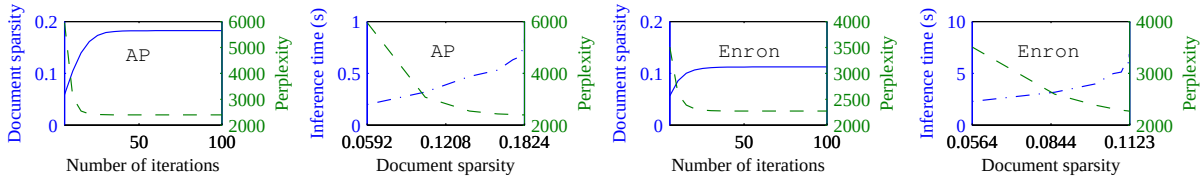


Figure 3.3: Illustration of trading off sparsity against time and quality. FW is able to reach convergence very quickly. After 20 iterations on average, its quality in terms of perplexity was almost stable, even though the number of topics is much larger ($K = 100$).

the documents were separated into clusters explicitly. We observed that inference by our framework often places very high probability on one topic, small probabilities on few more topics, and zero on others. This may be why, in the topical space, the documents are explicitly clustered. As a result, inference by our framework provides a better clustering of documents in the topical space.

3.3.2 Convergence rate and trade-off

When facing with large-scale settings including large corpora, extremely high dimensionality, and large number of topics, fast algorithms and compact storage demands are highly desired. Hence a principled way to trade off quality against time and storage requirement is sometimes necessary. Fortunately, the Frank-Wolfe algorithm can fulfill those desires for not only topic modeling but also other literatures. Indeed, it is provably fast and provides a simple way to decide the sparsity level of solutions, just by limiting the number of iterations.

We investigated further how quick FW reaches convergence in practice. The experiments were done with AP (small size) and Enron (average size), and on the learned LDA with $K = 100$ topics. Results are shown in Figure 3.3. One can realize that FW reached convergence very quickly. We found that in most cases, after 20 iterations on average the

quality was almost stable. Note that the dimension of the inference problem is $K = 100$ which is much larger than 20. The sparsity level of solutions got stable almost after 30 iterations. The same phenomenon was observed on other corpora. These facts suggest that FW can converge very quickly in practice despite of the loose bound in Corollary 2. This property is attractive for practical applications.

3.4 Application to supervised dimension reduction

In this section, we provide another evidence for the flexibility of our framework by encoding prior knowledge (or side information) into inference. In particular, we use FW to develop effective methods for supervised dimension reduction (SDR) for discrete data. This section only summarizes the key ideas and experimental results. For more detailed descriptions and analyses, we refer the readers to [102] (or [101] for brevity).

In SDR, we are asked to find a low-dimensional space which preserves the predictive information of the response variable. Projection on that space should keep the discrimination property of data in the original space. Existing methods for this problem often try to find directly a low-dimensional space that preserves separation of the data classes in the original space. For simplicity, we call that new space *discriminative space*.

3.4.1 The two-phase framework for SDR

We now describe our framework for SDR. Existing methods for this problem often try to find directly a low-dimensional space that preserves separation of the data classes in the original space. For simplicity, we call that new space *discriminative space*. Different approaches have been employed such as maximizing the conditional likelihood [56], minimizing the empirical loss by max-margin principle [130], or maximizing the joint likelihood of documents and labels [14]. Those are one-phase algorithms to find the discriminative space, and bear resemblance to existing methods for continuous data [77, 94]. Three remaining drawbacks are that learning is very slow, that scalability of unsupervised models is not appropriately exploited, and more seriously, the inherent local structure of data is not taken into consideration.

To overcome those limitations of supervised topic models, we propose a novel framework which consists of two phases. Loosely speaking, the first phase tries to find an initial topical space, while the second phase tries to utilize label information and local structure

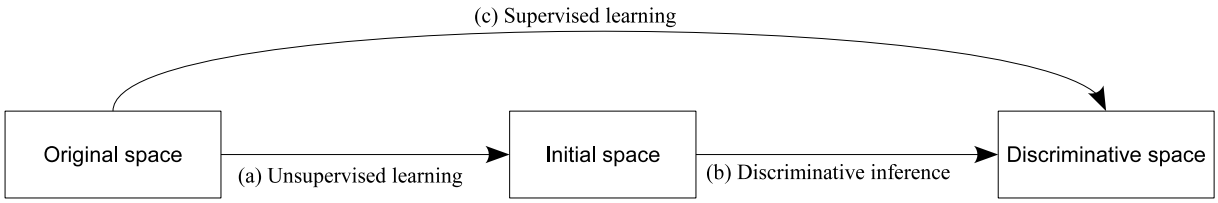


Figure 3.4: Sketch of approaches for SDR. Existing methods for SDR directly find the discriminative space, which is known as supervised learning (c). Our framework consists of two separate phases: (a) first find an initial space in an unsupervised manner; then (b) utilize label information and local structure of data to derive the final space.

of the training data to find the discriminative space. The first phase can be done by employing an unsupervised topic model [67, 100], and hence inherits scalability of unsupervised models. Label information and local structure in the form of neighborhood will be used to guide projection of documents onto the initial space, so that inner-class local structure is preserved and inter-class margin is widen. As a consequence, the discrimination property is not only preserved, but likely made better in the final space.

Figure 3.4 depicts graphically this framework, and a comparison with other one-phase methods. Note that we do not have to design entirely a learning algorithm as for existing approaches, but instead do one further inference phase for the training documents. Details of our framework are presented in Algorithm 3. Details for each step from (II.1) to (II.4) can be found in [101, 102].

3.4.2 Why is the framework good?

We next theoretically elucidate the main reasons for why our proposed framework is reasonable and can result in a good method for SDR. In our observations, the most important reason comes from the choice of the objective (3.4) for inference. Inference with that objective plays three crucial roles to preserve or make better the discrimination property of data in the topical space.

Preserving inner-class local structure

The first role is to preserve inner-class local structure of data. This is a result of using the additional term $\frac{1}{|N_d|}\sum_{\boldsymbol{d}'\in N_d} L(\widehat{\boldsymbol{d}'})$. Remember that projection of document $\boldsymbol{d}$ onto the unit simplex Δ is in fact a search for the point $\boldsymbol{\theta}_d \in \Delta$ that is closest to $\boldsymbol{d}$ in a certain

Algorithm 3 Two-phase framework for SDR

Phase I: learn an unsupervised model to get K topics $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K$.
Let $\mathfrak{A} = span\{\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K\}$ be the initial space.

Phase II: (finding discriminative space)

(II.1) for each class c, select a set S_c of topics that are potentially discriminative for c.

(II.2) for each document $\boldsymbol{d}$, select a set N_d of its nearest neighbors which are in the same class as $\boldsymbol{d}$.

(II.3) infer new representation $\boldsymbol{\theta}_d^*$ for each document $\boldsymbol{d}$ in class c using the Frank-Wolfe algorithm with the objective function

$$f(\boldsymbol{\theta}) = \lambda.L(\widehat{\boldsymbol{d}}) + (1-\lambda).\frac{1}{|N_d|}\sum_{\boldsymbol{d}' \in N_d} L(\widehat{\boldsymbol{d}'}) + R.\sum_{j \in S_c} \sin(\theta_j), \tag{3.4}$$

where $L(\widehat{\boldsymbol{d}})$ is the log likelihood of document $\widehat{\boldsymbol{d}} = \boldsymbol{d}/||\boldsymbol{d}||_1$; $\lambda \in [0,1]$ and R are nonnegative constants.

(II.4) compute new topics $\boldsymbol{\beta}_1^*, ..., \boldsymbol{\beta}_K^*$ from all $\boldsymbol{d}$ and $\boldsymbol{\theta}_d^*$. Finally, $\mathfrak{B} = span\{\boldsymbol{\beta}_1^*, ..., \boldsymbol{\beta}_K^*\}$ is the discriminative space.

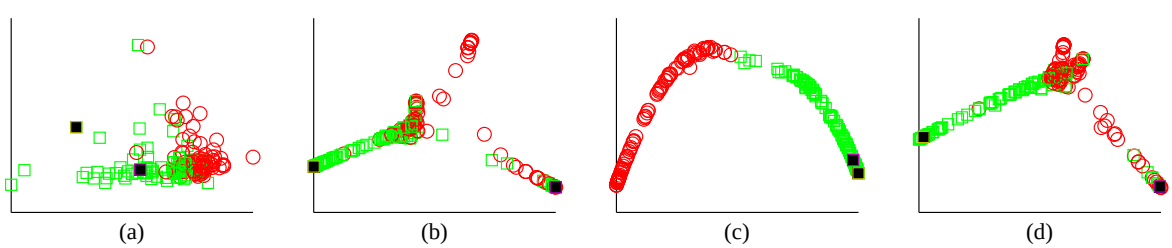


Figure 3.5: Laplacian embedding in 2D space. (a) data in the original space, (b) unsupervised projection, (c) projection when neighborhood is taken into account, (d) projection when topics are promoted. These projections onto the 60-dimensional space were done by FSTM and experimented on 20Newsgroups. The two black squares are documents in the same class.

sense.[9] Hence if $\boldsymbol{d}'$ is close to $\boldsymbol{d}$, it is natural to expect that $\boldsymbol{d}'$ is close to $\boldsymbol{\theta}_d$. To respect this nature and to keep the discrimination property, projecting a document should take its local neighborhood into account. As one can realize, the part $\lambda L(\widehat{\boldsymbol{d}}) + (1-\lambda)\frac{1}{|N_d|}\sum_{\boldsymbol{d}' \in N_d} L(\widehat{\boldsymbol{d}'})$ in the objective (3.4) serves well our needs. This part interplays goodness-of-fit and neighborhood preservation. Increasing λ means goodness-of-fit $L(\widehat{\boldsymbol{d}})$ can be improved, but local structure around $\boldsymbol{d}$ is prone to be broken in the low-dimensional space. Decreasing λ implies better preservation of local structure. Figure 3.5 demonstrates sharply these two extremes, $\lambda = 1$ for (b), and $\lambda = 0.1$ for (c). Projection by unsupervised models ($\lambda = 1$) often results in pretty overlapping classes in the topical space, whereas exploitation of local structure significantly helps us separate classes.

Since nearest neighbors N_d are selected within-class only, doing projection for $\boldsymbol{d}$ in step (II.3) is not intervened by documents from outside classes. Hence within-class local structure would be better preserved.

Widening the inter-class margin

The second role is to widen the inter-class margin, owing to the term $R\sum_{j \in S_c} \sin(\theta_j)$. As noted before, function $\sin(x)$ is monotonically increasing for $x \in [0, 1]$. It implies that the term $R\sum_{j \in S_c} \sin(\theta_j)$ promotes contributions of the topics in S_c when projecting document $\boldsymbol{d}$. In other words, the projection of $\boldsymbol{d}$ is encouraged to be close to the topics which are potentially discriminative for class c. Hence projection of class c is preferred to distributing around the discriminative topics of c. Increasing the constant R implies forcing projections to distribute more densely around the discriminative topics, and therefore making classes farther from each other. Figure 3.5(d) illustrates the benefit of this second role.

Reducing overlap between classes

The third role is to reduce overlap between classes, owing to the term $\lambda L(\widehat{\boldsymbol{d}}) + (1-\lambda)\frac{1}{|N_d|}\sum_{\boldsymbol{d}' \in N_d} L(\widehat{\boldsymbol{d}'})$ in the objective function (3.4). This is a very crucial role that helps the two-phase framework works effectively. Explanation for this role needs some insights into inference of $\boldsymbol{\theta}$.

In step (II.3), we have to do inference for the training documents. Let $\boldsymbol{u} = \lambda\widehat{\boldsymbol{d}} + (1-\lambda)\frac{1}{|N_d|}\sum_{\boldsymbol{d}' \in N_d} \widehat{\boldsymbol{d}'}$ be the convex combination of $\boldsymbol{d}$ and its within-class neighbors.[10] Note

[9] More precisely, the vector $\sum_k \theta_{dk}\boldsymbol{\beta}_k$ is closest to $\boldsymbol{d}$ in terms of KL divergence.

[10] More precisely, $\boldsymbol{u}$ is the convex combination of those documents in ℓ_1-normalized forms, since by notation $\widehat{\boldsymbol{d}} = \boldsymbol{d}/||\boldsymbol{d}||_1$.

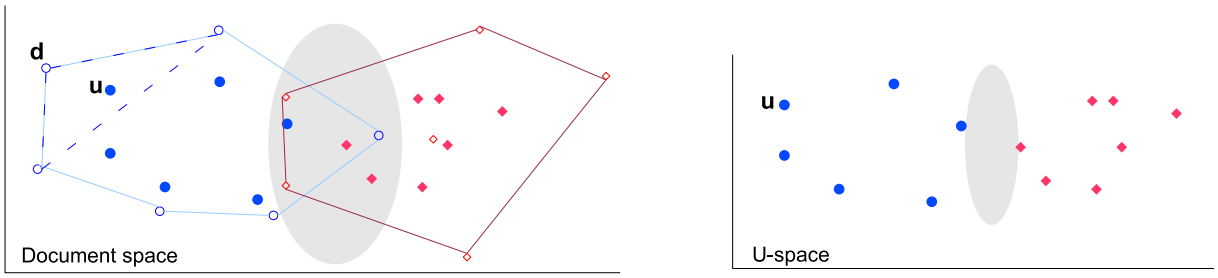


Figure 3.6: The effect of reducing overlap between classes. In Phase II, inferring $\boldsymbol{d}$ is reduced to inferring $\boldsymbol{u}$ which is the convex combination of $\boldsymbol{d}$ and its within-class neighbors. This means we are working in the U-space instead of the document space. Note that the classes in the U-space are often less overlapping than those in the document space.

that

$$\lambda L(\widehat{\boldsymbol{d}}) + (1-\lambda)\frac{1}{|N_d|}\sum_{\boldsymbol{d}'\in N_d} L(\widehat{\boldsymbol{d}'}) = \lambda\sum_{j=1}^{V}\widehat{d}_j \log\sum_{k=1}^{K}\theta_k\beta_{kj} + \frac{(1-\lambda)}{|N_d|}\sum_{\boldsymbol{d}'\in N_d}\sum_{j=1}^{V}\widehat{d'}_j \log\sum_{k=1}^{K}\theta_k\beta_{kj}$$
$$= \sum_{j=1}^{V}\left(\lambda\widehat{d}_j + (1-\lambda)\frac{1}{|N_d|}\sum_{\boldsymbol{d}'\in N_d}\widehat{d'}_j\right)\log\sum_{k=1}^{K}\theta_k\beta_{kj} = L(\boldsymbol{u}).$$

Hence, in fact we do inference for $\boldsymbol{u}$ by maximizing $f(\boldsymbol{\theta}) = L(\boldsymbol{u}) + R.\sum_{j\in S_c}\sin(\theta_j)$. It implies that we actually work with $\boldsymbol{u}$ in the U-space as depicted in Figure 3.6.

Those observations suggest that instead of working with the original documents in the document space, we do work with $\{\boldsymbol{u}_1, ..., \boldsymbol{u}_M\}$ in the U-space. Figure 3.6 shows that the classes in the U-space is less overlapping than those in the document space. Further, the overlap can sometimes be removed. Hence working in the U-space would be probably more effective than in the document space, in the sense of supervised dimension reduction.

3.4.3 Evaluation

This section is dedicated to investigation of effectiveness and efficiency of the two-phase framework in practice. We investigate three methods, PLSA^c, LDA^c, and FSTM^c, which are the results of adapting our framework to unsupervised models, PLSA [51], LDA [21], and FSTM [100], respectively. MedLDA [130] is taken as the state-of-the-art method for SDR into comparison.[11] We use 10 benchmark datasets for investigation which span over various domains including news in LA Times, biological articles, spam emails. Table 3.2

[11]MedLDA was retrieved from `www.ml-thu.net/~jun/code/MedLDAc/medlda.zip`
LDA was taken from `www.cs.princeton.edu/~blei/lda-c/`
FSTM was taken from `www.jaist.ac.jp/~s1060203/codes/fstm/`
PLSA was written by ourselves with the best effort.

Table 3.2: Statistics of data for experiments

Data	Training size	Testing size	Dimensions	Classes
LA1s	2566	638	13196	6
LA2s	2462	613	12433	6
News3s	7663	1895	26833	44
OH0	805	198	3183	10
OH5	739	179	3013	10
OH10	842	208	3239	10
OH15	735	178	3101	10
OHscal	8934	2228	11466	10
20Newsgroups	15935	3993	62061	20
Emailspam	3461	866	38729	2

shows some information about those data.[12]

In our experiments, we used the same criteria for topic models: relative improvement of the log likelihood (or objective function) is less than 10^{-4} for learning, and 10^{-6} for inference; at most 1000 iterations are allowed to do inference. The same criterion was used to do inference by the Frank-Wolfe algorithm in Phase 2 of our framework. MedLDA is a supervised topic model and is trained by minimizing a hinge loss. We used the best setting as studied by [130] for some other parameters: cost parameter $\ell = 32$, and 10-fold cross-validation for finding the best choice of the regularization constant C in MedLDA. These settings are chosen to avoid a possibly biased comparison.

It is worth noting that the two-phase framework plays the main role in searching for the discriminative space $\mathfrak{B}$. Hence, other works aftermath such as projection/inference new documents are done by unsupervised models. For instances, FSTM^c works as follows: we first train FSTM in an unsupervised manner to get an initial space $\mathfrak{A}$; we next do Phase 2 of Algorithm 3 to find the discriminative space $\mathfrak{B}$; projection of documents onto $\mathfrak{B}$ then is done by the inference method of FSTM which does not need label information.

Class separation and classification quality

Separation of classes in low-dimensional spaces is our first concern. A good method for SDR should preserve inter-class separation of data in the original space. Figure 3.7

[12] 20Newsgroups was taken from `www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`. Emailspam was taken from `csmining.org/index.php/spam-email-datasets-.html`. Other datasets were retrieved from the UCI repository.

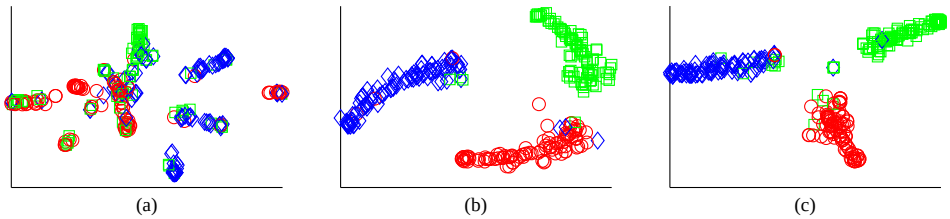


Figure 3.7: Projection of three classes of 20newsgroups onto the topical space by (a) FSTM, (b) FSTM^c, and (c) MedLDA. FSTM did not provide a good projection in the sense of class separation, since label information was ignored. FSTM^c and MedLDA actually found good discriminative topical spaces, and provided a good separation of classes. These embeddings were done with t-SNE [107].

depicts an illustration of how good different methods are. In this experiment, 60 topics were used to train FSTM and MedLDA.[13] One can observe that projection by FSTM can maintain separation between classes to some extent. Nonetheless, because of ignoring label information, a large number of documents have been projected onto incorrect classes. On the contrary, FSTM^c and MedLDA exploited seriously label information for projection, and hence the classes in the topical space separate very cleanly. The good preservation of class separation by MedLDA is mainly due to the training algorithm by max margin principle. Each iteration of the algorithm tries to widen the expected margin between classes. Hence such an algorithm implicitly inherits the discrimination property in the topical space. FSTM^c can separate the classes well owing to the fact that projecting documents has taken local neighborhood into account seriously, which very likely keeps inter-class separation of the original data. Furthermore, it also tries to widen the margin between classes as discussed in Section 3.4.2.

Classification quality: we next use classification as a means to quantify the goodness of the considered methods. The main role of methods for SDR is to find a low-dimensional space so that projection of data onto that space preserves or even makes better the discrimination property of data in the original space. In other words, predictiveness of the response variable is preserved or improved. Classification is a good way to see this preservation or improvement.

For each method, we projected the training and testing data ($\boldsymbol{d}$) onto the topical space, and then used the associated projections ($\boldsymbol{\theta}$) as inputs for multi-class SVM [54] to do classification.[14] MedLDA does not need to be followed by SVM since it can do

[13]For our framework, we set $N_d = 20, \lambda = 0.1, R = 1000$. This setting basically says that local neighborhood plays a heavy role when projecting documents, and that classes are very encouraged to be far from each other in the topical space.

[14]This classification method is included in Liblinear package which is available at `www.csie.ntu.edu.tw/~cjlin/liblinear/`

classification itself. Keeping the same setting as described before and varying the number of topics, the results are presented in Figure 3.8.

Observing the figure, one easily realizes that the supervised methods often performed substantially better than the unsupervised ones. This suggests that FSTM^c, LDA^c, and PLSA^c exploited well label information when searching for a topical space. FSTM^c, LDA^c, and PLSA^c performed better than MedLDA when the number of topics is relatively large (≥ 60). FSTM^c consistently achieved the best performance amongst topic-model-based methods, and sometimes reached 10% improvement over the-state-of-the-art MedLDA. In our observations, this improvement is mainly due to the fact that FSTM^c had taken seriously local structure of data into account whereas MedLDA did not.

There is a surprising behavior of MedLDA. Though being a supervised method, it performed comparably or even worse than unsupervised methods (PLSA, LDA, FSTM) for many datasets including LA1s, LA2s, OH10, and OHscal. In particular, MedLDA performed significantly worst for LA1s and LA2s. It seems that MedLDA lost considerable information when searching for a low-dimensional space. One of the main reasons for this surprising behavior could be that MedLDA ignores local structure. As evidenced by various researches, ignoring the inherent structure when searching for a topical space could harm or break the discrimination property of data. This could happen with MedLDA even though learning by max margin principle is well-known to keep good classification quality.

Learning time

The final measure for comparison is how quickly the methods do? We mostly concern on the methods for SDR including FSTM^c, LDA^c, PLSA^c, and MedLDA. Note that the time for learning a discriminative space by FSTM^c is the time to do 2 phases of Algorithm 3 which includes time to learn an unsupervised model, FSTM. The same holds for PLSA^c and LDA^c. Figure 3.9 summarizes the overall time for each method. Observing the figure, we find that MedLDA and LDA^c consumed intensive time, while FSTM^c and PLSA^c did substantially more speedily. One of the main reasons for slow learning of MedLDA and LDA^c is that inference by variational methods of MedLDA and LDA is often very slow. Inference in those models requires various evaluation of Digamma and gamma functions which are expensive. Further, MedLDA requires a further step of learning a classifier at each EM iteration, which is empirically slow in our observations. All of these contributed to the slow learning of MedLDA and LDA^c.

In contrast, FSTM has a linear time inference algorithm and requires simply a multi-

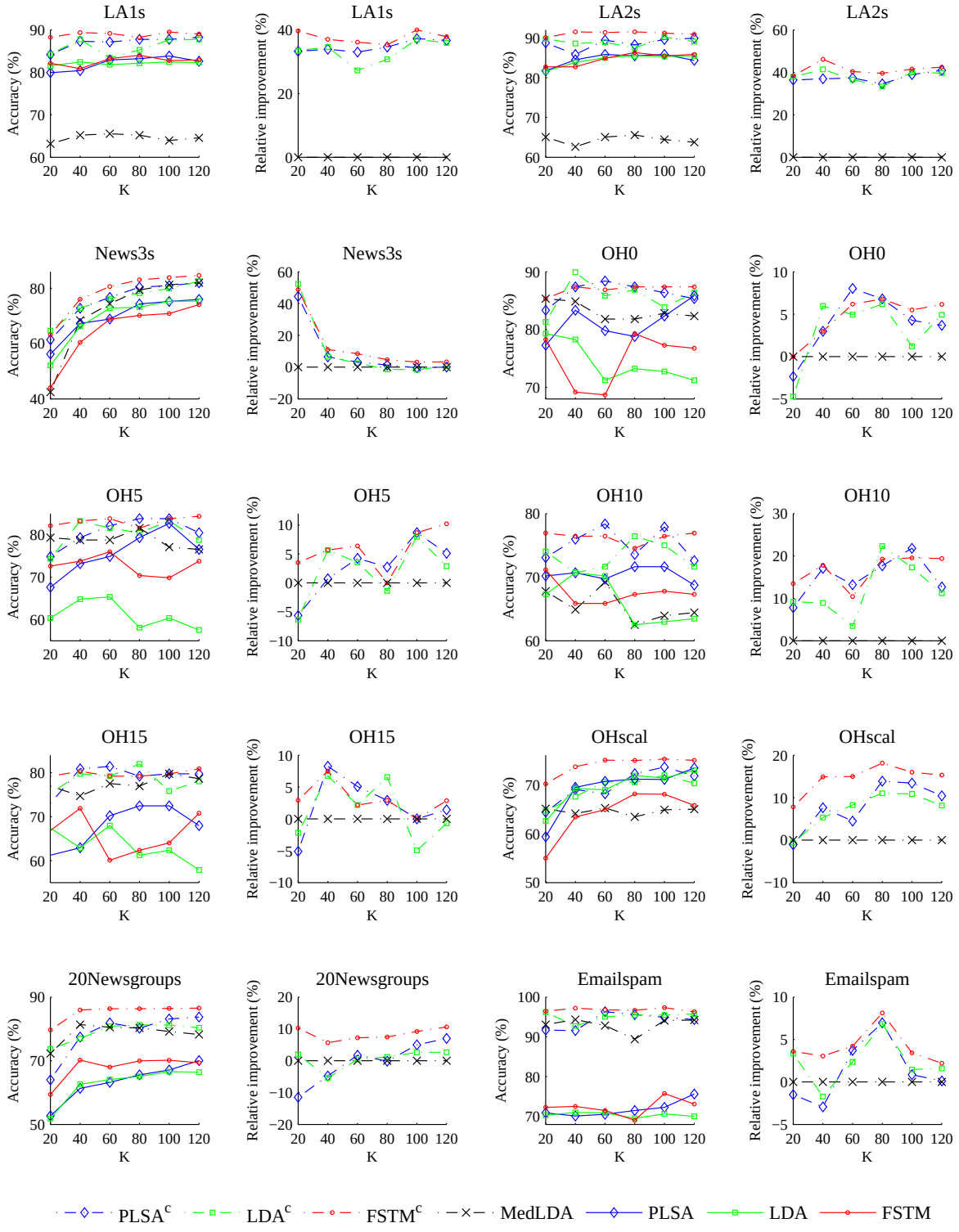


Figure 3.8: Accuracy of 7 methods as the number K of topics increases. Relative improvement is improvement of a method (A) over the-state-of-the-art MedLDA, and is defined as $\frac{accuracy(A)-accuracy(MedLDA)}{accuracy(MedLDA)}$.

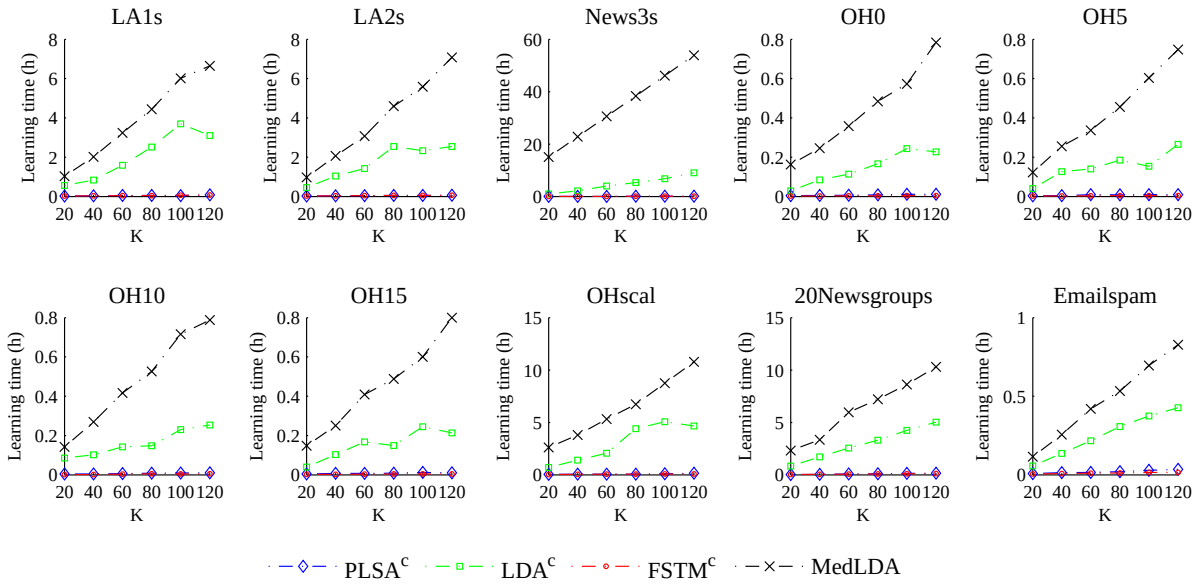


Figure 3.9: Necessary time to learn a discriminative space, as the number K of topics increases. FSTM^c and PLSA^c often performed substantially faster than MedLDA. As an example, for News3s and $K = 120$, MedLDA needed more than 50 hours to complete learning, whereas FSTM^c needed less than 8 minutes.

Table 3.3: Learning time in seconds when $K = 120$. For each dataset, the first line shows the learning time and the second line shows the corresponding accuracy. The best learning time is bold, while the best accuracy is italic.

Data	PLSA^c	LDA^c	FSTM^c	MedLDA
LA1s	287.05	11,149.08	**275.78**	23,937.88
	88.24	87.77	*89.03*	64.58
LA2s	**219.39**	9,175.08	238.87	25,464.44
	89.89	89.07	*90.86*	63.78
News3s	494.72	32,566.27	**462.10**	194,055.74
	82.01	82.59	*84.64*	82.01
OH0	39.21	816.33	**16.56**	2,823.64
	85.35	86.36	*87.37*	82.32
OH5	34.08	955.77	**17.03**	2,693.26
	80.45	78.77	*84.36*	76.54
OH10	37.38	911.33	**18.81**	2,834.40
	72.60	71.63	*76.92*	64.42
OH15	38.54	769.46	**15.46**	2,877.69
	79.78	78.09	*80.90*	78.65
OHscal	584.74	16,775.75	**326.50**	38,803.13
	71.77	70.29	*74.96*	64.99
20Newsgroups	556.20	18,105.92	**415.91**	37,076.36
	83.72	80.34	*86.53*	78.24
Emailspam	124.07	1,534.90	**56.56**	2,978.18
	94.34	95.73	*96.31*	94.23

plication of two sparse matrices for learning topics, while PLSA has a very simple learning formulation. Hence learning in FSTM and PLSA is unsurprisingly very fast [100]. The most time consuming part of FSTM^c and PLSA^c is to search nearest neighbors for each document. A modest implementation would requires $O(V.M^2)$ arithmetic operations, where M is the data size. Such a computational complexity will be problematic when the data size is large. Nonetheless, as empirically shown in Figure 3.9, the overall time of FSTM^c and PLSA^c was significantly less than that of MedLDA and LDA^c. Table 3.3 supports further this observation. Even for 20Newsgroups and News3s of average size, learning time of FSTM^c and PLSA^c is very competitive compared with MedLDA.

Summarizing, the above investigations demonstrate that the two-phase framework can result in very competitive methods for supervised dimension reduction. Three adapted methods, FSTM^c, LDA^c, and PLSA^c, mostly outperform their corresponding unsupervised models. LDA^c and PLSA^c often reached comparable performance with the state-of-the-art method, MedLDA. Amongst those adaptations, FSTM^c behaves superior in both classification performance and learning speed. We observe it often does 30-450 times faster than MedLDA.

3.4.4 Discussion

We have proposed the two-phase framework for doing dimension reduction of supervised discrete data. The framework was demonstrated to exploit well label information and local structure of the training data to find a discriminative low-dimensional space. Generality and flexibility of our framework was evidenced by adaptation to three unsupervised topic models, resulted in PLSA^c, LDA^c, and FSTM^c for supervised dimension reduction. These methods can perform qualitatively comparably with the state-of-the-art method, MedLDA. In particular, FSTM^c performed significantly best and can often achieve more than 10% improvement over MedLDA while enjoying 30-450 times faster speed. These results show that our framework can inherit scalability of unsupervised models to yield competitive methods for supervised dimension reduction.

The resulting methods (PLSA^c, LDA^c, and FSTM^c) are not limited to discrete data. They can work also on non-negative data, since their learning algorithms actually are very general. Hence in this paper, we contributed methods for not only discrete data but also non-negative real data. The code of these methods is freely available online at `www.jaist.ac.jp/~s1060203/codes/sdr/`

There is a number of possible extensions to our framework. First, one can easily mod-

ify the framework to deal with multilabel data. Second, the framework can be modified to deal with semi-supervised data. A key to these extensions is an appropriate utilization of labels to search for nearest neighbors, which is necessary for our framework. Other extensions can encode more prior knowledge into the objective function for inference. In our framework, label information and local neighborhood are encoded into the objective function and have been observed to work well. Hence, we believe that other prior knowledge can be used to derive good methods.

3.5 Summary

We make two contributions in this chapter. First, a framework (FW) for efficiently inferring sparse latent representations of documents is introduced. From theoretical and empirical analyses, the FW framework is shown to work significantly fast and always infer sparse solutions. Second, we propose an effective and scalable methods for doing supervised dimension reduction. In particular, one of the methods can perform consistently better in quality than the state-of-the-art methods, while enjoying 30-450 times faster speed.

Chapter 4

Fully Sparse Topic Models

Essentially, all models are wrong, but some are useful – George E. P. Box

In this chapter, a novel topic model is developed which has many interesting properties. It overcomes some severe limitations of existing models to better model corpora at a large scale. Inference in this model follows the framework discussed in Chapter 3.

4.1 Introduction

Topic modeling has been increasingly maturing to be an attractive research area. Originally motivated from textual applications, it has been going beyond far from text to touch upon many amazing applications in Computer Vision, Bioinformatics, Software Engineering, Forensics, to name a few. Recently, much interest in this community has focused on developing topic models for large-scale settings, e.g., [13, 48, 67, 71, 90, 115]. In our observations, the most common large-scale settings are:

(a) *the number of training documents is large;*

(b) *the number of topics to be learned is large;*

(c) *the vocabulary size (dimensionality) is large;*

(d) *a large number of documents need to be a posteriori inferred in a limited time budget.*

There are two fundamental issues to be addressed when dealing with large-scale settings: *inference speed* and *model complexity*. Scalable inference algorithms are highly

desired to resolve the settings (a) and (d), since inference plays the key role in many learning algorithms [21, 48]. Model complexity essentially refers to the number of effective parameters of a model. For a topic model like PLSA [51] or LDA [21], the hidden topics dominate its complexity, because each topic is often a dense distribution over the vocabulary. When dealing with the settings (b) and (c), the number of parameters in a topic model easily reaches billions. Such a huge model poses severe challenges for both learning and storage.

4.1.1 Our contribution

In this work, we present our initial step towards resolving the mentioned four large-scale settings. Our attempts attack the two fundamental issues mentioned before by seeking fast inference algorithms and sparse models.

Our first contribution is the introduction of *Fully Sparse Topic Model* (FSTM). Loosely speaking, FSTM is a simplified variant of LDA when relaxing the Dirichlet priors over hidden topics and over hidden topic proportions of documents. It is also a simplified variant of PLSA when removing the observed variable associated with each document. Nevertheless, FSTM has some following attractive properties:

- Inference is done by the Frank-Wolfe algorithm [28] which converges at a linear rate to the optimal solutions. The inference algorithm allows us to swiftly recover sparse topic proportions. Further, it provides a principled way to directly trade off sparsity of solutions against inference quality and running time.[1]
- Learning of topics amounts to multiplication of two sparse matrices. Hence topics are often very sparse. The sparsity level can be directly controlled.
- The complexity of the learning algorithm is near independent of dimensionality.[2]
- There is an implicit prior over topic proportions, though no explicit employment of priors. Such a prior can help FSTM avoid overfitting.

[1]Note that our reformulation of inference for FSTM can be applied to many variants of PLSA and LDA, and hence can help accelerate their inference. The reason is that such models often assume a document to be a mixture of topics.

[2]More precisely, the independence holds without taking into account the necessary number r of EM steps to reach convergence and an initial step which initiates necessary storage before learning. The EM algorithm often converges to stationary points at a linear rate [33]. However, to the best of our knowledge, there has been no rigorous analysis about relation between dimensionality and r. In practice, we experience that r does not depend on dimensionality.

Table 4.1: Theoretical comparison of 8 topic models: FSTM, PLSA, LDA, FTM [120], SparseTM [112], STC [129], SRS [89], RLSI [115]. V is the vocabulary size, K is the number of topics, $\bar{n}$ is the average length of documents. $\bar{K}$ is the average number of topics to which a term has nonzero contributions, $\bar{K} \leq K$. '-' denotes *'no'* or *'unspecified'*; '✓' means *'yes'* or *'taken in consideration'*.

Model	FSTM	PLSA	LDA	FTM	SparseTM	STC	SRS	RLSI
Document sparsity	✓	-	-	✓	-	✓	✓	-
Topic sparsity	✓	-	-	-	✓	-	✓	✓
Sparsity control	direct	-	-	indirect	indirect	indirect	indirect	indirect
Trade-off:								
sparsity vs. quality	✓	-	-	-	-	-	-	-
sparsity vs. time	✓	-	-	-	-	-	-	-
Dimension-free learning	✓	-	-	-	-	-	-	-
Inference complexity	$O(\bar{n}.\bar{K} + K)$	$O(\bar{n}.K)$	$O(\bar{n}.K)$	-	-	$O(\bar{n}.K)$	$O(\bar{n}.K)$	$O(V.\bar{K}^2 + K^3)$
Storage for topics	$V.\bar{K}$	$V.K$	$V.K$	-	-	$V.K$	$V.\bar{K}$	$V.\bar{K}$
Auxiliary parameters	0	0	0	0	0	3	2	2

For the first time in the topic modeling literature, FSTM is the model that couples the two interesting properties: near dimension-free learning algorithm, and ability to directly trade off sparsity of solutions against inference quality. The near independence of dimensionality implies that FSTM provides an almost optimal answer to the setting (c). It also implies that there exists a near dimension-free algorithm for doing dimensionality reduction (DR), since topic modeling is an approach to DR. These properties are crucial for dealing with data of extremely high dimensions. We hope that our results open a motivation for future studies to seek dimension-free algorithms for other problems.

The ability of FSTM to learn sparse topics and to infer sparse latent representations of documents allows us to save substantially memory for storage. Combined with a linear inference algorithm, FSTM overcomes severe limitations of existing probabilistic models and can deal well with the settings (b), (c), and (d). Fast learning of topics and fast inference of documents also enable us to deal well with the setting (a). To see more advantages of FSTM over existing models, we report some theoretical characteristics of some closely related models in Table 4.1.1.

Our second contribution is a distributed architecture for learning FSTM from large data. We employ both distributed scheme for data and task parallelism. Warm-start is further used to speed up learning, while keeping comparable quality. All of these provide a scalable learning algorithm that can handle very large corpora. In particular, we successfully learned a topic model with more than 33 billions of latent variables, from a large corpus with a vocabulary of 16 millions terms. This is the largest model that has been learned in the literature up to now.

Extensive experiments show that FSTM works well in practice. It significantly outperforms many models in terms of learning time, inference time, model complexity, and sparsity of latent representations of documents. The predictive power is observed to be

comparable with other models. In terms of generalization on unseen data, FSTM often does better. Qualitative performance of FSTM is also observed in application to classification, for both small and very large data.

4.1.2 Related work

Most previous works dealing with large data have focused mainly on the settings (a) and (b) by utilizing parallel/distributed/online architectures [13, 48, 67, 71, 90, 109]. Those works are breakthrough developments for learning LDA [21]. Their learning algorithms can work well with corpora of millions of documents and give an affirmative answer for the setting (a). However they remain three limitations: first, the LDA model itself is dense, which will consumes huge memory when vocabularies are very large. Second, the latent representations of documents are dense. Such dense representations will be problematic when stored for doing other tasks, e.g., information/image retrieval. The main reason is that the Dirichlet distribution employed by LDA prevents any zero contributions of terms to topics and of topics to documents. These limitations challenge deployment of LDA in practical applications with the settings (b) and (c).[3] Third, existing inference methods for LDA do not have any theoretical guarantee of neither inference quality nor inference time.

To reduce memory for efficient storage, some studies have introduced the notion of sparsity for topic models. Some researches try to reduce model complexity by encoding a spike-and-slap distribution [112] or using regularization [89, 115] to induce sparse topics. Furthermore, sparsity of topic proportions is also considered by employing Indian buffet processes [120] or by using regularization [89, 129]. Even though these models provide elegant solutions to the sparsity problem, they remain some drawbacks when dealing with large-scale settings. Indeed, the approaches by [112, 120] often result in much involved models and thus complicates learning and inference. Learning by existing sampling methods [112, 120] seem to be far from a touch upon large-scale settings. SRS [89] has no guarantee on convergence of inference/learning, and its scalability is unknown. STC [129] is problematic with learning, because learning of topics is to solve a optimization problem with a large number of variables which are inseparable. RLSI [115] has high complexity for both learning and inference, see Table 4.1.1. Finally, there are two common limitations of those non-probabilistic models (STC, SRS, RLSI): first, auxiliary parameters associated with regularization terms require us to do model selection, which is problematic in

[3] An example is topical exploration of huge corpora with extremely high dimensions, e.g. Google n-gram books (`http://aws.amazon.com/datasets/8172056142375670`). This application may requires learning tens of thousands of topics.

dealing with large-scale settings; second, one cannot directly trade off sparsity of solutions against time and quality.[4]

4.1.3 Roadmap

The main model will be presented in Section 4.2. Section 4.3 is devoted to analyzing some theoretical characteristics of FSTM, and to revealing why there is an implicit prior over latent representations. Evaluation and comparison on average corpora are discussed in details in Section 4.4. Section 4.5 presents a distributed architecture, and then experiments on large data. Some conclusions are in the final section.

4.2 Fully sparse topic models

Fully Sparse Topic Model assumes that a corpus is composed from K topics, $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K$, and each document $\boldsymbol{d}$ is generated by the following process:

Generate the jth word in $\boldsymbol{d}$ by:

- *First pick a latent topic z_k with probability $P(z_k|\boldsymbol{d}) = \theta_k$,*
- *Then generate a word w_j with probability $P(w_j|z_k) = \beta_{kj}$.*

It is straightforward to see that FSTM is a simplified variant of LDA. The main difference is that FSTM does not employ Dirichet prior over topic proportions, and deliberately allows only few topics to contribute to a document. This relaxation allows us to infer really sparse topic proportions of documents. No employment of Dirichlet prior over topics enables us to learn models of low complexity, i.e., sparse models. Figure 4.1 depicts the graphical representation of FSTM, accompanied by PLSA and LDA.

Motivated by large-scale settings, our focus is to design a fast inference algorithm and a fast learning algorithm that can learn sparse models. This is a nontrivial task, as evidenced in [112, 120, 129]. We tackle this task by first reformulating the inference problem as a concave maximization problem over the simplex of topics. This reformulation allows us to seamlessly employ the Frank-Wolfe algorithm to do inference and thus inherits

[4]For regularization techniques, one may expect to get sparser solutions by increasing the values of the auxiliary parameters. However, it is not always provably true. Hence such a control over sparsity is indirect.

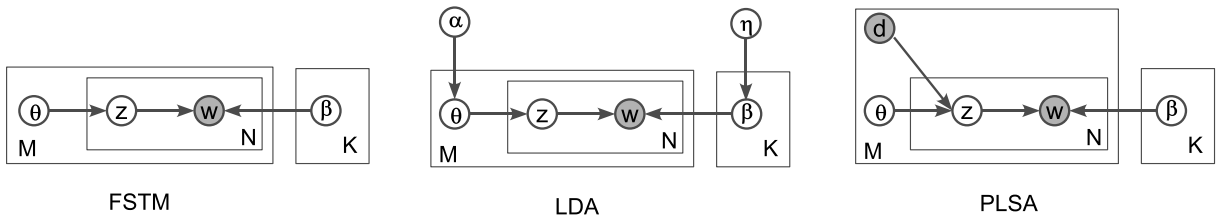


Figure 4.1: Graphical representations of three topic models.

its attractive properties. To obtain sparse topics, our idea is to exploit sparsity of original documents and sparsity of topic proportions. Hence we propose an approach so that learning topics amounts to multiplication of the original representation ($\boldsymbol{d}$) with the new representation ($\boldsymbol{\theta}$) of the corpus. Note that sparsity of topic proportions ($\boldsymbol{\theta}$) can be directly controlled just by limiting the number of iterations in the Frank-Wolfe algorithm. As a result, the learned topics are often very sparse and the sparsity level can be controlled.

In spite of no explicit prior over $\boldsymbol{\theta}$ in the model description, we will see in Section 4.3 that in fact there exists an implicit prior having density function $p(\boldsymbol{\theta}|\lambda) \propto \exp(-\lambda.||\boldsymbol{\theta}||_0)$, where $||\boldsymbol{\theta}||_0$ is the number of non-zero entries of $\boldsymbol{\theta}$. This property is a consequence of sparse inference in FSTM. Note that this property of FSTM is intriguing.

4.2.1 Inference

Given a document $\boldsymbol{d}$ and topics $\boldsymbol{\beta}$, the inference task in FSTM is to find which topics contribute to $\boldsymbol{d}$ and how much they contribute to $\boldsymbol{d}$. In other words, we have to infer $\boldsymbol{\theta}$. Unlike existing inference approaches for topic models, we will not make effort to infer directly $\boldsymbol{\theta}$. Instead, we reformulate the inference task as a concave maximization problem over the simplex of topics.

Lemma 4. *Consider FSTM with topics $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K$, and a given document $\boldsymbol{d}$. The inference problem can be reformulated as the following concave maximization problem, over the simplex $\Delta = conv(\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K)$,*

$$\boldsymbol{x}^* = \arg\max_{\boldsymbol{x} \in \Delta} \sum_{j \in I_d} d_j \log x_j. \tag{4.1}$$

Proof. For a given document $\boldsymbol{d}$, the probability that a term w_j appears in $\boldsymbol{d}$ can be expressed as $P(w_j|\boldsymbol{d}) = \sum_{k=1}^{K} P(w_j|z_k).P(z_k|\boldsymbol{d}) = \sum_{k=1}^{K} \theta_k \beta_{kj}$. Hence the log likelihood

Algorithm 4 Inference algorithm

Input: document $\boldsymbol{d}$ and topics $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K$.
Output: $\boldsymbol{\theta}_*$, for which $\sum_{k=1}^K \theta_{*,k}\boldsymbol{\beta}_k = \boldsymbol{x}_*$ maximizes $f(\boldsymbol{x}) = \sum_{j\in I_d} d_j \log x_j$.
Pick as $\boldsymbol{\beta}_r$ the vertex of $\Delta = conv(\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K)$ with largest f value.
Set $\boldsymbol{x}_0 := \boldsymbol{\beta}_r; \theta_{0,r} = 1; \theta_{0,k} = 0, \forall k \neq r$;
for $\ell = 0, ..., \infty$ **do**
  $i' := \arg\max_i \boldsymbol{\beta}_i^t \nabla f(\boldsymbol{x}_\ell)$;
  $\alpha' := \arg\max_{\alpha\in[0,1]} f(\alpha\boldsymbol{\beta}_{i'} + (1-\alpha)\boldsymbol{x}_\ell)$;
  $\boldsymbol{x}_{\ell+1} := \alpha'\boldsymbol{\beta}_{i'} + (1-\alpha')\boldsymbol{x}_\ell$;
  $\boldsymbol{\theta}_{\ell+1} := (1-\alpha')\boldsymbol{\theta}_\ell$; and then set $\theta_{\ell+1,i'} := \theta_{\ell+1,i'} + \alpha'$.
end for

of $\boldsymbol{d}$ is

$$\log P(\boldsymbol{d}) = \log \prod_{j\in I_d} P(w_j|\boldsymbol{d})^{d_j} \quad = \quad \sum_{j\in I_d} d_j \log P(w_j|\boldsymbol{d}) = \sum_{j\in I_d} d_j \log \sum_{k=1}^K \theta_k \beta_{kj}.$$

The inference task is the problem of searching for $\boldsymbol{\theta}$ to maximize the likelihood of $\boldsymbol{d}$. Denoting as $x_j = \sum_{k=1}^K \theta_k \beta_{kj}$ and $\boldsymbol{x} = (x_1, ..., x_V)^t$, we arrive at

$$\log P(\boldsymbol{d}) = \sum_{j\in I_d} d_j \log x_j. \tag{4.2}$$

Therefore optimization over $\boldsymbol{\theta}$ now is translated into that over $\boldsymbol{x}$. Note that $\boldsymbol{x} = (x_1, ..., x_V)^t = \sum_{k=1}^K \theta_k \boldsymbol{\beta}_k$. Combining this with the fact that $\sum_k \theta_k = 1$, $\theta_k \geq 0, \forall k$, one can easily realize that $\boldsymbol{x}$ is a convex combination of the K topics $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K$. It implies $\boldsymbol{x} \in \Delta$. As a result, the inference task is in turn the problem of finding $\boldsymbol{x} \in \Delta$ that maximizes the objective function (4.2). (Q.E.D.)

This lemma provides us a connection between inference and concave optimization, and allows us to seamlessly use the Frank-Wolfe algorithm for inference. An appropriate adaptation to the Frank-Wolfe algorithm results in an inference algorithm for FSTM, as presented in Algorithm 4. In our implementation, we solve for α by the gradient ascent approach.

4.2.2 Learning

The task of learning FSTM is to learn all topics $\boldsymbol{\beta}$, given a corpus $\mathcal{C}$. We use EM scheme to iteratively learn the model. Specifically, we repeat the following two steps until convergence:

E-step: do inference for each document of $\mathcal{C}$;

M-step: maximize the likelihood of $\mathcal{C}$ with respect to $\boldsymbol{\beta}$.

Note that the E-step for each document is discussed in the previous subsection. The remaining task is to solve for $\boldsymbol{\beta}$. Denoting as $\boldsymbol{\theta}_d$ the topic proportion of document $\boldsymbol{d} \in \mathcal{C}$ which has been inferred in the E-step, and the documents are i.i.d., we express the log likelihood of $\mathcal{C}$ as

$$\log P(\mathcal{C}) = \sum_{\boldsymbol{d}\in\mathcal{C}} \log P(\boldsymbol{d}) \quad = \quad \sum_{\boldsymbol{d}\in\mathcal{C}} \sum_{j\in I_d} d_j \log \sum_{k=1}^{K} \theta_{dk}\beta_{kj} \geq \sum_{\boldsymbol{d}\in\mathcal{C}} \sum_{j\in I_d} d_j \sum_{k=1}^{K} \theta_{dk} \log \beta_{kj}.$$

We have used Jensen's inequality to derive the last term, owing to the fact $\sum_k \theta_{dk} = 1, \theta_{dk} \geq 0, \forall k$.

Next we maximize the lower bound of $\log P(\mathcal{C})$ with respect to $\boldsymbol{\beta}$. In other words, we have to maximize

$$\begin{aligned} g(\boldsymbol{\beta}) = \sum_{\boldsymbol{d}\in\mathcal{C}} \sum_{j\in I_d} d_j \sum_{k=1}^{K} \theta_{dk} \log \beta_{kj} \quad &= \quad \sum_{k=1}^{K} \sum_{\boldsymbol{d}\in\mathcal{C}} \sum_{j\in I_d} d_j \theta_{dk} \log \beta_{kj}, \\ \text{s.t.} \qquad & \sum_{j=1}^{V} \beta_{kj} = 1, \beta_{kj} \geq 0, \forall k, j. \end{aligned} \tag{4.3}$$

It is worthwhile noticing that the vectors $\boldsymbol{\beta}_k$ are separable from each other in the objective function $g(\boldsymbol{\beta})$. Hence we can solve for each individually. Taking the Lagrange function into consideration and forcing its derivatives to be 0, we easily arrive at the following solution

$$\beta_{kj} \quad \propto \quad \sum_{\boldsymbol{d}\in\mathcal{C}} d_j \theta_{dk}. \tag{4.4}$$

Up to this point, we can learn FSTM by iterating E-step and M-step until convergence. In the E-step, each document is inferred by using the Frank-Wolfe algorithm, given the objective function as in (4.1) and topics $\boldsymbol{\beta}$. The M-step only does simple calculation according to (4.4) to update all topics.

4.3 Theoretical analysis

We will show that the inference algorithm for FSTM can provide provably good solutions. It requires modestly few arithmetic operations, linear in the length of the document to be inferred or in the number of topics. The learning algorithm has very low complexity which does not depend on the dimensionality V. Further, we can easily trade off quality of solution against sparsity and inference time. Existing topic models do not own these interesting properties.

4.3.1 Complexity and goodness of inference

Theorem 6. *Consider FSTM with K topics, and a document $\boldsymbol{d}$. Let C_f be defined as in Theorem 2 for the function $f(\boldsymbol{x}) = \sum_{j \in I_d} d_j \log x_j$. Then Algorithm 4 converges to the optimal solution with a linear rate. In addition, after L iterations, the inference error is at most $4C_f/(L+3)$, and the topic proportion $\boldsymbol{\theta}$ has at most $L+1$ non-zero components.*

Proof. Inference of FSTM is exactly the Frank-Wolfe algorithm for the function $f(\boldsymbol{x}) = \sum_{j \in I_d} d_j \log x_j$ which is twice differentiable at all $\boldsymbol{x}$ satisfying $x_j > 0, \forall j \in I_d$. Hence this theorem is a corollary of Theorem 2. (Q.E.D.)

Next we will analyze computational complexity of the inference algorithm. Common technique to store a sparse matrix is row-wise, i.e., we store all non-zero elements in a row of that matrix by an 1-dimensional array. This is beneficial to do multiplication of a sparse matrix with a vector. Indeed, consider a matrix $\boldsymbol{B}$ of size $m \times n$. Letting $\bar{m}$ be the average number of non-zero elements of a column of $\boldsymbol{B}$, computing $\boldsymbol{Bx}$ requires only $O(n.\bar{m} + m)$ arithmetic operations.

Theorem 7. *Each iteration of Algorithm 4 requires only $O(n.\bar{K} + K)$ arithmetic operations, where $\bar{K}$ is the average number of topics to which a term has non-zero contributions, $\bar{K} \leq K$, and $n = |I_d|$. Overall, after L iterations, Algorithm 4 requires $L.O(n.\bar{K} + K)$ arithmetic operations.*

Proof. Letting $\boldsymbol{a} = \nabla f(\boldsymbol{x})$, we have $\boldsymbol{\beta}^t \nabla f(\boldsymbol{x}) = \boldsymbol{\beta}^t \boldsymbol{a}$. Note that $\boldsymbol{a}$ is very sparse because of $a_i = \partial f / \partial x_i = 0$, for $i \notin I_d$. Hence only n columns of $\boldsymbol{\beta}^t$ involve in computation of $\boldsymbol{\beta}^t \boldsymbol{a}$. This implies that we need just $O(n.\bar{K} + K)$ arithmetic operations to compute $\boldsymbol{\beta}^t \boldsymbol{a}$ and to find the index i'. $O(n.\bar{K} + K)$ arithmetic operations are also sufficient to do the initial step of choosing $\boldsymbol{x}_0$, since the most expensive computations are to evaluate

f at the vertices of the simplex, which amounts to a multiplication of $(\log \boldsymbol{\beta})^t \boldsymbol{d}$, where $\log \boldsymbol{\beta} = (\log \beta_{ij})_{V \times K}$.

Searching for α can be done very quickly since the problem is concave in one variable. Each evaluation of $f(\boldsymbol{x})$ requires only $O(n)$ operations. Moreover $O(n.\bar{K} + K)$ arithmetic operations are sufficient to update other variables. (Q.E.D.)

Theoretically, L can be large in order to find good solutions. In particular, if we want to find an ϵ-approximate solution, the number of iterations should be $L > 4C_f/\epsilon - 3$ due to Theorem 6. This implies the complexity of the inference algorithm depends on the curvature constant C_f of the objective function. C_f shows how hard to maximize f over the simplex, and can sometimes depend on the dimensionality K of the optimization problem. However, L does not depends on V.

4.3.2 Complexity of learning

FSTM is learned by the EM scheme. Each EM iteration requires us to infer M training documents, and update topics according to formula (4.4). Note that update of topics simply does a multiplication of two very sparse matrices (one is the matrix representing the training corpus, and the other is the new representation of that corpus), and then does normalization. Hence it can be computed very fast. A simple implementation would require $O(M.\bar{n}.\bar{s})$ arithmetic operations to compute multiplication of two sparse matrices and then $O(V.K)$ to do normalization, where $\bar{n}$ is the average length of the original documents, and $\bar{s}$ is the average number of topics contributing to a document (average length of topic proportions, $\bar{s} \leq K$). Therefore, an EM iteration requires $O(V.K + M.\bar{n}.\bar{s} + M.\bar{L}.(\bar{n}.\bar{K} + K))$ operations, where $\bar{L}$ is the average number of iterations for the Frank-Wolfe algorithm to reach convergence. It is worth noticing that $\bar{s} \leq \bar{L}$.

Lemma 5. *An update of $\boldsymbol{\beta}$ by (4.4) can be done in $O(M.\bar{n}.\bar{s}.\bar{K})$ arithmetic operations.*

Proof. Letting $a_k = \sum_{j=1}^{V} \sum_{\boldsymbol{d} \in \mathcal{C}} d_j \theta_{dk} = \sum_{\boldsymbol{d} \in \mathcal{C}} \sum_{j \in I_d} d_j \theta_{dk}$, we have $\beta_{kj} = a_k^{-1} \sum_{\boldsymbol{d} \in \mathcal{C}} d_j \theta_{dk} = \sum_{\boldsymbol{d} \in \mathcal{C}} a_k^{-1} d_j \theta_{dk}$ for each k and j. Note that $a_1, ..., a_K$ can be computed in $O(M.\bar{n}.\bar{s} + K)$ requiring only one scan over the corpus. Hence computing $\boldsymbol{\beta}$ requires at most $O(M.\bar{n}.\bar{s}.\bar{K})$ operations when all $\boldsymbol{d}, \boldsymbol{\beta}, \boldsymbol{\theta}$ are represented in sparse format. (Q.E.D.)

Corollary 4. *Each EM iteration of the learning algorithm for FSTM can be computed in $O(M.\bar{n}.\bar{s}.\bar{K} + M.\bar{L}.(\bar{n}.\bar{K} + K))$ arithmetic operations.*

Learning FSTM requires only $r.O(M.\bar{n}.\bar{s}.\bar{K} + M.\bar{L}.(\bar{n}.\bar{K} + K))$ arithmetic operations, where r is the necessary number of EM iterations to reach convergence. Theoretically, the EM algorithm is known to have linear convergence rate [33]. Nevertheless, there has been no rigorous analysis about its complexity. The same situation remains with many other learning algorithms not only in topic modeling but also in Machine Learning. A very recent result by [2] for convex optimization shows that in the worst case the number of iterations for an algorithm to reach convergence depends on the dimension of the optimization problem. This pessimistic result suggests that the EM algorithm may have high complexity, and the necessary number of EM iterations may depend on dimensionality, V. The reason is that the objective of learning is the likelihood which is not concave, and thus likely harder than convex optimization in general.

If r does not depend on dimensionality, then so does the complexity of the learning algorithm for FSTM.[5] However, since r is not theoretically upper bounded, the learning algorithm is near independent of dimensionality. In practice, we find that r and $\bar{L}$ are often less than 40, even for large K, V and M. Hence they are much less than the vocabulary size V and corpus size M. Ignoring those constants, we loosely conclude that $M.O(\bar{n}.\bar{s}.\bar{K} + \bar{n}.\bar{K} + K)$ is the complexity for learning FSTM.

The near independence of dimensionality is an intriguing property of FSTM, which is crucial for dealing with large data of very high dimensions. To the best of our knowledge, this is the first time a near dimension-free learning algorithm for topic models have been proposed. Learning in other models such as PLSA, LDA, and RLSI is often linearly dependent on the dimensionality V. For example, learning algorithms for PLSA [51] and LDA [21] require $O(V.K+M.K+M.K.\bar{n})$ arithmetic operations. Such a linear dependence would cause some difficulties when working with data of extremely high dimensions.

4.3.3 Managing sparsity level and trade-off

Good solutions are often necessary for practical applications. In practice, we may have to spend intensive time and huge memory to search such solutions. This sometimes is not necessary or impossible in limited time/memory settings. Hence one would prefer to trading off quality of solutions against time/memory.

Searching for sparse solutions is a common approach in Machine Learning to reduce

[5] In fact, when the average length $\bar{n}$ of documents is large, i.e., documents are very long and dense, such an independence may not be true. However, in practice, a document written by human often has significantly few different terms compared to the vocabulary. For example, a news often has less than 1000 different terms; conversations on Facebook commonly have few tens of different terms.

memory for storage and efficient processing. Most previous works have tried to learn sparse solutions by imposing regularization which induces sparsity, e.g., L1 regularization [115, 129] and entropic regularization [89]. Nevertheless, those techniques are severely limited in the sense that we cannot directly control the sparsity level of solutions (e.g., one cannot decide how many non-zero components solutions should have). In other words, the sparsity level of solutions is a priori unpredictable. This limitation makes regularization techniques inferior in memory limited settings. It is also the case with other works that employ some probabilistic distributions to induce sparsity [112, 120] or that exploits sparsity of sufficient statistics of Gibbs samples [67].

Unlike prior topic models, the inference algorithm for FSTM naturally provides a principled way to control sparsity. Theorem 6 implies that if stopped at the Lth iteration, the inferred solution has at most $L + 1$ non-zero components. Hence one can control sparsity level of solutions by simply limiting the number of iterations. It means that we can predict a priori how sparse and how good the inferred solutions are. Less iterations, sparser (but probably worse) solutions of inference. Besides, we can trade off sparsity against inference time. More iterations imply more time and probably denser solutions.

4.3.4 Implicit prior over $\boldsymbol{\theta}$

In Section 4.2 we describe FSTM without any specific prior over latent representations $\boldsymbol{\theta}$. As well-known in the literature, no prior endowment may cause a model to be prone to overfitting. Nonetheless, it seems not the case with FSTM. Indeed, we argue that there is an implicit prior over $\boldsymbol{\theta}$ in the model.

Note that the inference algorithm of FSTM allows us to easily trade off sparsity of solutions against quality and time. If one insists on solutions with at most t nonzero components, the inference algorithm can be modified accordingly. In this case, it mimics that one is trying to find a solution to the problem $\max_{\boldsymbol{\theta}\in\Delta_K}\{f(\boldsymbol{\theta}) : ||\boldsymbol{\theta}||_0 \leq t\}$. We remark a well-known fact that the constraint $||\boldsymbol{\theta}||_0 \leq t$ is equivalent to addition of a penalty term $\lambda.||\boldsymbol{\theta}||_0$ to the objective function [69], for some constant λ. Therefore, one is trying to solve for $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\Delta_K}\{f(\boldsymbol{\theta}) - \lambda.||\boldsymbol{\theta}||_0\} = \arg\max_{\boldsymbol{\theta}\in\Delta_K} P(\boldsymbol{d}|\boldsymbol{\theta}).P(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}\in\Delta_K} P(\boldsymbol{\theta}|\boldsymbol{d})$, where $p(\boldsymbol{\theta}) \propto \exp(-\lambda.||\boldsymbol{\theta}||_0)$. Notice that the last problem, $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\Delta_K} P(\boldsymbol{\theta}|\boldsymbol{d})$, is an MAP inference problem. Hence, these observations basically show that inference by Algorithm 4 for sparse solutions mimics MAP inference. As a result, there exists an implicit prior, having density function $p(\boldsymbol{\theta};\lambda) \propto \exp(-\lambda.||\boldsymbol{\theta}||_0)$, over latent topic proportions. This is another characteristic that distinguishes FSTM from existing topic models.

4.3.5 The zero problem and solution

We have shown in Lemma 4 that the inference problem for FSTM can be reduced to that of maximizing the function $f(\boldsymbol{x}) = \sum_{j \in I_d} d_j \log x_j$ over the simplex $\Delta = conv(\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K)$. Note that $f(\boldsymbol{x})$ discourages solutions with zero entries, and encourages area inside Δ that ensures $f(\boldsymbol{x})$ to be well-defined. Hence it seems to prefer dense topics. Nonetheless, topics learned by (4.4) are likely very sparse, since the training documents ($\boldsymbol{d}$) and their new representations ($\boldsymbol{\theta}$) are often very sparse. These lead to a tension in which inference prefers dense topics, but learning of topics often results in sparse ones. To overcome such a situation, one possible way is to learn as sparse as possible topics while maintaining quality of inference for the training documents. Nonetheless, such an approach can only explain the given training data but forgets future ones, and thus is prone to overfitting. This exactly happens with FSTM.

Solution: for each document $\boldsymbol{d}$, instead of doing inference over Δ, we will do inference over the simplex $\Delta' = conv(\boldsymbol{\lambda}_1, ..., \boldsymbol{\lambda}_K)$ where $\lambda_{ki} \propto \beta_{ki} + \varepsilon$ for a very small constant $\varepsilon > 0$. The constant ε ensures that the objective function $f(\boldsymbol{x})$ is twice differentiable over Δ'. Hence inference can be done smoothly.[6] Such a simple modification really helps FSTM to overcome overfitting and to perform well on real data, as investigated in the next section.

4.4 Experimental evaluation

This section is devoted to investigating practical behaviors of FSTM to see clearly its characteristics. We will describe performance of our model on huge corpora in the next section. The investigation focuses mostly on some fundamental properties of FSTM and how good it is when applied to classification.

4.4.1 Sparsity and time

We first aim at answering the following questions: (1) *how sparse are topics and latent representations of documents?* (2) *how fast can the model infer/learn?* To this end, we chose 4 corpora for experiments: 2 small (AP, KOS), and 2 average (Grolier, Enron). Table 4.2 contains some information about these corpora. Four models are included

[6] With this modification, the topics of our model are in fact $\boldsymbol{\lambda}_1, ..., \boldsymbol{\lambda}_K$. However we do not have to store all, but instead store the most meaningful parts ($\boldsymbol{\beta}$). In our experiments, we set $\varepsilon = 10^{-10}$.

Table 4.2: Data for experiments. $\bar{n}$ is the average number of different terms in a document.

Data	M	Testing size	V	Classes	$\bar{n}$
AP	2,021	225	10,473	0	135
KOS	3,087	343	6,906	0	103
Grolier	23,044	6,718	15,276	0	80
Enron	35,875	3,986	28,102	0	96
20Newsgroups	15,935	3,993	62,061	20	80
Webspam	350,000	350,000	16,609,143	2	3,728

for comparison: FSTM, PLSA, LDA, and STC.[7] Despite of being a non-probabilistic model, STC is included for comparison because it was intentionally designed to model sparsity. In our experiments we used the same convergence criteria for these models: relative improvement of log likelihood (or objective functions in STC) is less than 10^{-6} for inference, and 10^{-4} for learning; at most 1000 iterations are allowed to do inference. We used default settings for some other auxiliary parameters of STC, relating to regularization terms.

Document sparsity: Figure 4.2 presents the results of experiments on four corpora. Document sparsity is used to see sparsity level of latent representations discovered by those models. Observing the first two rows of Figure 4.2, one can see that all models, except LDA, can discover sparse latent representations. PLSA interestingly can discover very sparse latent representations for testing data. It even often outperformed STC, which was intentionally designed for modeling sparsity. However, it seems that PLSA achieved sparse solutions by incident. Indeed, we rarely observed sparse topic proportions in the learning phase, but inference often resulted in sparse ones. One crucial reason for these contrary behaviors is that information was lost when saving the learned models, as we observed many nonzero elements of topics went to 0. STC can indeed discover sparse latent representations as expected. Nonetheless, the discovered sparsity level was not very high, i.e., new representations of documents were still pretty dense. Furthermore, the sparsity level seems to be inconsistent as the number of topics increases

On contrary, FSTM can discover very sparse latent representations in both learning and inference phases. The sparsity level consistently decreases as the number of topics increases. This implies that despite modeling a corpus with many topics, few topics actually contributes to a specific document. For example, on average, only 3 topics have non-zero contributions to a document of AP among 100 topics of the model; when mod-

[7]STC code was taken from `www.cs.cmu.edu/~junzhu/stc/`
PLSA was coded by ourselves with the best effort. SRS and RLSI were not included because of two reasons. First, there is no available code for these models. More importantly, there is an inconsistence in the update formula derived in [89] that prevents us from implementation; RLSI heavily needs involved distributed architectures.

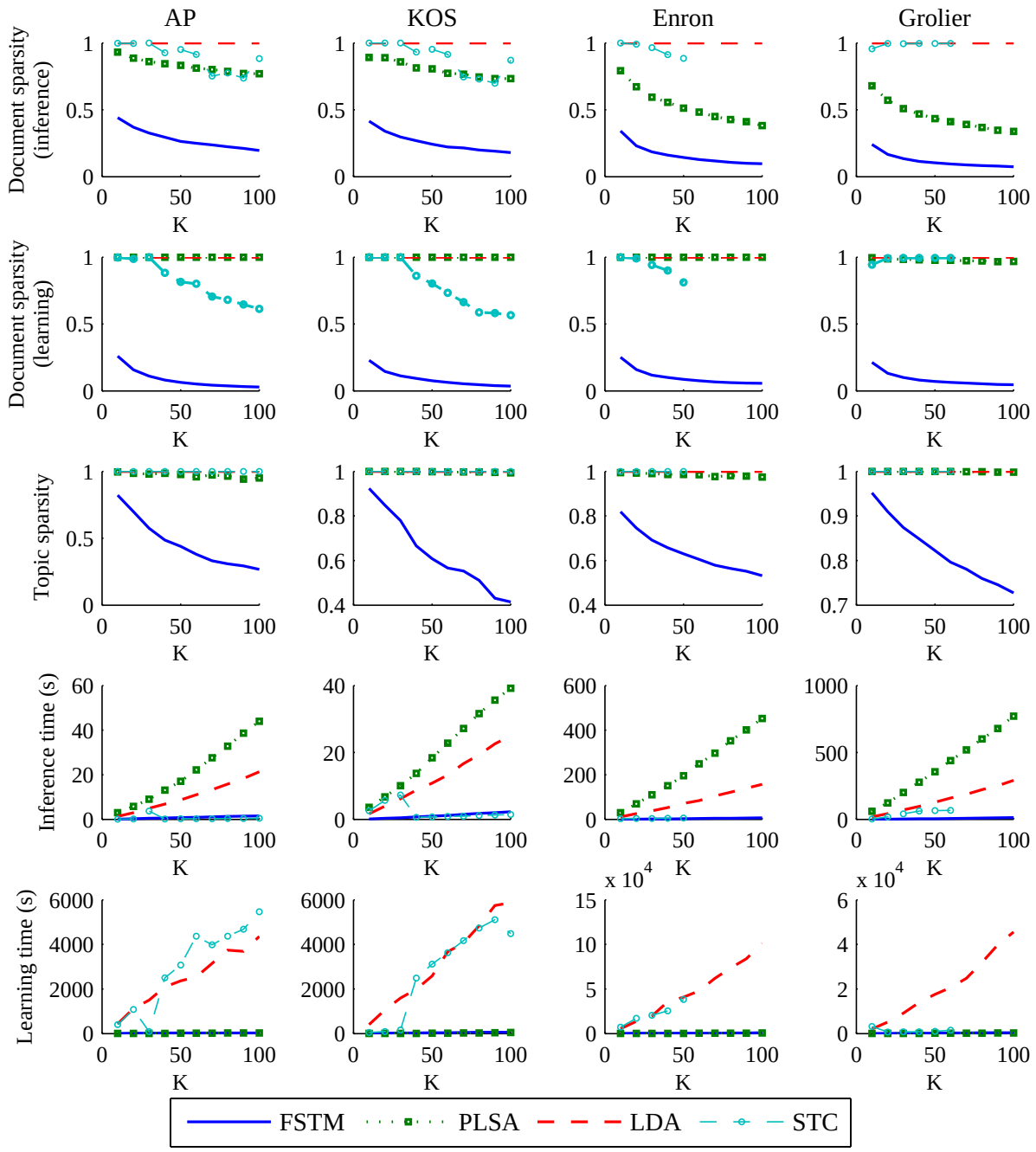


Figure 4.2: Experimental results as the number K of topics increases. Lower is better. For STC, there was a memory problem when dealing with Enron and Grolier for large K (e.g., when $K = 70$, STC has to solve a optimization problem with more than 20 millions of variables, and hence cannot be handled in a personal PC with 6Gb memory.) Hence we could not do experiments for such large K's.

eling with 10 topics, only 2 topics on average have non-zero contributions to a document. This behavior of FSTM is consistent with the fact that a document often says about few topics, independent of the number of topics a model is taking into account. Hence FSTM can discover very compact representations.

Topic sparsity: observing Figure 4.2, one easily realizes that most models could not discover sparse topics. LDA and STC are not surprising, because topics are assumed to be samples of Dirichlet distributions which implicitly prevent any zero contribution of terms to topics. PLSA could discover some sparse topics, but the sparsity level was insignificant. FSTM outperformed other models in this aspect, having discovered very sparse topics. The sparsity level of topics tends to increase as we model data with more topics. This achievement can be explained by the facts that new representations of documents inferred by FSTM are very sparse, that the original documents are sparse, and that topics are simply a product of these two sparse representations (see equation 4.4). Therefore, the learned models are often significantly compact.

Inference time: in Section 4.3, we have shown theoretically that inference of FSTM is in linear time. This is further supported by our experiments, as depicted in Figure 4.2. Both FSTM and STC worked comparably in practice. PLSA inferred most slowly by the folding-in technique. LDA can infer much more quickly by fast variational Bayesian methods [21]. Nevertheless, it still worked much more slowly than FSTM, often tens of times slower. There are at least two reasons for this slow inference: first, the inference problem in LDA is inherently NP-hard [91] and thus may require much time to reach at good solutions; second, the variational Bayesian algorithm has to do many computations relating to logarithm, exponent, gamma, and digamma functions which are expensive. In contrast, inference in FSTM can be done in linear time, and the objective function (likelihood) is relatively cheap to compute. In addition, the learned topics are often very sparse. All of these contribute to speeding up inference in FSTM.

Learning time: observing the last row of Figure 4.2, one can see that LDA and STC learned really slowly, often hundreds/thousands of times slower than FSTM and PLSA.[8] Slow learning of STC can be explained by the fact that learning of topics in this model is very expensive, since we have to solve a optimization problem with a large number, $K.V$, of variables which are inseparable. LDA learned slowly because its inference algorithm is slow, and it has to solve optimization problems requiring various evaluations of Gamma and Digamma functions which are often expensive. PLSA learned fastest due to its

[8] At some settings, we observe that STC did stop learning very early after only 4 or 5 iterations, but inference after that paid more time to do than usual. Otherwise, it needed many iterations (often more than 30) to reach convergence. Hence we suppose that those early terminations were caused by some internal issues.

Table 4.3: Example of topics learned by FSTM when $K = 100$. Shown for each topic are words that have highest probabilities.

Enron	
1	power, company, project, energy, india, government, electricity
2	corp, contract, message, party, review, offer, receive, prohibited
3	paper, pulp, mill, received, market, oct, story, office, press, release,
4	company, financial, stock, investor, partnership, billion, credit
5	energy, market, customer, wind, power, pjm, generation, prices
6	request, sap, approval, application, resource, security, data, access
7	travel, roundtrip, fares, hotel, city, visit, special, sheraton, sale
8	gas, contract, capacity, point, shipper, storage, firm, allocation
9	bill, michael, david, karen, mike, meeting, thomas, paul, mark
10	game, yard, fantasy, defense, allowed, against, point, updated
Grolier	
1	tax, government, public, property, insurance, income, business
2	family, marriage, crime, children, united, law, social, people, divorce
3	philosophy, world, knowledge, human, god, theory, nature
4	mental, disorders, behavior, children, disorder, treatment, sexual
5	species, birds, animals, mammals, million, world, behavior, animal
6	ballet, dance, company, de, theater, dances, ballets, dancer, american
7	water, coal, oil, energy, gas, power, fuel, united, steel, production
8	food, world, united, production, land, agricultural, plant, corn, cattle
9	architecture, style, art, building, century, gothic, architect, buildings
10	century, music, children, folk, tales, literature, poetry, poems, written

simple learning formulations. There is a seemingly contrary behavior of PLSA, in which learning is fastest but inference is slowest. The main reason is that inference by folding-in [51] is an adaptation of learning, and more importantly learning does not require doing separately inference of documents which differs from other models. FSTM can learn very fast, comparably with PLSA. One reason for such a fast learning is the fast inference algorithm. Another reason is that the inferred topic proportions and topics themselves are very sparse, and hence help further speed up learning.

4.4.2 Quality and trade-off

We next consider how good FSTM is. Table 4.3 shows some random examples from 100 topics learned by FSTM on Enron and Grolier. We can observe that those learned topics are very understandable and each indicates clearly a specific meaning. We further observed that for the same setting of K and the same corpus, most topics learned by one model can be learned by the others (among FSTM, LDA, and PLSA). These observations suggests that FSTM can provide us qualitative topics.

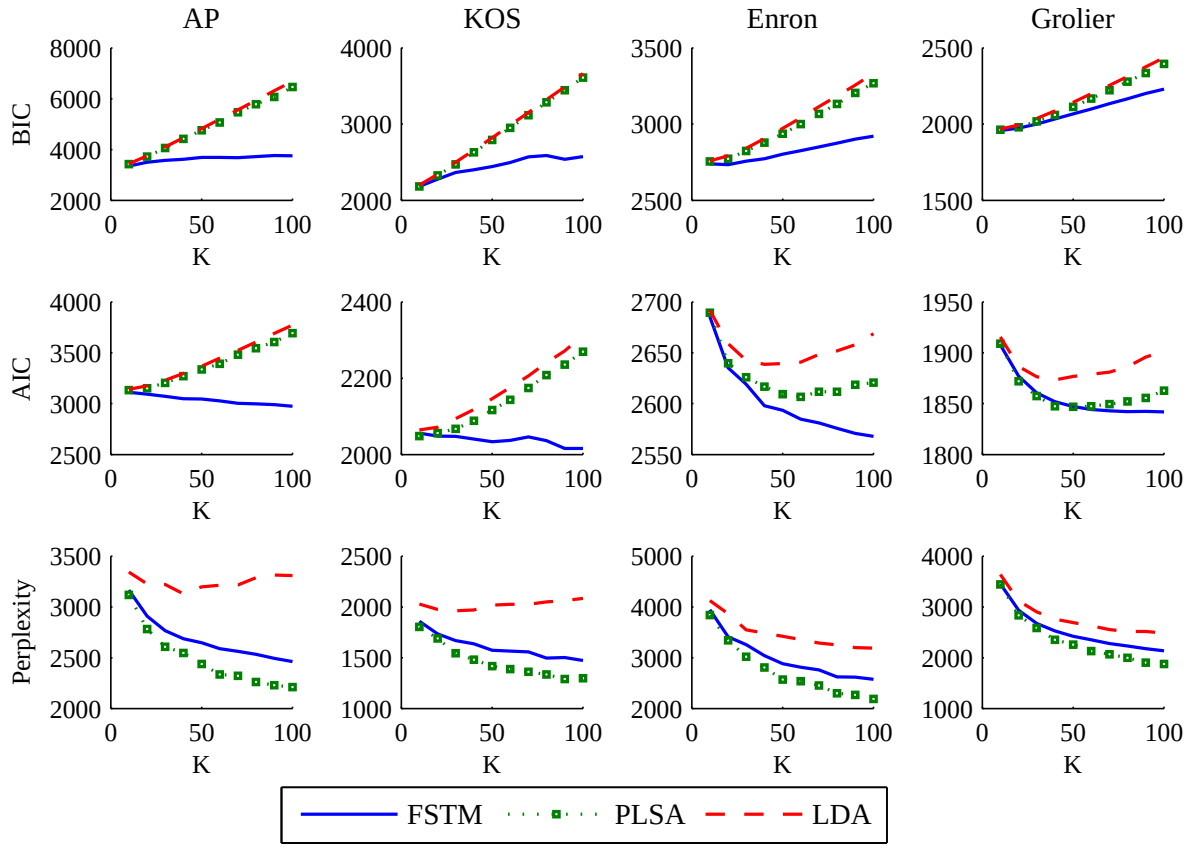


Figure 4.3: Quality of three models as the number of topics increases. Lower is better.

Next, we use three measures to quantify the quality: *Bayesian Information Criterion* (BIC), *Akaike Information Criterion* (AIC) [39], and Perplexity [21]. BIC and AIC are popular measures for model selection in Machine Learning.[9] They measure both simplicity and goodness-of-fit of the considered models; the simpler is preferred when two models have comparable quality of fitting data. A model with larger BIC/AIC is more likely to overfit the data [39]. Perplexity is also a common measure in topic modeling literature to compare predictive power of different models.[10]

Figure 4.3 presents the quality of three models on four corpora. (STC was not included in this investigation, because the objective function in learning is a regularized one, and hence different in manner with probabilistic topic models.) Observing the first two rows of the figure, one can easily realize that BIC and AIC of FSTM were significantly better than those of LDA and PLSA for most experiments. Note that FSTM can learn very sparse

[9] $AIC = (-2\log\mathfrak{L} + 2p)/M$, and $BIC = (-2\log\mathfrak{L} + p\log M)/M$, where $\mathfrak{L}$ is the achieved likelihood, and p is the number of free parameters of the model. Note that free parameters in the considered topic models basically correspond to the entries of topics, and one more for LDA. Hence $p = (V-1)K + 1$ for LDA, while $p - K$ for FSTM/PLSA is the number of non-zero entries of the learned topics.

[10] Perplexity of a model $\mathfrak{M}$ is calculated on the testing set $\mathcal{D}$ by $Perp(\mathcal{D}|\mathfrak{M}) = \exp\left(-\sum_{\boldsymbol{d}\in\mathcal{D}} \log P(\boldsymbol{d}|\mathfrak{M}) / \sum_{\boldsymbol{d}\in\mathcal{D}} |\boldsymbol{d}|\right)$.

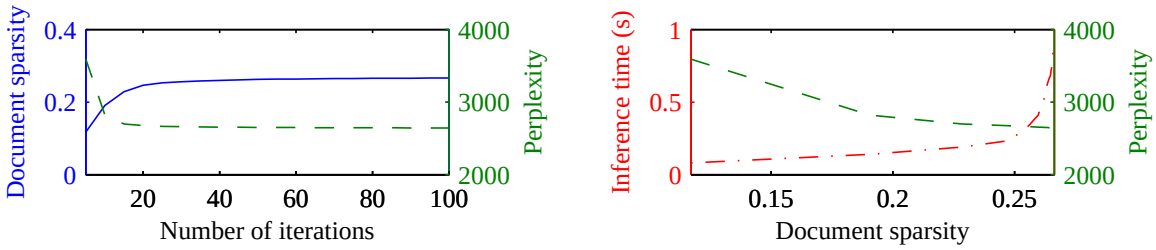


Figure 4.4: Illustration of trading off sparsity against quality and time. More iterations imply better quality, but probably denser topic proportions. Inference was done on AP, where FSTM had been learned with 50 topics.

topics as previously discussed. In addition, we observed that the likelihoods achieved by FSTM were often comparable with those by PLSA, while those by LDA were often worst. Hence FSTM was evaluated better than other models according to BIC/AIC. For PLSA and LDA, despite using more free parameters (dense topics) to model data, the achieved likelihoods were not very significantly greater than those of FSTM. Therefore, they are more likely prone to overfitting. The ability to avoid overfitting of FSTM in these experiments supports further the theoretical analysis in Section 4.3, where an implicit prior is argued to keep FSTM from overfitting.

The last row of Figure 4.3 shows perplexity obtained by three models. We observe that PLSA consistently achieved better perplexity than LDA and FSTM. This seems unusual since LDA is a Bayesian extension of PLSA and thus is often expected to have better predictive power. Nonetheless, in our observations, at least two factors had contributed to this inferior predictiveness: first, the variational Bayesian method [21] is not guaranteed to find good solutions; second, the objective of inference in LDA is posterior probability $P(\boldsymbol{\theta}|\boldsymbol{d})$, not the likelihood $P(\boldsymbol{d})$, while perplexity is mainly about likelihood. FSTM achieved good predictive power. The inference algorithm of FSTM played a crucial role in this good power, since it is guaranteed to find provably good solutions.

Trade-off: Figure 4.4 illustrates how FSTM trades off sparsity of solutions against inference quality (measured by perplexity) and running time. Unsurprisingly, more iterations means better quality but probably denser topic proportions. Note that the upper bound on inference error in Theorem 6 is quite loose. However, in practice inference converged very quickly, as observed in Figure 4.4. After 20 iterations on average, the quality and sparsity level were almost stable. We rarely observed inference needed more than 100 iterations to reach convergence. This is an interesting behavior of FSTM and is appealing to resolving large-scale settings.

4.4.3 Classification

Next we investigate how well FSTM works in practical applications. Classification is a traditional basic problem in Machine Learning. When applying topic models to classification, it is common that unsupervised topic models often play the role as dimensionality reduction. That is, we employ topic models to learn a new representation of data in the topical space (of lower dimensions); then learning a classifier and classifying new documents are done on the reduced representation. We want to see how well FSTM keeps important information of data for classification when doing dimensionality reduction.

To this end, we took 20Newsgroups and Webspam in consideration. Webspam is a large dataset for large-scale classification, and hence we will deal with it later. For comparison with PLSA and LDA, only 20Newsgroup was selected as it can be handled by the learning algorithms of PLSA [51] and LDA [21]. With the same settings for topic models as described before, and using the same classification algorithm [38], Figure 4.4.3 shows the accuracy as the number of topics increases.

Observing Figure 4.4.3, we realize that FSTM performed better than PLSA and LDA in most cases. The achievement of FSTM was sometimes 6% improvement over those of PLSA and LDA. Such a practical behavior demonstrates that FSTM can learn meaningful representation of data. There are at least two reasons for the good performance of FSTM. First, as investigated before, FSTM can generalize well on unseen data while keeping low model complexity and comparable predictive power. Second, while inferring topic proportions for documents, FSTM does simultaneously two nice jobs: feature extraction and selection. Feature space here is the topical space, whereas selection relates to sparse inference by the Frank-Wolfe algorithm. The ability to infer sparse solutions, which are provably good, implies that inference in FSTM does feature selection. Such a selection is local for each document. Figure 4.4.3 shows that FSTM always uses very few features to represent documents whereas PLSA and LDA use most features. The better performance in terms of classification accuracy illustrates that FSTM can do feature selection considerably well.

4.5 Large-scale learning

We have seen that FSTM consistently inferred very sparse representations of documents and learned sparse topics. It has a linear time inference algorithm, and learning of topics amounts to multiplication of two sparse matrices. Therefore, those characteristics con-

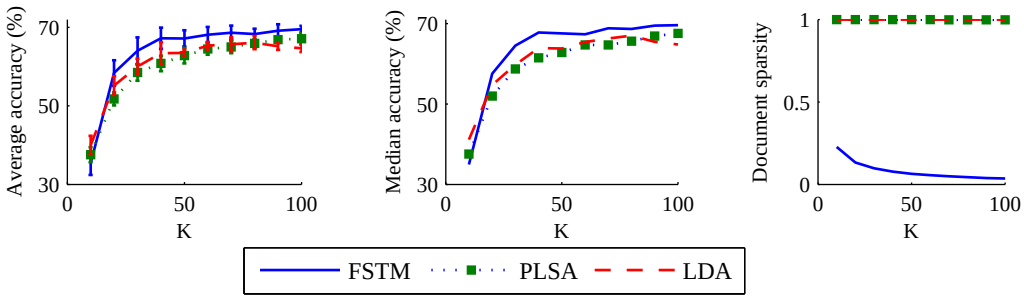


Figure 4.5: Classification on 20Newsgroups when topic models do dimensionality reduction. On the right shows how sparse the learned topic proportions are. We see that FSTM used few features to represent documents while PLSA and LDA used most features. For each number K of topics, document sparsity is averaged over 10 random runs.

tribute crucially to the scalability of FSTM. Moreover, we observe in Section 4.4 that FSTM works qualitatively in practice. Therefore, we take a further step towards dealing with very big models and large data.

In this section, we first describe a distributed architecture for learning large models from large corpora. To further speed up learning for FSTM, we discuss the use of warm-start for inference of documents. Empirical experiments in subsection 4.5.2 show that such a technique boosts learning speed significantly while maintaining comparable quality. Finally, we discuss some attractive results when learning FSTM with more than 33 billions of variables from Webspam, and then doing large-scale classification.

4.5.1 A distributed architecture

We describe a distributed architecture, which is implemented using OpenMP, for large-scale learning. Even though OpenMP is a shared memory model, we employ both data parallelism and task parallelism schemes. This is possible because the learning algorithm for FSTM is naturally distributable. Specifically,

- CPUs are grouped into clusters.
- There is a master which plays the key role as globally updating topics.
- Data is distributed across clusters.
- Each cluster c has its own subset $\mathcal{C}_c$ of data and subtopics. The cluster mainly does inference for its data.
- Communication of a cluster with the master is only its subtopics.

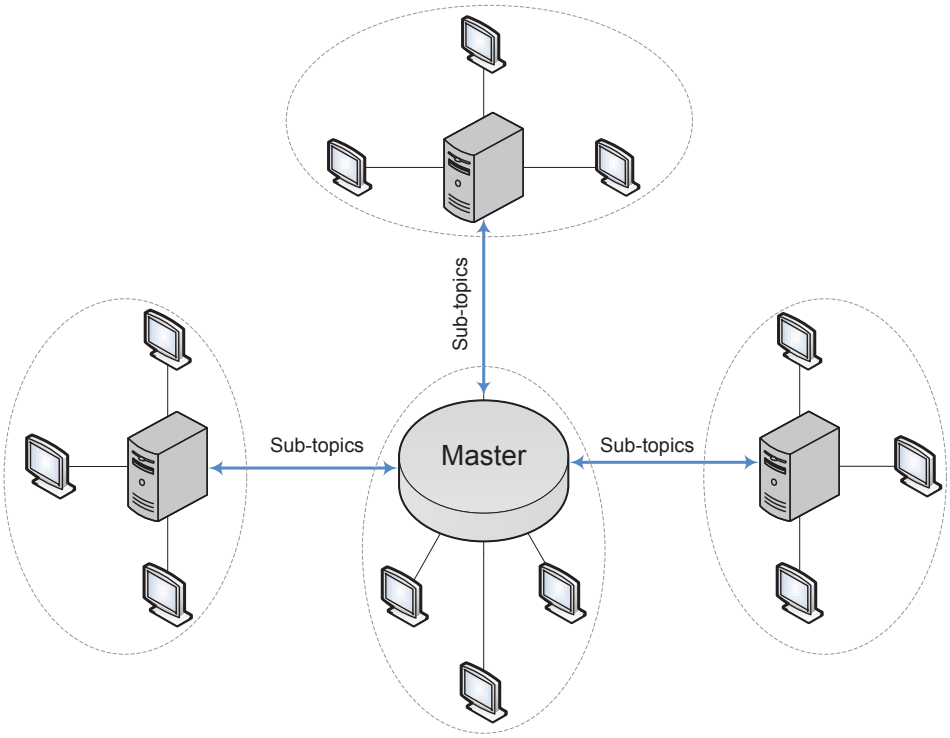


Figure 4.6: A distributed architecture for learning FSTM from large data.

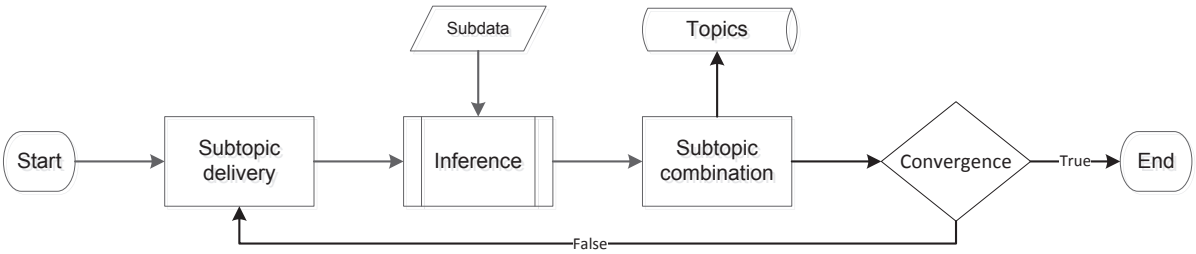


Figure 4.7: Workflow of the EM algorithm on the distributed architecture.

Here a subtopics refers to the parts of topics which are necessary to do inference for documents of the associated cluster. Note that topics in FSTM are often very sparse. Hence communication of subtopics are often compact. Figure 4.6 shows the proposed architecture.

The workflow of the learning process is shown in Figure 4.7. Learning FSTM is to repeat the EM iterations until convergence. Data is distributed to clusters before starting learning. Each EM iteration consists of the following steps:

- All clusters retrieve necessary subtopics from the master.
- Each cluster c then does inference for its data, $\mathcal{C}_c$, in parallel.

- After inference, each cluster c computes the statistic $a^c_{kj} = \sum_{\boldsymbol{d} \in \mathcal{C}_c} d_j \theta_{dk}$, and then sends it to the master.

- The master collects statistics from all clusters and then reconstructs the global topics as $\beta_{kj} \propto \sum_c a^c_{kj}$.

In our implementation, the number of clusters can be decided by the users. The more clusters, the less data in each cluster and hence the faster inference. However, the overall memory to store subtopics will increase consequently. Each cluster may have many CPUs and thus parallel computation is exploited to do inference.

4.5.2 Boosting inference with warm-start

Warm-start is a popular technique for iterative algorithms. The core idea is to exploit results of previous steps to improve computation/quality of the current step. A suitable exploitation can help us significantly reduce unnecessary computations in iterative algorithms. We use this technique to further improve the learning speed for FSTM.

Our focus is on doing inference for documents. Note that for each EM iteration, we have to do inference for each document individually. Hence a slight improvement for the inference algorithm would be significant, especially for large data. Our idea to reduce computations for inference is a novel replacement of the initial step of the inference algorithm. For a document $\boldsymbol{d}$, instead of choosing a vertex in the simplex of topics, we select some of the topics appearing in $\boldsymbol{\theta}_d$, which has been inferred in the previous E-step. Such an inheritance amounts to doing many steps in the inference algorithm. As a consequence, we could save many computations for doing inference of a document. In our implementation, at the initial step we remove all components of $\boldsymbol{\theta}_d$ except ones which are greater than a prescribed threshold ξ. From experiments, we find that $\xi = 0.001$ works well.

Figure 4.8 shows some results when using warm-start. After 3 iterations, the time to do an EM step decreases considerably as the number of iterations increases. Note that the improvement over the original algorithm is often considerable. Meanwhile, the quality in terms of likelihood is maintained comparably. Note further that the speed of reaching convergence did not change even when warm-start is used. These phenomena were also observed on other corpora. Besides, we find that when using warm-start, the affect on sparsity of topics and topic proportions is negligible. Hence, those observations support for the reliability of using warm-start to speed up learning.

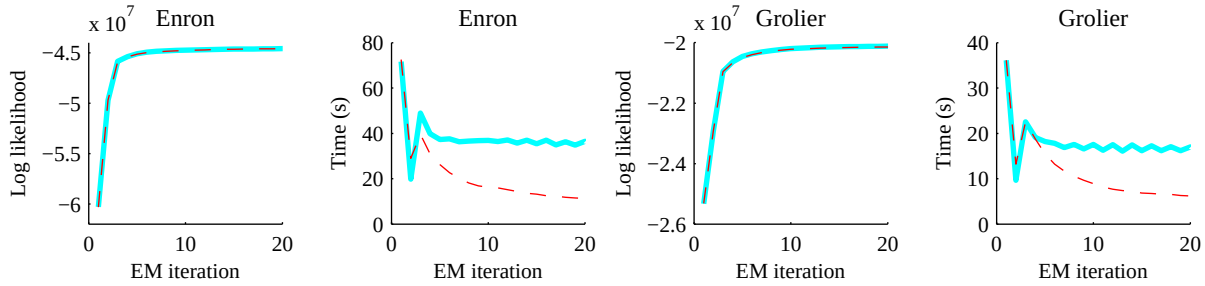


Figure 4.8: Quality and time as the number of EM iterations increase, when $K = 100$. Bold lines show performance of the original learning algorithm. Dash lines show performance when warm-start is used.

Table 4.4: Results of learning FSTM from Webspam.

Number of topics	1000	2000
Time per EM iteration	28 minutes	65 minutes
EM iterations to reach convergence	17	16
Topic sparsity (compared with dense models)	0.0165 (60 times smaller)	0.0114 (87 times smaller)
Document sparsity (compared with dense models)	0.0054 (185 times smaller)	0.0028 (357 times smaller)
Storage for the new representation ($\boldsymbol{\theta}$) (compared with the original corpus)	31.5 Mb (757 times smaller)	33.2 Mb (718 times smaller)
Average length of topic proportions, $\bar{s}$ (compared with dense representations)	5.4 (185 times smaller)	5.6 (357 times smaller)

4.5.3 Large-scale experiments and classification

We now demonstrate the scalability of our implementation on large data of very high dimensions. Webspam is selected for experiments. This corpus consists of 350K documents with more than 16 millions of terms, and hence serves well our purpose.

We learn FSTM with 1000 and 2000 topics, respectively. We emphasize that with 2000 topics, the model will consist of more than 33 billions of latent variables. Learning such a model is really nontrivial. With the same setting, dense models such as LDA or PLSA would pose various difficulties for storage, since all those 33 billions of variables consume at least 130Gb of memory. Topics in FSTM are often very sparse, and thus is manageable.

128 CPUs (each with 2.9 GHz) were used and grouped into 32 clusters. Each cluster had to process about 11000 documents. Results of learning are presented in Table 4.4. Observing the table, we find that learning reached convergence quickly. Topics and topic proportions are very sparse. Note that the average number of topics contributing to

Table 4.5: Large-scale classification on Webspam. Though reducing the dimensionality drastically, the quality of classification is still comparably maintained.

Data	Dimensions	Storage	Accuracy	Classified by
Original Webspam	16609143	23.3 Gb	*99.15*%	BMD [Yu et al. 2012]
When reducing dimensionality with FSTM				
1000 topics	1000	31.5 Mb	*98.877*%	FSTM + Liblinear
2000 topics	2000	33.2 Mb	*99.146*%	FSTM + Liblinear

a documents changes insignificantly when increasing the number of topics from 1000 to 2000. This behavior is consistent with the fact that a document often talks about only few topics, independent of how many topics the corpus is modeled. Since topic proportions are very sparse, storage for the new representation of data is hundreds of times smaller than the original one (23Gb).

Since Webspam is a supervised dataset, we conducted experiments for classification either. We use the new representation of the corpus previously learned by FSTM to be the input for Liblinear [38], resulting in FSTM + Liblinear method for classification where FSTM plays the role as a dimensionality reduction subroutine. Using 5-folds cross-validation and default settings for Liblinear, the results are shown in Table 4.5. One can easily realize that in both cases, 1000 and 2000 topics, the average accuracy is very good and is comparable with that by the most recent advanced method [124] which did classification on the original data with dimensionality of 16 millions. These promising results imply that information was not significantly lost when reducing dimensions from 16 millions to 2000. Note that FSTM used only few features to represent documents. These observations suggest that FSTM can do well feature selection.

The promising result on large-scale classification and the good performance on average data, investigated in Section 4.4, suggest that FSTM can work well in practice and can infer very meaningful representations of documents. As a result, FSTM can provide us a useful tool, not only a model of linguistic data but also a promising dimensionality reduction approach, to efficiently deal with large-scale settings.

4.6 Summary

We have introduced *fully sparse topic model* (FSTM) for modeling large collections of documents with very high dimensionality. FSTM overcomes many limitations of existing topic models, and has been demonstrated to work qualitatively on real data. The

scalability of our model enables us to easily deal with large-scale settings.

Learning algorithms which are independent of dimensionality of the problems of interest are highly desired. Those dimension-free algorithms are even much more crucial when facing with data of extremely high dimensions. A discovery of such algorithms would be of practically significant and would make a great progress for Machine Learning and Data Mining. The near dimension-free learning algorithm, developed in this work, provides an evidence for the existence of such dimension-free algorithms. Exploitation of sparsity of data seems to be one of the keys when seeking such algorithms.

An implementation of FSTM is freely available at `www.jaist.ac.jp/~s1060203/codes/fstm`.

Chapter 5

Probable convexity and application to Correlated Topic Models

In this chapter we make a fresh view on posterior estimation in probabilistic models by looking at the "practical properties" of the problem. Chapter 3 introduces the FW framework for speedily doing a posteriori inference of topic mixtures. However, FW is limited to models that the inference problem is naturally concave. For many topic models, posterior estimation may not be concave and hence precludes the use of FW. This chapter introduces a new framework for this situation, and then study CTM and related nonconjugate models.

5.1 Introduction

Estimation of posterior distributions plays a central role when developing probabilistic graphical models. With conjugate priors, we are likely able to derive efficient sampling algorithms for estimation [44, 78]. When nonconjugate priors are used, the estimation problem is much more difficult, as observed in the topic modeling literature by Ahmed and Xing [4], Blei and Lafferty [18, 20], Putthividhya et al. [80, 81], Salomatin et al. [86]. A popular approach is to cast estimation as an optimization problem. Nonetheless, the resulting problems are often non-convex. Non-convexity poses various obstacles for designing efficient algorithms, and does not allow us to directly exploit the nice theory of convex optimization.

In this work, we introduce the framework of *probable convexity* that aims at two targets: (1) to reveal how hard an optimization problem in practice is; (2) to support us smoothly

employ efficient methods of convex optimization to deal with non-convex problems. The probable convexity of a family $\mathfrak{F}$ of real functions essentially says that most members of $\mathfrak{F}$ are convex. With such families, in practice we probably rarely meet non-convex functions from $\mathfrak{F}$. We remark that in many situations (e.g., posterior estimation in graphical models) we often has to deal with not only one but many members of a family at once. Hence some appearances of non-convex members may not affect significantly the overall result. Hence a direct employment of convex optimization is possible and beneficial. In other words, we could do minimization efficiently for functions of $\mathfrak{F}$ in practice.

We next use the framework to investigate estimation of posterior distributions in the *Correlated Topic Model* (CTM) [18] and related non-conjugate models. In particular, we study the problem of a posteriori estimating theta (topic mixture) for a given document: $\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \Pr(\boldsymbol{\theta}|\boldsymbol{d})$. This is an MAP problem and is intractable for many models in the worst case [91]. We show that under certain conditions, the objective function of this MAP problem is in fact *probably concave*, i.e., concave with high probability. This suggests that posterior estimation of theta may be tractable in practice. Similar results are obtained for related nonconjugate topic models.

The cornerstone of our analyses of nonconjugate models is the logistic-normal function which originates from the logistic-normal distribution [5]. We show in this work that the logistic-normal function is probable concave under certain conditions. This result may be of interest elsewhere and beneficial in practical applications, because the logistic-normal distribution is used as an effective prior in many contexts including topic modeling [18, 20, 64, 80, 81, 86] and grammar induction [29, 30].

As a consequence of our analysis, a novel algorithm for learning CTM is proposed. This algorithm is surprisingly simple in which posterior estimation of theta is done by Online Frank-Wolfe [47]. From empirical experiments we find that the new algorithm is significantly faster than existing ones, while maintaining or making better the quality of the learned models. This further suggests that even though MAP inference for CTM is intractable in the worst case, most instances in practice may be resolved efficiently.

Organization: We present the concepts of probable convexity in Section 5.2. Section 5.3 presents our analysis of the logistic-normal function. The study of CTM and related nonconjugate models is presented in Section 5.4. The new algorithm for learning CTM and experimental results are discussed in Section 5.5. The final section is for further discussion and conclusion.

5.2 Probable convexity

Let $\mathfrak{F}(x;a)$ be a family of real functions defined on a set $X \subset \mathbb{R}^K$, parameterized by a. Each value of a determines a function $f(x;a)$ of $\mathfrak{F}(x;a)$.

Definition 6 (probable convexity)**.** *Let $\mathfrak{F}(x;a)$ be a family of functions defined on a set $X \subset \mathbb{R}^K$, parameterized by a. Family $\mathfrak{F}(x;a)$ is said to be* probably convex *if there exists a positive constant p such that any element of $\mathfrak{F}(x;a)$ is convex on X with probability at least p. Equivalently, $\mathfrak{F}(x;a)$ is said to be* p-convex *if any element of $\mathfrak{F}(x;a)$ is convex on X with probability at least p.*

By definition, a family of convex functions is probably convex with probability 1. The family $\mathfrak{F}(x;a,b,c) = \{ax^2 + bx + c : a,b,c \in \mathbb{R}\}$ is probably convex with probability 1/2, since convexity of this family is decided by the sign of a.

Definition 7 (almost sure convexity)**.** *Let $\mathfrak{F}(x;a)$ be a family of functions defined on a set $X \subset \mathbb{R}^K$, parameterized by a. Family $\mathfrak{F}(x;a)$ is said to be* almost surely convex *if any element of $\mathfrak{F}(x;a)$ is convex on X with probability* 1.

It is easy to see that a family of convex functions is almost surely convex. By definition, the family $\mathfrak{F}(x;a,b,c)$ is not almost surely convex. If a family is almost surely convex, almost all of its members are convex.

A family $\mathfrak{F}(x;a)$ is said to be *p-concave* if the family $-\mathfrak{F}(x;a) = \{-f(x;a) : f(x;a) \in \mathfrak{F}(x;a)\}$ is p-convex. One can easily realize that if $\mathfrak{F}(x;a)$ is *p-concave*, then $-\mathfrak{F}(x;a)$ is *p-convex* and vice versa.

The concept of probable convexity applies equally to the cases of only one function. A function $f(x)$ is said to be p-convex in X if it is convex in X with probability at least p. Similarly, function $f(x)$ is said to be p-concave in X if it is concave in X with probability at least p.

Convex optimization refers to minimizing a convex function over a convex domain. It is also refers to maximizing a concave function over a convex domain. It has a long history and has a rich foundation. Convex problems are often considered as being easy since there exist various fast algorithms. The book by Boyd and Vandenberghe [24] provides an excellent introduction to the field.

5.3 Concavity of the logistic-normal function

We first consider probable convexity of the following function which is called *logistic-normal*:

$$LN(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = -\frac{1}{2}(\log\tilde{\boldsymbol{x}}-\boldsymbol{\mu})^t\boldsymbol{\Sigma}^{-1}(\log\tilde{\boldsymbol{x}}-\boldsymbol{\mu}) - \sum_{k=1}^{K}\log x_k, \tag{5.1}$$

where $\boldsymbol{\mu}\in\mathbb{R}^{K-1}, \boldsymbol{\Sigma}\in\mathbb{S}_+^{K-1}$; $\boldsymbol{x}\in\overline{\Delta}_K$ such that $\log\tilde{\boldsymbol{x}}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$. This function naturally originates from the logistic-normal distribution [5], whose density is $p(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})\propto\exp(LN(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}))$. Due to the broad use of this distribution in probabilistic modeling, the logistic-normal function plays an important role in many contexts. Nonetheless, the function itself is neither convex nor concave in $\overline{\Delta}_K$. This is one of the main reasons for why posterior estimation in nonconjugate models is often intractable.

By a thorough analysis of this function, we found the following property.

Theorem 8. *Denote $p = 1-e^{2\log(K-1)-0.5(\lambda-1)^2/\sigma}$ for $\lambda=\lambda_{K-1}(\boldsymbol{\Sigma}^{-1})$ and $\sigma=\max_i\Sigma_{ii}^{-1}$. Function $LN(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$ is p-concave over $\overline{\Delta}_K$ if $\lambda\geq 1$.*

This theorem essentially says that LN is in fact concave under some conditions. Note that the quantity $(\lambda-1)^2/\sigma$ is not always small. Indeed, letting $\lambda_k(\boldsymbol{\Sigma}^{-1})$ be the kth eigenvalue of $\boldsymbol{\Sigma}^{-1}$, we have $\mathrm{Tr}\,(\boldsymbol{\Sigma}^{-1})=\sum_{k=1}^{K-1}\lambda_k(\boldsymbol{\Sigma}^{-1})=\sum_{k=1}^{K-1}\Sigma_{kk}^{-1}$. When the condition number of $\boldsymbol{\Sigma}^{-1}$ is not large, $\lambda_{K-1}(\boldsymbol{\Sigma}^{-1})$ and σ may be of the same order. This observation suggests that the probability bound obtained in Theorem 8 is significant.

Corollary 5. *With notations as in Theorem 8, function $LN(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$ is almost surely concave as $\lambda^2/\sigma\to+\infty$.*

In the case that the least eigenvalue λ is much larger than $\log(K-1)$, function LN is concave with high probability. More concretely, if $\lambda^2=\omega(\sigma\log K)$, i.e., $\lambda^2/\sigma\log K\to+\infty$ as $K\to+\infty$, then $\exp\{2\log(K-1)-0.5(\lambda-1)^2/\sigma\}$ goes to 0. Hence the following result holds.

Corollary 6. *With notations as in Theorem 8, assume that $\lambda^2=\omega(\sigma\log K)$. Function $LN(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$ is almost surely concave as $K\to+\infty$.*

5.3.1 Proof of Theorem 8

We will show probable concavity of LN by investigating concavity in common sense. Note that the domain $\overline{\Delta}_K$ is convex, and function LN is twice differentiable over $\overline{\Delta}_K$. Hence, to see concavity, it suffices to show that the second derivative is negative semidefinite [24].

Let Σ_i^{-1} be the ith row of $\boldsymbol{\Sigma}^{-1}$. The first and second partial derivatives of the function w.r.t the variables are:

$$\frac{\partial LN}{\partial x_i} = \begin{cases} -\frac{1}{x_i}\Sigma_i^{-1}(\log\tilde{\boldsymbol{x}}-\boldsymbol{\mu})-\frac{1}{x_i}, & i<K \\ \frac{1}{x_K}\sum_{h=1}^{K-1}\Sigma_h^{-1}(\log\tilde{\boldsymbol{x}}-\boldsymbol{\mu})-\frac{1}{x_K}, & i=K \end{cases}$$

$$\frac{\partial^2 LN}{\partial x_i \partial x_j} = \begin{cases} -\frac{\Sigma_{ij}^{-1}}{x_i x_j}, & i<K, i\neq j, j<K \\ \frac{1}{x_i^2}\Sigma_i^{-1}(\log\tilde{\boldsymbol{x}}-\boldsymbol{\mu})-\frac{\Sigma_{ii}^{-1}}{x_i^2}+\frac{1}{x_i^2}, & i<K, i=j \\ \frac{1}{x_i x_K}\sum_{h=1}^{K-1}\Sigma_{ih}^{-1}, & i<K, j=K \\ \frac{1}{x_j x_K}\sum_{h=1}^{K-1}\Sigma_{hj}^{-1}, & i=K, j<K \\ -\frac{1}{x_K^2}\sum_{h=1}^{K-1}\Sigma_h^{-1}(\log\tilde{\boldsymbol{x}}-\boldsymbol{\mu})-\frac{1}{x_K^2}\sum_{h=1}^{K-1}\sum_{t=1}^{K-1}\Sigma_{ht}^{-1}+\frac{1}{x_K^2}, & i=j=K. \end{cases}$$

Denote $\boldsymbol{S} = \begin{pmatrix} \boldsymbol{\Sigma}^{-1} & \boldsymbol{s}_K^t \\ \boldsymbol{s}_K & s_{KK} \end{pmatrix}; \boldsymbol{U} = \begin{pmatrix} \boldsymbol{\Sigma}^{-1} \\ \boldsymbol{s}_K \end{pmatrix}$, where $\boldsymbol{s}_K = -\sum_{t=1}^{K-1}\Sigma_t^{-1}$ is the sum of the rows of $\boldsymbol{\Sigma}^{-1}$, and s_{KK} is the sum of all elements of $\boldsymbol{\Sigma}^{-1}$. We can express the second derivative of LN as

$$\begin{aligned} LN'' &= diag\frac{1}{\boldsymbol{x}}.diag[\boldsymbol{U}(\log\tilde{\boldsymbol{x}}-\boldsymbol{\mu})].diag\frac{1}{\boldsymbol{x}} - diag\frac{1}{\boldsymbol{x}}.\boldsymbol{S}.diag\frac{1}{\boldsymbol{x}} + diag\frac{1}{\boldsymbol{x}}.diag\frac{1}{\boldsymbol{x}} \\ &= diag\frac{1}{\boldsymbol{x}}.\left(\boldsymbol{I}_K - \boldsymbol{S} + diag[\boldsymbol{U}(\log\tilde{\boldsymbol{x}}-\boldsymbol{\mu})]\right).diag\frac{1}{\boldsymbol{x}}. \end{aligned} \tag{5.2}$$

A classical result in Algebra [1, exercise 8.28] says that for any symmetric $\boldsymbol{A}$ and nonsingular $\boldsymbol{Y}$, the product $\boldsymbol{Y}\boldsymbol{A}\boldsymbol{Y}^t$ is positive semidefinite if and only if $\boldsymbol{A}$ is positive semidefinite. Consequently, the matrix $\boldsymbol{I}_K - \boldsymbol{S} + diag[\boldsymbol{U}(\log\tilde{\boldsymbol{x}}-\boldsymbol{\mu})]$ decides negative semidefiniteness of LN''.

Lemma 6. *Denote $\boldsymbol{z} = \boldsymbol{\Sigma}^{-1}(\log\tilde{\boldsymbol{x}}-\boldsymbol{\mu})$. LN'' is negative semidefinite if $z_1+\cdots+z_{K-1}\geq 1$ and $\boldsymbol{I}_{K-1} - \boldsymbol{\Sigma}^{-1} + diag(\boldsymbol{z}) \leq 0$.*

Proof. As discussed before, matrix $\boldsymbol{I}_K - \boldsymbol{S} + diag[\boldsymbol{U}(\log\tilde{\boldsymbol{x}}-\boldsymbol{\mu})]$ decides negative definiteness of LN''. Letting $z_K = -z_1 - \cdots - z_{K-1}$ and $\boldsymbol{1} = (1,...,1)^t \in \mathbb{R}^{K-1}$, we have

$$
\begin{aligned}
\boldsymbol{A} &= \boldsymbol{I}_K - \boldsymbol{S} + diag[\boldsymbol{U}(\log \tilde{\boldsymbol{x}} - \boldsymbol{\mu})] \\
&= I_K - (\boldsymbol{I}_{K-1} \;\; \boldsymbol{1})^t \boldsymbol{\Sigma}^{-1} (\boldsymbol{I}_{K-1} \;\; \boldsymbol{1}) + diag(z_1, ..., z_K) \\
&= (\boldsymbol{I}_{K-1} \;\; \boldsymbol{1})^t \left[\boldsymbol{I}_{K-1} - \boldsymbol{\Sigma}^{-1} + diag(\boldsymbol{z})\right] (\boldsymbol{I}_{K-1} \;\; \boldsymbol{1}) + \begin{pmatrix} \boldsymbol{0} & -(\boldsymbol{z}+\boldsymbol{1}) \\ -(\boldsymbol{z}+\boldsymbol{1})^t & z_K + 1 \end{pmatrix} \quad (5.3)
\end{aligned}
$$

Consider the last term $\boldsymbol{C} = \begin{pmatrix} \boldsymbol{0} & -(\boldsymbol{z}+\boldsymbol{1}) \\ -(\boldsymbol{z}+\boldsymbol{1})^t & z_K + 1 \end{pmatrix}$. This matrix is of size $K \times K$, but has rank 2. It is not hard to see that all principle minors of $\boldsymbol{C}$ are 0, except the ones which associate with the last two rows and columns. Those principle minors are $z_K + 1$ and $\begin{vmatrix} 0 & -z_i - 1 \\ -z_i - 1 & z_K + 1 \end{vmatrix} = z_K + 1 - (z_i + 1)^2$ for $i \in \{1, ..., K-1\}$. According to a classical result in Algebra [1, exercise 8.32], $\boldsymbol{C} \leq 0$ if and only if all of its principle minors are non-positive. Therefore $\boldsymbol{C} \leq 0$ if and only if $z_K + 1 \leq 0$.

If $\boldsymbol{C}$ and $\boldsymbol{I}_{K-1} - \boldsymbol{\Sigma}^{-1} + diag(\boldsymbol{z})$ are negative semidefinite, so are $\boldsymbol{A}$ and LN''. This suggests that if $z_K + 1 \leq 0$ and $\boldsymbol{I}_{K-1} - \boldsymbol{\Sigma}^{-1} + diag(\boldsymbol{z}) \leq 0$, then $LN'' \leq 0$ which completes the proof. (Q.E.D.)

Next we want to see under what conditions, matrix $\boldsymbol{I}_{K-1} - \boldsymbol{\Sigma}^{-1} + diag(\boldsymbol{z}) \leq 0$ with the constraint of $z_1 + \cdots + z_{K-1} \geq 1$. The following theorem reveals a property whose detailed proof is presented in section 5.3.2.

Theorem 9. *Let $\boldsymbol{z}$ be a Gaussian random variable with mean 0 and covariance matrix $\boldsymbol{A} \in \mathbb{S}_+^{K-1}$, and $\sigma = \max_i A_{ii}$. For a fixed $\boldsymbol{S} \in \mathbb{S}_+^{K-1}$, consider $\boldsymbol{B} = \boldsymbol{I}_{K-1} - \boldsymbol{S} + diag(\boldsymbol{z})$. Assuming $\lambda_{K-1}(\boldsymbol{S}) \geq 1$, we have*

$$
\Pr(\lambda_1(\boldsymbol{B}) \geq 0 | z_1 + \cdots + z_{K-1} \geq 1) \leq \exp\left\{2\log(K-1) - 0.5(1 - \lambda_{K-1}(\boldsymbol{S}))^2/\sigma\right\}.
$$

This theorem essentially says that under certain assumption, matrix $\boldsymbol{B}$ is negative semidefinite with probability at least $1 - \exp\left\{2\log(K-1) - 0.5(1 - \lambda_{K-1}(\boldsymbol{S}))^2/\sigma\right\}$. Hence we have enough tools to prove Theorem 8.

Proof of Theorem 8. Consider the logistic-normal function $LN(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, and denote $\lambda = \lambda_{K-1}(\boldsymbol{\Sigma}^{-1})$ and $\sigma = \max_i \Sigma_{ii}^{-1}$. As discussed before, concavity of this function over $\overline{\Delta}_K$ is decided by its second partial derivative LN''. Lemma 6 suggests that $LN(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is concave if $z_1 + \cdots + z_{K-1} \geq 1$ and $\boldsymbol{I}_{K-1} - \boldsymbol{\Sigma}^{-1} + diag(\boldsymbol{z}) \leq 0$, where $\boldsymbol{z} = \boldsymbol{\Sigma}^{-1}(\log \tilde{\boldsymbol{x}} - \boldsymbol{\mu})$.

Note that $\mathbb{E}\boldsymbol{z} = 0$ and $cov(z) = \boldsymbol{\Sigma}^{-1}$ since $\mathbb{E}\log\tilde{\boldsymbol{x}} = \boldsymbol{\mu}$ and $cov(\log\tilde{\boldsymbol{x}}) = \boldsymbol{\Sigma}$. Theorem 9 implies that with the constraint of $z_1 + \cdots + z_{K-1} \geq 1$, $\boldsymbol{I}_{K-1} - \boldsymbol{\Sigma}^{-1} + diag(\boldsymbol{z}) \leq 0$ holds with probability at least $1 - \exp\left\{2\log(K-1) - 0.5(1-\lambda)^2/\sigma\right\}$ if $\lambda \geq 1$. This means assuming $\lambda \geq 1$, function $LN(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is concave with probability at least $1 - \exp\left\{2\log(K-1) - 0.5(1-\lambda)^2/\sigma\right\}$. (Q.E.D.)

5.3.2 Proof of Theorem 9

To prove this theorem we need the following result.

Lemma 7. *Consider a Gaussian random vector $\boldsymbol{z}$ with mean 0 and covariance matrix $\boldsymbol{A} \in \mathbb{S}_+^K$. Let $\sigma_i = A_{ii}$ be the ith diagonal entry of $\boldsymbol{A}$, and $\sigma = \max_i \sigma_i$. Then for any real $a > 0$, we have $\mathbb{E}\,\mathrm{Tr}\, e^{a.diag(\boldsymbol{z})} = \sum_{k=1}^K e^{a^2\sigma_k/2} \leq Ke^{a^2\sigma/2}$.*

Proof. Note that

$$\begin{aligned}
\mathrm{Tr}\, e^{a.diag(\boldsymbol{z})} &= \mathrm{Tr} \sum_{i=0}^{\infty} \frac{a^i}{i!} diag^i(\boldsymbol{z}) \\
&= \mathrm{Tr} \sum_{i=0}^{\infty} \frac{a^i}{i!} diag(z_1^i, ..., z_K^i) \\
&= \sum_{i=0}^{\infty} \frac{a^i}{i!} \mathrm{Tr}\, diag(z_1^i, ..., z_K^i) \\
&= \sum_{i=0}^{\infty} \frac{a^i}{i!} \sum_{k=1}^{K} z_k^i = \sum_{k=1}^{K} \sum_{i=0}^{\infty} \frac{a^i}{i!} z_k^i = \sum_{k=1}^{K} e^{az_k}
\end{aligned}$$

Hence $\mathbb{E}\,\mathrm{Tr}\, e^{a.diag(\boldsymbol{z})} = \mathbb{E}\sum_{k=1}^K e^{a.z_k} = \sum_{k=1}^K \mathbb{E}e^{a.z_k}$.

By assumption, z_k is a Gaussian variable with mean 0 and variance σ_k. Using the generating function of Gaussian, we have $\mathbb{E}e^{a.z_k} = e^{a^2\sigma_k/2}$. So substituting these quantities into the expectation in the last paragraph completes the proof. (Q.E.D.)

Proof of Theorem 9. We have

$$\begin{aligned}
\Pr(\lambda_1(\boldsymbol{B}) \geq 0 | z_1 + \cdots + z_{K-1} \geq 1) &\leq \Pr(\lambda_1(\boldsymbol{B}) \geq 0) \\
&\leq \inf_{a>0} \left\{\mathbb{E}\,\mathrm{Tr}\, e^{a\boldsymbol{B}}\right\} \\
&\quad \text{(Laplace transform method)} \\
&= \inf_{a>0} \left\{\mathbb{E}\,\mathrm{Tr}\, e^{a[\boldsymbol{I}_{K-1} - \boldsymbol{S} + diag(\boldsymbol{z})]}\right\}
\end{aligned}$$

$$
\begin{aligned}
\Pr(\lambda_1(\boldsymbol{B}) \geq 0 | z_1 + \cdots + z_{K-1} \geq 1) &\leq \inf_{a>0} \left\{ \mathbb{E} \left(\operatorname{Tr} e^{a[\boldsymbol{I}_{K-1} - \boldsymbol{S}]} . \operatorname{Tr} e^{a.diag(\boldsymbol{z})} \right) \right\} \\
&\quad \text{(Corollary 1)} \\
&= \inf_{a>0} \left\{ \operatorname{Tr} e^{a[\boldsymbol{I}_{K-1} - \boldsymbol{S}]} . \mathbb{E} \operatorname{Tr} e^{a.diag(\boldsymbol{z})} \right\} \\
&\leq \inf_{a>0} \left\{ \operatorname{Tr} e^{a[\boldsymbol{I}_{K-1} - \boldsymbol{S}]} . (K-1) . e^{a^2\sigma/2} \right\} \\
&\quad \text{(Lemma 7)} \\
&= \inf_{a>0} \left\{ (K-1) . e^{a^2\sigma/2} . \operatorname{Tr} e^{a[\boldsymbol{I}_{K-1} - \boldsymbol{S}]} \right\} \\
&\leq \inf_{a>0} \left\{ (K-1) . e^{a^2\sigma/2} . K . \lambda_1 (e^{a[\boldsymbol{I}_{K-1} - \boldsymbol{S}]}) \right\} \\
&\leq \inf_{a>0} \left\{ (K-1)^2 . e^{a^2\sigma/2} . e^{\lambda_1 (a[\boldsymbol{I}_{K-1} - \boldsymbol{S}])} \right\} \\
&\quad \text{(Spectral mapping theorem)} \\
&= \inf_{a>0} \left\{ (K-1)^2 . e^{a^2\sigma/2} . e^{a - a\lambda_{K-1}(\boldsymbol{S})} \right\} \\
&= \inf_{a>0} \left\{ (K-1)^2 . e^{a^2\sigma/2 + a - a\lambda_{K-1}(\boldsymbol{S})} \right\} \\
&= (K-1)^2 \exp \left\{ - \frac{(1 - \lambda_{K-1}(\boldsymbol{S}))^2}{2\sigma} \right\} .
\end{aligned}
$$

Note that the last equality is obtained by minimizing the function $a^2 \frac{\sigma}{2} + a - a\lambda_{K-1}(\boldsymbol{S})$ for $a > 0$ conditioned on $1 \leq \lambda_{K-1}(\boldsymbol{S})$. (Q.E.D.)

5.4 MAP inference of topic mixtures in CTM

We next study convexity of a family originated from the topic modeling literature. In particular, we are interested in the problem of estimating topic mixtures (posterior distributions) in correlated topic models (CTM) [18]. This problem is intractable by traditional approaches [4, 18]. We will show that in fact this problem is tractable under some conditions, by showing probable concavity of the objective function.

The correlated topic model assumes that a corpus is composed from K topics $\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_K$, and a document $\boldsymbol{d}$ arises from the following generative process:

1. Draw $\boldsymbol{x} | \boldsymbol{\mu}, \boldsymbol{\Sigma} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$
2. For the nth word of $\boldsymbol{d}$:
   - draw topic assignment $z_{dn} | \boldsymbol{x} \sim \mathcal{M}(f(\boldsymbol{x}))$
   - draw word $w_{dn} | z_{dn}, \boldsymbol{\beta} \sim \mathcal{M}(\boldsymbol{\beta}_{z_{dn}})$.

where $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$; $\mathcal{M}(\cdot)$ is the multinomial distribution; $f(\boldsymbol{x})$ maps a natural parameterization of the topic proportion to the mean parameterization:

$$\boldsymbol{\theta} = f(\boldsymbol{x}) = \frac{e^{\boldsymbol{x}}}{\sum_{k=1}^{K} e^{x_k}}. \tag{5.4}$$

This logistic transformation maps a K-dimensional vector $\boldsymbol{x}$ to a $(K-1)$-dimensional vector $\boldsymbol{\theta}$. Hence various $\boldsymbol{x}$'s can correspond to a single $\boldsymbol{\theta}$. Fixing $x_K = 0$, the transformation (5.4) means that $\boldsymbol{\theta}$ follows the logistic-normal distribution [17]. According to Aitchison and Shen [5], the density function of $\boldsymbol{\theta}$ is thus

$$p(\boldsymbol{\theta}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{\sqrt{\det(2\pi\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\log\tilde{\boldsymbol{\theta}} - \boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\log\tilde{\boldsymbol{\theta}} - \boldsymbol{\mu}) - \sum_{k=1}^{K} \log\theta_k \right), \tag{5.5}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{K-1}, \boldsymbol{\Sigma} \in \mathbb{S}_+^{K-1}$. Note that $\boldsymbol{\theta}$ is derived from $\boldsymbol{x}$ by (5.4). Hence $\log\tilde{\boldsymbol{\theta}}$ is a normal random variable with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

One of the most interesting tasks in this model is the posterior estimation of topic mixtures for documents. More concretely, given the model parameters $\Upsilon = \{\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$, we are interested in the following problem for a given document $\boldsymbol{d}$:

$$\begin{aligned} \boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}\in\Delta_K} \Pr(\boldsymbol{\theta}|\boldsymbol{d}, \Upsilon) \\ &= \arg\max_{\boldsymbol{\theta}\in\Delta_K} \Pr(\boldsymbol{\theta}, \boldsymbol{d}|\Upsilon) \end{aligned} \tag{5.6}$$

Lemma 8. *Given a CTM model with parameters $\Upsilon = \{\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ and a document $\boldsymbol{d}$, the MAP problem (5.6) can be reformulated as*

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\overline{\Delta}_K} \sum_j d_j \log \sum_{k=1}^{K} \theta_k \beta_{kj} - \frac{1}{2}(\log\tilde{\boldsymbol{\theta}} - \boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\log\tilde{\boldsymbol{\theta}} - \boldsymbol{\mu}) - \sum_{k=1}^{K} \log\theta_k. \tag{5.7}$$

Proof. We have

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\Delta_K} \Pr(\boldsymbol{\theta}, \boldsymbol{d}|\Upsilon) = \arg\max_{\boldsymbol{\theta}\in\Delta_K} \log\Pr(\boldsymbol{\theta}, \boldsymbol{d}|\Upsilon) = \arg\max_{\boldsymbol{\theta}\in\Delta_K} \log\Pr(\boldsymbol{d}|\boldsymbol{\theta}, \Upsilon) + \log\Pr(\boldsymbol{\theta}|\Upsilon).$$

Note that $\Pr(\boldsymbol{d}|\boldsymbol{\theta}, \Upsilon) = \sum_j d_j \log \sum_{k=1}^{K} \theta_k \beta_{kj} + c$ for some constant c, and the density of the logistic-normal distribution is given in (5.5). Hence

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}\in\Delta_K} \sum_j d_j \log \sum_{k=1}^{K} \theta_k \beta_{kj} - \frac{1}{2}(\log\tilde{\boldsymbol{\theta}} - \boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\log\tilde{\boldsymbol{\theta}} - \boldsymbol{\mu}) - \sum_{k=1}^{K} \log\theta_k - \frac{1}{2}\log\det(2\pi\boldsymbol{\Sigma}) + c.$$

Since any point on the boundary of Δ_K makes the objective function undefined and hence is not optimal. Therefore, ignoring the boundary of Δ_K and the constant in the objective function completes the proof. (Q.E.D.)

Loosely speaking, Lemma 8 says that posterior estimation of topic mixtures in CTM is in fact an optimization problem. The objective function is well-defined on $\overline{\Delta}_K$. We would like to remark that this function is neither concave nor convex in general. Hence maximizing it over $\overline{\Delta}_K$ theoretically is intractable.

5.4.1 Some results

Let the model parameters $\Upsilon = \{\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$ be fixed, where $\boldsymbol{\beta}_k \in \Delta_V, \boldsymbol{\mu} \in \mathbb{R}^{K-1}, \boldsymbol{\Sigma} \in \mathbb{S}_+^{K-1}$. Consider the following family, parameterized by $\boldsymbol{d}$:

$$CTM(\boldsymbol{\theta}; \boldsymbol{d}, \Upsilon) = \{f(\boldsymbol{\theta}; \boldsymbol{d}, \Upsilon) : \boldsymbol{\theta} \in \overline{\Delta}_K, \log \tilde{\boldsymbol{\theta}} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})\}. \tag{5.8}$$

where $f(\boldsymbol{\theta}; \boldsymbol{d}, \Upsilon) = \sum_j d_j \log \sum_{k=1}^K \theta_k \beta_{kj} - \frac{1}{2}(\log \tilde{\boldsymbol{\theta}} - \boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\log \tilde{\boldsymbol{\theta}} - \boldsymbol{\mu}) - \sum_{k=1}^K \log \theta_k$. This family contains all possible instances of the problem (5.7). Hence, analyzing this family means analyzing the problem of estimating topic mixtures in CTM.

Consider a member $f(\boldsymbol{\theta}; \boldsymbol{d}, \Upsilon)$. Note that $\boldsymbol{d}$ and $\boldsymbol{\beta}$ are always nonnegative in practices of topic modeling. Hence the first term in $f(\boldsymbol{\theta}; \boldsymbol{d}, \Upsilon)$ is always concave over $\overline{\Delta}_K$. It implies that concavity of $f(\boldsymbol{\theta}; \boldsymbol{d}, \Upsilon)$ is heavily determined by the logistic-normal term $y = -\frac{1}{2}(\log \tilde{\boldsymbol{\theta}} - \boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\log \tilde{\boldsymbol{\theta}} - \boldsymbol{\mu}) - \sum_{k=1}^K \log \theta_k$. If this term is concave, then $f(\boldsymbol{\theta}; \boldsymbol{d}, \Upsilon)$ is concave. Combining these observations with Theorem 8, Corollary 5, and Corollary 6, we arrive at the following results for CTM.

Theorem 10. *Let Υ be fixed, $\sigma = \max_i \Sigma_{ii}^{-1}, \lambda = \lambda_{K-1}(\boldsymbol{\Sigma}^{-1})$, and $p = 1 - e^{2\log(K-1) - 0.5(\lambda-1)^2/\sigma}$. Assuming $\lambda \geq 1$, family $CTM(\boldsymbol{\theta}; \boldsymbol{d}, \Upsilon)$ is p-concave over $\overline{\Delta}_K$.*

Corollary 7. *With notations as in Theorem 10, family $CTM(\boldsymbol{\theta}; \boldsymbol{d}, \Upsilon)$ is almost surely concave as $\lambda^2/\sigma \to +\infty$.*

Corollary 8. *With notations as in Theorem 10, assume that $\lambda^2 = \omega(\sigma \log K)$. Family $CTM(\boldsymbol{\theta}; \boldsymbol{d}, \Upsilon)$ is almost surely concave as $K \to +\infty$.*

5.4.2 Implication to related models

Many nonconjugate models employ the Gaussian distribution to model correlation of hidden topics, including those by Blei and Lafferty [20], Miao et al. [64], Putthividhya et al. [80, 81], Salomatin et al. [86]. The analysis for CTM is very general for the case of logistic-normal priors. Therefore, the results for CTM can be easily derived for other nonconjugate topic models. Here we take DTM [20] and IFTM [80] into consideration as two specific examples.

The *Independent Factor Topic Model* (IFTM) by Putthividhya et al. [80] is a variant of CTM in which $\boldsymbol{\mu}$ is replaced with $\boldsymbol{\mu}' = \boldsymbol{A}\boldsymbol{s} + \boldsymbol{\mu}$ to model independent sources that compose correlated topics. A slight modification to our analysis would yield interesting results for the corresponding family, denoting $\Upsilon' = \{\boldsymbol{\beta}, \boldsymbol{\mu}', \boldsymbol{\Sigma}\}$,

$$IFTM(\boldsymbol{\theta}; \boldsymbol{d}, \Upsilon') = \{f(\boldsymbol{\theta}; \boldsymbol{d}, \Upsilon') : \boldsymbol{\theta} \in \overline{\Delta}_K, \log \tilde{\boldsymbol{\theta}} \sim \mathcal{N}(\boldsymbol{\mu}', \boldsymbol{\Sigma})\}.$$

Theorem 11. *Let Υ' be fixed, $\sigma = \max_i \Sigma_{ii}^{-1}, \lambda = \lambda_{K-1}(\boldsymbol{\Sigma}^{-1})$, and $p = 1 - e^{2\log(K-1) - 0.5(\lambda-1)^2/\sigma}$. Assuming $\lambda \geq 1$, family $IFTM(\boldsymbol{\theta}; \boldsymbol{d}, \Upsilon')$ is p-concave over $\overline{\Delta}_K$.*

The *Dynamic Topic Model* (DTM) by Blei and Lafferty [20] also employs Gaussian priors to model correlation. Those priors are separable, i.e., having diagonal covariance matrices. Let $DTM(\boldsymbol{\theta}; \boldsymbol{d}, \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma)$ be defined similarly with (5.8), where $\boldsymbol{\Sigma}^{-1} = diag(\sigma, ..., \sigma)$. For this family, note that $\lambda_{K-1}(\boldsymbol{\Sigma}^{-1}) = \sigma$. Hence, Theorem 10 implies

Theorem 12. *For fixed $\{\boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma\}$, if $\sigma \geq 1$ then family $DTM(\boldsymbol{\theta}; \boldsymbol{d}, \boldsymbol{\beta}, \boldsymbol{\alpha}, \sigma)$ is probably concave with probability at least $1 - e^{2\log(K-1) - 0.5\sigma - 0.5/\sigma + 1}$.*

5.5 A fast algorithm for learning CTM

In this section we discuss an application of the findings in Section 5.4 to designing an efficient algorithm for learning CTM. Nonconjugacy of the prior over $\boldsymbol{\theta}$ poses various drawbacks and precludes using sampling techniques [18]. Hence Blei and Lafferty [18] proposed to use variational Bayesian methods to approximate the posterior distributions of latent variables. Variational Bayesian methods have been employed heavily for learning many other nonconjugate models [20, 64, 80, 81, 86]. The use of simplified distributions to approximate the true posterior often results in more parameters to be optimized when learning a model. (For example, the method by Blei and Lafferty [18] maintains K

Gaussian distributions for each document.) Hence it could be problematic when the corpus is large.

Learning CTM and other related models can be made significantly simpler by using our analysis. Indeed, to estimate the posterior ($P(\boldsymbol{\theta}|\boldsymbol{d}, \Upsilon)$) of topic mixtures, one can exploit fast algorithms for convex optimization. The analysis in Section 5.4 provides a theoretically reasonable justification for such an exploitation. Once $\boldsymbol{\theta}$ had been inferred for each document in the training data, one can follow the approach as in Chapter 4 to estimate topics $\boldsymbol{\beta}$. A Gaussian prior is also easily estimated when all $\boldsymbol{\theta}$ of the training documents are known.

5.5.1 Derivation of the algorithm

Our proposed algorithm for learning CTM is presented in Algorithm 5 which is an alternative algorithm similar to EM. This algorithm tries to maximize the following regularized joint likelihood of the training corpus $\mathcal{C}$:

$$\begin{aligned} L(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \sum_{\boldsymbol{d} \in \mathcal{C}} \log \Pr(\boldsymbol{\theta}, \boldsymbol{d}|\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) - \frac{M}{2}\alpha \operatorname{Tr} \boldsymbol{\Sigma}^{-1} \\ &= \sum_{\boldsymbol{d} \in \mathcal{C}} \sum_j d_j \log \sum_{k=1}^{K} \theta_k \beta_{kj} - \frac{1}{2} \sum_{\boldsymbol{d} \in \mathcal{C}} (\log \tilde{\boldsymbol{\theta}} - \boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\log \tilde{\boldsymbol{\theta}} - \boldsymbol{\mu}) \\ &\quad - \frac{M}{2} \log \det \boldsymbol{\Sigma} - \frac{M}{2}\alpha \operatorname{Tr} \boldsymbol{\Sigma}^{-1} + \textit{constant}. \end{aligned}$$

The main reason for imposing a regularization term $\alpha \operatorname{Tr} \boldsymbol{\Sigma}^{-1}$ on the joint likelihood is to control the eigenvalues of the learned $\boldsymbol{\Sigma}^{-1}$. Large α often prevents the eigenvalues of $\boldsymbol{\Sigma}^{-1}$ from increasing. On the other hand, small values of α play the role as promoting large eigenvalues of $\boldsymbol{\Sigma}^{-1}$. In the latter case, Corollary 7 and Corollary 8 suggest that estimation of topic mixtures ($\boldsymbol{\theta}$) is more likely to be a concave problem, and thus can be done efficiently.

In Step 1 which does posterior inference for each document, we use the Online Frank-Wolfe algorithm [47] to maximize the joint probability $\Pr(\boldsymbol{\theta}, \boldsymbol{d}|\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. This algorithm theoretically converges to the optimal solutions, provided that the optimization problem is concave.[1]

In Step 2, we fix $\boldsymbol{\theta}_d$ which has been inferred for each document $\boldsymbol{d} \in \mathcal{C}$ in Step 1,

[1]In practice we can approximate $\overline{\Delta}_K$ by $\Delta_\epsilon = \{\boldsymbol{\theta} : \sum_{k=1}^{K} \theta_k = 1, \theta_i \geq \epsilon, \forall i\}$ for a very small constant ϵ, says $\epsilon = 10^{-10}$. Hence the online Frank-Wolfe algorithm should be slightly modified accordingly.

Algorithm 5 fCTM: a fast algorithm for learning correlated topic models

Input: a corpus $\mathcal{C} = \{\boldsymbol{d_1}, ..., \boldsymbol{d}_M\}$, and a positive constant α.
Output: $\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$.
Initialize $\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}$, and then alternate the following two steps until convergence.
Step 1: for each document $\boldsymbol{d}$, use Algorithm 6 to solve for

$$\boldsymbol{\theta}_d = \arg\max_{\boldsymbol{\theta} \in \overline{\Delta}_K} \log \Pr(\boldsymbol{\theta}, \boldsymbol{d} | \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \tag{5.9}$$

Step 2: compute

$$\beta_{kj} \propto \sum_{\boldsymbol{d} \in \mathcal{C}} d_j \theta_{dk}, \tag{5.10}$$

$$\boldsymbol{\mu} = \frac{1}{M} \sum_{\boldsymbol{d} \in \mathcal{C}} \log \tilde{\boldsymbol{\theta}}_d, \tag{5.11}$$

$$\boldsymbol{\Sigma} = \alpha \boldsymbol{I}_{K-1} + \frac{1}{M} \sum_{\boldsymbol{d} \in \mathcal{C}} (\log \tilde{\boldsymbol{\theta}}_d - \boldsymbol{\mu})(\log \tilde{\boldsymbol{\theta}}_d - \boldsymbol{\mu})^t. \tag{5.12}$$

Algorithm 6 Online Frank-Wolfe

Input: document $\boldsymbol{d}$, and model $\Upsilon = \{\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$.
Output: $\boldsymbol{\theta}$ that maximizes
$f(\boldsymbol{\theta}) = \sum_j d_j \log \sum_{k=1}^K \theta_k \beta_{kj} - \frac{1}{2}(\log \tilde{\boldsymbol{\theta}} - \boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\log \tilde{\boldsymbol{\theta}} - \boldsymbol{\mu}) - \sum_{k=1}^K \log \theta_k$.
Initialize $\boldsymbol{\theta}_1$ arbitrarily in $\overline{\Delta}_K$.
for $\ell = 1, ..., \infty$ **do**
  Pick f_ℓ uniformly from
    $\{\sum_j d_j \log \sum_{k=1}^K \theta_k \beta_{kj};\ \ -\frac{1}{2}(\log \tilde{\boldsymbol{\theta}} - \boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\log \tilde{\boldsymbol{\theta}} - \boldsymbol{\mu}) - \sum_{k=1}^K \log \theta_k\}$
  $F := \frac{1}{\ell} \sum_{h=1}^{\ell} f_h$
  $i' := \arg\max_i \nabla F(\boldsymbol{\theta}_\ell)_i$; (maximal partial gradient)
  $\alpha := 2/(\ell + 2)$;
  $\boldsymbol{\theta}_{\ell+1} := \alpha \boldsymbol{e}_{i'} + (1 - \alpha) \boldsymbol{\theta}_\ell$.
end for

and maximize $L(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ to estimate the model parameters. Solving for $\boldsymbol{\beta}$ can be done independently of $\boldsymbol{\mu}, \boldsymbol{\Sigma}$. Hence by using the same argument as Than and Ho [100], we can arrive at the formula (5.10) for updating topics. Maximizing the term relating to $\boldsymbol{\mu}$ in $L(\boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$ will lead to (5.11) for updating $\boldsymbol{\mu}$.

Take $\boldsymbol{\Sigma}$ into consideration: $L_\alpha = -\frac{1}{2} \sum_{\boldsymbol{d} \in \mathcal{C}} (\log \tilde{\boldsymbol{\theta}}_d - \boldsymbol{\mu})^t \boldsymbol{\Sigma}^{-1} (\log \tilde{\boldsymbol{\theta}}_d - \boldsymbol{\mu}) - \frac{M}{2} \log \det \boldsymbol{\Sigma} - \frac{M}{2} \alpha \operatorname{Tr} \boldsymbol{\Sigma}^{-1}$. Its derivative with respect to $\boldsymbol{\Sigma}^{-1}$ is $\nabla L_\alpha = -\frac{1}{2} \sum_{\boldsymbol{d} \in \mathcal{C}} (\log \tilde{\boldsymbol{\theta}}_d - \boldsymbol{\mu})(\log \tilde{\boldsymbol{\theta}}_d - \boldsymbol{\mu})^t + \frac{M}{2} \boldsymbol{\Sigma} - \frac{M}{2} \alpha \boldsymbol{I}_{K-1}$. Solving $\nabla L_\alpha = 0$, one can derive (5.12) for updating $\boldsymbol{\Sigma}$.

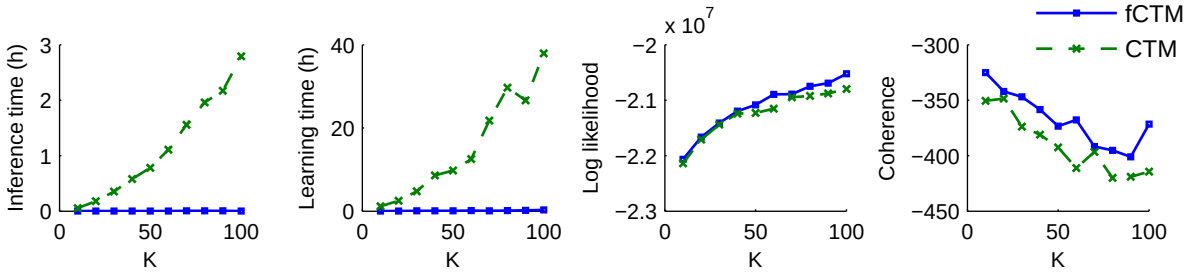


Figure 5.1: Performance of fCTM and CTM on Grolier as the number of topics increases. Lower is better for inference/learning time, whereas higher is better for likelihood and coherence. For $K = 100$, CTM needed approximately 38 hours for learning, while fCTM needed less than 19 minutes to complete learning.

5.5.2 Experiments

To see advantages of our algorithm (fCTM), we took the variational Bayesian method (denoted as CTM) by Blei and Lafferty [18] into comparison. Four measures were used for comparison: *learning time, likelihood* of the training data, *inference time* (Step 1) for the testing data, and *coherence* [66] for measuring the quality of the learned topics.

The Grolier corpus was considered, for which 23024 documents were used for training, and 6718 documents were held out for testing.[2]

Runtime. Figure 5.1 records some statistics from learning/inference results. We observed that fCTM learn significantly faster than CTM. Similar behavior holds when doing inference for each document. In our observations, fCTM often learns 65-135 times faster than CTM. Speedy learning of fCTM can be explained by the fact that Step 1 is done efficiently by the Online Frank-Wolfe algorithm. CTM did slowly because many auxiliary parameters need to be optimized. Furthermore, the variational method for doing inference is not guaranteed to converge quickly. Figure 5.1 shows that CTM often needs intensive time to do inference.

Model quality. Likelihood and coherence are used to see the quality of models learned from data. Coherence is used to assess quality (goodness and interpretability) of individual topics. It has been observed to reflect well human assessment [66].

To calculate the coherence of a topic k, we first choose the set $V^k = \{v_1^k, ..., v_t^k\}$ of the

[2]We used the same convergence criteria for fCTM and CTM: relative improvement of objective functions is less than 10^{-6} for inference of each document, and 10^{-3} for learning; at most 100 iterations are allowed to do inference. We used default settings for some other parameters of CTM, and set $\alpha = 100$ in fCTM.

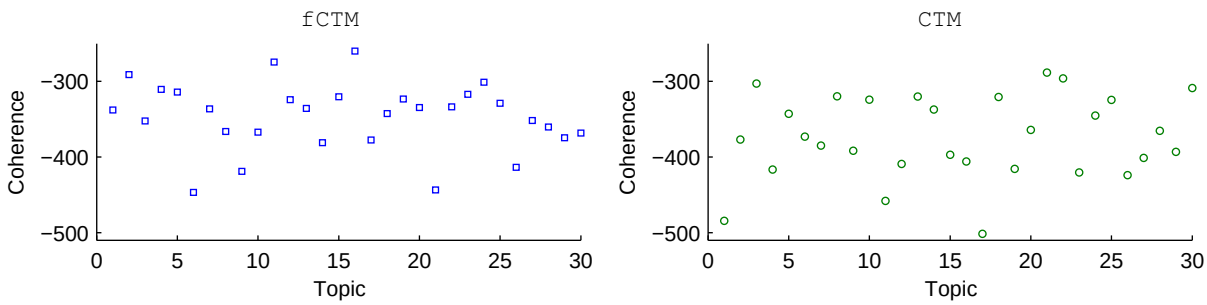


Figure 5.2: Quality of individual topics learned by fCTM and CTM on Grolier for models with 30 topics. Higher is better.

top t terms that have highest probabilities in that topic, and then compute

$$C(k, V^k) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{D(v_m^k, v_l^k) + 1}{D(v_l^k)}$$

where $D(v)$ is the document frequency of term v, $D(u, v)$ is the number of documents that contain both terms u and v. In our experiments, we chose top $t = 20$ terms for investigation, and coherence of individual topics is averaged: $coherence = \frac{1}{K} \sum_{k=1}^{K} C(k, V^k)$.

Figure 5.1 shows the quality of the learned models. We observe that the two learning methods performed comparably in terms of likelihood. However, the topic quality of CTM is inferior to that of fCTM in terms of coherence. Note that topics learned by fCTM consistently better than CTM. When investigating individual topics we find that topics learned by CTM vary significantly in quality. Some topics seem not to be good enough as depicted in Figure 5.2. In contrast, the quality of topics learned by fCTM does not vary much.

Models of hidden interactions. Figure 5.3 and 5.4 shows parts of the full model with 100 topics learned by fCTM from Grolier. Figure 5.3 shows positive correlations between topics, while Figure 5.4 shows negative correlations. We observe that the learned topics are interpretable and the discovered correlations are reasonable. Those further support that fCTM can learn qualitative models with a significantly faster speed than CTM.

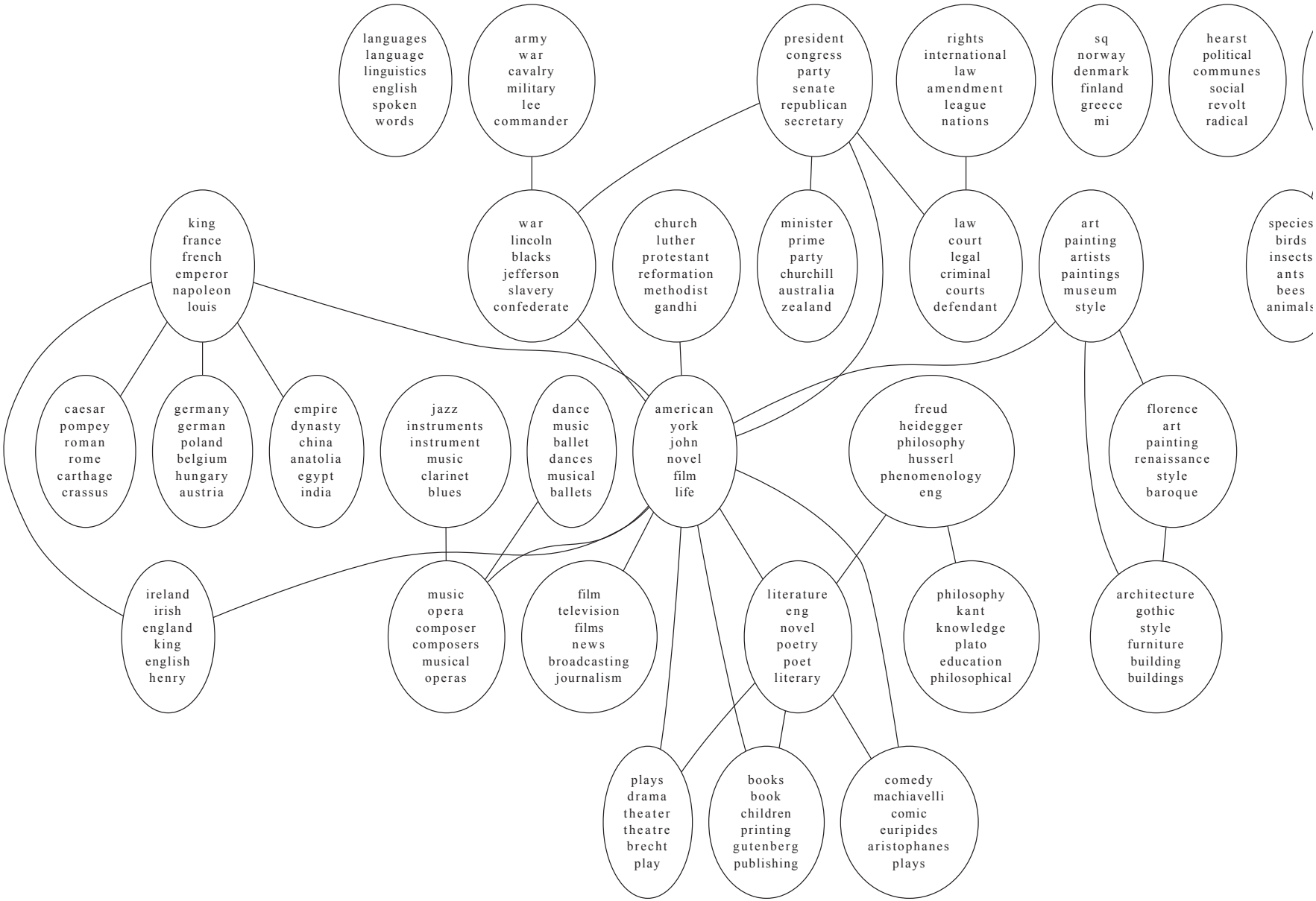


Figure 5.3: Illustration of the correlated topic model with 100 topics learned from Grolier articles. An edge connecting two topics shows that the two topics *very likely appear together* in a document. This visualization was drawn with Graphviz [40]

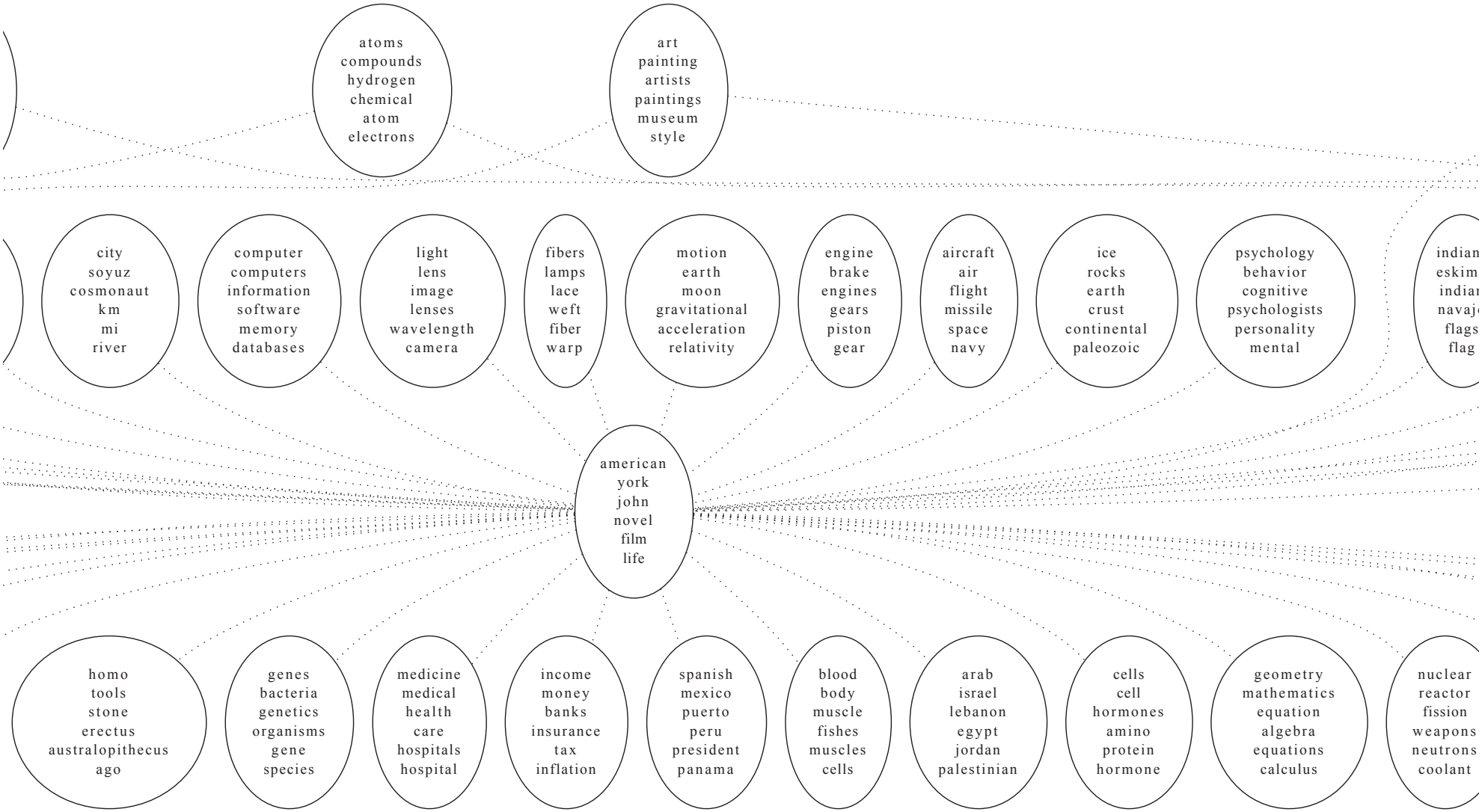


Figure 5.4: Illustration of the correlated topic model with 100 topics learned from Grolier. An edge connecting two topics shows that the two topics *unlikely appear together* in a document. This visualization was drawn with Graphviz [40]

5.6 Summary

We have introduced a framework with the concept of probable convexity to analyze convexity of real functions. The analysis can be used in many situations where the function family of interest is not convex. It could help us to identify a subset of convex functions from that non-convex family, and hence we could deal with the family efficiently in practice. Hence probable convexity provides a feasible way to deal with non-convexity of real problems such as posterior estimation in probabilistic graphical models.

When applied to the problem of estimating topic mixtures in CTM [18], we found that this problem is in fact concave with some probability. Hence in practice we can exploit results from convex optimization to design efficient algorithms. The same results were discussed with many nonconjugate models. Finally, we proposed a novel algorithm for learning CTM which can work significantly faster than existing methods, while keeping or making better the quality of the learned models.

Chapter 6

Conclusion and future research

The thesis has systematically studied to model large collections of documents. To this end, the thesis elucidated two fundamental problems that have to be resolved in order to successfully develop scalable learning algorithms. The first problem is posterior inference of topic mixtures, which essentially asks for uncovering what topics and how significantly each contributes to a document. The second problem is model complexity which refers to learn sparse topic models.

Chapter 3 introduces the FW framework for inferring sparse topic mixtures. The framework has some properties that are attractive for large-scale modeling such as fast convergence rate, a provable guarantee on inference quality, and the ability to directly trade off sparsity of topic mixtures against quality. Goodness and flexibility of FW have been demonstrated by two specific applications: designing effective methods for supervised dimension reduction, and designing scalable learning algorithms for FSTM and CTM.

Chapter 5 discusses a novel theory of probable convexity to deal with the cases that inference is inherently intractable. We have shown that posterior inference of topic mixtures in CTM and many nonconjugate models are tractable in practices, although the problem is well-known intractable in the worse case. Benefits of this study is a novel algorithm for learning correlated topic models. The scalability of this algorithm enables us to easily make a large-scale analysis of hidden interactions of latent topics.

Chapter 4 introduces FSTM for modeling large corpora. The model overcomes many limitations of existing topic models. More importantly, its learning algorithm is near independent of the dimensionality of data. This property is very attractive as recent applications often face with extremely high dimensional data. The distributed architecture enables us to learn thousands of hidden topics from collections with millions of documents.

Despite of significant achievements, the research remains some limitations and leaves open many opportunities for future study. The followings are some of the limitations.

Fast inference of posteriors other than topic mixtures

The thesis has focused mostly on estimation of topic mixtures. However, for some models and in some applications, it is necessary to estimate some other posterior distributions. We left open the estimation of the z variable in topic models which are sometimes important to know, because it shows what topic generates a specific occurrence of terms.

Incorporation of prior knowledge

FSTM is in a very initial form of topic models. It can be included as the core in more complex models. Also there is no mechanism in FSTM to exploit domain (or prior) knowledge to refine its quality. Hence it is an open direction for future research to include prior knowledge into FSTM. An easy way is to encode prior knowledge into the objective function of inference and then use the framework in Chapter 3 to do inference.

Learning when data and model do not fit in memory

Although the learning algorithm for FSTM is scalable, in its current form it is still hard to manage oversize text collections as it requires data to be loaded into memory. Hence an interesting research would be to propose online algorithms that can learn from sequential data or extremely large collections.

From practical observations, we found that the learned topics in FSTM are pretty dense. This would be problematic when we want to learn hundreds of thousands of topics for the aim of topical exploration. Therefore, model sparsity should be studied further in order to learn really big models from data.

Publications

[1] Khoat Than, Tu Bao Ho. Modeling the diversity and log-normality in data. *Intelligent Data Analysis: An International Journal*, Vol. 18(6), 2014.

[2] Khoat Than, Tu Bao Ho, and Duy Khuong Nguyen. An effective framework for supervised dimension reduction. *Neurocomputing*, accepted, 2013.

[3] Khoat Than and Tu Bao Ho. Fully sparse topic models. *Journal of Machine Learning Research*, submitted, 2013.

[4] Khoat Than and Tu Bao Ho. Managing sparsity, time, and quality of inference in topic models. *arXiv:1210.7053 [stat.ML]*, 2013.

[5] Khoat Than and Tu Bao Ho. Probable convexity and its application to Correlated Topic Models. *Technical report*, 2013.

[6] Khoat Than and Tu Bao Ho. A geometric interpretation of Bayesian classification methods. *Technical report*, 2013.

[7] Khoat Than, Thi Nhan Le, Tatsuo Kanda, Tu Bao Ho. Classification and Discriminative Analysis of Short Sequences using Topic Modeling. *Tech. report*, 2013.

[8] Duy Khuong Nguyen, Khoat Than, and Tu Bao Ho. Simplicial non-negative matrix factorization. In *Proceedings of the 10th IEEE RIVF*, 2013.

[9] Khoat Than, Tu Bao Ho, Duy Khuong Nguyen, and Ngoc Khanh Pham. Supervised dimension reduction with topic models. In *ACML*, volume 25 of *Journal of Machine Learning Research: W&CP*, pages 395–410, 2012.

[10] Khoat Than and Tu Bao Ho. Fully sparse topic models. In *ECML-PKDD*, volume 7523 of *Lecture Notes in Computer Science*, pages 490–505. Springer, 2012.

[11] Khoat Than and Tu Bao Ho. Automatic construction of knowledge from large unstructured data with topic modeling. In *Proceedings of the 13th International Symposium on Knowledge and System Sciences (KSS)*, Japan, 2012.

Bibliography

[1] Karim M. Abadir and Jan R. Magnus. *Matrix Algebra*. Cambridge University Press, 2005.

[2] Alekh Agarwal, Peter L. Bartlett, Pradeep Ravikumar, and Martin J. Wainwright. Information-theoretic lower bounds on the oracle complexity of stochastic convex optimization. *Information Theory, IEEE Transactions on*, 58(5):3235 –3249, may 2012. ISSN 0018-9448. doi: 10.1109/TIT.2011.2182178.

[3] Deepak Agarwal and Bee-Chung Chen. fLDA: matrix factorization through latent dirichlet allocation. In *The third ACM International Conference on Web Search and Data Mining*, pages 91–100. ACM, 2010.

[4] Amr Ahmed and Eric Xing. On tight approximate inference of the logistic-normal topic admixture model. In *AISTATS*, volume 2 of *Journal of Machine Learning Research: W&CP*, pages 19–26, 2007.

[5] J Aitchison and Sheng M Shen. Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261–272, 1980.

[6] David Aldous. Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII 1983*, volume 1117 of *Lecture Notes in Mathematics*, pages 1–198. Springer Berlin / Heidelberg, 1985.

[7] Anima Anandkumar, Dean Foster, Daniel Hsu, Sham Kakade, and Yi-Kai Liu. A spectral algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, volume 25, pages 926–934, 2012.

[8] David Andrzejewski, Xiaojin Zhu, and Mark Craven. Incorporating domain knowledge into topic modeling via dirichlet forest priors. In *The 26th International Conference on Machine Learning (ICML)*, 2009.

[9] Sanjeev Arora, Rong Ge, and Ankur Moitra. Learning topic models–going beyond svd. In *IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1–10. IEEE, 2012. doi: 10.1109/FOCS.2012.49.

[10] Sanjeev Arora, Rong Ge, Yoni Halpern, David Mimno, Ankur Moitra, David Sontag, Yichen Wu, and Michael Zhu. A practical algorithm for topic modeling with provable guarantees. In *ICML*, volume 28 of *Journal of Machine Learning Research: W&CP*, pages 280–288, 2013.

[11] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007. URL `http://www.ics.uci.edu/$\sim$mlearn/{MLR}epository.html`.

[12] A. Asuncion, M. Welling, P. Smyth, and Y.W. Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 27–34, 2009.

[13] Arthur U. Asuncion, Padhraic Smyth, and Max Welling. Asynchronous distributed estimation of topic models for document analysis. *Statistical Methodology*, 8(1): 3–17, 2011.

[14] David Blei and Jon McAuliffe. Supervised topic models. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

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

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

[17] David M. Blei and John Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 1(1):17–35, 2007.

[18] David M. Blei and John Lafferty. A correlated topic model of science. *The Annals of Applied Statistics*, 1(1):17–35, 2007.

[19] David M Blei and John Lafferty. Topic models. In Mehran Sahami Ashok Srivastava, editor, *Text mining: classification, clustering, and applications*, chapter 4, pages 71–93. Taylor & Francis, 2009.

[20] David M. Blei and John D. Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120. ACM, 2006.

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

[22] David M. Blei, Thomas L. Griffiths, Michael I. Jordan, and Joshua B. Tenenbaum. Hierarchical topic models and the nested chinese restaurant process. In *Advances in Neural Information Processing Systems*, volume 16, pages 106–114, 2004.

[23] David M. Blei, Thomas L. Griffiths, and Michael I. Jordan. The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies. *Journal of the ACM*, 57(2):7:1–7:30, 2010. doi: 10.1145/1667053.1667056.

[24] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[25] Qing Cao, Wenjing Duan, and Qiwei Gan. Exploring determinants of voting for the helpfulness of online user reviews: A text mining approach. *Decision Support Systems*, 50(2):511–521, 2011.

[26] Jonathan Chang, Jordan Boyd-Graber, Sean Gerrish, Chong Wang, and David Blei. Reading tea leaves: how humans interpret topic models. In *Advances in Neural Information Processing Systems (NIPS)*, volume 22, 2009.

[27] Mung Chiang. Geometric programming for communication systems. *Foundations and Trends in Communications and Information Theory*, 2(1-2):1–153, 2005.

[28] Kenneth L. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Trans. Algorithms*, 6:63:1–63:30, 2010. ISSN 1549-6325. doi: http://doi.acm.org/10.1145/1824777.1824783. URL `http://doi.acm.org/10.1145/1824777.1824783`.

[29] Shay B Cohen and Noah A Smith. Shared logistic normal distributions for soft parameter tying in unsupervised grammar induction. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 74–82. ACL, 2009.

[30] Shay B Cohen and Noah A Smith. Covariance in unsupervised learning of probabilistic grammars. *The Journal of Machine Learning Research*, 11:3017–3051, 2010.

[31] Ali Daud, Juanzi Li, Lizhu Zhou, and Faqir Muhammad. Knowledge discovery through directed probabilistic topic models: a survey. *Frontiers of Computer Science in China*, 4(2):280–301, 2010.

[32] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[33] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 1–38, 1977.

[34] Chris Ding. A probabilistic model for latent semantic indexing. *Journal of the American Society for Information Science and Technology*, 56(6):597–608, 2005.

[35] Gabriel Doyle and Charles Elkan. Accounting for burstiness in topic models. In *The 26th International Conference on Machine Learning (ICML)*, 2009.

[36] Alexander Van Esbroeck, Chih-Chun Chia, and Zeeshan Syed. Heart rate topic models. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*. AAAI, 2012.

[37] Ali Faisal, Jussi Gillberg, Gayle Leen, and Jaakko Peltonen. Transfer learning using a nonparametric sparse topic model. *Neurocomputing*, 2013.

[38] R.E. Fan, K.W. Chang, C.J. Hsieh, X.R. Wang, and C.J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[39] Malcolm R. Forster. Key concepts in model selection: Performance and generalizability. *Journal of Mathematical Psychology*, 44(1):205–231, 2000.

[40] Emden R. Gansner and Stephen C. North. An open graph visualization system and its applications to software engineering. *SOFTWARE - PRACTICE AND EXPERIENCE*, 30(11):1203–1233, 2000.

[41] Sean Gerrish and David Blei. How they vote: Issue-adjusted models of legislative behavior. In *Advances in Neural Information Processing Systems*, volume 25, pages 2762–2770, 2012.

[42] Samuel J Gershman and David M Blei. A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1):1–12, 2012.

[43] Thomas L Griffiths, Mark Steyvers, and Joshua B Tenenbaum. Topics in semantic representation. *Psychological Review*, 114(2):211–244, 2007.

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

[45] Justin Grimmer. A bayesian hierarchical topic model for political texts: Measuring expressed agendas in senate press releases. *Political Analysis*, 18(1):1–35, 2010. doi:

10.1093/pan/mpp034. URL http://pan.oxfordjournals.org/content/18/1/1.abstract.

[46] 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*, pages 363–371. ACL, 2008.

[47] Elad Hazan and Satyen Kale. Projection-free online learning. In *Proceedings of the 29th Annual International Conference on Machine Learning (ICML)*, 2012.

[48] Matthew D. Hoffman, David M. Blei, and Francis Bach. Online learning for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, volume 23, pages 856–864, 2010.

[49] Leah Hoffmann. Looking back at big data. *Communications of the ACM*, 56(4): 21–23, 2013. doi: 10.1145/2436256.2436263.

[50] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42(1):177–196, 2001.

[51] Thomas Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, 42:177–196, 2001. ISSN 0885-6125. URL http://dx.doi.org/10.1023/A:1007617005950.

[52] Go Irie, Takashi Satou, Akira Kojima, Toshihiko Yamasaki, and Kiyoharu Aizawa. Affective audio-visual words and latent topic driving model for realizing movie affective scene classification. *IEEE Transactions on Multimedia*, 12(6):523–535, 2010.

[53] Alan J Izenman. Introduction to random-matrix theory, 2012.

[54] S.S. Keerthi, S. Sundararajan, K.W. Chang, C.J. Hsieh, and C.J. Lin. A sequential dual method for large scale multi-class linear svms. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 408–416. ACM, 2008.

[55] Christian Kleiber and Samuel Kotz. *Statistical Size Distributions in Economics and Actuarial Sciences*. Wiley-Interscience, 2003.

[56] S. Lacoste-Julien, F. Sha, and M.I. Jordan. Disclda: Discriminative learning for dimensionality reduction and classification. In *Advances in Neural Information Processing Systems (NIPS)*, volume 21, pages 897–904. MIT, 2008.

[57] Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133:365–397, 2012. ISSN 0025-5610. doi: 10.1007/s10107-010-0434-y. URL `http://dx.doi.org/10.1007/s10107-010-0434-y`.

[58] Thomas Landauer and Susan Dumais. A solution to platos problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104(2):211–240, 1997.

[59] Martin O. Larsson and Johan Ugander. A concave regularization technique for sparse mixture models. In J. Shawe-Taylor, R.S. Zemel, P. Bartlett, F.C.N. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1890–1898. 2011.

[60] Ackhard Limpert, Werner A. Stahel, and Markus Abbt. Log-normal distributions across the sciences: Keys and clues. *BioScience*, 51(5):341–352, may 2001.

[61] B. Liu, L. Liu, A. Tsykin, G.J. Goodall, J.E. Green, M. Zhu, C.H. Kim, and J. Li. Identifying functional mirna–mrna regulatory modules with correspondence latent dirichlet allocation. *Bioinformatics*, 26(24):3105, 2010.

[62] Dong C. Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

[63] Xian-Ling Mao, Zhao-Yan Ming, Tat-Seng Chua, Si Li, Hongfei Yan, and Xiaoming Li. Sshlda: a semi-supervised hierarchical topic model. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 800–809. ACL, 2012.

[64] Gengxin Miao, Ziyu Guan, Louise E. Moser, Xifeng Yan, Shu Tao, Nikos Anerousis, and Jimeng Sun. Latent association analysis of document pairs. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 1415–1423, New York, NY, USA, 2012. ACM. doi: 10.1145/2339530.2339752. URL `http://doi.acm.org/10.1145/2339530.2339752`.

[65] David Mimno. Computational historiography: Data mining in a century of classics journals. *Journal on Computing and Cultural Heritage*, 5(1):3, 2012.

[66] 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*, pages 262–272. Association for Computational Linguistics, 2011.

[67] David Mimno, Matthew D. Hoffman, and David M. Blei. Sparse stochastic inference for latent dirichlet allocation. In *Proceedings of the 29th Annual International Conference on Machine Learning*, 2012.

[68] I. Mukherjee and D.M. Blei. Relative performance guarantees for approximate inference in latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, volume 21, pages 1129–1136, 2009.

[69] Gill P. Wright M.: Murray, W. *Practical optimization.* Academic Press, 1981.

[70] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

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

[72] David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin. Automatic evaluation of topic coherence. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 100–108. ACL, 2010.

[73] David Newman, Youn Noh, Edmund Talley, Sarvnaz Karimi, and Timothy Baldwin. Evaluating topic models for digital libraries. In *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, pages 215–224. ACM, 2010.

[74] Frank Nielsen and Vincent Garcia. Statistical exponential families: A digest with flash cards. *CoRR*, abs/0911.4863, 2009.

[75] Frank Nielsen and Richard Nock. Clustering multivariate normal distributions. In *Emerging Trends in Visual Computing*, number 5416 in LNCS, pages 164–174. Springer-Berlin / Heidelberg, 2009.

[76] Christos H Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences*, 61(2):217–235, 2000.

[77] Nathan Parrish and Maya R. Gupta. Dimensionality reduction by local discriminative gaussian. In *Proceedings of the 29th Annual International Conference on Machine Learning*, 2012.

[78] Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2):945–959, 2000.

[79] D. Putthividhya, H. T. Attias, and S. Nagarajan. Independent factor topic models. In *The 26th International Conference on Machine Learning (ICML)*, 2009.

[80] D. Putthividhya, H. T. Attias, and S. Nagarajan. Independent factor topic models. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, 2009.

[81] D. Putthividhya, H.T. Attias, and S.S. Nagarajan. Topic regression multi-modal latent dirichlet allocation for image annotation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3408 –3415, 2010. doi: 10.1109/CVPR.2010.5540000.

[82] Daniel Ramage, Susan Dumais, and Dan Liebling. Characterizing microblogs with topic models. In *International AAAI Conference on Weblogs and Social Media*, 2010.

[83] Daniel Ramage, Susan Dumais, and Dan Liebling. Characterizing microblogs with topic models. In *International AAAI Conference on Weblogs and Social Media*, 2010.

[84] Michael Redmond and Alok Baveja. A data-driven software tool for enabling cooperative information sharing among police departments. *European Journal of Operational Research*, 141(3):660–678, 2002.

[85] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 487–494, 2004.

[86] Konstantin Salomatin, Yiming Yang, and Abhimanyu Lad. Multi-field correlated topic modeling. In *Proceedings of the SIAM International Conference on Data Mining (SDM)*, pages 628–637. SIAM, 2009.

[87] H Andrew Schwartz, Johannes C Eichstaedt, Lukasz Dziurzynski, Margaret L Kern, Martin EP Seligman, Lyle H Ungar, Eduardo Blanco, Michal Kosinski, and David Stillwell. Toward personality insights from language exploration in social media. In *AAAI Spring Symposium Series*, 2013.

[88] Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.

[89] Madhusudana Shashanka, Bhiksha Raj, and Paris Smaragdis. Sparse overcomplete latent variable decomposition of counts data. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.

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

[91] David Sontag and Daniel M. Roy. Complexity of inference in latent dirichlet allocation. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[92] Mark Steyvers and Tom Griffiths. Probabilistic topic models. In Thomas K. Landauer, Danielle S. McNamara, Simon Dennis, and Walter Kintsch, editors, *Handbook of Latent Semantic Analysis*, chapter 21, pages 427–448. Psychology Press, 2007.

[93] Mark Steyvers, Padhraic Smyth, Michal Rosen-Zvi, and Thomas Griffiths. Probabilistic author-topic models for information discovery. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 306–315. ACM, 2004.

[94] Masashi Sugiyama. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *The Journal of Machine Learning Research*, 8:1027–1061, 2007.

[95] Xiaoling Sun and Hongfei Lin. Topical community detection from mining user tagging behavior and interest. *Journal of the American Society for Information Science and Technology*, 64(2):321–333, 2013. ISSN 1532-2890. doi: 10.1002/asi.22740. URL `http://dx.doi.org/10.1002/asi.22740`.

[96] Edmund M Talley, David Newman, David Mimno, Bruce W Herr II, Hanna M Wallach, Gully APC Burns, AG Miriam Leenders, and Andrew McCallum. Database of nih grants using machine-learned categories and graphical clustering. *Nature Methods*, 8(6):443–444, 2011.

[97] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[98] Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[99] Y.W. Teh, D. Newman, and M. Welling. A collapsed variational bayesian inference algorithm for latent dirichlet allocation. In *Advances in Neural Information Processing Systems*, volume 19, page 1353, 2007.

[100] Khoat Than and Tu Bao Ho. Fully sparse topic models. In Peter Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery*

in Databases, volume 7523 of *Lecture Notes in Computer Science*, pages 490–505. Springer, 2012.

[101] Khoat Than, Tu Bao Ho, Duy Khuong Nguyen, and Ngoc Khanh Pham. Supervised dimension reduction with topic models. In *ACML*, volume 25 of *Journal of Machine Learning Research: W&CP*, pages 395–410, 2012.

[102] Khoat Than, Tu Bao Ho, and Duy Khuong Nguyen. An effective framework for supervised dimension reduction. *Neurocomputing (accepted)*, 2013.

[103] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[104] Craig A Tracy and Harold Widom. Introduction to random matrices. In *Geometric and quantum aspects of integrable systems*, pages 103–130. Springer, 1993.

[105] Joel Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

[106] Flora S. Tsai. A tag-topic model for blog mining. *Expert Systems with Applications*, 38(5):5330 – 5335, 2011. doi: 10.1016/j.eswa.2010.10.025.

[107] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[108] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Y. Eldar and G. Kutyniok, editors, *Compressed Sensing: Theory and Applications*, chapter 5, pages 210–268. Cambridge University Press, 2012.

[109] Mirwaes Wahabzada and Kristian Kersting. Larger residuals, less work: Active document scheduling for latent dirichlet allocation. In Dimitrios Gunopulos, Thomas Hofmann, Donato Malerba, and Michalis Vazirgiannis, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 6913 of *LNCS*, pages 475–490. Springer Berlin / Heidelberg, 2011.

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

[111] Hanna M. Wallach, David Mimno, and Andrew McCallum. Rethinking lda: why priors matter. In *Neural Information Processing Systems (NIPS)*, 2009.

[112] Chong Wang and David M. Blei. Decoupling sparsity and smoothness in the discrete hierarchical dirichlet process. In *Advances in Neural Information Processing Systems*, volume 22, pages 1982–1989, 2009.

[113] Chong Wang, David Blei, and David Heckerman. Continuous time dynamic topic models. In *The 24th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2008.

[114] Chong Wang, Bo Thiesson, Christopher Meek, and David M. Blei. Markov topic models. In *Neural Information Processing Systems (NIPS)*, 2009.

[115] Quan Wang, Jun Xu, Hang Li, and Nick Craswell. Regularized latent semantic indexing. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 685–694. ACM, 2011. ISBN 978-1-4503-0757-4. doi: 10.1145/2009916.2010008.

[116] Quan Wang, Jun Xu, Hang Li, and Nick Craswell. Regularized latent semantic indexing: A new approach to large-scale topic modeling. *ACM Trans. Inf. Syst.*, 31 (1):5:1–5:44, 2013. doi: 10.1145/2414782.2414787.

[117] 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*, pages 178–185. ACM, 2006.

[118] J. Weng, E.P. Lim, J. Jiang, and Q. He. Twitterrank: finding topic-sensitive influential twitterers. In *Proceedings of the third ACM International Conference on Web Search and Data Mining*, pages 261–270. ACM, 2010.

[119] Avi Wigderson and David Xiao. Derandomizing the ahlswede-winter matrix-valued chernoff bound using pessimistic estimators, and applications. *Theory of Computing*, 4(1):53–76, 2008.

[120] Sinead Williamson, Chong Wang, Katherine A. Heller, and David M. Blei. The ibp compound dirichlet process and its application to focused topic modeling. In *International Conference on Machine Learning (ICML)*, 2010.

[121] K. Xu, W. Yang, G. Liu, and H. Sun. Unsupervised satellite image classification using markov field topic model. *Geoscience and Remote Sensing Letters, IEEE*, 10 (1):130–134, 2013. doi: 10.1109/LGRS.2012.2194770.

[122] Zhong P. Yang and Xiao X. Feng. A note on the trace inequality for products of hermitian matrix power. *Journal of Ineuqualities in Pure and Applied Mathematics*, 3(5):78:1–78:12, 2002.

[123] 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, pages 937–946. ACM, 2009.

[124] Hsiang-Fu Yu, Cho-Jui Hsieh, Kai-Wei Chang, and Chih-Jen Lin. Large linear classification when data cannot fit in memory. *ACM Trans. Knowl. Discov. Data*, 5(4):23:1–23:23, February 2012. ISSN 1556-4681. doi: 10.1145/2086737.2086743. URL `http://doi.acm.org/10.1145/2086737.2086743`.

[125] Xiaotong Yuan and Shuicheng Yan. Forward basis selection for sparse approximation over dictionary. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 22 of *Journal of Machine Learning Research: W&CP*, pages 1377–1388, 2012.

[126] Ke Zhai and Jordan Boyd-Graber. Online topic models with infinite vocabulary. In *ICML*, volume 28 of *Journal of Machine Learning Research: W&CP*, pages 561–569, 2013.

[127] Tong Zhang. Sequential greedy approximation for certain convex optimization problems. *IEEE Transactions on Information Theory*, 49(3):682 – 691, 2003. ISSN 0018-9448. doi: 10.1109/TIT.2002.808136.

[128] Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Trans. Math. Softw.*, 23(4):550–560, 1997. ISSN 0098-3500. doi: http://doi.acm.org/10.1145/279232.279236.

[129] Jun Zhu and Eric P. Xing. Sparse topical coding. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence (UAI)*, 2011.

[130] Jun Zhu, Amr Ahmed, and Eric P. Xing. Medlda: maximum margin supervised topic models. *The Journal of Machine Learning Research*, 13:2237–2278, 2012.

Appendix A

Modeling the diversity and log-normality in data

This part presents a study to model two properties of data. The main result is a new topic model for which posterior inference for documents is accelerated by employing variational methods.

A.1 Introduction

Topic models often consider a given corpus to be composed of latent topics, each of which turns out to be a distribution over words. A document in that corpus is a mixture of these topics. These in some models imply that the order of the documents in a corpus does not play an important role. Further, the order of the words in a specific document is often discarded.

One of the most influential models having the above-mentioned assumptions is the *Latent Dirichlet Allocation* model (LDA) [21]. LDA assumes that each latent topic is a sample drawn from a Dirichlet distribution, and that the topic proportions in each document are samples drawn from a Dirichlet distribution as well. This interpretation of topic-word distributions has been utilized in many other models, such as the *Correlated Topic Model* (CTM) [17], the *Independent Factor Topic Model* (IFTM) [79], DCMLDA [35], Labeled LDA [82], and fLDA [3].

A.1.1 Forgotten characteristics of data

Geologists have shown that the concentration of elements in the Earth's crust distributes very skewed and fits the lognormal distribution well. The latent periods of many infectious diseases also follow lognormal distributions. Moreover, the occurrences of many real events have been shown to be log-normally distributed, see [60] and [55] for more information. In linguistics, the number of words per sentence, and the lengths of all words used in common telephone conversations, fit lognormal distributions. Recently, the number of different words per document in many collections has been observed to very likely follow the lognormal distribution as well [34]. These observations suggest that log-normality is present in many data types.

Another inherent property of data is the "diversity" of features (or attributes). Loosely speaking, diversity of a feature in a dataset is essentially the number of different values of that feature observed in the records of that dataset. For a text corpus, high diversity of a word means a high number of different frequencies observed in the corpus.[1] The high diversity of a word in a corpus reveals that the word may play an important role in that corpus. The diversity of a word varies significantly among different corpora with respect to the importance of that word. Nonetheless, to the best of our knowledge, this phenomenon has not been investigated previously in the machine learning literature.

In the topic modeling literature, log-normality and diversity have not been under consideration up to now. We will see that despite the inherent importance of the diversity of data, existing topic models are still far from appropriately capturing it. Indeed, in our investigations, the most popular LDA behaved inconsistently with respect to diversity. Higher diversity did not necessarily assure a consistently better performance or a consistently worse performance. Beside, LDA tends to favor data of low diversity. This phenomenon may be reasonably explained by the use of the Dirichlet distribution to generate topics. Such a distribution often generates samples of low diversity, see Section A.4 for detailed discussions. Hence the use of the Dirichlet distribution implicitly sets a severe setback on LDA in modeling data with high diversity.

[1] For example, the word "learning" has 71 different frequencies observed in the NIPS corpus [11]. This fact suggests that "learning" appears in many documents of the corpus, in fact 1153 documents, and that many documents contain this word with very high frequencies, e.g. more than 50 occurrences. Hence, this word would be important in the topics of NIPS.

A.1.2 Our contributions

In this work, we address those issues by using the lognormal distribution. A rationale for our approach is that such distribution often allows its samples to have high variations, and hence is able to capture well the diversity of data. For topic models, we posit that the topics of a corpus are samples drawn from the lognormal distribution. Such an assumption has two aspects: one is to capture the lognormal properties of data, the other is to better model the diversity of data. Also, this treatment leads to a new topic model, named *Dirichlet-Lognormal topic model* (DLN).

By extensive experiments, we found that the use of the lognormal distribution really helps DLN to capture the log-normality and diversity of real data. The greater the diversity of the data, the better prediction by DLN; the more log-normally distributed the data is, the better the performance of DLN. Further, DLN worked consistently with respect to diversity of data. For these reasons, the new model overcomes the above-mentioned drawbacks of LDA. Summarizing, our contributions are as follows:

- We introduce and carefully investigate an inherent property of data, named "diversity". Diversity conveys many important characteristics of real data. In addition, we extensively investigate the existence of log-normality in real datasets.
- We investigate the behaviors of LDA, and find that LDA behaves inconsistently with respect to diversity. These investigations highlight the fact that "diversity" is not captured well by existing topic models, and should be paid more attention.
- We propose a new variant of LDA, called DLN. The new model can capture well the diversity and log-normality of data. It behaves much more consistently than LDA does. This shows the benefits of the use of the lognormal distribution in topic models.

Roadmap: In the next section, some notations and definitions will be introduced. Some characteristics of some real datasets will be investigated in Section A.3. By those investigations, we will see the necessity of more attention to diversity and log-normality of data. Insights into the lognormal and Dirichlet distributions will be discussed in Section A.4. Also we will see the rationales of using the lognormal distribution to cope with diversity and log-normality. Section A.5 is dedicated to presenting the DLN model. Our experimental results and comparisons will be described in Section A.6. Further discussions are in Section A.7. Section A.8 is a brief review of the literature. The last section will show our conclusions.

A.2 Definitions

Each dataset $\mathcal{D} = \{d_1, d_2, ..., d_D\}$ is a set of D records, composed from a set of features, $\mathcal{A} = \{A_1, A_2, ..., A_V\}$; each record $d_i = (d_{i1}, ..., d_{iV})$ is a tuple of which d_{ij} is a specific value of the feature A_j.

Diversity is the main focus of this article. Here we define it formally in order to avoid confusion with the other possible meanings of this word.

Definition 8 (Observed value set)**.** *Let $\mathcal{D} = \{d_1, d_2, ..., d_D\}$ be a dataset, composed from a set $\mathcal{A}$ of features. The observed value set of a feature $A \in \mathcal{A}$, denoted $OV_{\mathcal{D}}(A)$, is the set of all values of A observed in $\mathcal{D}$.*

Note that the observed value set of a feature is very different from the domain that covers all possible values of that feature.

Definition 9 (Diversity of feature)**.** *Let $\mathcal{D}$ be a dataset, and be composed from a set $\mathcal{A}$ of features. The diversity of the feature A in the data set $\mathcal{D}$ is*

$$Div_{\mathcal{D}}(A) = \frac{|OV_{\mathcal{D}}(A)|}{|\mathcal{D}|}$$

Clearly, diversity of a feature defined above is the normalized version of the number of different values of that feature in the data set. This concept is introduced in order to compare different datasets.

The diversity of a dataset is defined via averaging the diversities of the features of that dataset. This number will provide us an idea about how variation a given dataset is.

Definition 10 (Diversity of dataset)**.** *Let $\mathcal{D}$ be a dataset, composed from a set $\mathcal{A}$ of features. The diversity of the dataset $\mathcal{D}$ is*

$$Div_{\mathcal{D}} = average\{Div_{\mathcal{D}}(A) : A \in \mathcal{A}\}$$

Note that the concept of diversity defined here is completely different from the concept of variance. Variance often relates to the variation of a random variable from the true statistical mean of that variable whereas diversity provides the extent of variation in general of a variable. Furthermore, diversity only accounts for a given dataset, whereas variance does not. The diversity of the same feature may vary considerably among different datasets.

By means of averaging over all features, the diversity of a dataset surfers from outliers. In other words, the diversity of a dataset may be overly dominated by very few features, which have very high diversities. In this case, the diversity is not a good measure of the variation of the considered dataset. Overcoming this situation will be our future work.

We will often deal with textual datasets in this work. Hence, for the aim of clarity, we adapt the above definitions for text and discuss some important observations regarding such a data type.

If the dataset $\mathcal{D}$ is a text corpus, then the observed value set is defined in terms of frequency. We remark that in this work each document is represented by a sparse vector of frequencies, each component of which is the number of occurrences of a word occurred in that document.

Definition 11 (Observed frequency set)**.** *Let $\mathcal{C} = \{d_1, d_2, ..., d_M\}$ be a text corpus of size M, composed from a vocabulary $\mathcal{V}$ of V words. The observed frequency set of the word $w \in \mathcal{V}$, denoted $OV_{\mathcal{C}}(w)$, is the set of all frequencies of w observed in the documents of $\mathcal{C}$.*

$$OV_{\mathcal{C}}(w) = \{freq(w) : \exists d_i \textit{ that contains exactly } freq(w) \textit{ occurrences of } w\}$$

In this definition, there is no information about how many documents have a certain $freq(w) \in OV_{\mathcal{C}}(w)$. Moreover, if a word w appears in many documents with the same frequency, the frequency will be counted only once. The observed frequency set tells much about the behavior and stability of a word in a corpus. If $|OV_{\mathcal{C}}(w)|$ is large, w must appear in many documents of $\mathcal{C}$. Moreover, many documents must have high frequency of w. For example, if $|OV_{\mathcal{C}}(w)| = 30$, w must occur in at least 30 documents, many of which contain at least 20 occurrences of w.

Definition 12 (Diversity of word)**.** *Let $\mathcal{C}$ be a corpus, composed from a vocabulary $\mathcal{V}$. The diversity of the word $w \in \mathcal{V}$ in the corpus is*

$$Div_{\mathcal{C}}(w) = \frac{|OV_{\mathcal{C}}(w)|}{|\mathcal{C}|}$$

Definition 13 (Diversity of corpus)**.** *Let $\mathcal{C}$ be a corpus, composed from a vocabulary $\mathcal{V}$. The diversity of the corpus is*

$$Div_{\mathcal{C}} = average\{Div_{\mathcal{C}}(w) : w \in \mathcal{V}\}$$

It is easy to see that if a corpus has high diversity, a large number of its words would have a high number of different frequencies, and thus have high variations in the corpus.

These facts imply that such kind of corpora seem to be hard to deal with. Moreover, provided that the sizes are equal, a corpus with higher diversity has higher variation, and hence may be more difficult to model than a corpus with lower diversity. Indeed, we will see this phenomenon in the later analyses.

In this work, we will often mention lognormal and Dirichlet distributions. Hence we include here their mathematical definitions. The lognormal distribution of a random variable $\boldsymbol{x} = (x_1, ..., x_n)^T$, with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, has the following density function

$$LN(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{|\boldsymbol{\Sigma}|}x_1...x_n} \exp\{-\frac{1}{2}(\log \boldsymbol{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\log \boldsymbol{x} - \boldsymbol{\mu})\}.$$

Similarly, the density function of the Dirichlet distribution is

$$Dir(\boldsymbol{x}; \alpha_1, \ldots, \alpha_n) = \frac{\Gamma\left(\sum_{i=1}^n \alpha_i\right)}{\prod_{i=1}^n \Gamma(\alpha_i)} \prod_{i=1}^n x_i^{\alpha_i - 1},$$

where $\sum_{i=1}^n x_i = 1, x_i > 0$. The constraint means that the Dirichlet distribution is in fact in $(n-1)$-dimensional space.

A.3 Diversity and Log-normality of real data

We first describe our initial investigations on 5 real datasets from the UCI Machine Learning Repository [11] and Blei's webpage.[2] Some information on these datasets is reported in Table A.1, in which the last two rows have been averaged. In fact, the Communities and Crime dataset (Comm-Crime for short) is not a usual text corpus. This data set contains 1994 records each of which is the information of a US city. There are 123 attributes, some of which are missing for some cities [84]. In our experiments, we removed the attributes from all records if they are missing in some records. Also, we removed the first 5 non-predictive attributes, and the remainings consist of only 100 real attributes including crime.

Our initial investigations studied the diversity of the above data sets. These three textual corpora, AP, NIPS, and KOS, were preprocessed to remove all function words and stopwords, which are often assumed to be meaningless to the gists of the documents. The remaining are content words. Some statistics are given in Table A.2.

[2]The AP corpus: http://www.cs.princeton.edu/∼blei/lda-c/ap.tgz

Table A.1: Datasets for experiments.

Data set	AP	NIPS	KOS	SPAM	Comm-Crime
Number of documents	2246	1500	3430	4601	1994
Vocabulary size	10473	12419	6906	58	100
Document length	194.05	1288.24	136.36		
#unique words per doc	134.48	497.54	102.96		

Table A.2: Statistics of the 3 corpora. Although NIPS has least documents among the three corpora, all of its statistics here are much greater than those of the other two corpora.

Data set	AP	KOS	NIPS
Diversity	0.0012	0.0011	0.004
No. of words with $\|OV\| \geq 5$	1267	1511	5900
No. of words with $\|OV\| \geq 10$	99	106	1633
No. of words with $\|OV\| \geq 20$	1	4	345
Three greatest $\|OV\|$'s	{25; 19; 19}	{26; 21; 21}	{86; 80; 71}

One can easily realize that the diversity of NIPS is significantly larger than that of AP and KOS. Among 12419 words of NIPS, 5900 words have at least 5 different frequencies; 1633 words have at least 10 different frequencies.[3] These facts show that a large number of words in NIPS vary significantly within the corpus, and hence may cause considerable difficulties for topic models.

AP and KOS are comparable in terms of diversity. Despite this fact, AP seems to have quite greater variation compared with KOS. The reason is that although the number of documents in AP is nearly 10/15 of that in KOS, the number of words with $|OV| \geq 5$ in AP is approximately 12/15 of that in KOS. Furthermore, KOS and AP have nearly the same number of words with $|OV| \geq 10$. Another explanation for the larger variation of AP over KOS is is that the documents in AP are much longer on average than those of KOS, see Table A.1. Longer documents would generally provide more chances for occurrences of words, and thus would probably encourage greater diversity for a corpus.

Comm-Crime and SPAM are non-textual datasets. Their diversities are 0.0458 and 0.0566, respectively. Almost all attributes have $|OV| \geq 30$, except one in each data set, and the greatest $|OV|$ in SPAM is 2161 which is far greater than that in the textual counterparts. The values of attributes are mostly real numbers, and vary considerably. This is why their diversities are much larger than those of textual corpora.

The next investigations were on how individual content words distribute in a corpus.

[3]The three words which have greatest number of different frequencies, $|OV|$, are "network", "model", and "learning". Each of these words appears in more than 1100 documents of NIPS. To some extent, they are believed to compose the main theme of the corpus with very high probability.

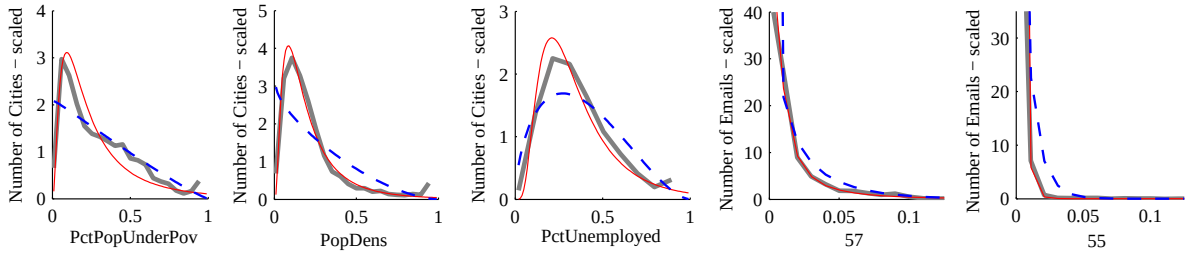


Figure A.1: Distributions of some attributes in Comm-Crime and SPAM. Bold curves are the histograms of the attributes. Thin curves are the best fitted Lognormal distributions; dashed curves are the best fitted Beta distributions.

We found that many words (attributes) of SPAM and Comm-Crime very likely follow lognormal distributions. Figure A.1 shows the distributions of some representative words. To see whether or not these words are likely log-normally distributed, we fitted the data with lognormal distributions by maximum likelihood estimation. The solid thin curves in the figure are density functions of the best fitted lognormal distributions. We also fitted the data with the Beta distribution.[4] Interestingly, Beta distributions, as plotted by dashed curves, fit data very badly. By more investigations, we found that more than 85% of attributes in Comm-Crime very likely follow lognormal distributions. This amount in SPAM is 67%. For AP, NIPS and KOS, not many words seem to be log-normally distributed.

A.4 Insights into the Lognormal and Dirichlet distributions

The previous section provided us an overview on the diversity and log-normality of the considered datasets. Diversity differs from dataset to dataset, and in some respects represents characteristics of data types. Textual data often have much less diversity than non-textual data. There are non-negligible differences in terms of diversity between text corpora. We also have seen that many datasets have many log-normally distributed properties. These facts raise an important question of how to model well diversity and log-normality of real data.

Taking individual attributes (words) into account in modeling data, one may immediately think about using the lognormal distribution to deal with the log-normality of data. This naive intuition seems to be appropriate in the context of topic modeling. As

[4]Note that Beta distributions are 1-dimensional Dirichlet distributions. We fitted the data with this distribution for the aim of comparison in terms of goodness-of-fit between the Dirichlet and lognormal distributions.

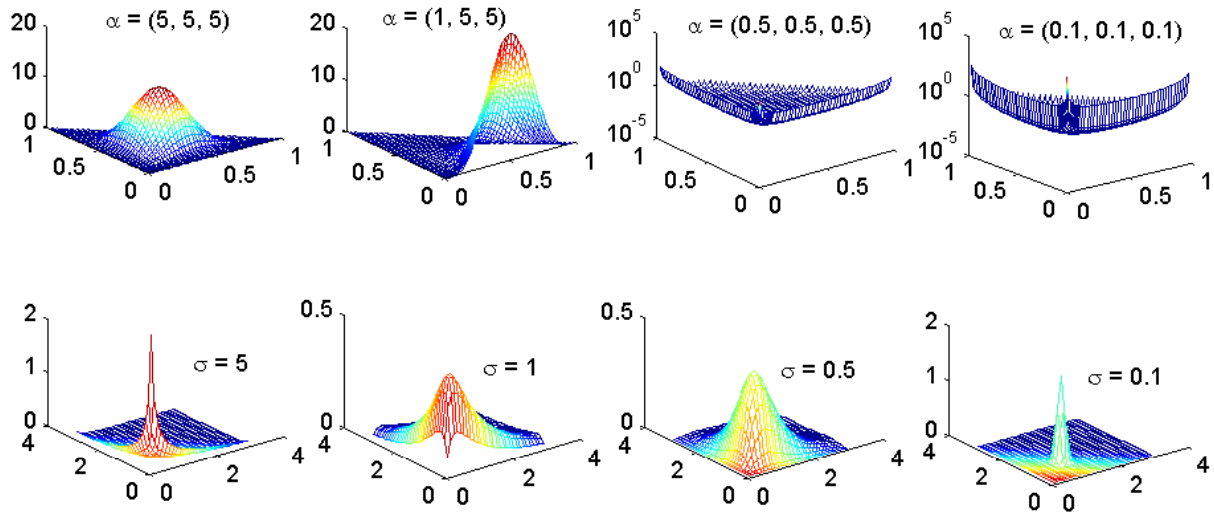


Figure A.2: Illustration of two distributions in the 2-dimensional space. The top row are the Dirichlet density functions with different parameter settings. The bottom row are the Lognormal density functions with parameters set as $\boldsymbol{\mu} = 0, \boldsymbol{\Sigma} = Diag(\sigma)$.

we shall see, the lognormal distribution is not only able to capture log-normality, but also able to model well diversity. Justifications for those abilities may be borrowed from the characteristics of the distribution.

Attempts to understand the lognormal and Dirichlet distributions were initiated. We began by illustrating the two distributions in 2-dimensional space. Depicted in Figure A.2 are density functions with different parameter settings.

As one can easily observe, the mass of the Dirichlet distribution will shift from the center of the simplex to the corners as the values of the parameters decrease. Conversely, the mass of the lognormal distribution will shift from the origin to regions which are far from the origin as σ decreases. From more careful observations, we realized that the lognormal distribution often has long (thick) tails as σ is large, and has quickly-decreased thin tails as σ is small. Nonetheless, the reverse phenomenon is the case for the Dirichlet distribution.

The tails of a density function tell us much about that distribution. A distribution with long (thick) tails would often generate many samples which are outside of its mass. This fact suggests that the variations of individual random variables in such a multivariate distribution might be large. As a consequence, such probability distributions often generate samples of high diversity.

Unlike distributions with long tails, those with short (thin) tails considerably restrict

Table A.3: Synthetic datasets originated from the Beta and lognormal distributions. As shown in this table, the Beta distribution very often yielded the same samples. Hence it generated datasets with diversity which is often much less than the number of attributes. Conversely, the lognormal distribution sometimes yielded repeated samples, and thus resulted in datasets with very high diversity.

Dataset	Drawn from	#Documents	#Attributes	Diversity
1	lognormal	1000	200	193.034
2	beta	1000	200	82.552
3	lognormal	5000	200	193.019
4	beta	5000	200	82.5986
5	lognormal	5000	2000	1461.6
6	beta	5000	2000	456.6768

variations of theirs samples. This implies that individual random variables in such distributions may be less free in terms of variation than those in long-tail distributions. Therefore, probability distributions with short thin tails are likely to generate samples of low diversity.

The above arguments suggest at least two implications. First, the lognormal distribution probably often generates samples of high diversity, and hence is capable of modeling high diversity data, since it often has long (thick) tails. Second, the Dirichlet distribution is appropriate to model data of low diversity like text corpora. As a result, it seems to be inferior in modeling data of high diversity, compared with the lognormal distribution.

With the aim of illustrating the above conclusions, we simulated an experiment as follows. Using tools from Matlab, we made 6 synthetic datasets from samples organized into documents. 3 datasets were constructed from samples drawn from the Beta distribution with parameters $\alpha = (0.1, 0.1)$; the others were from 1-dimensional lognormal distribution with parameters $\mu = 0, \sigma = 1$. All samples were rounded to the third decimal. Note that the Beta distribution is the 1-dimensional Dirichlet distribution. Some information of the 6 synthetic datasets is reported in Table A.3. Observe that with the same settings, the lognormal distribution gave rise to datasets with significantly higher diversity than the Beta distribution. Hence, this simulation supports further our conclusions above.

A.5 The DLN model

We have discussed in Section A.4 that the Dirichlet distribution seems to be inappropriate with data of high diversity. It will be shown empirically in the next section that this distribution often causes a topic model to be inconsistent with respect to diversity. In

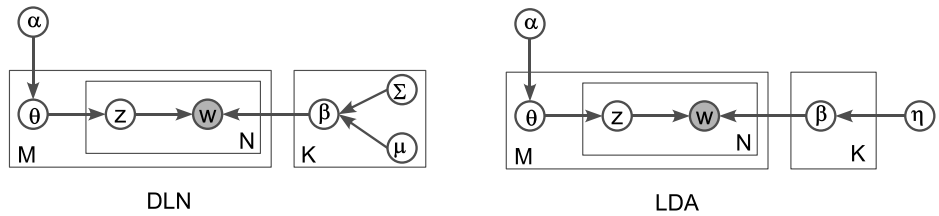


Figure A.3: Graphical model representations of DLN and LDA.

addition, many datasets seem to have log-normally distributed properties. Therefore, it is necessary to derive new topic models that can capture well diversity and log-normality. In this section, we describe a new variant of LDA, in which the Dirichlet distribution used to generate topics is replaced with the lognormal distribution.

Similar with LDA, the DLN model assumes the bag-of-words representations for both documents and corpus. Let $\mathcal{C}$ be a given corpus that consists of M documents, composed from the vocabulary $\mathcal{V}$ of V words. Then the corpus is assumed to be generated by the following process:

1. For each topic $k \in \{1, ..., K\}$, choose
   $\boldsymbol{\beta}_k | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k \sim LN(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
2. For each document d in the corpus:
   (a) Choose topic proportions $\boldsymbol{\theta}_d | \alpha \sim Dir(\alpha)$
   (b) For the nth word w_{dn} in the document,
      - Choose topic index $z_{dn} | \boldsymbol{\theta}_d \sim Mult(\boldsymbol{\theta}_d)$
      - Generate the word
        $w_{dn} | \boldsymbol{\beta}, z_{dn} \sim Mult(f(\boldsymbol{\beta}_{z_{dn}}))$.

Here $f(\cdot)$ is a mapping which maps $\boldsymbol{\beta}_k$ to parameters of multinomial distributions. In DLN, the mapping is

$$f(\boldsymbol{\beta}_k) = \frac{\boldsymbol{\beta}_k}{\sum_{j=1}^{V} \beta_{kj}}.$$

The graphical representation of the model is depicted in Figure A.3. We note that the distributions used to endow the topics are the main differences between DLN and LDA. Using the lognormal distribution also results in various difficulties in learning the model and inferring new documents. To overcome those difficulties, we used variational methods.

A.5.1 Variational method for learning and posterior inference

There are many learning approaches to a given model. Nonetheless, the lognormal distribution used in DLN is not conjugate with the multinomial distribution. So learning the parameters of the model is much more complicated than that of LDA. We use variational methods [110] for our model.

The main idea behind variational methods is to use simpler variational distributions to approximate the original distributions. Those variational distributions should be tractable to learn their parameters, but still give good approximations.

Let $\mathcal{C}$ be a given corpus of M documents, say $\mathcal{C} = \{\boldsymbol{w}_1, ..., \boldsymbol{w}_M\}$. $\mathcal{V}$ is the vocabulary of the corpus and has V words. The jth word of the vocabulary is represented as the jth unit vector of the V-dimensional space $\mathcal{R}^V$. More specifically, if w_j is the jth word in the vocabulary $\mathcal{V}$ and w_j^i is the ith component of w_j, then $w_j^i = 0$ for all $i \neq j$, and $w_j^j = 1$. These notations are similar to those in [21] for ease of comparison.

The starting point of our derivation for learning and inference is the joint distribution of latent variables for each document d, $P(\boldsymbol{z}_d, \boldsymbol{\theta}_d, \boldsymbol{\beta}|\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. This distribution is so complex that it is intractable to deal with. We will approximate it by the following variational distribution:

$$
\begin{aligned}
Q(\boldsymbol{z}_d, \boldsymbol{\theta}_d, \boldsymbol{\beta}|\boldsymbol{\phi}_d, \boldsymbol{\gamma}_d, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}) &= Q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d) Q(\boldsymbol{z}_d|\boldsymbol{\phi}_d) \prod_{k=1}^{K} Q(\boldsymbol{\beta}_k|\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k) \\
&= Q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d) \prod_{n=1}^{N_d} Q(z_{dn}|\boldsymbol{\phi}_{dn}) \prod_{k=1}^{K} \prod_{j=1}^{V} Q(\beta_{kj}|\widehat{\mu}_{kj}, \widehat{\sigma}_{kj}^2)
\end{aligned}
$$

Where $\widehat{\boldsymbol{\Sigma}}_k = diag(\widehat{\sigma}_{k1}^2, ..., \widehat{\sigma}_{kV}^2), \widehat{\boldsymbol{\mu}}_k = (\widehat{\mu}_{k1}, ..., \widehat{\mu}_{kV})^T \in \mathcal{R}^V$. The variational distribution of discrete variable z_{dn} is specified by the K-dimensional parameter $\boldsymbol{\phi}_{dn}$. Likewise, the variational distribution of continuous variable $\boldsymbol{\theta}_d$ is specified by the K-dimensional parameter $\boldsymbol{\gamma}_d$. The topic-word distributions are approximated by much simpler variational distributions $Q(\boldsymbol{\beta}_k|\widehat{\boldsymbol{\mu}}_k, \widehat{\boldsymbol{\Sigma}}_k)$ which are decomposable into 1-dimensional lognormals.

We now consider the log likelihood of the corpus $\mathcal{C}$ given the model $\{\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$.

$$
\begin{aligned}
\log P(\mathcal{C}|\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \sum_{d=1}^{M} \log P(\boldsymbol{w}_d|\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\
&= \sum_{d=1}^{M} \log \int d\boldsymbol{\theta}_d \int d\boldsymbol{\beta} \sum_{z_d} P(\boldsymbol{w}_d, \boldsymbol{z}_d, \boldsymbol{\theta}_d, \boldsymbol{\beta}|\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \\
&= \sum_{d=1}^{M} \log \int d\boldsymbol{\theta}_d \int d\boldsymbol{\beta} \sum_{z_d} P(\boldsymbol{w}_d, \Xi|\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \frac{Q(\Xi|\Lambda)}{Q(\Xi|\Lambda)}.
\end{aligned}
$$

Where we have denoted $\Xi = \{\boldsymbol{z}_d, \boldsymbol{\theta}_d, \boldsymbol{\beta}\}$, $\Lambda = \{\boldsymbol{\phi}_d, \boldsymbol{\gamma}_d, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}\}$. By Jensen's inequality [110] we have

$$
\begin{aligned}
\log P(\mathcal{C}|\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &\geq \sum_{d=1}^{M} \int d\boldsymbol{\theta}_d \int d\boldsymbol{\beta} \sum_{z_d} Q(\Xi|\Lambda) \log \frac{P(\boldsymbol{w}_d, \Xi|\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma})}{Q(\Xi|\Lambda)} \\
&\geq \sum_{d=1}^{M} \left[\boldsymbol{E}_Q \log P(\boldsymbol{w}_d, \Xi|\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma}) - \boldsymbol{E}_Q \log Q(\Xi|\Lambda) \right]. \qquad \text{(A.1)}
\end{aligned}
$$

The task of the variational EM algorithm is to optimize the equation (A.1), i.e., to maximize the lower bound of the log likelihood. The algorithm alternates E-step and M-step until convergence. In the E-step, the algorithm tries to maximize the lower bound w.r.t variational parameters. Then for fixed values of variational parameters, the M-step maximizes the lower bound w.r.t model parameters. In summary, the EM algorithm for the DLN model is as follows.

- ***E-step***: *maximize the lower bound in (A.1) w.r.t* $\boldsymbol{\phi}, \boldsymbol{\gamma}, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}$.
- ***M-step***: *maximize the lower bound in (A.1) w.r.t* $\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma}$.
- *Iterate these two steps until convergence.*

Note that DLN differs from LDA only in topic-word distributions. Thus $\boldsymbol{\phi}, \boldsymbol{\gamma}$, and α can be learnt as in [21], with a slightly different formula for $\boldsymbol{\phi}$.

$$
\phi_{dni} \propto \left[\widehat{\mu}_{i\nu} - \log \sum_{t=1}^{V} \exp(\widehat{\mu}_{it} + \frac{1}{2}\widehat{\sigma}_{it}^2) \right] \exp\left(\Psi(\gamma_{di}) - \Psi(\sum_{j=1}^{K} \gamma_{dj}) \right) \qquad \text{(A.2)}
$$

To complete the description of the learning algorithm for DLN, we next deal with the remaining variational parameters and model parameters. For the aim of clarity, we begin with the lower bound in (A.1).

$$\begin{aligned}\boldsymbol{E}_Q \log P(\boldsymbol{w}_d, \Xi|\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma}) &= \boldsymbol{E}_Q \log P(\boldsymbol{w}_d|\boldsymbol{z}_d, \boldsymbol{\beta}) + \boldsymbol{E}_Q \log P(\boldsymbol{z}_d|\boldsymbol{\theta}_d) \\ &\quad + \boldsymbol{E}_Q \log P(\boldsymbol{\theta}_d|\alpha) + \boldsymbol{E}_Q \log P(\boldsymbol{\beta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})\end{aligned}$$

$$\begin{aligned}\boldsymbol{E}_Q \log Q(\Xi|\boldsymbol{\phi}_d, \boldsymbol{\gamma}_d, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}) &= \boldsymbol{E}_Q \log Q(\boldsymbol{z}_d|\boldsymbol{\phi}_d) + \boldsymbol{E}_Q \log Q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d) \\ &\quad + \sum_{i=1}^{K} \boldsymbol{E}_Q \log Q(\boldsymbol{\beta}_i|\widehat{\boldsymbol{\mu}}_i, \widehat{\boldsymbol{\Sigma}}_i)\end{aligned}$$

Thus the log likelihood now is

$$\begin{aligned}\log P(\mathcal{C}|\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma}) \geq & \sum_{d=1}^{M} \boldsymbol{E}_Q \log P(\boldsymbol{w}_d|\boldsymbol{z}_d, \boldsymbol{\beta}) \\ & - \sum_{d=1}^{M} \left[KL\left(Q(\boldsymbol{z}_d|\boldsymbol{\phi}_d)||P(\boldsymbol{z}_d|\boldsymbol{\theta}_d)\right) - KL\left(Q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d)||P(\boldsymbol{\theta}_d|\alpha)\right)\right] \\ & - \sum_{d=1}^{M}\sum_{i=1}^{K} KL\left(Q(\boldsymbol{\beta}_i|\widehat{\boldsymbol{\mu}}_i, \widehat{\boldsymbol{\Sigma}}_i)||P(\boldsymbol{\beta}_i|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right) \qquad \text{(A.3)}\end{aligned}$$

Where $KL(\cdot||\cdot)$ is the Kullback-Leibler divergence of two distributions. Since $Q(\boldsymbol{z}_d|\boldsymbol{\phi}_d)$ and $P(\boldsymbol{z}_d|\boldsymbol{\theta}_d)$ are multinomial distributions, according to [74], we have

$$KL\left(Q(\boldsymbol{z}_d|\boldsymbol{\phi}_d)||P(\boldsymbol{z}_d|\boldsymbol{\theta}_d)\right) = \sum_{n=1}^{N_d}\sum_{i=1}^{K} \phi_{dni} \log \phi_{dni} - \sum_{n=1}^{N_d}\sum_{i=1}^{K} \phi_{dni} \left[\Psi(\gamma_{di}) - \Psi(\sum_{t=1}^{K} \gamma_{dt})\right] \qquad \text{(A.4)}$$

Where $\Psi(\cdot)$ is the digamma function. Note that the first term is the expectation of $\log Q(\boldsymbol{z}_d|\boldsymbol{\phi}_d)$, and the second one is the expectation of $\log P(\boldsymbol{z}_d|\boldsymbol{\theta}_d)$ for which we have used the expectation of the sufficient statistics $\boldsymbol{E}_Q[\log \theta_{di}|\gamma_d] = \Psi(\gamma_{di}) - \Psi(\sum_{t=1}^{K} \gamma_{dt})$ for the Dirichlet distribution [21].

Similarly, for Dirichlet distributions as implicitly shown in [21],

$$
\begin{aligned}
&KL\left(Q(\boldsymbol{\theta}_d|\boldsymbol{\gamma}_d)||P(\boldsymbol{\theta}_d|\alpha)\right) = \\
&-\log\Gamma(\sum_{i=1}^{K}\alpha_i) + \sum_{i=1}^{K}\log\Gamma(\alpha_i) - \sum_{i=1}^{K}(\alpha_i - 1)\left(\Psi(\gamma_{di}) - \Psi(\sum_{t=1}^{K}\gamma_{dt})\right) \\
&+\log\Gamma(\sum_{j=1}^{K}\gamma_{dj}) - \sum_{i=1}^{K}\log\Gamma(\gamma_{di}) + \sum_{i=1}^{K}(\gamma_{di} - 1)\left(\Psi(\gamma_{di}) - \Psi(\sum_{t=1}^{K}\gamma_{dt})\right)
\end{aligned} \tag{A.5}
$$

By a simple transformation, we can easily show that the KL divergence of two log-normal distributions, $Q(\boldsymbol{\beta}|\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}})$ and $P(\boldsymbol{\beta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, is equal to that of other normal distributions, $Q^*(\boldsymbol{\beta}|\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}})$ and $P^*(\boldsymbol{\beta}|\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Hence using the KL divergence of two Normals as in [75], we obtain the divergence of two lognormal distributions.

$$
\begin{aligned}
&KL\left(Q(\boldsymbol{\beta}_i|\widehat{\boldsymbol{\mu}}_i, \widehat{\boldsymbol{\Sigma}}_i)||P(\boldsymbol{\beta}_i|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right) \\
&= \frac{1}{2}\log|\widehat{\boldsymbol{\Sigma}}_i^{-1}\boldsymbol{\Sigma}_i| + \frac{1}{2}Tr\left((\widehat{\boldsymbol{\Sigma}}_i^{-1}\boldsymbol{\Sigma}_i)^{-1}\right) - \frac{V}{2} + \frac{1}{2}(\widehat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\widehat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i)
\end{aligned} \tag{A.6}
$$

Where $Tr(A)$ is the trace of the matrix A.

The remaining term in (A.3) is the expectation of the log likelihood of the document $\boldsymbol{w}_d$. To find more detailed representations, we observe that, since $\boldsymbol{\beta}_i$ is a log-normally random variable,

$$
\begin{aligned}
\boldsymbol{E}_Q \log\beta_{ij} &= \widehat{\mu}_{ij}, j \in \{1, ..., V\} \\
\boldsymbol{E}_Q \log\sum_{t=1}^{V}\beta_{it} &= \log\exp\left(\boldsymbol{E}_Q \log\sum_{t=1}^{V}\beta_{it}\right) && \text{(A.7)} \\
&\leq \log\boldsymbol{E}_Q\sum_{t=1}^{V}\beta_{it} && \text{(A.8)} \\
&\leq \log\sum_{t=1}^{V}\exp(\widehat{\mu}_{it} + \widehat{\sigma}_{it}^2/2) && \text{(A.9)}
\end{aligned}
$$

Note that the inequality (A.8) has been derived from (A.7) using Jensen's inequality. The last inequality (A.9) is simply another form of (A.8), replacing the expectations of individual variables by their detailed formulas [55].

From those observations, we have

$$
\begin{aligned}
&\boldsymbol{E}_Q \log P(\boldsymbol{w}_d|\boldsymbol{z}_d, \boldsymbol{\beta}) \\
&= \sum_{n=1}^{N_d} \boldsymbol{E}_Q \log P(w_{dn}|z_{dn}, \boldsymbol{\beta}) && \text{(A.10)} \\
&= \sum_{n=1}^{N_d} \sum_{i=1}^{K} \sum_{j=1}^{V} \phi_{dni} w_{dn}^j \boldsymbol{E}_Q \left[\log \beta_{ij} - \log \sum_{t=1}^{V} \beta_{it} \right] && \text{(A.11)} \\
&\geq \sum_{n=1}^{N_d} \sum_{i=1}^{K} \sum_{j=1}^{V} \phi_{dni} w_{dn}^j \left[\widehat{\mu}_{ij} - \log \sum_{t=1}^{V} \exp(\widehat{\mu}_{it} + \widehat{\sigma}_{it}^2/2) \right] && \text{(A.12)}
\end{aligned}
$$

There is a little strange in the right-hand side of (A.11) resulting from (A.10). The reason is that in DLN each topic $\boldsymbol{\beta}_i$ has to be transformed by the mapping $f(\cdot)$ into parameters of the multinomial distribution. Hence the derived formula is more complicated than that of LDA.

A lower bound of the log likelihood of the corpus $\mathcal{C}$ is finally derived from combining (A.3), (A.4), (A.5), (A.6), and (A.12). We next have to incorporate this lower bound into the variational EM algorithm for DLN by describing how to maximize the lower bound with respect to the parameters.

Variational parameters:

First, we would like to maximize the lower bound by variational parameters, $\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}$. Note that the term containing $\widehat{\boldsymbol{\mu}}_i$ for each $i \in \{1, ..., K\}$ is

$$
\begin{aligned}
\mathcal{L}[\widehat{\boldsymbol{\mu}}_i] = &- \frac{M}{2} (\widehat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i)^T \boldsymbol{\Sigma}_i^{-1} (\widehat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i) \\
&+ \sum_{d=1}^{M} \sum_{n=1}^{N_d} \sum_{j=1}^{V} \phi_{dni} w_{dn}^j \left[\widehat{\mu}_{ij} - \log \sum_{t=1}^{V} \exp(\widehat{\mu}_{it} + \widehat{\sigma}_{it}^2/2) \right].
\end{aligned}
$$

Since log-sum-exp functions are convex in their variables [27], $\mathcal{L}[\widehat{\boldsymbol{\mu}}_i]$ is a concave function in $\widehat{\boldsymbol{\mu}}_i$. Therefore, we can use convex optimization methods to maximize $\mathcal{L}[\widehat{\boldsymbol{\mu}}_i]$. In particular, we use LBFGS [62] to find the maximum of $\mathcal{L}[\widehat{\boldsymbol{\mu}}_i]$ with the following partial derivatives

$$
\frac{\partial \mathcal{L}}{\partial \widehat{\mu}_{ij}} = -M \boldsymbol{\Sigma}_{ij}^{-1} (\widehat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i) + \sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi_{dni} w_{dn}^j - \sum_{d=1}^{M} \sum_{n=1}^{N_d} \phi_{dni} \frac{\exp(\widehat{\mu}_{ij} + \widehat{\sigma}_{ij}^2/2)}{\sum_{t=1}^{V} \exp(\widehat{\mu}_{it} + \widehat{\sigma}_{it}^2/2)}
$$

Where $\boldsymbol{\Sigma}_{ij}^{-1}$ is the jth row of $\boldsymbol{\Sigma}_i^{-1}$.

The term in the lower bound of (A.3) that contains $\widehat{\boldsymbol{\Sigma}}_i$ for each i is

$$\mathcal{L}[\widehat{\boldsymbol{\Sigma}}_i] \;=\; \frac{M}{2}\log|\widehat{\boldsymbol{\Sigma}}_i| - \frac{M}{2}Tr(\boldsymbol{\Sigma}_i^{-1}\widehat{\boldsymbol{\Sigma}}_i) - \sum_{d=1}^{M}\sum_{n=1}^{N_d}\phi_{dni}\log\sum_{t=1}^{V}\exp(\widehat{\mu}_{it}+\widehat{\sigma}_{it}^2/2)$$

We use LBFGS-B [128] to find its maximum subject to the constraints $\widehat{\sigma}_{ij}^2 > 0, \forall j \in \{1, ..., V\}$, with the following derivatives

$$\frac{\partial\mathcal{L}}{\partial\widehat{\sigma}_{ij}^2} = \frac{M}{2\widehat{\sigma}_{ij}^2} - \frac{M}{2}\sigma_{ij}^{-2} - \frac{1}{2}\sum_{d=1}^{M}\sum_{n=1}^{N_d}\phi_{dni}\frac{\exp(\widehat{\mu}_{ij}+\widehat{\sigma}_{ij}^2/2)}{\sum_{t=1}^{V}\exp(\widehat{\mu}_{it}+\widehat{\sigma}_{it}^2/2)}$$

Where σ_{ij}^{-2} is the jth element on the diagonal of $\boldsymbol{\Sigma}_i^{-1}$.

Model parameters:

We now want to maximize the lower bound of (A.3) with respect to the model parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, for the M-step of the variational EM algorithm. The term containing $\boldsymbol{\mu}_i$ for each i is

$$\mathcal{L}[\boldsymbol{\mu}_i] = -\frac{M}{2}(\widehat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\widehat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i)$$

The maximum of this function is reached at

$$\boldsymbol{\mu}_i = \widehat{\boldsymbol{\mu}}_i \tag{A.13}$$

The term containing $\boldsymbol{\Sigma}_i^{-1}$ that is to be maximized is

$$\begin{aligned}\mathcal{L}[\boldsymbol{\Sigma}_i^{-1}] \;=\; & \frac{M}{2}\log|\boldsymbol{\Sigma}_i^{-1}| - \frac{M}{2}Tr(\boldsymbol{\Sigma}_i^{-1}\widehat{\boldsymbol{\Sigma}}_i) \\ & -\frac{M}{2}(\widehat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\widehat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i)\end{aligned}$$

And its derivative is

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{\Sigma}_i^{-1}} = \frac{M}{2}\boldsymbol{\Sigma}_i - \frac{M}{2}\widehat{\boldsymbol{\Sigma}}_i - \frac{M}{2}(\widehat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i)(\widehat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i)^T$$

Setting this to 0, we can find the maximum point:

$$\boldsymbol{\Sigma}_i \;=\; \widehat{\boldsymbol{\Sigma}}_i + (\widehat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i)(\widehat{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i)^T \tag{A.14}$$

We have derived how to maximize the lower bound of the log likelihood of the corpus $\mathcal{C}$ in (A.1) with respect to the variational parameters and model parameters. The variational EM algorithm now proceeds by maximizing the lower bound w.r.t $\boldsymbol{\phi}, \boldsymbol{\gamma}, \widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Sigma}}$ under the fixed values of the model parameters, and then by maximizing w.r.t $\alpha, \boldsymbol{\mu}, \boldsymbol{\Sigma}$ under the fixed values of variational parameters. Iterate these two steps until convergence. In our experiments, the convergence criterion is that the relative change of the log likelihood was no more than 10^{-4}.

For inferences on each new document, we can use the same iterative procedure as described in [21] using the formula (A.2) for $\boldsymbol{\phi}$. The convergence threshold for the inferences of each document was 10^{-6}.

A.6 Evaluation

This section is dedicated to presenting evaluations and comparisons for the new model. The topic model that will be used to compare with DLN is LDA. As previously mentioned, LDA is very popular and is the core of various topic models, where the topic-word distributions are endowed with the Dirichlet distribution. This view on topics is the only point in which DLN differs from LDA. Hence, any advantages of DLN over LDA can be applied to other variants of LDA. However, any LDA-based model can be readily modified to become a DLN-based model. From these observations, it is reasonable to compare performances of DLN and LDA.

Our strategy is as follows:

- We want to see how good the predictive power of DLN is in general. Perplexity will be used as a standard measure for this task.
- Next, stability of topic models with respect to diversity will be considered. Additionally, we will also study whether LDA and DLN likely favor data of low or high diversity. See subsection A.6.2.
- Finally, we want to see how well DLN can model data having log-normality and high diversity. This will be measured via classification on two non-textual datasets, Comm-Crime and SPAM. Details are in subsection A.6.3.

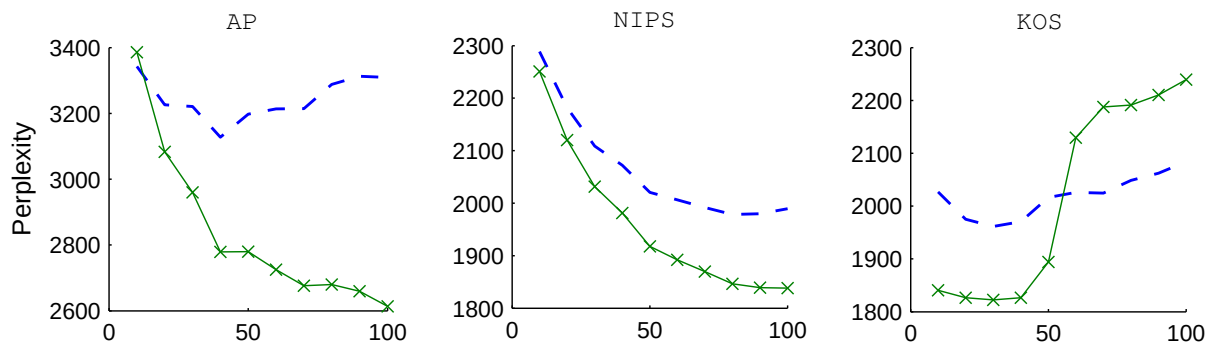


Figure A.4: Perplexity as the number of topics increases. Solid curves are DLN, dashed curves are LDA. The lower is the better.

A.6.1 Perplexity as a goodness-of-fit measure

We first use perplexity as a standard measure to compare LDA and DLN. Perplexity is a popular measure which evaluates the goodness-of-fit of a statistical model, and is widely used in the language modeling community. It is known to correlate closely with the precision-recall measure in information retrieval [50]. The measure is often used to compare predictive powers of different topic models as well.

Let $\mathcal{C}$ be the training data, and $\mathcal{D} = \{\boldsymbol{w}_1, ..., \boldsymbol{w}_T\}$ be the test set. Then perplexity is calculated by

$$Perp(\mathcal{D}|\mathcal{C}) = \exp\left(-\frac{\sum_{d=1}^{T} \log P(\boldsymbol{w}_d|\mathcal{C})}{\sum_{d=1}^{T} |\boldsymbol{w}_d|}\right).$$

The data for this task were the 3 text corpora. The two non-textual data sets were not considered, since perplexity is implicitly defined for text. For each of the 3 text corpora, we selected randomly 90% of the data to train DLN and LDA, and the remainings were used to test their predictive powers. Both models used the same convergence settings for both learning and inference. Figure A.4 shows the results as the number of topics increases. We can see clearly that DLN achieved better perplexity for AP and NIPS than LDA. However, it behaved worse than LDA on the KOS corpus.

Remember that NIPS has the greatest diversity among these 3 corpora as investigated in Section A.3. That is, the variations of the words in that corpus are very high. Besides, the lognormal distribution seems to favor data of high diversity as analyzed in Section A.4. The use of this distribution in DLN aims to capture the diversity of individual words better. Hence the better perplexity of DLN over LDA for the NIPS corpus is apparently justified.

The better result of DLN on NIPS also suggests more insights into the LDA model.

In Section A.4 we have argued that the Dirichlet distribution seems to favor data of low diversity, and seems inappropriate for high diversity data. These hypotheses are further supported by our experiments in this section.

Note that AP and KOS have nearly equal diversity. Nevertheless, the performances of both models on these corpora were quite different. DLN was much better than LDA on AP, but not on KOS. This phenomenon should be further investigated. In our opinion, some explanations for this may be borrowed from some observations in Section A.3. Notice that although the number of documents of KOS is approximately 50% larger than that of AP, the number of words having at least 5 different frequencies ($|OV| \geq 5$) in KOS is only about 20% larger than that of AP. This fact suggests that the words in AP seem to have higher variations than those in KOS. Besides, $Div_{\text{AP}} > Div_{\text{KOS}}$. Combining these observations, we can conclude that AP has higher variation than KOS. This is probably the reason why DLN performed better than LDA on AP.

A.6.2 Stability in predictive power

Next we would like to see whether the two models can work stably with respect to diversity. The experiments described in the previous subsection are not good enough to see this. The reason is that both topic models were tested on corpora of different numbers of documents, each with different document length. It means comparisons across various corpora by perplexity would not be fair if based on those experiments. Hence we need to conduct other experiments for this task.

Perplexity was used again for this investigation. To arrive at fair comparisons and conclusions, we need to measure perplexity on corpora of the same size and same document length. In order to have such corpora, we did as follows. We used 3 text corpora as above. For each corpus, 90% were randomly chosen for training, and the remaining were used for testing. In each testing set, each document was randomly cut off to remain only 100 occurrences of words in total. This means the resulting documents for testing were of the same length across testing sets. Additionally, we randomly removed some documents to remain only 100 documents in each testing set. Finally, we have 3 testing sets which are equal in size and document length.

After learning both topic models, the testing sets were inferred to measure their predictive powers. The results are summarized in Figure A.5. As known in Section A.3, the diversity of NIPS is greater than those of AP and KOS. However, LDA performed inconsistently in terms of perplexity on these corpora as the number of topics increased.

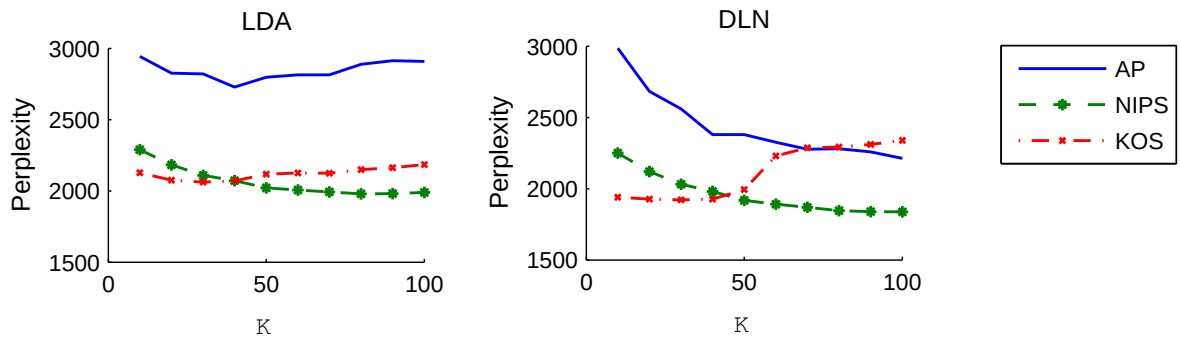


Figure A.5: Sensitivity of LDA and DLN against diversity, measured by perplexity as the number of topics increases. The testing sets were of same size and same document length in these experiments. Under the knowledge of $Div_{\text{NIPS}} > Div_{\text{AP}} > Div_{\text{KOS}}$, we can see that LDA performed inconsistently with respect to diversity; DLN performed much more consistently.

Higher diversity led to neither consistently better nor consistently worse perplexity. This fact suggests that LDA cannot capture well the diversity of data.

In comparison with LDA, DLN worked more consistently on these corpora. It achieved the best perplexity on NIPS, which has the largest diversity among 3 corpora. The gap in perplexity between NIPS and the others is quite large. This implies that DLN may capture well data of high diversity. However, since the perplexity for AP was worse than that for KOS while $Div_{\text{AP}} = 0.0012 > Div_{\text{KOS}} = 0.0011$, we do not know clearly whether DLN can cope well with data of low diversity or not. Answers for this question require more sophisticated investigations.

Another observation from the results depicted in Figure A.5 is that LDA seems to work well on data of low diversity, because its perplexity on KOS was consistently better than on other corpora. A reasonable explanation for this behavior is the use of the Dirichlet distribution to generate topics. Indeed, such distribution favors low diversity, as analyzed in Section A.4. Nonetheless, it is still unclear to conclude that LDA really works well on data of low diversity, because its perplexity for KOS was much better than that for AP while $Div_{\text{AP}} \simeq Div_{\text{KOS}}$.

A.6.3 Document classification

Our next experiments were to measure how well the two models work, via classification tasks, when data have high diversity and log-normality. As is well-known, topic models are basically high-level descriptions of data. In other words, the most interesting characteristics of data are expected to be captured in topic models. Hence we can consider

them as other representations of data. This interpretation implicitly allows us to apply them to many other applications, such as classification.

The datasets for these tasks are SPAM and Comm-Crime. We used micro precision [88] as a measure for comparison. Loosely speaking, precision can be interpreted as the extent of our confidence in assigning labels to documents. It is believed, at least in the text categorization community, that this measure is more reliable than the accuracy measure for classification [88]. Thus it is reasonable to use it for our tasks in this section.

SPAM is straightforward to understand, and is very suitable for the classification task. The main objective is to predict whether a given document is spam or not. Thus, we keep the spam attribute unchanged, and multiply all values of other attributes in all records by 10000 to make sure that the obtained values are integers. Resulting records are regarded as documents in which each value of an attribute is the frequency of the associated word.

The nature of Comm-Crime is indirectly related to classification. The goal of Comm-Crime is to predict how many violent crimes will occur per 100K population. In this corpus, all cities have these values that can be used to train or test a learning algorithm. Since predicting an exact number of violent crimes is unrealistic, we predicted the interval in which the number of violent crimes of a city most probably falls.[5]

Since all crime values in the original data were normalized to be in [0,1], two issues arise when performing classification on this dataset. First, how many intervals are appropriate? Second, how to represent crime values, each belonging to exactly one interval, as class labels. The first issue is easier to deal with in practice than the latter. In our experiments, we first tried 10 intervals, and then 15 intervals. For the second issue, we did as follows: each attribute was associated with a word except crime. The values of the attributes were scaled by the same number to make sure that all are integers, and then were regarded as frequencies of the associated words. For the crime attribute, we associated each interval with each class label. Each record then corresponds to a document, where the crime value is associated with a class label.

We considered performances on Comm-Crime of 3 approaches: SVM, DLN+SVM, LDA+SVM. Here we used multi-class SVM implemented in the package by Joachims.[6] It was trained and tested on the original dataset to ensure fair comparisons. DLN+SVM (and LDA+SVM) worked in the same way as in previous works [21], i.e., we first modeled

[5]Be aware that this dataset is also suitable to be used in regression, since the data were previously normalized to be in [0, 1]. However, this section is devoted to comparing topic models in terms of how well they can capture diversity and log-normality of data. SPAM and Comm-Crime are good datasets for these tasks, because they both have high diversity and many likely log-normally distributed attributes.

[6]Available from http://svmlight.joachims.org/svm_multiclass.html

Table A.4: Average precision in crime prediction.

#intervals	SVM	DLN+SVM	LDA+SVM
10	0.56	**0.61**	0.58
15	0.43	**0.48**	0.46

the data by DLN (LDA) to find hidden representations of the documents in terms of topic proportions vectors, and then used them as feature vectors for SVM. For topic models, the number of topics, K, should be chosen appropriately. In [111], Wallach et al. empirically showed that LDA may work better as the number of topics increases. Nevertheless, the Subsections A.6.1 and A.6.2 have indicated that large values of K did not lead to consistently better perplexity for LDA. Moreover, the two models did not behave so badly at $K = 50$. Hence we chose 20 topics for both topic models in subsequent experiments. We used 5-fold cross-validation for the candidates. The results are presented in Table A.4.

Among the 3 approaches, DLN+SVM consistently performed best. These results suggest that DLN worked better than LDA did. We remark that Comm-Crime has very high diversity and seems to have plenty of log-normality. Hence the better performance of DLN over LDA suggests that the new model can capture well log-normality of data, and can work well on data of high diversity.

One can realize that the precisions obtained from these approaches were quite low. In our opinion, this may be due to the inherent nature of that data. To provide evidence for our belief, we conducted separately regression on the original Comm-Crime dataset with two other well-known methods, Bagging and Linear Regression implemented in Weka.[7] Experiments with these methods used default parameters and used 5-fold cross-validation. Mean absolute errors from these experiments varied from 0.0891 to 0.0975. Note that all values of the attributes in the dataset had been normalized to be in [0, 1]. Therefore the resulting errors are problematic. After scaling and transforming the regression results to classification, the consequent precisions vary from 0.3458 to 0.4112. This variation suggests that Comm-Crime seems to be difficult for current learning methods.

The above experiments on Comm-Crime provide some supporting evidence for the good performance of DLN. We next conducted experiments for classification on SPAM. We used the same settings as above, 50 topics for topic models and 5-fold cross-validation. The results are described in Table A.5. One can easily observe the consistently better performance of our new model over LDA, working in combination with SVM. Note that precisions for SPAM are much greater than those for Comm-Crime. The reasons are

[7]Version 3.7.2 at http://www.cs.waikato.ac.nz/∼ml/weka/

Table A.5: Average precision in spam filtering.

SVM	DLN+SVM	LDA+SVM
0.81	**0.95**	0.92

that SPAM is inherently for binary classification, which is often easier than multi-class counterparts, and that the training set for SPAM is much bigger than that for Comm-Crime which better enables learning.

A.7 Discussion

In summary, we now have strong evidence from the empirical results and analyses for the following conclusions. First, DLN can get benefits from data that have many likely log-normally distributed properties. It seems to capture well log-normality of data. Second, DLN is more suitable than LDA on data of high diversity, since consistently better performances have been observed. Third, topic models are able to model well data that are non-textual, since the combinations of topic models with SVM often got better results than SVM did alone in our experiments.

LDA and DLN have been compared in various evaluations. The performance of DLN was consistent with the diversity of data, whereas LDA was inconsistent. Furthermore, DLN performed consistently better than LDA on data that have high diversity and many likely log-normally distributed properties. Note that in our experiments, the considered datasets have different diversities. This treatment aimed to ensure that each conclusion will be strongly supported. In addition, the lognormal distribution is likely to favor data of high diversity as demonstrated in Section A.4. Hence, the use of the lognormal distribution in our model really helps the model to capture diversity and log-normality of real data.

Although the new model has many distinguishing characteristics for real applications, it suffers from some limitations. First, due to the complex nature of the lognormal distribution, learning the model from real data is complicated and time-consuming. Second, the memory for practical implementation is large, $O(K.V.V + M.V + K.M)$, since we have to store K different lognormal distributions corresponding to K topics. Hence it is suitable with corpora of average vocabularies, and datasets with average numbers of attributes.

Some concerns may arise when applying DLN in real applications: what characteristics of data ensure the good performance of DLN? Which data types are suitable for DLN?

The followings are some of our observations.

- For non-textual datasets, DLN is very suitable if diversity is high. Our experiments suggest that the higher diversity the data have, the better DLN can perform. Note that diversity is basically proportional to the number of different values of attributes observed in a dataset. Hence, by intuition, if there are many attributes that vary significantly in a dataset, then the diversity of that dataset would be probably high, and thus DLN would be suitable.

- Log-normality of data is much more difficult to see than diversity.[8] Nonetheless, if once we know that a given dataset has log-normally distributed properties, DLN would probably work better on it than LDA.

- For text corpora, the diversity of a corpus is essentially proportional to the number of different frequencies of words observed in the corpus. Hence if a corpus has words that vary significantly, DLN would probably work better than LDA. The reason is that DLN favors data of high diversity.

- A corpus whose documents are often long will allow high variations of individual words. This implies that such a corpus is very likely to have high diversity. Therefore, DLN would probably work better than LDA, as observed in the previous section. Corpora with short documents seem to be suitable for LDA.

- A corpus that is made from different sources with different domains would very likely have high diversity. As we can see, each domain may result in a certain common length for its documents, and thus the average document length would vary significantly among domains. For instance, scientific papers in NIPS and news in AP differ very much in length; conversations in blogs are often shorter than scientific papers. For such mixed corpora, DLN seems to work well, but LDA is less favorable.

A.8 Related work

In the topic modeling literature, many models assume a given corpus to be composed of some hidden topics. Each document in that corpus is a mixture of those topics. The first

[8] In principle, checking the presence of log-normality in a dataset is possible. Indeed, checking the log-normality property is equivalent to checking the normality property. This is because if a variable x follows the normal distribution, then $y = e^x$ will follow the log-normal distribution [55], [60]. Hence, checking the log-normality property of a dataset $\mathcal{D}$ can be reduced to checking the normality property of the logarithm version of $\mathcal{D}$.

generative model of this type is known as *Probabilistic Latent Semantic Analysis* (pLSA) proposed by Hofmann [50]. Assuming pLSA models a given corpus by K topics, then the probability of a word w appearing in document d is

$$P(w|d) = \sum_z P(w|z)P(z|d), \tag{A.15}$$

where $P(w|z)$ is the probability that the word w appears in the topic $z \in \{1, ..., K\}$, and $P(z|d)$ is the probability that the topic z participates in the document d. However, pLSA regards the topic proportions, $P(z|d)$, to be generated from some discrete and document-specific distributions.

Unlike pLSA, the topic proportions in each document are assumed to be samples drawn from Dirichlet distributions in LDA [21]. Such assumption is strongly supported by the de Finetti theorem on exchangeable random variables [6]. Amazingly, LDA has been reported to be successful in many applications.

Many subsequent topic models have been introduced since then that differ from LDA in endowing distributions on topic proportions. For instance, CTM and IFTM treat the topic proportions as random variables which follow logistic distributions; *Hierarchical Dirichlet Process* (HDP) considers these vectors as samples drawn from a Dirichlet process [97]. Few models differ from LDA in view of topic-word distributions, i.e., distributions over words. Some candidates in this line are *Dirichlet Forest* (DF) [8], *Markov Topic Model* (MTM) [114], and *Continuous Dynamic Topic Model* (cDTM) [113].

Unlike those approaches, we endowed the topic-word distributions with the lognormal distribution. Such treatment aimed to tackle diversity and log-normality of real datasets. Unlike the Dirichlet distribution used by other models, the lognormal distribution seems to allow high variation of its samples, and thus can capture well high diversity data. Hence it is a good candidate to help us cope with diversity and log-normality.

A.9 Summary

In this work, we studied a fundamental property of real data, phrased as "diversity", which has not been paid enough attention from the machine learning community. Loosely speaking, diversity measures average variations of attributes within a dataset. We showed that diversity varies significantly among different data types. Textual corpora often have much less diversity than non-textual datasets. Even within text, diversity varies significantly among different types of text collections.

We empirically showed that diversity of real data non-negligibly affects performance of topic models. In particular, the well-known LDA model [21] worked inconsistently with respect to diversity. In addition, LDA seems not to model well data of high diversity. This fact raises an important question of how to model well the diversity of real corpora.

To deal with the inherent diversity property, we proposed a new variant of LDA, called DLN, in which topics are samples drawn from the lognormal distribution. In spite of being a simple variant, DLN was demonstrated to model well the diversity of data. It worked consistently and seemingly proportionally as diversity varies. On the other hand, the use of the lognormal distribution also allows the new model to capture lognormal properties of many real datasets [60], [34].

Finally, we remark that our approach here can be readily applied to various topic models since LDA is their core. In particular, the Dirichlet distribution used to generate topics can be replaced with the lognormal distribution to cope with diversity of data.